

UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE INGENIERIA

Sistema para controlar la información que se maneja en el área de personal de una Universidad

Tesis profesional que para obtener el titulo de:

Ingeniero en Computación

Presenta:

Gabriela Barajas Ruiz

Director: Ing. Carlos Saucedo Maciel

Ciudad Universitaria **Marzo 2006**



AGRADECIMIENTOS:

Dedicada con todo cariño a mis queridos padres,
quienes me apoyaron incondicionalmente con todo su amor
a lo largo de mi vida profesional.

Con toda mi admiración le dedico este trabajo a mi hermana
la M.. en A. Liliana Barajas Ruiz ya que ha sido un ejemplo de disciplina,
dedicación y constancia a lo largo de toda mi vida.

Gracias a mi esposo Oscar A. Pérez Aguirre
quien me apoyó en todos los sentidos para la realización de este proyecto.

Con especial dedicatoria a mi amado hijo, ya que con su llegada a este mundo
me ha impulsado enormemente a superarme día con día y desde ahora y para siempre
se ha convertido en mi fuente de inspiración. Te amo Miguelito.

Gracias a mi querida Facultad de Ingeniería
y a todos los profesores que contribuyeron en mi formación profesional

PANORAMA GENERAL

Para nadie es un secreto el enorme crecimiento de la tecnología, en todos los ámbitos y actividades empresariales en nuestros días y más aún lo que se vislumbra en un futuro cercano; y por supuesto el área de Recursos Humanos no podía quedar al margen de ello.

El desarrollo tecnológico ha alcanzado niveles sorprendentes, tal vez más allá de lo que hace unos cuantos años podríamos haber imaginado. Y que decir de la optimización, automatización y agilización de los medios empleados en Recursos Humanos.

Los beneficios que ofrecen los sistemas se ven reflejados en:

- Mayor equilibrio del presupuesto de Recursos Humanos
- Optimización de tiempos
- Administración de recursos
- Planeación de actividades entre otros puntos

Todas las innovaciones informáticas están orientadas a facilitar las labores en los diferentes aspectos de la industria, sin embargo, con todo lo impresionante y poderoso que es y pudiera llegar a ser el avance informático, existe un factor que nunca será ni deberá ser sustituido por la tecnología y es precisamente el “Elemento Humano”.

Este es el factor fundamental en el crecimiento de toda empresa, tal vez por ello se ha iniciado un cambio en el concepto propio de Recursos Humanos por el de Capital Humano o Capital Intelectual.

Tampoco es posible ver a la gente como máquinas productoras de dinero, ya que no es el dinero lo que le da valor a las empresas sino su Capital Humano. No se puede concebir una empresa “robotizada y mecanizada” exitosa, sino más bien se conciben empresas más humanas con un exacto equilibrio en lo tecnológico para alcanzar el éxito.

Junto a la tecnología, los Recursos Humanos son el aspecto que mayor crecimiento en importancia ha tenido en el estudio moderno de las organizaciones. Poco a poco, todas las

ciencias y técnicas jurídicas, económicas, empresariales y sociales tienen como materia de estudio y como objetivo de sus leyes al ser humano y a los grupos sociales. La Dirección y Gestión de Recursos Humanos surge como la disciplina que estudia el factor humano en las organizaciones en todas sus dimensiones: la motivación, el papel de los grupos informales, la comunicación, el poder, la cultura, el liderazgo y los estilos de dirección, entre otros.

La Gestión de Recursos Humanos tiene importancia vital en el cumplimiento de las metas de la organización. Una de sus principales tareas es proporcionar las capacidades humanas requeridas por la organización y desarrollar habilidades y aptitudes del individuo para hacerlo más satisfactorio a él y a la colectividad en que se desenvuelve. Estos elementos, se pueden garantizar en gran medida con el diseño de un sistema de selección de personal que provea a la organización del material humano que necesita, brindando a la par, posibilidades de desarrollo y satisfacción personal. Un proceso de selección eficiente, que permita cubrir los puestos de trabajo de tal forma que la organización pueda ser operada de manera competente tanto en el presente, como en el futuro, puede ser la clave para el cumplimiento de la misión organizacional.

INTRODUCCIÓN

El sistema que se describe en el presente trabajo, fue desarrollado para una universidad de prestigio, por razones de confidencialidad no se mencionará el nombre original. Cuando se haga referencia a ella se utilizara el nombre ficticio **"Universidad Virtual" (UV).** A continuación se detallan sus principales características:

La Universidad UV tiene como misión contribuir –en un ambiente de participación responsable, apertura, libertad y respeto–- al desarrollo y la difusión del conocimiento, y a la formación de profesionales e investigadores con calidad humana y académica, que se comprometan en el servicio a los demás para el logro de una sociedad más libre, productiva, justa y solidaria.

A lo largo de más de 60 años, la Universidad UV no sólo ha sido testigo sino agente activo del desarrollo de México. Su colaboración –cifrada en las tres funciones sustantivas de docencia, investigación y difusión–, ha estado sustentada en claros principios de libertad y responsabilidad social.

La Universidad UV, cuenta con un amplio número de empleados dedicado cada uno de ellos a colaborar de manera activa en el cumplimiento de las funciones y misión de la Universidad.

El personal administrativo de la UV, esta dividido en unidades de organización, mismas que están formadas de acuerdo a los siguientes criterios:

1. El trabajo que se debe hacer
2. Las personas de que se puede disponer
3. Los lugares en que dicho trabajo se puede realizar

A continuación, se presenta un organigrama general de la UV con las principales unidades de organización ó entidades en que se encuentra dividida:

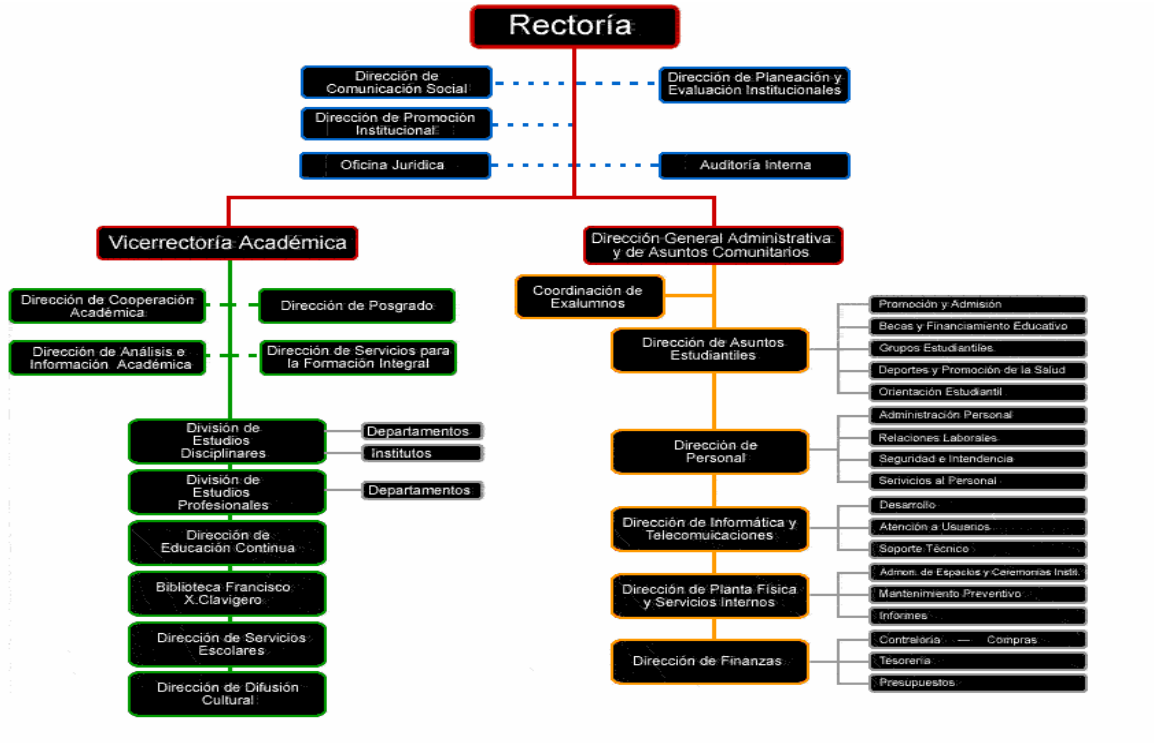


Figura 1 *Entidades de la UV*

Su sistema de organización es lineal, es decir, cada individuo dentro de la empresa tiene un solo jefe para todos los aspectos, por consiguiente, únicamente recibe órdenes y le reporta a el.

Debido a que la UV es una organización de servicio, su dirección de mayor importancia es la de personal, ya que de la inteligencia, la experiencia laboral y la creatividad de su personal, dependerá la fuerza de la empresa.

La dirección de personal de la UV, se divide de la siguiente forma:

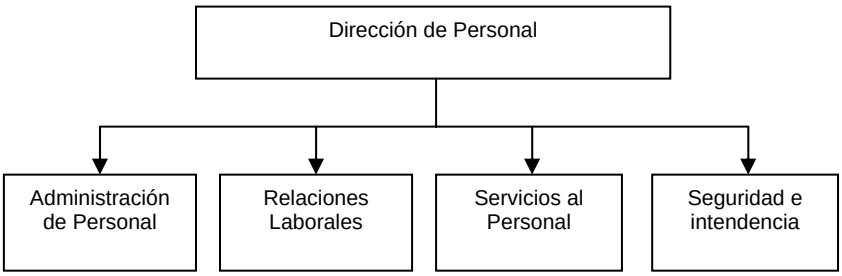


Figura 2 *División de la dirección de personal*

El presente trabajo, se enfoca en la entidad de administración de personal de la UV, cuyas funciones principales son:

- Análisis de puestos y plazas
- Reclutamiento Selección y Contratación de personal
- Análisis Organizacional
- Control de personal
- Control de Nómina

Muchos de los procesos que se llevan a cabo en el área de administración de personal, se realizan sin ayuda de un sistema que permita automatizar y centralizar la información de candidatos, personal y dependencias que se manejan.

Para ingresar a la UV, es necesario cumplir con una serie de requisitos, en este proceso muchos candidatos pueden ser rechazados por diferentes circunstancias ya sea por invalides de documentos, falta de vacante o carencia de alguno de los requisitos solicitados.

Sin embargo, para la UV, es necesario llevar un control de todas estas solicitudes, con el fin de dar un seguimiento al proceso, desde la entrega de solicitud hasta la firma o rechazo del contrato.

Así mismo, es de suma importancia para la UV saber como esta organizado el personal dentro de ella, conocer el número de empleados, el número de puestos y plazas ocupados y vacantes así como la estructura organizacional de cada una de sus entidades con el fin de tener un control de las contrataciones que se pueden hacer, los recursos con que se cuenta y el costo por dependencia.

En resumen, el problema se puede acotar en los siguientes puntos:

1. Necesidad de registro y control de información de candidatos
2. Necesidad de control en altas de puestos y plazas.
3. Necesidad de control de las diversas etapas para la contratación de personal.
4. Necesidad de control de la estructura organizacional de una empresa.
5. Necesidad de presentar gráficamente la estructura organizacional mediante métodos más sencillos.

OBJETIVO GENERAL

Desarrollar un sistema para apoyar al área de Administración de personal de la UV en el proceso de Selección y Contratación de su personal. Así mismo, el sistema deberá apoyar a optimizar el proceso de representación gráfica de la estructura organizacional por medio de organigramas y a la administración de puestos y plazas de nueva creación.

OBJETIVOS ESPECIFICOS

- Presentar un panorama general de las funciones que se desarrollan en el área de recursos humanos, de teoría de programación y de bases de datos.

- Conocer los principales modelos y métodos para el desarrollo de software, sus ventajas y desventajas para finalmente seleccionar el que mas se apega a las necesidades del sistema.

- Realizar un estudio de factibilidad y establecer las características específicas que el sistema debe cubrir mediante la definición de requerimientos.

- Definir la arquitectura del sistema, realizar el diseño de la base de datos, el diccionario de datos y establecer los usuarios y políticas de respaldo.

- Presentar el diseño de las pantallas creadas y el código fuente de las funciones principales para el desarrollo del sistema de la UV

- Describir las estrategias utilizadas para realizar las pruebas por módulo y globales del sistema para asegurar que cumple con los requerimientos planteados por la UV.

INDICE

Panorama General 6

Introducción 8

Capitulo 1 Antecedentes y Marco Teórico

1.1 Reclutamiento Selección y Contratación 14
1.1.1 Organización 14
1.1.2 Unidades de Organización 15
1.1.3 Sistema de Organización Lineal 16
1.1.4 Organigramas 16
1.1.5 Análisis de puestos 17
1.1.5.1 Actividades involucradas en el análisis de puestos 19
1.1.6 Reclutamiento de personal 19
1.1.6.1 Fuentes de reclutamiento 20
1.1.6.2 Medios de reclutamiento 23
1.1.7 Selección de Personal 24
1.1.7.1 Elaboración del perfil 25
1.1.7.2 Solicitud 26
1.1.7.3 Evaluación técnica 27
1.1.7.4 Evaluación psicológica 28
1.1.7.5 Evaluación socio-económica 29
1.1.7.6 Entrevista en la selección 29
1.1.7.7 Examen médico 32
1.1.7.8 Conclusión 33
1.1.7.9 Decisión 33
1.1.8 Administración de personal 33
1.1.8.1 Definición 33
1.1.8.2 Objetivos 34
1.1.8.3 Principales funciones 34
1.1.8.4 Principios de la selección de personal 36
1.2 Teoría sobre lenguajes de programación
1.2.1 Definición 37
1.2.2 Antecedentes 38
1.2.3 Tipos de Lenguajes 41
1.2.4 Estilos de programación 43
1.2.5 Tendencias 46
1.3 Bases de Datos
1.3.1 Conceptos Básicos 46

1.3.1.1 Conceptos básicos de bases de datos 46
1.3.1.2 Conceptos básicos de archivos computacionales 47
1.3.2 Definición 47
1.3.3 Objetivos de los sistemas de bases de datos 47
1.3.4 Niveles 50
1.3.5 Modelo de datos 51
1.3.5.1 Clasificación de modelos 51
1.5.6 Manejador de bases de datos 54
1.5.6.1 Características de los RDBMS 54
1.5.6.2 Funciones principales 55
1.5.6.3 Esquemas de seguridad en el RDBMS 56
1.5.6.4 Componentes de un RDBMS 57
1.5.7 Conclusión 60

Capitulo 2 Análisis y definición de requerimientos

2.1 Modelos del proceso de software 62
2.1.1 Modelo cascada o ciclo de vida clásico 64
2.1.2 Desarrollo evolutivo 67
2.1.3 Desarrollo formal de sistemas 68
2.1.4 Desarrollo orientado a reutilización 70
2.1.5 Desarrollo Incremental 71
2.1.6 Desarrollo en espiral 74
2.2 Métodos de la ingeniería de software 75
2.2.1 Método de desarrollo por análisis estructurado 75
2.1.2 Diseño estructurado 78
2.3 Estudio de factibilidad 80
2.4 Requerimientos 85
2.4.1 Definición 86
2.4.2 Clasificación de requerimientos 86
2.4.3 Requerimientos de la UV 87

Capítulo 3 Diseño del sistema

3.1 Arquitectura del sistema 103
3.2 Diccionario de datos 111
3.3 Modelo Entidad-Relación 116
3.3.1 Convenciones básicas y definiciones empleadas 117
3.3.2 Atributos 119
3.3.3 Identificadores únicos 119
3.3.4 Diagrama entidad-relación de la UV 120
3.4 Usuarios 121

3.5 Respaldos de la información 125

Capitulo 4 Desarrollo del sistema

4.1 Funciones generales 128
4.2 Conexión a la base de datos 134
4.3 Procedimientos almacenados 134
4.4 Pantallas del sistema 138

Capitulo 5

5.1 Pruebas de defectos 153
5.1.1 Pruebas de caja negra 153
5.1.2 Particiones de equivalencia 154
5.1.3 Pruebas Estructurales 155
5.1.4 Pruebas de trayectorias 156
5.2 Pruebas de integración 157
5.2.1 Pruebas descendentes y ascendentes 158
5.2.2 Pruebas de interfaces 159
5.2.3 Pruebas de esfuerzo 160
5.3 Herramienta McCabe 160
5.3.1 Complejidad Ciclomática 161
5.3.2 Complejidad esencial 162
5.4 Conclusiones Finales 167

Anexo A Manual del Usuario 168

Anexo B Glosario 194

Bibliografía 196

CAPITULO 1

A N T E C E D E N T E S

Y

M A R C O T E Ó R I C O

Objetivos:

- Mostrar un panorama general de las funciones que se desarrollan en el área de Recursos Humanos.
- Proporcionar un marco teórico en donde se describan las principales teorías de programación y conceptos generales de bases de datos.

1.1 RECLUTAMIENTO SELECCIÓN Y CONTRATACIÓN

1.1.1 ORGANIZACIÓN

[1] El hombre es incapaz de cubrir todas sus necesidades por sí mismo debido a que carece de la capacidad, tiempo, fuerza y perseverancia necesarias, por ello requiere apoyarse en los demás. Cuando varias personas toman conciencia de que al coordinar sus esfuerzos pueden conseguir más de lo que podrían hacer por sí mismas, se está en ese momento hablando de organización.

El concepto básico que sustenta la idea de organización es el de una coordinación de esfuerzos, cuyo objetivo es la ayuda mutua.

El concepto de organización implica que hay que alcanzar algunos objetivos o finalidades comunes a través de la coordinación de actividades. Al efectuar la coordinación de actividades se requiere de una distribución equitativa de las mismas, en función de la especialidad que respalde a cada persona. A esto se le denomina “división del trabajo”.

La organización se refiere a estructurar, es esto quizá la parte más típica de los elementos que corresponden a la mecánica administrativa. Nos dice en concreto quién va a hacer cada cosa

(en el sentido de que puesto) y cómo lo va a hacer. Cuando la organización está terminada, solo resta actuar integrando, dirigiendo y controlando.

La organización, por ser el elemento final del aspecto teórico: recoge, complementa y lleva hasta sus últimos detalles todo lo que la previsión y la planeación han señalado con respecto a cómo debe ser una empresa.

1.1.2 UNIDADES DE ORGANIZACIÓN

Se forman agrupando las funciones en cada línea básica, de acuerdo con tres criterios básicos principales:

1. El trabajo que se debe hacer
2. Las personas concretas de que se puede disponer
3. Los lugares en que dicho trabajo se debe realizar.

Aún cuando la nomenclatura no suele ser uniforme en las distintas empresas, la más general considera los grupos del siguiente modo:

- Gerencia, en el primer nivel jerárquico
- Departamento, en el segundo
- Oficina, en el tercero
- Sección, en el cuarto
- Grupo y subgrupos, en el quinto
- Unidad y sub-unidades, en el sexto.

Las denominaciones cambian de una empresa a otra.

---

[1]Granados, Jaime A (2004). Reclutamiento, selección, contratación de personal. México, El manual moderno (Pág. 20).

1.1.3 SISTEMA DE ORGANIZACIÓN LINEAL

La organización lineal es aquella donde la autoridad y responsabilidad correlativas se transmiten íntegramente por una sola línea para cada persona o grupo. Existe solo un jefe para todos los aspectos, por lo que sólo se reciben órdenes de él.

Las ventajas de este sistema son:

- Es sencillo y claro
- No existen conflictos de autoridad ni fugas de responsabilidad
- Se facilita la rapidez de acción
- Se crea una firme disciplina
- Es más fácil y útil en la pequeña empresa.

1.1.4 ORGANIGRAMAS

Los sistemas de organización se representan de manera intuitiva y con objetividad en los llamados organigramas, conocidos también como cartas o gráficas de organización.

Consisten en hojas o cartulinas en las que cada puesto de jefe se representa por un cuadro que encierra el nombre de ese puesto (y en ocasiones de quien lo ocupa); la unión de los cuadros mediante líneas indica los canales de autoridad y responsabilidad. Esto ayuda a mostrar:

- La división de funciones
- Niveles jerárquicos
- Líneas de autoridad y responsabilidad
- Canales formales de comunicación

- Naturaleza lineal o staff del departamento
- Jefes de cada grupo
- Relaciones entre los puestos de la empresa, departamento o sección.

Dependiendo la cantidad de niveles jerárquicos, los organigramas pueden ser verticales, horizontales o aplanados.

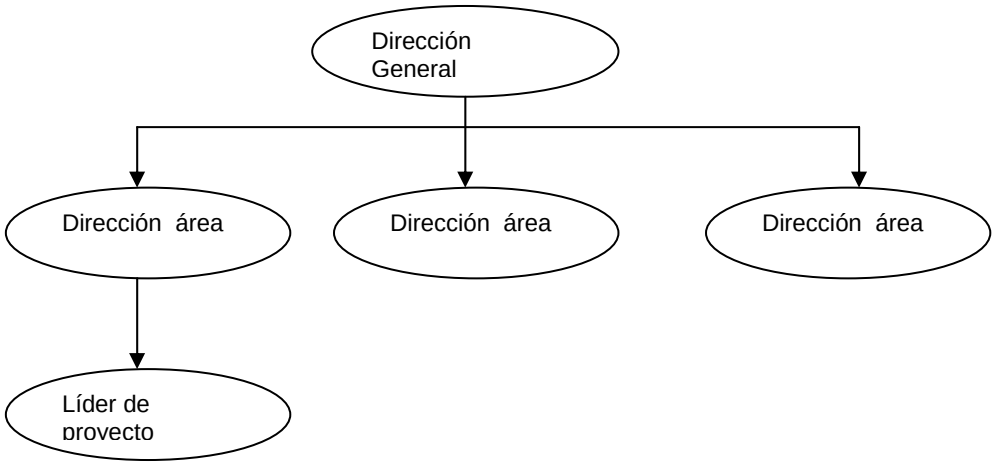


Figura 1.1 *Organigrama jerárquico vertical*

1.1.5 ANÁLISIS DE PUESTOS

Esta técnica estudia en detalle las organizaciones para las cuales se realiza el trabajo de reclutamiento y selección de personal, y tiene como finalidad conocer lo que cada trabajador hace y las aptitudes que requiere para desempeñar de manera eficiente el puesto. Dessler (1994) define el análisis de puestos como “el procedimiento para determinar las obligaciones y habilidades requeridas por un puesto de trabajo, así como el tipo de individuo idóneo para ocuparlo”. El puesto de trabajo determina en gran medida el rol que las personas juegan en las organizaciones. En términos generales un puesto se puede definir como un conjunto de operaciones, cualidades, responsabilidades y condiciones que forman una unidad de trabajo específica.

Para la empresa, es la única forma de diseñar un organigrama representativo con estructura, metas, objetivos y sin duplicación de funciones. El concepto de puesto implica dos elementos esenciales, que son:

1. Su contenido, determinado por el conjunto de funciones concretas.
2. Sus requisitos, determinados por los factores de habilidad, esfuerzo, responsabilidad y condiciones de trabajo.

El análisis de los puestos de trabajo es un proceso objetivo, en la medida en que no tiene en consideración a la persona que ocupa el puesto de trabajo, sino al puesto en sí (Chiavenato, 1988).

En varias ocasiones, la experiencia permite observar que las personas crean y dan vida e importancia al puesto, y gracias a la capacidad y habilidad el individuo puede:

- Apegarse al puesto
- Puede crecer al puesto hasta redimensionarlo.

Es muy importante tener claro que existe una gran diferencia entre una descripción de puesto y el análisis de puestos. La descripción es un resumen de las principales responsabilidades, funciones y actividades del puesto. El análisis es un estudio detallado del puesto para determinar el grado en que se da cada uno de los factores de valuación. La descripción, en consecuencia, es una herramienta útil para analizar un puesto, pero no es el análisis de puesto.

1.1.5.1 ACTIVIDADES INVOLUCRADAS EN EL ANÁLISIS DE PUESTOS

Los pasos involucrados en el proceso de análisis incluyen las siguientes actividades:

- Aprobación por parte de la Gerencia
- Fijar objetivos

- Establecimiento de un programa de acción
- Seleccionar al personal que realizará el análisis
- Entrenar a los analistas
- Definir métodos de trabajo y diseñar instrumentos de recopilación de datos
- Selección de los titulares de puesto, de quienes se obtendrá la información

1.1.6 RECLUTAMIENTO DE PERSONAL

Una vez analizado el aspecto de la estructura de una empresa y el análisis de puestos, el psicólogo podrá proceder a buscar los elementos humanos que requiere la empresa, utilizando para ello la técnica o método que más convenga a sus necesidades.

Antes de mencionar los métodos, se definirá qué es ***Reclutamiento***: es la técnica encaminada a proveer de recursos humanos a la empresa u organización en el momento oportuno.

En el caso de una empresa ya establecida, con una estructura definida, podría plantearse la siguiente pregunta: ¿cuándo se necesita un reclutado? Se puede requerir en dos casos:

- Cuando es un puesto de nueva creación
- Cuando se tiene un puesto vacante

El psicólogo reclutador tendrá que recurrir a lugares específicos, o bien utilizar medios de difusión o comunicación para obtenerlos. A estos lugares se les denomina fuentes de reclutamiento, y a los medios de comunicación o difusión, como su nombre lo indica, medios de reclutamiento.

Los aspectos más importantes en el proceso de la obtención de recursos humanos son los que se presentan en seguida.

1.1.6.1 FUENTES DE RECLUTAMIENTO

Existen dos tipos de fuentes de reclutamiento, las internas y las externas.

A) FUENTES INTERNAS

Son aquellas que sin necesidad de recurrir a personas o lugares fuera de la empresa, proporcionan el personal requerido en el momento oportuno, entre ellas se encuentran:

- **Sindicatos.** Funciona en el momento que la empresa requiera personal para una vacante o vacantes que entran en el dominio del sindicato.

- **Archivo o cartera de personal.** Se integra con las solicitudes de los candidatos que se presentaron en periodos de reclutamiento anteriores y que fueron archivados, probablemente en una base de datos computarizada, sin haberse utilizado por que se cubrió la vacante en cuestión. La desventaja es que es probable que esta base de datos no se encuentre actualizada desde la última vez que se consultó y podría ser que los datos ya no coincidan con la persona o que la misma ya no esté disponible.

- **Familiares y recomendados.** Se hace uso de esta fuente cuando se boletinan las vacantes entre los propios trabajadores, si es que las políticas de la empresa lo permiten.

- **Promoción o transferencia interna de personal.** Sucede cuando se estudia las plantillas o la historia laboral del personal, con el objetivo de determinar si existe o no un candidato viable para ser promovido al puesto vacante.

B) FUENTES EXTERNAS

Son aquellas a las que el psicólogo puede recurrir, de acuerdo con las características requeridas en cuanto al aspecto profesional, educativo o técnico, y que son totalmente ajenas a la empresa, algunas son las siguientes:

- **Profesionales y educativas.** Son instituciones que se dedican a la preparación de profesionales en las diferentes áreas científicas y tecnológicas, y pueden ser universidades, escuelas comerciales, profesionales, tecnológicos, institutos, etc. Existen también, dentro de algunas instituciones educativas, ferias de reclutamiento, reclutamiento de amateurs, congresos y seminarios.

- **Asociaciones profesionales.** Cuando se requiere personal con experiencia en determinado campo científico, tecnológico o profesional, se puede tener contacto con asociaciones profesionales, como: Asociación de Ingenieros Mecánicos Electricistas, Asociación Nacional de Ingenieros Químicos, Colegio de Contadores, etc.

- **Bolsas de Trabajo.** Son organizaciones que se dedican a proporcionar información con respecto a las vacantes en diversas empresas o centros de trabajo, y que prestan sus servicios de manera gratuita para el trabajador, la empresa puede recurrir a ellas para solicitar candidatos.

- **Agencias de colocación.** Estas agencias no cobran al candidato, sino a la empresa. La tarifa consiste en un mes de salario de la persona que colocan. Cuando el empleo es a nivel ejecutivo, cobran desde 10 hasta 30% del salario anual integrado.

- **Out-sourcing.** Se trata de empresas que no solo facilitan la obtención de personal especializado en puestos de alta tecnología, difíciles de conseguir; sino que ahora se están utilizando como sustitutos patronales mientras prueban la pericia y actitudes de los empleados o como una manera de evitar tener relación laboral con los mismos y así aparentar tener menos gente en las plantillas.

Otra modalidad es la de reclutar jóvenes que están en condiciones de realizar sus prácticas profesionales o servicio social.

- **La puerta de la calle.** Ocurre cuando un candidato se presenta sin que haya existido algún aviso para él por parte de la empresa; es decir, el candidato acude de manera espontánea. Por ejemplo las instituciones bancarias, empresas comerciales y empresas departamentales, entre otras.

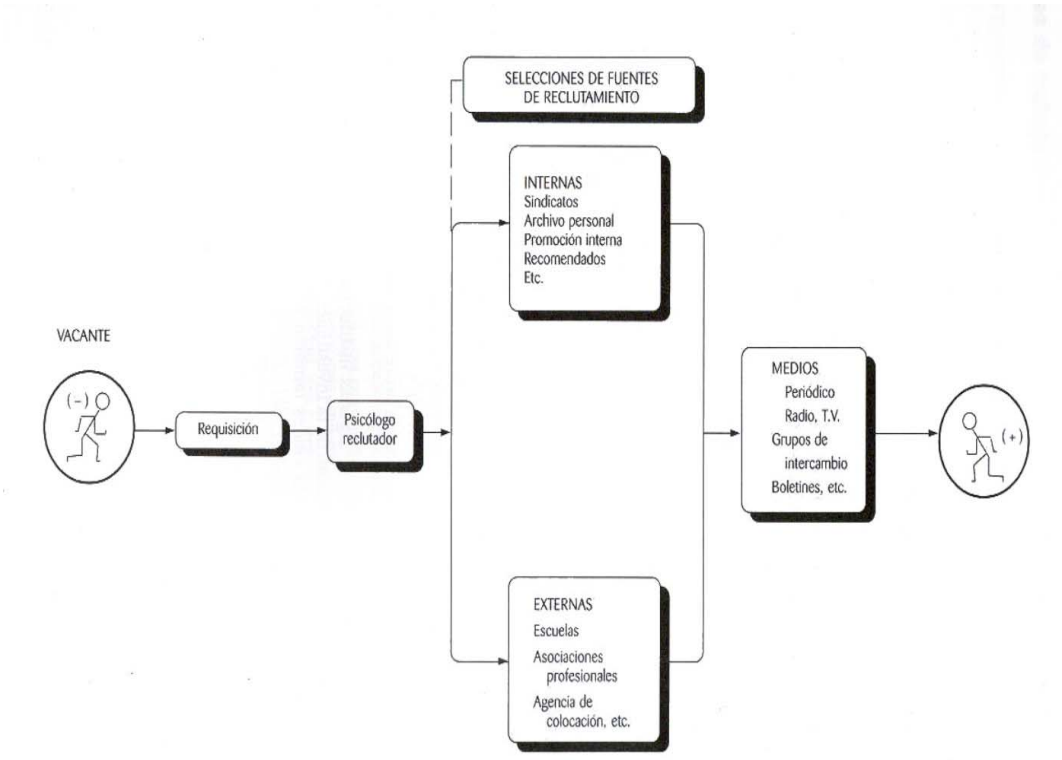


Figura 1.2 *Fuentes de reclutamiento*

1.1.6.2 MEDIOS DE RECLUTAMIENTO

Son aquellos medios de difusión que el psicólogo utiliza para dar a conocer un puesto vacante. Estos medios son de gran utilidad por su potencialidad de difusión, pues penetran en distintos ambientes. Son capaces de transmitir información tanto a estructuras socioeconómicas bajas o

altas, y también a élites profesionales y tecnológicas, organizaciones de mano de obra o de personal altamente calificado. Los principales medios son:

- **Medios impresos.** La prensa es el medio más utilizado en reclutamiento de personal. Se emplean anuncios en primera plana o en anuncios interiores que tienen la ventaja de ser los más económicos. La información presentada en dichos anuncios debe contener como mínimo: puesto vacante, escolaridad requerida, experiencia requerida, características personales específicas, prestaciones, dirección, teléfono, a donde dirigirse.

- **Radio y Televisión.** El alto costo de estos medios no permite que se utilicen con frecuencia. Se puede emplear la radio cuando el reclutamiento sea en provincia mediante la radiodifusora de la localidad. Los anuncios de televisión tienen un costo elevadísimo.

- **Grupos de intercambio**. Se constituyen a través de los encargados de reclutamiento de diferentes empresas, los cuales proporcionan información sobre las diferentes vacantes de la organización. Se boletinan candidatos viables de ocupar vacantes en otras organizaciones cuando no se tiene algún puesto que ofrecer de acuerdo con sus características. Tiene la ventaja de poder conseguir recursos humanos de modo inmediato y sin costo.

- **Reclutamiento ON-LINE (vía INTERNET).** La Internet es una nueva forma de compartir información de cualquier tipo. Los beneficios de reclutar a través de este medio son: reducción de los costos de reclutamiento de la empresa cliente, acceso a una base de datos digital, reducción del tiempo de búsqueda y acceso las 24 horas del día, los 365 días del año.

1.1.7 SELECCIÓN DE PERSONAL

La importancia de una buena selección radica en tener las mejores expectativas de un trabajador en el desempeño de sus actividades dentro de la empresa, la evaluación debe realizarse de manera secuencial y precisa, contemplando los siguientes aspectos:

- Experiencia laboral
- Trayectoria académica
- Aspectos psicológicos

Estos últimos son componentes básicos de la personalidad de un individuo que hacen posible su óptimo desarrollo dentro de una sociedad y específicamente en su ámbito laboral. Puede decirse entonces, que la selección es la serie de técnicas encaminadas a encontrar a la persona adecuada para el puesto adecuado.

Se habla de selección de personal en el momento en que se tienen los candidatos probables para ocupar la vacante; éstos deberán ser evaluados de modo secuencial.

1.1.7.1 ELABORACIÓN DE PERFIL

El psicólogo al recibir una requisición, deberá elaborar un perfil psicológico de acuerdo con las necesidades de la empresa. Cuando se tramita una requisición al departamento de personal, deben estar claros los datos del candidato.

Las características intelectuales estarán determinadas por los factores de capacidad, en términos de análisis, síntesis, nivel de pensamiento, integración, organización, etc., que el puesto requiera. Las características de habilidades se refieren a condiciones técnicas determinadas por el tipo de labor a desempeñar. Las de personalidad se refieren a los factores internos como motivación, intereses, estabilidad emocional, juicio, etc.

1.1.7.2 SOLICITUD

Es un cuestionario debidamente estructurado, que deberá contener los datos relevantes del candidato bajo los siguientes rubros:

A) DATOS PERSONALES

Son aquellos que proporcionan conocimientos específicos de la persona, como son: nombre, edad, sexo, estado civil, dirección, teléfono, etc.

B) DATOS ESCOLARES

Son los relacionados con la preparación escolar del candidato, desde su inicio hasta su fin; si cursó la primaria, secundaria, bachillerato, profesional; donde, cuado, en que escuela, etc.

C) EXPERIENCIA OCUPACIONAL

Proporciona una visión general de la experiencia del candidato en el desarrollo de una actividad profesional técnica o de mano de obra; que tiempo la ha desempeñado, que otro tipo de actividades ha realizado, donde ha laborado, causas de la renuncia, nivel ocupacional, etc.

D) DATOS FAMILIARES

Son aquellos que muestran la situación del candidato en su núcleo familiar, como son: numero de hermanos, actividades y escolaridad de ellos, posición que ocupa entre ellos, contribución económica, etc.

E) REFERENCIAS PERSONALES

Son aquellas personas que pueden proporcionar información acerca del candidato, como es el tiempo que tienen de conocerlo, además de identificar sus hábitos y manera de proceder en su interacción con la gente que le rodea. De preferencia se pide que no sean familiares.

F) GENERALES

Son datos que pueden ser de utilidad para la empresa y que van de acuerdo con las políticas, necesidades, etc., de la misma. Algunas empresas piden, por ejemplo, que el candidato redacte una pequeña carta en la que solicite empleo a dicha compañía.

El diseño de la solicitud esta determinado por las políticas y necesidades de la empresa, y puede ser de diversas formas, tamaños, etc.

1.1.7.3 EVALUACIÓN TÉCNICA

Todo puesto requiere de ciertos conocimientos o pautas conductuales para ser desempeñado de manera adecuada .A estos conocimientos, adquiridos en algún plantel educativo ó institución, se le denomina preparación académica. Los candidatos recién egresados de algún centro profesional no han adquirido dicha experiencia, pero si tienen conocimientos que en un momento determinado puedan aplicar en el desempeño de una labor.

El psicólogo, dentro del proceso de selección, evalúa dichos conocimientos mediante los instrumentos adecuados que el diseña.

Es importante subrayar que la evaluación técnica debe realizarse antes que la psicológica, para que la empresa ahorre tiempo y costos.

1.1.7.4 EVALUACIÓN PSICOLÓGICA

En ella se reúnen los datos de capacidad intelectual y emocional del candidato. Se realiza por medio de baterías psicológicas que deben seleccionarse tomando en consideración los siguientes aspectos: Nivel de aplicación, características del perfil, tiempo de aplicación y costo.

Los tres niveles más importantes en el proceso de reclutamiento y selección son los siguientes:

- Nivel obrero u operativo
- Nivel medio o mandos intermedios
- Nivel ejecutivo o gerencial

Una batería de pruebas psicológicas incluye los siguientes aspectos básicos:

INTELIGENCIA Se refiere a la medición de la capacidad para resolver situaciones del entorno a través de los siguientes factores:

- Factor general de la inteligencia (factor g)
- Factores específicos de la inteligencia, como son análisis y síntesis, nivel de pensamiento, etc.

***HABILIDAD*.** Se refiere a la medición de las capacidades más desarrolladas como la comunicación oral o escrita, ventas, etc.

PERSONALIDAD. Medición de los factores del estilo de afrontamiento a la vida de la persona.

1.1.7.5 EVALUACIÓN SOCIOECONOMICA

Permite verificar de manera directa y objetiva las condiciones en las que se desenvuelve el solicitante. En ella se confrontan los datos proporcionados por el candidato en la solicitud y durante la entrevista inicial.

Es de suma importancia conocer el nivel económico del solicitante así como sus condiciones de vida, sus relaciones familiares, las referencias personales y las de trabajo.

La información socioeconómica generalmente se obtiene por medio de un cuestionario aplicado por una trabajadora social, por el personal del área de recursos humanos de la empresa, o bien, como se hace en la mayor parte de los casos, a través de despachos u organizaciones especializados en esta actividad.

1.1.7.6 ENTREVISTA EN LA SELECCIÓN

La entrevista es una forma estructurada de comunicación interpersonal, generalmente entre dos personas (entrevistador y entrevistado), debidamente planeada, con un objetivo determinado y con la finalidad de obtener información relevante para tomar decisiones benéficas para ambas partes.

La finalidad de la entrevista dentro del proceso de selección de personal es valorar los datos obtenidos en los pasos anteriores; es decir, a través del currículum vital, la solicitud, los exámenes psicológicos, etc. Dicha información debe ser ordenada y jerarquizada para determinar qué áreas quedan poco claras. Además, sirve para profundizar en información relevante para el puesto en cuestión, la cual muchas veces no es posible obtener sino mediante una entrevista adecuada. Asimismo, sirve para evaluar el grado de probabilidad de que el candidato se adapte y funcione de modo adecuado dentro de las condiciones relacionadas con el puesto que se requiere cubrir.

Resulta de gran ayuda contar con una guía de entrevista, como un apoyo para evitar la omisión de áreas importantes para garantizar una secuencia lógica de la misma.

TIPOS DE ENTREVISTA

Existen diferentes términos para llamar a los tipos de entrevista, entre ellos se encuentran:

- Entrevista directa
- Entrevista indirecta
- Entrevista mixta
- Entrevista Acertada

A) ENTREVISTA DIRECTA

Es aquella en la que el entrevistador tiene mayor actuación o desempeña más actividades. Se realizan preguntas de acuerdo con un interrogatorio previamente diseñado, dichas preguntas van encaminadas a obtener información sobre áreas más específicas. Además sirve como filtro para ver qué candidatos cubren los requisitos mínimos para el puesto y, por tanto, pueden continuar la siguiente fase del proceso de selección.

B) ENTREVISTA INDIRECTA

El entrevistado es el que toma la parte más activa, ya que el entrevistador formula preguntas abiertas y hasta cierto punto indefinidas, con el objetivo de señalar sólo el área que interesa y dejar a aquel la iniciativa de que hable de la forma que desee.

El entrevistador no sólo tiene que atender a las respuestas verbales del individuo, sino también observar de manera cuidadosa todas las reacciones que tenga el entrevistado ante las preguntas que se le formulan. Por reacciones se entienden los movimientos de las diferentes partes del cuerpo, gestos, cambios de tono de voz o silencios excesivos, así como el grado de coherencia entre las expresiones corporales y las verbales.

C) ENTREVISTA MIXTA

Es una combinación de la entrevista directa y la indirecta. Al inicio se hacen preguntas directas pero conforme se va desarrollando un clima propicio se le va dando la pauta al entrevistado.

Durante el curso de la entrevista el psicólogo deberá ser lo suficientemente hábil para no dejar caer el clima de confianza establecido en un principio. Esto puede lograrse de diferentes maneras; es decir, existen ciertas tácticas de la entrevista que permiten conservar la atención e interés del solicitante a lo largo de ella. Entre estas tácticas se encuentran las de agrado-desagrado, confrontación, presión, eco, por qué, silencio, por mencionar algunas.

D) ENTREVISTA ACERTADA

Una de las modalidades es la entrevista acertada o dimensional, que a lo largo de los años se ha utilizado cada vez con mayor frecuencia dentro del ámbito laboral.

El proceso de entrevista acertada permite al ejecutivo tomar decisiones más adecuadas en la selección del personal, pudiéndose dividir en cuatro etapas:

1. Tratar de identificar el comportamiento pasado del candidato para predecir su conducta futura.

2. Identificar los requisitos más significativos que el puesto exige.

3. Utilizar técnicas y habilidades eficaces de la entrevista.

4. Intercambiar opiniones con otros entrevistadores en relación con el candidato.

ERRORES MÁS COMUNES DURANTE LA ENTREVISTA

El error más conocido es el efecto de halo, es decir, la tendencia a sentir simpatía o antipatía por algún tipo de persona, sin que exista una justificación objetiva para ello.

Otro error es hacer entrevistas prolongadas, es decir, con más tiempo del fijado de antemano.

Otra equivocación que se puede cometer es la falta de interés en algunos aspectos o momentos de la entrevista, generando un cúmulo de datos cuantitativos y un descuido total de los cualitativos. Cada entrevistado es diferente por lo que cada entrevista debe ser tratada como única, individual y personal.

Otros errores son el no aclarar en el momento la información que se está recibiendo, hacer más de una pregunta a la vez, interrumpir al entrevistado y exagerar la toma de notas durante a entrevista.

1.1.7.7 EXAMEN MEDICO

Se utiliza para conocer si el candidato posee la capacidad física para desempeñar su trabajo sin consecuencias negativas para él o para las personas que lo rodean. En caso de encontrar anomalías, se le orientará para corregirlas o para realizar exámenes periódicos y prevenir consecuencias más graves.

En ocasiones, los resultados del examen médico condicionan el ingreso de una persona a la organización, ya que ésta última no podrá ingresar, hasta que no solucione por cuenta propia aquellos problemas que pudieron haber sido detectados en el examen medico.

1.1.7.8 CONCLUSIÓN

Esta parte integra los datos obtenidos a través de todo el proceso de selección, analizándose todas las características del aspirante, comparándolas siempre con el perfil de puesto, de tal modo que pueda llegarse a la emisión de un juicio valorativo sobre la aceptación o no aceptación del aspirante.

Existen tres modos de presentar las conclusiones:

a) Conclusión abierta

b) Conclusión cerrada

c) Conclusión mixta

1.1.7.9 DECISIÓN

Constituye el hecho de aceptar o rechazar al candidato. Depende de la organización de la empresa, la asignación de la persona que toma esta decisión final; podría ser el jefe directo, el jefe de personal o inclusive el dueño de la misma.

1.1.8 ADMINISTRACIÓN DE PERSONAL

1.1.8.1 DEFINICIÓN

La Administración de personal es el proceso de Admón. Aplicado al acrecentamiento y conservación del esfuerzo, la experiencia, la salud, los conocimientos, las habilidades, etc. de los miembros de la organización, en beneficio del individuo, de la propia organización y del país en general. Consiste en desarrollar y administrar políticas, programas y procedimientos para prever una estructura administrativa eficiente, trabajadores capaces, oportunidades de progreso, satisfacción en el trabajo y una adecuada seguridad en sí mismo.

[2]Granados, Jaime A (2004). Reclutamiento, selección, contratación de personal. México, El manual moderno.

1.1.8.2 OBJETIVOS

Los objetivos son los parámetros que miden las acciones que lleva a cabo la persona encargada del personal dentro de la empresa. En el área de Personal, los objetivos de mayor importancia son los siguientes:

- Contribuir al mejoramiento de la productividad interna
- Ofrecer un mayor bienestar a los empleados
- Diseñar las funciones básicas a desempeñar

Los objetivos forman en conjunto la base de un sólido departamento de personal. La mayoría de negocios busca que su organización sea productiva y para lograr este propósito, deben contar con un personal altamente motivado. De esta manera, las tareas se desarrollarán con esmero y eficacia.

La primera tarea para el empresario deberá ser la identificación clara de los puestos claves, antes de colocar a la gente que tendrá que desempeñarlos. De esta manera se facilitará la selección y contratación posterior del personal más idóneo para cada uno.

1.1.8.3 PRINCIPALES FUNCIONES

Entre las principales funciones de la administración de personal se encuentran:

1. *Planeación de personal:* Determina las necesidades de personal en la empresa, determina los objetivos, políticas, procedimientos y programas de administración de personal dentro de la empresa. Consiste en realizar estudios tendientes a la proyección de la estructura de la organización en el futuro, incluyendo análisis de puestos proyectados y estudio de las posibilidades de desarrollo de los trabajadores para ocupar estas, a fin de determinar programas de capacitación y desarrollo, llegado el caso de reclutamiento y selección.
2. *Empleo (reclutamiento, selección, contratación e inducción*)**:** Lograr que todos los puestos sean cubiertos por personal idóneo, de acuerdo a una adecuada planeación de recursos humanos.

    a. Reclutamiento: buscar y atraer solicitantes capaces para cubrir las vacantes que se presenten: Técnica encaminada a proveer de recursos humanos a la empresa u organización en el momento oportuno.

    b. Selección: Analizar las habilidades y capacidades de los solicitantes a fin de decidir, sobre bases objetivas, cuál tiene mayor potencial para el desempeño de un puesto y posibilidades de un desarrollo futuro, tanto personal como de la organización. Proceso que trata no solamente de aceptar o rechazar

candidatos, sino conocer sus aptitudes y cualidades con objeto de colocarlo en el puesto más fin a sus características.

c. Contratación: Formalizar con apego a la Ley la futura relación de trabajo para garantizar los intereses, derechos y deberes tanto del trabajador como de la empresa.

d. Inducción: Dar toda la información necesaria al nuevo trabajador y realizar todas las actividades pertinentes para lograr su rápida incorporación a los grupos sociales que existan en su medio de trabajo, a fin de lograr una identificación entre el nuevo miembro y la organización. Consiste en llevar al individuo al puesto que va a ocupar, presentarlo con su superior y compañeros con el objeto de lograr una adaptación de grupo que evite una baja en el rendimiento, que obtenga una visión de la empresa. Así mismo se le mostraran las instalaciones de la empresa y principalmente de su área de trabajo.

3. Capacitación y desarrollo**:** Tiene por objeto ampliar, desarrollar y perfeccionar al hombre para su crecimiento profesional en determinado puesto en la empresa o para estimular su eficiencia y productividad. Debe basarse en el análisis de necesidades que parta de una comparación del desempeño y la conducta actual con la conducta y desempeño que se desean. Con base a este análisis, se identifican los métodos y necesidades de capacitación para superar las delicias.

4. Administración de sueldos y salarios: Parte de la Admón., de personal que estudia los principios o técnicas para lograr que la remuneración global que recibe el trabajador sea adecuado a la importancia de su puesto, a su eficiencia personal, a sus necesidades y a las posibilidades de la empresa. Consiste en asignar valores monetarios a los puestos, en tal forma que sean justos y equitativos en relación a otras posiciones de la organización y a puestos similares en el mercado de trabajo.

5. Prestaciones y servicio de personal. Son todas aquellas actividades que realiza la empresa enfocadas a proporcionar al trabajador un beneficio, ya sea en dinero o en especie. Satisfacer las necesidades de los trabajadores que laboran en la organización y tratar de ayudarles en los problemas relacionados a su seguridad y bienestar personal.

1.1.8.4 PRINCIPIOS DE LA SELECCIÓN DE PERSONAL

Colocación: Parte de la tarea del seleccionador es tratar de incrementar los recursos humanos de la organización por medio del descubrimiento de habilidades como actitudes que puedan aprovechar los candidatos en su propio beneficio y en el de la organización. Así pues, si un candidato no tiene las habilidades necesarias para un determinado puesto, pero se le considera parcialmente un buen prospecto por otras características personales, es necesario descubrir otras habilidades, las cuales puedan ser requeridas en otra parte de la organización o en otra ocupación dentro de la misma.

Orientación: Tradicionalmente se ha considerado a la organización como un sistema aislado de su medio. Por tanto si un candidato no era aceptado, simplemente se rechazaba; pero no hay que olvidar que la organización se encuentra dentro de un sistema económico, social, cultural, político, etc., y que por ello cumplirá con sus objetivos sociales si ayuda a resolver los problemas del país. Es necesario considerar la labor del seleccionador ya no como limitada por las fronteras de la organización, sino en un sentido mas amplio y teniendo como marco de referencia los problemas de su ocupación y desempleo en el país. Entonces, en caso de que no sea posible aceptar un candidato es importante orientarlo; es decir; dirigirlo hacia otras fuentes de empleo.

Ética profesional: El proceso de selección de personal implica una serie de decisiones que pueden afectar a la vida futura del candidato. Si no es aceptado, si se le coloca en un puesto para el cual no tiene habilidades; para el cual tiene más capacidad de la necesaria,
etcétera, son circunstancias que pueden convertirse en fuente de frustraciones para el candidato y que, por tanto, pueden minar su salud mental y la de la gente que los rodea.

1.2 TEORÍA SOBRE LENGUAJES DE PROGRAMACIÓN

1.2.1 DEFINICIÓN

Los lenguajes son sistemas de comunicación. Un lenguaje de programación consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas "comunicarse" con las computadoras. Existen por lo menos varios cientos de lenguajes y dialectos de programación diferentes. Algunos se crean para una aplicación especial, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones. En todo caso los lenguajes de programación deben tener instrucciones que pertenecen a las categorías ya familiares de entrada/salida, cálculo/manipulación de textos, lógica/comparación, almacenamiento y recuperación.

No obstante, aunque todos los lenguajes de programación tienen un conjunto de instrucciones que permiten realizar dichas operaciones, existe una marcada diferencia en los símbolos, caracteres y sintaxis de los lenguajes de máquina, lenguajes ensambladores y lenguajes de alto nivel.

1.2.2 ANTECEDENTES

Al desarrollarse las primeras computadoras electrónicas, se vio la necesidad de programarlas, es decir, de almacenar en memoria la información sobre la tarea que iban a ejecutar. Las primeras se usaban como calculadoras simples; se les indicaban los pasos de cálculo, uno por uno.

John Von Neumann desarrolló el modelo que lleva su nombre, para describir este concepto de "programa almacenado". En este modelo, se tiene una abstracción de la memoria como un conjunto de celdas, que almacenan simplemente números. Estos números pueden representar dos cosas: los datos, sobre los que va a trabajar el programa; o bien, el programa en sí.

¿Cómo es que describimos un programa como números? Se tenía el problema de representar las acciones que iba a realizar la computadora, y que la memoria, al estar compuesta por switches correspondientes al concepto de bit, solamente nos permitía almacenar números binarios.

La solución que se tomó fue la siguiente: a cada acción que sea capaz de realizar nuestra computadora, asociarle un número, que será su código de operación (opcode). Por ejemplo, una calculadora programable simple podría asignar los opcodes:

1 = SUMA, 2 = RESTA, 3 = MULTIPLICA, 4 = DIVIDE.

Podemos ver que con esta representación, es simple expresar las operaciones de las que es capaz el hardware (en este caso, nuestra calculadora imaginaria), en la memoria.

La descripción y uso de los opcodes es lo que llamamos lenguaje de máquina. Es decir, la lista de códigos que la máquina va a interpretar como instrucciones, describe las capacidades de programación que tenemos de ella; es el lenguaje más primitivo, depende directamente del hardware, y requiere del programador que conozca el funcionamiento de la máquina al más bajo nivel.

Los lenguajes más primitivos fueron los lenguajes de máquina. Esto, ya que el hardware se desarrolló antes del software, y además cualquier software finalmente tiene que expresarse en el lenguaje que maneja el hardware.

La programación en esos momentos era sumamente tediosa, pues el programador tenía que "bajarse" al nivel de la máquina y decirle, paso a pasito, cada punto de la tarea que tenía que realizar. Además, debía expresarlo en forma numérica; y por supuesto, este proceso era propenso a errores, con lo que la productividad del programador era muy limitada. Sin embargo, hay que recordar que en estos momentos, simplemente aún no existía alternativa.

El primer gran avance que se dio, como ya se comentó, fue la abstracción dada por el Lenguaje Ensamblador, y con él, el nacimiento de las primeras herramientas automáticas para generar el código máquina. Esto redujo los errores triviales, como podía ser el número que correspondía a una operación, que son sumamente engorrosos y difíciles de detectar, pero fáciles de cometer. Sin embargo, aún aquí es fácil para el programador perderse y cometer errores de lógica, pues debe bajar al nivel de la forma en que trabaja el CPU, y entender bien todo lo que sucede dentro de él.

Con el desarrollo en los 50s y 60s de algoritmos de más elevado nivel, y el aumento de poder del hardware, empezaron a entrar al uso de computadoras científicos de otras ramas; ellos conocían mucho de Física, Química y otras ramas similares, pero no de Computación, y por supuesto, les era sumamente complicado trabajar con lenguaje ensamblador en vez de fórmulas. Así, nació el concepto de Lenguaje de Alto Nivel, con el primer compilador de FORTRAN (FORmula TRANslation), que, como su nombre indica, inició como un "simple" esfuerzo de traducir un lenguaje de fórmulas, al lenguaje ensamblador y por consiguiente al lenguaje de máquina. A partir de FORTRAN, se han desarrollado innumerables lenguajes, que siguen el mismo concepto: buscar la mayor abstracción posible, y facilitar la vida al programador, aumentando la productividad, encargándose los compiladores o intérpretes de traducir el lenguaje de alto nivel, al lenguaje de computadora.

Hay que notar la existencia de lenguajes que combinan características de los de alto nivel y los de bajo nivel (es decir, Ensamblador). Como ejemplo podemos mencionar el lenguaje C: contiene estructuras de programación de alto nivel, y la facilidad de usar librerías que también son características de alto nivel; sin embargo, fue diseñado con muy pocas instrucciones, las cuales son sumamente sencillas, fáciles de traducir al lenguaje de la máquina; y requiere de un entendimiento apropiado de cómo funciona la máquina, el uso de la memoria, etcétera. Por ello, muchas personas consideramos a lenguajes como C (que fue diseñado para hacer sistemas operativos), lenguajes de nivel medio.

1.2.3 TIPOS DE LENGUAJES

a) Lenguaje de Máquina

El lenguaje de máquina de una computadora consta de cadenas de números binarios (ceros y unos) y es el único que "entienden" directamente los procesadores. Todas las instrucciones preparadas en cualquier lenguaje de máquina tienen por lo menos dos partes. La primera es el comando u operación, que dice a la computadora cuál es la función que va a realizar.

La segunda parte de la instrucción es el operando, que indica a la computadora donde hallar o almacenar los datos y otras instrucciones que se van a manipular; el número de operandos de una instrucción varía en las distintas computadoras.

En el principio de la computación este era el lenguaje que tenía que "hablar" el ser humano con la computadora y consistía en insertar en un tablero miles de conexiones y alambres y encender y apagar interruptores.

Aunque en la actualidad ya no se emplea, es importante reconocer que ya no es necesario que nos comuniquemos en este lenguaje de "unos" y "ceros", pero es el que internamente una computadora reconoce o "habla".

b) Lenguaje Ensamblador

A principios de la década de 1950, y con el fin de facilitar la labor de los programadores, se desarrollaron códigos mnemotécnicos para las operaciones y direcciones simbólicas. Uno de los primeros pasos para mejorar el proceso de preparación de programas fue sustituir los

códigos de operación numéricos del lenguaje de máquina por símbolos alfabéticos, que conforman un lenguaje mnemotécnico. Todas las computadoras actuales tienen códigos mnemotécnicos aunque, naturalmente, los símbolos que se usan varían en las diferentes marcas y modelos. La computadora sigue utilizando el lenguaje de máquina para procesar los datos, pero los programas ensambladores traducen antes los símbolos de código de operación especificados a sus equivalentes en lenguaje de máquina.

Los lenguajes ensambladores tienen ventajas sobre los lenguajes de máquina. Ahorran tiempo y requieren menos atención a detalles. Se incurren en menos errores y los que se cometen son más fáciles de localizar. Además, los programas en lenguaje ensamblador son más fáciles de modificar que los programas en lenguaje de máquina. Pero existen limitaciones. La codificación en lenguaje ensamblador es todavía un proceso lento. Además, una desventaja importante de estos lenguajes es que tienen una orientación a la máquina. Es decir, están diseñados para la marca y modelo específico de procesador que se utiliza.

c) Lenguajes de Alto Nivel

Los primeros programas ensambladores producían sólo una instrucción en lenguaje de máquina por cada instrucción del programa fuente. Para agilizar la codificación, se desarrollaron programas ensambladores que podían producir una cantidad variable de instrucciones en lenguaje de máquina por cada instrucción del programa fuente. Dicho de otra manera, una sola macroinstrucción podía producir varias líneas de código en lenguaje de máquina.

El desarrollo de las técnicas mnemotécnicas y las macroinstrucciones condujo, a su vez, al desarrollo de lenguajes de alto nivel que a menudo están orientados hacia una clase determinada de problemas de proceso.

A diferencia de los programas de ensamble, los programas en lenguaje de alto nivel se pueden utilizar con diferentes marcas de computadoras sin tener que hacer modificaciones considerables. Esto permite reducir sustancialmente el costo de la reprogramación cuando se adquiere equipo nuevo. Otras ventajas de los lenguajes de alto nivel son:

- Son más fáciles de aprender que los lenguajes ensambladores.
- Se pueden escribir más rápidamente.
- Permiten mejor documentación.

1.2.4 ESTILOS DE PROGRAMACIÓN

Los lenguajes de programación pueden ser muchos y también pueden ser tomados de diferentes perspectivas. Un programador debe saber que conceptos utilizar u omitir al momento de programar. Por lo tanto es importante tener claro las características o estilos de programación para lograr determinar cual es la herramienta o lenguaje ideal según las características del sistema a implementar .Existen 4 estilos de programación fuertemente conocidos:

- Programación Imperativa/Secuencial/Estructurada
- Programación Orientado al Objeto
- Programación Orientada al Evento
- Programación Funcional

a) Programación Imperativa

Esta basada en el modelo Von Neumann, en donde un conjunto de operaciones primitivas realizan una ejecución secuencial. Realiza una abstracción en el manejo de variables, expresiones e instrucciones y para programar es necesario declarar las variables necesarias y diseñar una secuencia adecuada de instrucciones (asignaciones).

b) Programación Orientada al Objeto (POO)

La programación orientada al objeto esta basada en los objetos, clase, método, envió y recepción de mensajes, herencia y polimorfismo. En donde:

Objetos: es cualquier cosa que se ofrece a la vista y afecta a los sentidos. También se define como una entidad tangible que exhibe algún comportamiento bien definido. En términos de programación, un objeto no necesariamente es algo tangible (por ejemplo: un proceso). Lo que sí puede decirse de todo objeto es que tiene estado, comportamiento e identidad.

Clases: Una clase es una colección de datos y métodos que operan sobre los datos y sirven para definir el contenido y capacidades de algunos objetos.
Encapsulación: Modularidad y ocultamiento de información (datos y métodos).

Polimorfismo: es la cualidad que poseen los objetos para responder en distinto modo ante el mismo mensaje.

Herencia: la herencia permite definir nuevas clases y comportamientos basados en clases.

Algunos de los lenguajes de este tipo son C++ y Java

c) Programación Orientada Al Evento

Esta programación es el resultado de la programación orientada al objeto. En este tipo de programación permite trabajar con objetos y clases Standard previamente definidas por la aplicación, las cuales manejan los conceptos de encapsulación. Las herramientas que trabajan de esta forma por lo general se trabajan con código original de lenguajes imperativos. Algunas herramientas de este tipo son Visual Basic (Basic), Delphi (Pascal) y Power Builder (C).

d) Programación Funcional

El Programa es una Función (o un grupo de funciones). Usualmente compuesto de Funciones más simples. La relación entre las funciones es muy simple:
Una Función puede llamar a otra Función, o el resultado de una Función puede ser usado como el argumento de otra Función.

Las variables, comandos y efectos laterales son exclusivos. Los programas son escritos enteramente dentro del lenguaje de expresiones, funciones y declaraciones.

Ejemplo: Función factorial en ML
fun Factorial loop(n,f)= if n>0 then factorial loop(n-1,f*n)
Dos de estos lenguajes son Scheme y ML.

[3]Levine Guillermo (1989). Introducción a la computación y a la programación estructurada. México Mc Graw Hill

1.2.5 TENDENCIAS

Los lenguajes de programación tenderán a hacer más fáciles las tareas del programador generalizando los comandos de voz en lenguaje natural, brindando herramientas para el trabajo fluido, dinámico y flexible, se busca la interacción de hombre y máquina.

Las interfaces estarán orientadas a un lenguaje más comprensible y amigable explorando al máximo los avances tecnológicos del hardware, aprovechando así más su capacidad.

Para comprender la prospectiva en este campo necesitamos hablar de las dos herramientas de trabajo: El software y hardware. Actualmente el desarrollo de los lenguajes con respecto al hardware se encuentra a nivel del sistema, es la capa superior de este, lo que se prevé es que el software llegue al nivel de la microelectrónica y con el desarrollo de la cibernética avance al nivel del registro de transferencia, es decir que el software este representado en microchips que logre autorregularse: Aprender, recordar y captar los diferentes datos.

1.3 BASES DE DATOS

1.3.1 CONCEPTOS BÁSICOS

1.3.1.1 Conceptos Básicos de Bases de Datos

Para empezar y poder comprender mejor lo que es una base de datos se empezará por definir algunos conceptos básicos que involucran a las bases de datos.

Dato: Conjunto de caracteres con algún significado, pueden ser numéricos, alfabético, o alfanuméricos.

Información: Es un conjunto ordenado de datos los cuales son manejados según la necesidad del usuario, para que un conjunto de datos pueda ser procesado eficientemente y pueda dar lugar a información, primero se debe guardar lógicamente en archivos.

1.3.1.2 Conceptos básicos de archivos computacionales

Campo: Es la unidad mas pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto.

Registro: Colección de campos de iguales o de diferentes tipos

Archivo: Colección de registros almacenados siguiendo una estructura homogénea.

1.3.2 DEFINICIÓN

Una base de datos es en esencia una colección de datos relacionados entre sí, de la cual los usuarios pueden extraer información sin considerar las fronteras de los datos.

Un objetivo importante de un sistema de base de datos es proporcionara los usuarios una visión abstracta de los datos, es decir el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos. Sin embargo para que el sistema sea manejable, los datos se deben extraer eficientemente.

1.3.3 OBJETIVOS DE LOS SISTEMAS DE BASES DE DATOS.

Los objetivos principales de un sistema de bases de datos es disminuir los siguientes aspectos.

- **Redundancia e inconsistencia de datos.**

  Puesto que los archivos que mantienen almacenada la información son creados por diferentes tipos de programas de aplicación existe la posibilidad de que si no se controla detalladamente el almacenamiento, se pueda originar un duplicado de información, es decir que la misma información sea mas de una vez en un dispositivo de almacenamiento y acceso a los datos, es decir diversas copias de un mismo dato no concuerdan entre si, por ejemplo: que se actualiza la dirección de un cliente en un archivo y que en otros archivos permanezca la anterior.

- **Dificultad para tener acceso a los datos.**

Un sistema de base de datos debe contemplar un entorno de datos que le facilite al usuario el manejo de los mismos. Si suponemos un banco, en el que uno de los gerentes necesita averiguar los nombres de todos los clientes que viven dentro del código postal 03720 de la ciudad. El gerente pide al departamento de procesamiento de datos que genere la lista correspondiente. Puesto que esta situación no fue prevista en el diseño del sistema, no existe ninguna aplicación de consulta que permita este tipo de solicitud, esto ocasiona una deficiencia del sistema.

- **Aislamiento de los datos.**

Puesto que los datos están repartidos en varios archivos, y estos no pueden tener diferentes formatos, es difícil escribir nuevos programas de aplicación para obtener los datos apropiados.

- **Anomalías del acceso concurrente.**

Para mejorar el funcionamiento global del sistema y obtener un tiempo de respuesta más rápido, muchos sistemas permiten que múltiples usuarios actualicen los datos simultáneamente.

En un entorno así la interacción de actualizaciones concurrentes puede dar por resultado datos inconsistentes. Para prevenir esta posibilidad debe mantenerse alguna forma de supervisión en el sistema.

- **Problemas de seguridad.**

La información de toda empresa es importante, aunque unos datos son más importantes que otros, por tal motivo se debe considerar el control de acceso a los mismos, no todos los usuarios pueden visualizar alguna información, por tal motivo para que un sistema de base de datos sea confiable debe mantener un grado de seguridad que garantice la autentificación y protección de los datos. En un banco por ejemplo, el personal de nóminas sólo necesita ver la parte de la base de datos que tiene información acerca de los distintos empleados del banco y no a otro tipo de información.

- **Problemas de integridad.**

Los valores de datos almacenados en la base de datos deben satisfacer cierto tipo de restricciones de consistencia. Estas restricciones se hacen cumplir en el sistema añadiendo códigos apropiados en los diversos programas de aplicación.

1.3.4 NIVELES

Existen diferentes niveles de abstracción para simplificar la interacción de los usuarios con el sistema, estos son:

- **Nivel Físico.** Es la representación del nivel mas bajo de abstracción, en éste se describe en detalle la forma la forma en como se almacenan los datos en los dispositivos de almacenamiento (por ejemplo, mediante señaladores o índices para el acceso aleatorio a los datos).

- **Nivel Conceptual.** El siguiente nivel más alto de abstracción describe que datos son almacenados realmente en la base de datos y las relaciones que existen entre los mismos,

describe la base de datos completa en términos de su estructura de diseño. El nivel conceptual de abstracción lo usan los administradores de bases de datos, quienes deben decidir qué información se va a guardar en la base de datos. Consta de las siguientes definiciones:

1. **Definición de los datos:** Se describen el tipo de datos y la longitud de campo todos los elementos direccionables e base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos y registros conceptuales (entidades).

2. **Relaciones entre datos:** Se definen las relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

En el nivel conceptual la base de datos aparece como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad los archivos conceptuales no existen físicamente. La transformación de registros conceptuales a registros físicos para el almacenamiento se lleva a cabo por el sistema y es transparente al usuario.

- **Nivel de visión.** Nivel más alto de abstracción, es lo que el usuario final puede visualizar del sistema terminado, describe solo una parte de la base de datos al usuario acreditado para verla. El sistema puede proporcionar muchas visiones para la misma base de datos.

1.3.5 MODELO DE DATOS

Modelo: Es una representación de la realidad que contiene las características generales de algo que se va a realizar. En base de datos, esta representación la elaboramos de forma gráfica.

Modelo de Datos: Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia.

1.3.5.1 Clasificación de Modelos

Los modelos de datos se dividen en tres grupos:

- Modelos lógicos basados en objetos
- Modelos lógicos basados en registros.
- Modelos físicos de datos.

A) Modelos Lógicos basados en objetos

Se usan para describir datos en los niveles de conceptual y de visión, es decir, con este modelo se representan los datos de la misma forma como nosotros los captamos en el mundo real, tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

Modelo Entidad-Relación.

Denominado por sus siglas como: E-R, este modelo representa a la realidad a través de **entidades**, que son objetos que existen y que se distinguen de otros por sus características, por ejemplo: un alumno se distingue de otro por sus características particulares como el nombre, o el número de control asignado al entrar a una institución educativa, así mismo, un empleado, una materia, etc.

Las características de las entidades en base de datos se llaman **atributos**, por ejemplo el nombre, dirección teléfono, grado, grupo, etc. son atributos de la entidad alumno, Clave, número de seguro social, departamento, etc., son atributos de la entidad empleado. A su vez una entidad se puede asociar o relacionar con más entidades a través de **relaciones.**

B) Modelos lógicos basados en registros.

Se utilizan para describir datos en los niveles conceptual y físico.

Estos modelos utilizan registros e instancias para representar la realidad, así como las relaciones que existen entre estos registros (ligas) o apuntadores. A diferencia de los modelos de datos basados en objetos, se usan para especificar la estructura lógica global de una base de datos y proporcionar una descripción a nivel más alto de la implementación.

Los tres modelos de datos más ampliamente aceptados son:

- Modelo Relacional
- Modelo de Red
- Modelo Jerárquico

Modelo Relacional

En este modelo se representan los datos y las relaciones entre estos, a través de una colección de tablas, en las cuales los renglones (tuplas) equivalen a los de cada uno de los registros que contendrá la base de datos y las columnas corresponden a las características (atributos) de cada registro localizado en la tupla.

En este modelo, existen dos formas de representar las relaciones entre las entidades; pero para ello se necesita definir que es una **llave primaria:** Es un atributo el cual queda definido

como atributo principal, es una forma única de identificar a una entidad. Por ejemplo, el RFC de un empleado se distingue de otro por que los RFC no pueden ser iguales.

Modelo de Red

Este modelo representa los datos mediante colecciones de registros y de sus relaciones se representan por medio de ligas o enlaces, los cuales pueden verse como punteros. Los registros se organizan en un conjunto de gráficas arbitrarias.

Modelo Jerárquico

Es similar al modelo de red en cuanto a las relaciones y datos, ya que estos se representan por medio de registros y sus ligas. La diferencia radica en que están organizados por conjuntos de árboles en lugar de gráficas arbitrarias.

Modelos físicos de datos.

Se usan para describir a los datos en el nivel más bajo, aunque existen muy pocos modelos de este tipo, básicamente capturan aspectos de la implementación de los sistemas de base de datos. Existen dos clasificaciones de este tipo que son:

- Modelo unificador.
- Memoria de elementos.

1.3.6 MANEJADOR DE BASES DE DATOS

Entre la base de datos física (es decir, los datos tal y como están almacenados en la realidad) y los usuarios del sistema, existe un nivel de programas, denominado, manejador de bases de datos (MBD) o, en la mayoría de los casos, el sistema administrador de bases de datos RDBMS (Data Base Management System).

Un RDBMS es el conjunto de programas que permiten la definición, manipulación y control de acceso para una o varias bases de datos.

1.3.6.1 Características de los RDBMS

- Facilitan la integridad, seguridad y acceso de los datos.
- Los datos se almacenan como mínima redundancia.
- Las aplicaciones son independientes del almacenamiento físico de los datos.

1.3.6.2 Funciones principales

- Crear y organizar la Base de datos.

- Establecer y mantener las trayectorias de acceso a la base de datos de forma que los datos puedan ser accesados rápidamente.

- Manejar los datos de acuerdo a las peticiones de los usuarios.

- Registrar el uso de las bases de datos.

- Interacción con el manejador de archivos. Esto a través de las sentencias en DML al comando del sistema de archivos. Así el manejador de base de datos es el responsable del verdadero almacenamiento de los datos.

- Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.

- Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos

.

- Seguridad e integridad. Consiste en contar con mecanismos que permitan el control de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

Un RDBMS se puede ver como la interface entre la base de datos física y las peticiones del usuario. El RDBMS interpreta las peticiones de entrada/salida del usuario y las manda al sistema operativo para la transferencia de datos entre la unidad de memoria secundaria y la memoria principal.

En sí, un sistema manejador de base de datos es el corazón de la base de datos ya que se encarga del control total de los posibles aspectos que la pueden afectar.

1.3.6.3 Esquemas de seguridad en el RDBMS

Dentro del Sistema Manejador de Base de Datos (DBMS) podemos encontrar un acceso multicapas, como el que se muestra a continuación:

- El usuario final debe tener una cuenta válida dentro de la capa del servidor (DBMS). *Seguridad a Nivel Servidor*
- El usuario final debe ser un usuario válido dentro de la capa de la base de datos. *Seguridad a Nivel de Base de Datos*
- El usuario final deberá tener permiso dentro de la capa de los datos. *Seguridad a Nivel de Permisos sobre Objetos y Comandos.*

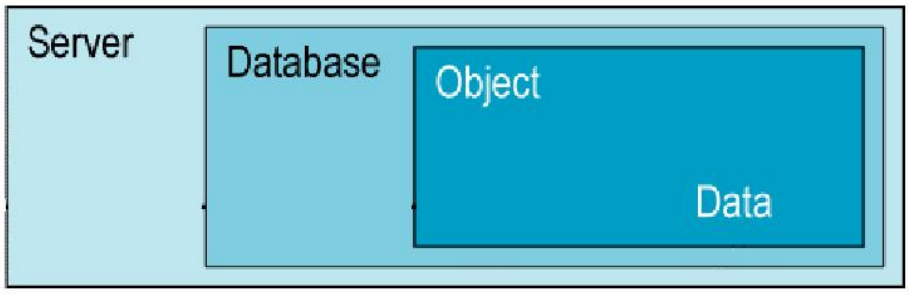


Figura 1.3 *Seguridad en Base de Datos*

1.3.6.4 componentes de un RDBMS

DDL o Lenguaje de Definición de Datos: Se utiliza para crear, eliminar o modificar tablas, índices, vistas, triggers, procedimientos; es decir, nos permite definir la estructura de la base de datos mediante comandos como crear (*Create*), eliminar (*Drop*), o alterar (*Alter*).

-create. Utilizado para crear nuevas bases de datos, tablas, campos, índices, vistas, defaults, reglas, procedimientos, procedimientos, triggers.

-alter. Utilizado para modificar la estructura de una tabla para agregar campos o constrainst.

-drop. Utilizado para eliminar bases de datos, tablas, campos, índices, vistas, defaults, reglas, procedimientos, procedimientos, triggers

DML o Lenguaje de Manipulación de Datos: Se utiliza para realizar la consulta y edición de la información contenida en la base de datos, esto implica: seleccionar, insertar, borrar, modificar.

Los DML se distinguen por sus sublenguajes de recuperación subyacentes; se pueden distinguir dos tipos de DML, el procedural y el no procedural. La principal diferencia entre ambos es que en los lenguajes procedurales se tratan los registros individualmente, mientras que en uno no procedural se opera sobre un conjunto de registros.

Las instrucciones relacionadas con este componente son:

-select. Permite realizar consultas a la base de datos.

-insert. Empleado para agregar registros a una tabla.

-update. Utilizado para modificar los valores de los campos de una tabla.

-delete. Utilizado para modificar los valores de los campos de una tabla.

DCL o Lenguaje de Control de Datos: Se utiliza para la definición de los privilegios de control de acceso y edición a los elementos que componen la base de datos (seguridad), es decir, permitir o revocar el acceso.

Los permisos a nivel base de datos pueden otorgarse a usuarios para ejecutar ciertos comandos dentro de la base o para que puedan manipular objetos y los datos que puedan contener estos.

Las instrucciones relacionadas con este componente son:

-**grant.** Permite otorgar permisos a los usuarios sobre los objetos definidos en la base de datos, así como las operaciones a utilizar sobre ellos.

-**revoke.** Permite revocar permisos sobre los objetos definidos en la base de datos y las operaciones sobre los mismos.

DD o Diccionario de Datos: El contenido del diccionario puede considerarse como “datos acerca de los datos” (los cuales comúnmente reciben el nombre de metadatos), es decir, definiciones de otros objetos de la base de datos.

En particular, todos los diversos esquemas (externo, conceptual e interno), se almacenan físicamente en el diccionario, tanto en forma fuente como en forma objeto. Un diccionario amplio incluirá también las referencias cruzadas que indican, por ejemplo que partes de datos utiliza cada programa, que informes necesita cada departamento, etc. De hecho, el diccionario puede integrarse a la base de datos que describe, y, por tanto, incluir su propia descripción.

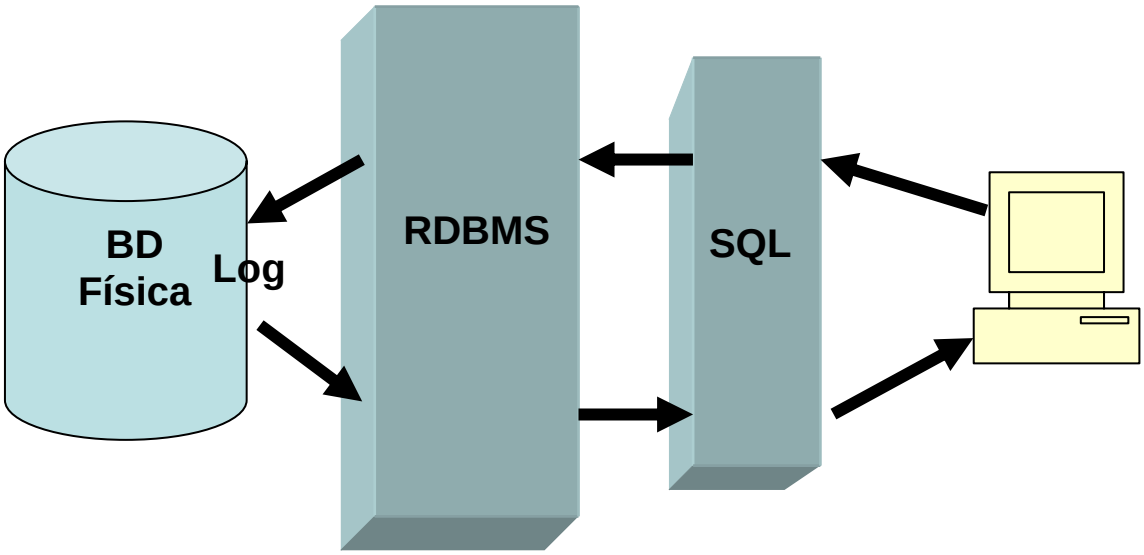


Figura 1.4 *Esquema Cliente Servidor*

[4]Diplomado en Bases de Datos impartido por la UNAM (DGSCA 2005).

Conclusión

Una vez que se han dado a conocer los conceptos generales de las teorías que se utilizarán para el desarrollo del presente trabajo, con el fin de tener un panorama general de los conceptos utilizados en recursos humanos, lenguajes de programación y bases de datos, se comenzará a realizar un análisis detallado del sistema, mediante la definición de requerimientos, métodos y factibilidad para realizar el sistema de la UV, de esto se hablará mas a detalle en el siguiente capítulo (Análisis y definición de requerimientos).

CAPITULO 2

ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS

Objetivos:

- Conocer que son los modelos del proceso de software, sus ventajas y desventajas para finalmente poder seleccionar el que mas se apega a cubrir las necesidades del sistema.

- Conocer el método de desarrollo por análisis estructurado, sus ventajas y desventajas

- Realizar un estudio de factibilidad de acuerdo a los requerimientos y los objetivos que debe cumplir el área de administración de personal de la UV para conocer de que manera contribuye el sistema a la optimización de los procesos que actualmente se llevan a cabo.

- Plantear los requerimientos generales y específicos del sistema.

2.1 MODELOS DEL PROCESO DE SOFTWARE

Un modelo del proceso del software es una representación abstracta de un proceso del software. Cada modelo de proceso representa un proceso desde una perspectiva particular por lo que sólo provee información parcial acerca de ese proceso.

Estos modelos generales no son descripciones definitivas de los procesos del software. Más bien, son abstracciones útiles que se pueden utilizar para explicar diferentes enfoques para desarrollar software. Para sistemas muy grandes, no se usa un proceso de software sencillo, para desarrollar diversas partes del sistema, se utilizan diferentes procesos.

Los principales modelos de procesos son:

1. *El modelo cascada ó ciclo de vida clásico*. Toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, como especificación de requerimientos, diseño de software, implementación, pruebas, etc.

2. *Desarrollo evolutivo*. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un primer sistema se desarrolla rápidamente a partir de especificaciones abstractas. Entonces, se refina con la ayuda del cliente para producir un sistema que satisfaga sus necesidades.

3. *Desarrollo formal de sistemas*. Este enfoque se basa en la producción de una especificación matemática formal del sistema y en la transformación de esta especificación, utilizando métodos matemáticos, para construir un programa.

4. *Desarrollo basado en la reutilización*. Este enfoque se basa en la existencia de un número significante de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en integrar estos componentes en el sistema más que en desarrollarlos desde cero.

Los procesos que se basan en el modelo de cascada y en el desarrollo evolutivo se utilizan ampliamente en el desarrollo de sistemas prácticos. El desarrollo formal de sistemas se ha utilizado exitosamente en varios proyectos, pero pocas organizaciones utilizan los procesos del modelo. La reutilización informal es común en muchos procesos, pero la mayoría de las organizaciones no orientan el desarrollo de software alrededor de esta. Este enfoque tendrá mucha influencia en el siglo XXI, ya que construir sistemas a partir de componentes reutilizables es fundamental para el desarrollo rápido de software.

2.1.1 MODELO CASCADA O CICLO DE VIDA CLÁSICO

[5] El primer proceso de desarrollo de software que se publicó se derivó de otros procesos de ingeniería, este se conoce como “modelo de cascada” o como ciclo de vida de software. Las principales etapas de este modelo se transforman en actividades fundamentales de desarrollo:

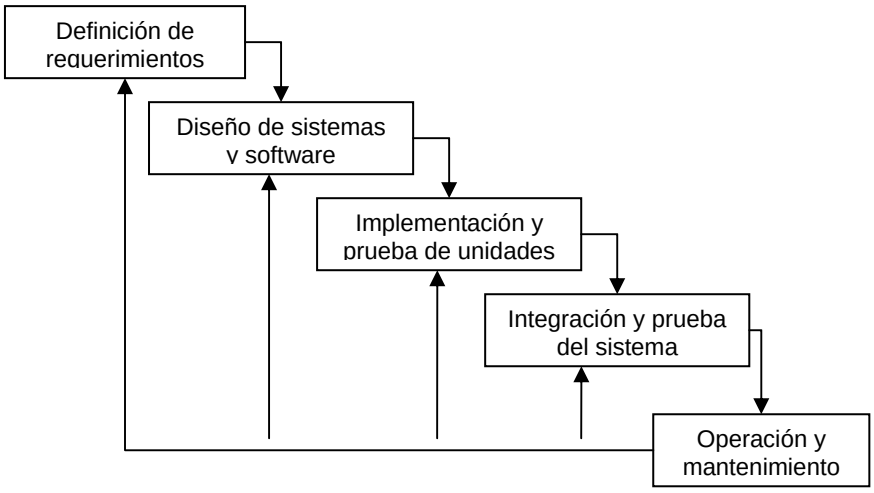


Figura 2.1 *Modelo cascada*

1. Análisis y definición de requerimientos. Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema.

2. Diseño de sistemas y de software. El proceso de diseño de sistemas divide los requerimientos en sistemas de hardware o de software. Establece una arquitectura completa del sistema. El diseño de software identifica y describe las abstracciones fundamentales del sistema de software y sus relaciones.

3. Desarrollo de Software. Los encargados de desarrollo de software pueden instalar (o modificar y después instalar) software comprado a terceros o escribir programas diseñados a la medida del solicitante. La elección depende del costo de cada alternativa, del tiempo disponible para escribir el software y de la disponibilidad de los

programadores. Por regla general, los programadores (o analistas programadores) que trabajan en las grandes organizaciones pertenecen a un grupo permanente de profesionales. En empresas pequeñas, donde no hay programadores, se pueden contratar servicios externos de programación.

Los programadores también son responsables de la documentación de los programas y de proporcionar una explicación de cómo y por qué ciertos procedimientos se codifican en determinada forma. La documentación es esencial para probar el programa y llevar a cabo el mantenimiento una vez que la aplicación se encuentra instalada.

4. Prueba de Sistemas. 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. En ocasiones se permite que varios usuarios utilicen el sistema para que los analistas observen si tratan de emplearlo en formas no previstas. Es preferible descubrir cualquier sorpresa antes de que la organización implante el sistema y dependa de él.

   En muchas organizaciones, las pruebas son conducidas por personas ajenas al grupo que escribió los programas originales; con esto se persigue asegurar, por una parte, que las pruebas sean completas e imparciales y, por otra, que el software sea más confiable.

5. Operación y mantenimiento. Por lo general (aunque no necesariamente), ésta es la fase más grande del ciclo de vida. El sistema se instala y pone en uso práctico.

El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubren nuevos requerimientos

El resultado de cada fase es uno o más documentos aprobados. La siguiente fase empieza hasta que las otras hayan finalizado. En la práctica estas etapas se traslapan y proporcionan información a las otras.

Durante el diseño se identifican los problemas con los requerimientos, durante el diseño del código se hace la misma identificación y así sucesivamente. El proceso de software no es un modelo lineal simple sino que implica una serie de iteraciones de las actividades de desarrollo.

Durante la fase final del ciclo de vida (operación y mantenimiento) el software se pone en funcionamiento. Se descubren errores y omisiones en los requerimientos originales del software, por lo tanto el sistema debe evolucionar para mantenerse útil. Hacer estos cambios implica repetir algunas o todas las etapas previas del proceso.

El problema con el modelo de cascada es su inflexibilidad al dividir el proyecto en estas etapas. Se deben hacer compromisos en las etapas iniciales, por lo que es difícil responder a los cambios en los requerimientos del cliente. Por lo tanto, este modelo solo se debe utilizar cuando los requerimientos se comprendan del todo.

Ventajas

- Existe una comprensión amplia de la definición de requerimientos antes de comenzar las siguientes etapas
- En la práctica, las etapas en que esta dividido este modelo se traslapan y proporcionan información a las otras.

Desventajas

- Se deben de hacer compromisos en las etapas iniciales, lo que significa que es difícil responder a los cambios en los requerimientos del cliente.

2.1.2 DESARROLLO EVOLUTIVO

Se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un

sistema adecuado. Las actividades se llevan a cabo concurrentemente y tienen retroalimentación rápida a lo largo del proceso.

Existen dos tipos de desarrollo evolutivo:

1. Desarrollo exploratorio. El objetivo del proceso es trabajar con el cliente para explorar sus requerimientos y entregar un sistema final. El desarrollo empieza con las partes del sistema que se comprenden mejor. El sistema evoluciona agregando nuevos atributos acordes con las propuestas del cliente.
2. Prototipos desechables. El objetivo del proceso de desarrollo evolutivo es comprender los requerimientos del cliente y entonces desarrollar una definición mejorada de los requerimientos para el sistema. El prototipo se centra en experimentar con aquellas partes de los requerimientos del cliente que no se comprenden del todo.

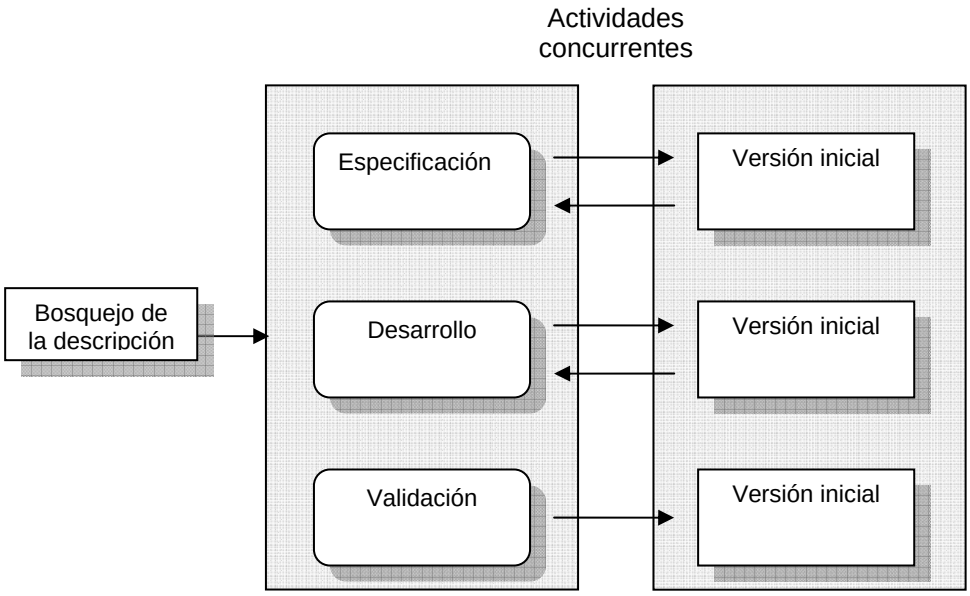


Figura 2.2 *Proceso del desarrollo evolutivo*

Para sistemas pequeños (menores de 100,000 líneas de código) o para sistemas de tamaño medio (hasta 500,000 líneas de código) con un periodo de vida relativamente corto, el enfoque evolutivo de desarrollo es el mejor. Sin embargo, para sistemas grandes con un periodo de vida largo, los problemas en el desarrollo evolutivo son particularmente agudos. Para ellos se recomienda un proceso mixto del modelo de cascada y el de desarrollo evolutivo.

Ventajas

- la especificación se puede desarrollar de forma creciente. Tan pronto como los usuarios desarrollen un mejor entendimiento de su problema, éste se reflejará en el sistema de software.

Desventajas

- El proceso no es visible. Los administradores tienen que hacer entregas regulares para medir el progreso. Si los sistemas se desarrollan rápidamente, es muy costoso producir documentos que reflejen cada versión del sistema.
- A menudo los sistemas tienen una estructura deficiente. Los cambios continuos tienden a corromper la estructura del software. Incorporar cambios en él se convierte en una tarea difícil y costosa.
- Se requieren herramientas y técnicas especiales. Éstas permiten un desarrollo rápido pero son incompatibles con otras herramientas o técnicas y relativamente pocas personas tienen las habilidades necesarias para utilizarlas.

2.1.3 DESARROLLO FORMAL DE SISTEMAS

Este es un enfoque para desarrollar software que tiene algo en común con el modelo de cascada, pero donde el modelo de desarrollo se basa en la transformación matemática formal de una especificación de un problema a un programa ejecutable.

Las diferencias críticas entre este enfoque y el modelo de cascada son:

1. La especificación de requerimientos de software se refinan en una especificación formal detallada que se expresa en notación matemática.
2. Los procesos de desarrollo de diseño, implementación y pruebas de unidades se reemplazan con un proceso de desarrollo por transformaciones donde la especificación formal se refina, a través de una serie de transformaciones, hasta llegar a un programa.

En el proceso de transformación, la representación matemática formal del sistema se convierte de forma sistemática en una representación mas detallada del sistema, pero matemáticamente correcta. Cada paso agrega detalle hasta que la especificación formal se convierte en un programa equivalente. Las transformaciones son lo suficiente cercanas como para que el

esfuerzo de verificar la transformación no sea excesivo. Por lo tanto se puede garantizar, suponiendo que no existen errores en la verificación, que el programa es una implementación real de la especificación.

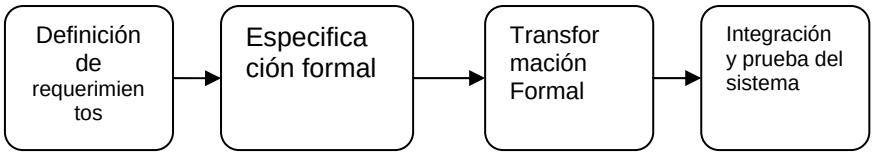


Figura 2.3 *Desarrollo formal de sistemas*

Ventajas

- Simplifica la creación de pruebas de protección o seguridad para demostrar a los clientes o a los organismos de certificación que el sistema cumple los requerimientos de protección de seguridad

Desventajas

- Pruebas largas y poco prácticas para sistemas de gran escala
- Se requieren ciertas habilidades para elegir que transformación aplicar, asimismo es difícil probar la correspondencia entre transformaciones

2.1.4 DESARROLLO ORIENTADO A LA REUTILIZACIÓN

En la mayoría de los proyectos de software existe algo de reutilización de software. Por lo general, esto pasa cuando las personas que trabajan en el proyecto conocen diseños o código similares al requerido. Los buscan, los modifican acorde a lo requerido y los incorporan en el sistema. En el enfoque evolutivo, la reutilización es a menudo indispensable para el desarrollo rápido de sistemas.

Esta reutilización informal es independiente del proceso genérico que se utilice. Sin embargo, en años recientes, ha surgido un enfoque de desarrollo de software (ingeniería de software

basada en componentes) que se basa en la reutilización, el cual se está utilizando de forma amplia.

Este enfoque basado en la reutilización se compone de un gran número de componentes de software reutilizable, así como de marcos de trabajo para estos. Algunas veces estos componentes son sistemas por si mismos (Sistemas Comerciales) que se utilizan para proveer funcionalidad específica, como dar formato al texto, efectuar cálculos numéricos, etc.

Aunque la etapa de especificación de requerimientos y la validación son comparables con otros procesos, las etapas intermedias en el proceso orientado a la reutilización son diferentes.

Estas etapas son:

1. Análisis de Componentes: Dar la especificación de requerimientos, se buscan los componentes para implementar esta especificación. Por lo general, no existe una paridad y los componentes que se utilizan solo proporcionan algún tipo de funcionalidad esperada.
2. Modificación de Requerimientos. En esta etapa, los requerimientos se analizan utilizando información a cerca de los componentes que se van descubriendo. Entonces, estos componentes se modifican para reflejar los componentes disponibles. Si las modificaciones no son posibles, la actividad de análisis de componentes se lleva a cabo nuevamente para buscar soluciones alternativas.
3. Diseño de Sistemas con Reutilización En esta fase se diseña o se reutiliza un marco de trabajo para el sistema. Los diseñadores toman en cuenta los componentes que se reutilizan y organizan el marco de trabajo para que se ajuste a ellos. Si los componentes reutilizables no están disponibles, se diseña un nuevo software.
4. Desarrollo e Integración para crear el sistema, el software que no se puede comprar se desarrolla, y los componentes y los sistemas COTS se integran. En este modelo, la integración de sistemas es parte del proceso de desarrollo, más que una actividad separada.

Ventajas

- reduce la cantidad de software a desarrollarse y también reduce los costos y los riesgos
- Conduce a una entrega más rápida del software

Desventajas

- Debido a la reutilización de código, este método puede conducir a un sistema que no cumple las necesidades reales de los usuarios. Mas aún, si las nuevas versiones de los componentes reutilizables no están bajo la vigilancia de la organización que los utiliza, se pierde el control sobre la evolución del sistema.

2.1.5 DESARROLLO INCREMENTAL

El modelo de desarrollo en cascada requiere que los clientes de un sistema cumplan un conjunto de requerimientos antes de que inicie el diseño, y que el diseñador cumpla estrategias particulares de diseño antes de la implementación. Los cambios de requerimientos durante el desarrollo implican rehacer el trabajo de captura de éstos, de diseño e implementación. Sin embargo, las ventajas de este modelo son que se administra fácilmente y que su separación en el diseño y la implementación conduce a sistemas robustos susceptibles al cambio.

En contraste, un enfoque de desarrollo evolutivo permite que los requerimientos y las decisiones de diseño se retrasen, y conduce a un software débilmente estructurado y difícil de comprender y mantener. El desarrollo incremental es un enfoque intermedio que cambia las ventajas de ambos modelos.

En proceso de desarrollo incremental, los clientes identifican los servicios que proveerá el sistema, identifican cuales servicios son más importantes y cuáles menos. Entonces se definen varios incrementos en donde cada uno proporciona un subconjunto de funcionalidad del sistema. Los servicios de prioridad más alta son los que se entregan primero al cliente.

Durante el desarrollo, se lleva a cabo un análisis adicional de requerimientos para los requerimientos posteriores, pero no se aceptan cambios en los requerimientos para el incremento en cuestión.

Cuando un incremento se completa y entrega, los clientes pueden ponerlo en servicio. Experimentan con él, lo cual les ayudará a clarificar sus requerimientos para los incrementos posteriores y para las últimas versiones del incremento actual. La funcionalidad del sistema mejora con cada incremento.

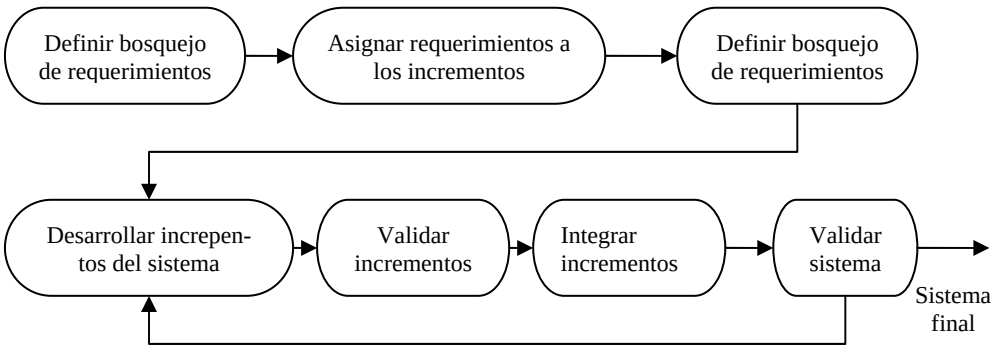


Figura 2.4 *Desarrollo Incremental*

No es necesario usar el mismo proceso de desarrollo para cada incremento. Se puede utilizar un modelo de desarrollo en cascada si los servicios en un incremento tienen una especificación bien definida. Si la especificación no es clara, se puede utilizar un desarrollo evolutivo.

Ventajas

- El cliente no tiene que esperar hasta que se entregue el sistema completo para sacar provecho de él.

- Los clientes pueden utilizar los incrementos iniciales como un prototipo para obtener experiencia sobre los requerimientos de los incrementos posteriores del sistema.

- Existe un bajo riesgo de fallar en el proyecto total; puesto que los servicios de alta prioridad se entregan primero y los incrementos posteriores se integran a ellos, por lo que significa que los clientes encontrarán menos fallas en el software en las partes más importantes del sistema.

Desventajas

- los incrementos deben ser relativamente pequeños (no más de 20,000 líneas de código) y cada uno debe entregar alguna funcionalidad del sistema.

- Debido a que los requerimientos no se definen en detalle hasta que un incremento se implementa, es difícil identificar los recursos comunes que requieren todos los incrementos.

2.1.6 DESARROLLO EN ESPIRAL

Más que representar el proceso del software como una sucesión de actividades con retrospectiva de una actividad a otra, se representa como una espiral. El ciclo más interno podría referirse a la factibilidad del sistema, el siguiente ciclo a la definición de requerimientos del sistema, el siguiente ciclo al diseño del sistema, y así sucesivamente.

Cada ciclo de espiral se divide en cuatro sectores:

1. Definición de objetivos. Para esta fase del proyecto se definen los objetivos específicos. Se identifican las restricciones del proceso y el producto, y se estipula un plan detallado de administración. Se identifican los riesgos del proyecto. Dependiendo de esos riesgos, se planean estrategias alternativas.

2. Evaluación y reducción de riesgos. Se lleva a cabo un análisis detallado para cada uno de los riesgos del proyecto. Se definen los pasos para reducir dichos riesgos. Por ejemplo, si existe el riesgo de tener requerimientos inapropiados, se desarrolla un prototipo del sistema.

3. Desarrollo y validación. Después de la evaluación de riesgos, se elige un modelo para el desarrollo del sistema. Por ejemplo, si los riesgos en la interfaz de usuarios son dominantes, un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si los riesgos de protección son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado, y así sucesivamente. El modelo de cascada es el más apropiado para el desarrollo si el mayor riesgo identificado es la integración de los subsistemas.

---

[5]Ian Sommerville (2002). Ingeniería de software .México, Pearson Educación.

4. Planeación. El proyecto se revisa y se toma la decisión de si se debe continuar con un ciclo posterior de la espiral. Si se decide continuar, se debe planear la siguiente fase del proyecto.

Ventajas

- No existen fases fijas como la especificación o el diseño
- Este modelo puede contener otros modelos
- La construcción de prototipos se utiliza para resolver las dudas en los requerimientos y así reducir el riesgo.

Desventajas

- Al contener otros modelos puede heredar alguna de las desventajas de estos.

2.2 MÉTODOS DE LA INGENIERÍA DE SOFTWARE

2.2.1 MÉTODO DE DASARROLLO POR ANÁLISIS ESTRUCTURADO

Muchos especialistas en sistemas de información reconocen la dificultad de comprender de manera completa sistemas grandes y complejos. El método de desarrollo del análisis estructurado tiene como finalidad superar esta dificultad por medio de:

1) La división del sistema en componentes
2) La construcción de un modelo de sistema

El método incorpora elementos tanto de análisis como de diseño.

ANÁLISIS ESTRUCTURADO

[6] El análisis estructurado se concentra en especificar lo que se requiere que haga el sistema o la aplicación. No se establece como se cumplirán los requerimientos o la forma en que implantará la aplicación. Más bien permite que las personas observen los elementos lógicos (lo que hará el sistema) separado de los componentes físicos (computadoras, terminales, sistemas de almacenamiento, etc.) Después de esto se puede desarrollar un diseño físico eficiente para la situación donde será utilizado.

El análisis estructurado se combina, con bastante frecuencia con el método ya presentado de ciclo de vida clásico de desarrollo de sistemas, por ejemplo, los analistas pueden optar por desarrollar diagramas de flujo de datos como una forma para documentar las relaciones entre componentes durante la investigación detallada de algún sistema existente. Asimismo, se pueden definir los archivos y datos en un diccionario centralizado de datos de acuerdo con las reglas del análisis estructurado.

Algunas veces se opta por no utilizar este método de desarrollo ya que el desarrollo de diagramas y esquemas es una tarea que consume mucho tiempo, sobre todo si el sistema es grande y complejo. Para superar esto se han desarrollado herramientas asistidas por computadora. También se señala que los elementos que faltan, tales como las personas y los procedimientos de control son parte del sistema mismo y no pueden omitirse en la descripción de éste.

ELEMENTOS DEL ANÁLISIS ESTRUCTURADO

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

A) Descripción gráfica

Una de las formas de describir un sistema es preparar un bosquejo que señale sus características, identifique la función para la que sirve e indique cómo éste interactúa con otros elementos, entre otras cosas. Sin embargo, escribir de esta manera un sistema grande es un proceso tedioso y propenso a errores ya que es fácil omitir algún detalle o dar una explicación que quizás los demás no entiendan.

En lugar de las palabras el análisis estructurado utiliza símbolos, o íconos, para crear un modelo gráfico del sistema. Los modelos de este tipo muestran los detalles del sistema pero sin introducir procesos manuales o computarizados, archivos en cinta o disco magnéticos, o procedimientos operativos y de programas. Si se selecciona los símbolos y notación correctos, entonces casi cualquier persona puede seguir la forma en que los componentes se acomodarán entre sí para formar el sistema.

B) Diagramas de flujo de datos

El modelo del sistema recibe el nombre de Diagrama de Flujo de Datos (DFD). La descripción completa de un sistema está formada por un conjunto de diagramas de flujo de datos.

Para desarrollar una descripción del sistema por el método de análisis estructurado, se sigue un proceso descendente (top-down). En el modelo original se detallan diagramas de bajo nivel que muestran características adicionales del sistema. Cada proceso puede desglosarse en diagramas de flujo de datos cada vez más detallados. Esta secuencia se repite hasta que se obtienen suficientes detalles que permiten al analista comprender en su totalidad la parte del sistema que se encuentra bajo investigación.

C) Diccionario de datos

Todas las definiciones de los elementos en el sistema – flujos de datos, procesos y almacenes de datos – están descritos en forma detallada en el diccionario de datos. Si algún miembro del equipo encargado del proyecto desea saber alguna definición del nombre de un dato o el contenido particular de un flujo de datos, esta información debe encontrarse disponible en el diccionario de datos.

2.2.2 DISEÑO ESTRUCTURADO

El diseño estructurado, otro elemento del análisis estructurado que emplea la descripción gráfica, se enfoca en el desarrollo de especificaciones del software. La meta del diseño estructurado es crear programas formados por módulos independientes unos de otros desde el punto de vista funcional. Este enfoque no solo conduce hacia mejores programas sino que facilita el mantenimiento de los mismos cuando surja la necesidad de hacerlo.

El diseño estructurado es una técnica específica para el diseño de programas y no un método de diseño de comprensión. Es decir, no indica nada relacionado con el diseño de archivos o bases de datos, la presentación de entradas o salidas, la secuencia de procesamiento o el hardware que dará soporte a la aplicación. Esta técnica conduce a la especificación de módulos de programa que son funcionalmente independientes.

La herramienta fundamental del diseño estructurado es el diagrama estructurado. Al igual que los diagramas de flujo de datos, los diagramas estructurados son de naturaleza gráfica y evitan cualquier referencia relacionada con el hardware o detalles físicos. Su finalidad no es mostrar la lógica de los programas (que es la tarea de los diagramas de flujo). Los diagramas estructurados describen la interacción entre módulos independientes junto con los datos que un módulo pasa a otro cuando interacciona con él. Estas especificaciones funcionales para los módulos se proporcionan a los programadores antes que dé comienzo la fase de escritura de código.

Ventajas

- Se puede combinar con el método de ciclo de vida clásico de desarrollo de sistemas.
- Contempla un diccionario de datos para la captura de detalles tales como atributos e identificadores únicos.
- Es ideal para aplicaciones en tiempo real.

Desventajas

- El desarrollo de diagramas y esquemas es una tarea que consume mucho tiempo.
- No ofrece una técnica propia para el modelado de entidades o diseño de datos, pero recomienda el uso de la técnica de Entidad-Relación.

OBSERVACIONES FINALES

Después del estudio y análisis comparativo de diversos modelos de desarrollo de sistemas, se llegó a la conclusión de utilizar el modelo de cascada ó ciclo de vida básico para generar una primera versión del sistema que cumpla con los requerimientos definidos por el cliente.

Se realizarán prototipos iniciales del sistema para resolver incertidumbres en la especificación de requerimientos en las etapas iniciales. Una vez que las partes del sistema hayan quedado bien comprendidas se pueden especificar y desarrollar utilizando un proceso basado en el

modelo de cascada. En caso de que surjan más requerimientos críticos en alguna etapa avanzada del diseño en cascada, se contemplarán hasta que la primera versión haya quedado liberada y se implementarán en una versión posterior.

[6] Kendall & Kendall (1997). Técnicas de análisis y diseño de sistemas. México, Mc Graw Hill.

2.3 ESTUDIO DE FACTIBILIDAD

El área de administración de personal de la UV requiere un sistema que les ayude a automatizar el manejo de la información de los candidatos potenciales a ocupar un puesto dentro de la institución. Se requiere dar un seguimiento desde que el candidato llega a entregar su documentación hasta la contratación o rechazo del mismo. También se requiere automatizar el proceso de altas y cambios de puestos y plazas dentro de la institución y optimizar la elaboración de organigramas de la empresa.

Actualmente la universidad cuenta con un sistema compuesto por dos módulos, uno para el control de empleados y de toda la información relacionada con los mismos y el otro para llevar el control de nóminas y prestaciones.

Dicho sistema fue desarrollado con el lenguaje Visual Basic 6.0, el administrador de Bases de Datos que utilizan es SQL Server 2000 y se cuenta con una base de datos relacional con la información de puestos, plazas y empleados.

Se requiere que el sistema sea completamente compatible con el ya existente y que la información se almacene en la base de datos con que actualmente cuenta la empresa. Así mismo se requiere que el nuevo sistema cumpla con los estándares de interfaz que se manejan actualmente en la universidad.

Los principales problemas en los procesos actuales son:

- Para el proceso de reclutamiento, selección y contratación, no existe un sistema que ayude a optimizar la información de candidatos potenciales. La documentación de los candidatos se encuentra en papel y es archivada en carpetas.

- El proceso de altas y modificaciones de puestos y plazas se realiza mediante formatos en papel que requieren de varias validaciones para ser autorizados y realizar la modificación en la base de datos directamente por el área de sistemas, durante este proceso se puede perder el formato además de que no se tiene un control de la cantidad de movimientos en puestos y plazas y el estatus en que se encuentran dichos movimientos.
- No existe una forma de visualizar la información de los contratos eventuales, por lo que se pierde el control en las fechas de vencimiento.
- No se cuenta con un sistema que permita obtener de manera automática los organigramas de la UV. Actualmente se realizan en Excel y la información se ingresa manualmente.

Los principales objetivos del área de administración de personal de la UV son:

1. Administrar todos los procesos relacionados con los empleados de la UV, tales como: altas, bajas, cambio de área, cambio de puesto, cambio de jerarquía y control de información general de empleados.

2. Controlar el proceso de pago de nómina y prestaciones de los empleados.

3. Tener un control de los puestos existentes en la universidad, tener identificado a que entidad pertenece cada puesto, así como un control del proceso de solicitud de puesto

de nueva creación, de todas las solicitudes de nuevos puestos y el estatus en que éstas se encuentran.

4. Controlar el proceso para dar de alta nuevas plazas, saber con cuantas plazas cuenta la UV, cuales se encuentran ocupadas y cuales vacantes e identificar a que puesto pertenecen.

5. Llevar a cabo el proceso de reclutamiento, selección y contratación de nuevo personal, además de tener identificada la cartera de candidatos potenciales para ingresar en la Universidad.

6. Representar de forma gráfica la estructura organizacional de la universidad ya sea por entidades o por empleados.

Después de analizar los objetivos, los sistemas ya existentes en la UV, las problemáticas y los requerimientos generales de la organización se plantea un sistema que de manera general tendrá las siguientes características:

Una primer parte, estará dedicada a controlar toda la información relacionada con el manejo de puestos de nueva creación en la UV, así como modificaciones y bajas de los anteriores.

Una segunda parte, estará dedicada a controlar toda la información relacionada con el manejo de plazas de nueva creación en la UV, así como modificaciones y bajas de los anteriores.

La tercera parte, se enfocará a optimizar el proceso de reclutamiento, selección y contratación de candidatos potenciales para ocupar plazas vacantes.

La cuarta parte es un control de los contratos que por diferentes circunstancias solo son válidos para un tiempo determinado, definido al momento de su contratación.

La quinta y última parte tendrá la función de generar diagramas organizacionales de la UV a partir de diferentes criterios definidos por el usuario.

Se plantea desarrollar el sistema, bajo la tecnología con la que actualmente se cuenta en la Universidad (SQL Server y Visual Basic 6) debido a que el sistema deberá interactuar y compartir información con los ya existentes en el área de Administración de Personal. Adicionalmente, para el desarrollo de la parte de estructura organizacional, se pretende utilizar el programa Visio 2000 debido a su compatibilidad con Visual Basic 6.0 y a las ventajas que brinda para el diseño de diagramas.

A continuación se describen las principales características del software a utilizar:

VISIO 2000

Visio 2000 es un programa para la creación de diagramas que ayuda a crear diagramas empresariales y técnicos en los que se documenten y organicen ideas, procesos y sistemas complejos. Los diagramas creados en Visio 2000 permiten clara, concisa y eficazmente visualizar y comunicar información, de unas formas no posibles utilizando exclusivamente texto y números. Visio 2000 también automatiza la visualización de los datos al sincronizarse directamente con los orígenes de datos para proporcionar diagramas actualizados, y se puede personalizar para cubrir las necesidades de la organización.

VENTAJAS

- Ayuda al usuario a transmitir una imagen más profesional, clara y concisa de sus ideas.
- Diagramas definidos, con mejoras de gráficos, textos y color, así como la habilidad para importar y manipular imágenes de otras aplicaciones.
- Se integra dentro de la familia Microsoft Office, simplifica las funciones más habituales, el uso de la interfaz y mejora la interoperabilidad.

- Mejora la productividad, en cuanto a reportes y conectividad de bases de datos, así como las capacidades de búsqueda.

DESVENTAJAS

- Pueden ser demasiadas opciones para quien quiera diagramas sencillos de un solo tipo
- Compatibilidad limitada con otros lenguajes diferentes de Visual Basic

SQL SERVER 2000

SQL Server es un sistema manejador de bases de datos relacionales en modo cliente-servidor. Diferente de las bases de datos basadas en archivos, el motor de la base de datos cliente-servidor hace operaciones de lectura-grabación en la base de datos. Por esta razón, las bases de datos que usan esta tecnología pueden manejar muchos mas usuarios concurrentes y cantidades de datos enormemente mayores.

VENTAJAS

- Manejo de Bases de Datos Distribuidas
- Manejo de varias particiones físicas para almacenamiento de datos flexible
- Permite realizar algunas tareas de mantenimiento y administración sin tener que darla de baja
- Conectividad con clientes ODBC
- Maneja la arquitectura cliente-servidor lo que permite separar las tareas en aquellas que se realizarán en el cliente y aquellas que se realizan en el servidor.
-

DESVENTAJAS

- únicamente es compatible con sistemas operativos de la familia Windows
- El manejo de la seguridad no es sumamente confiable.

VISUAL BASIC 6.0

Es un sistema de desarrollo diseñado especialmente para crear aplicaciones con interfaz gráfica, de una forma rápida y sencilla. Para soportar este tipo de desarrollos, Visual Basic utiliza fundamentalmente dos herramientas, una que permite realizar los diseños gráficos y un lenguaje de alto nivel.

VENTAJAS

sencillez para programar aplicaciones de cierta complejidad para Windows.

DESVENTAJAS

Su necesidad de archivos adicionales además del ejecutable y cierta lentitud en comparación con otros lenguajes.

Cabe mencionar que los procesos no serán modificados al elaborar el sistema, únicamente serán automatizados.

Se puede concluir, que el sistema contribuirá de manera directa en la automatización de los procesos que se llevan a cabo en el área de administración de personal de la UV, contribuyendo en el cumplimiento de los objetivos (puntos 3 al 6) listados anteriormente.

2.4 REQUERIMIENTOS

A continuación se da una breve definición de lo que es un requerimiento así como su clasificación.

2.4.1 Definición Los requerimientos del sistema son los servicios que este proveerá y las restricciones para que funcione. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se llama ingeniería de requerimientos.

2.4.2 Clasificación de requerimientos

***Requerimientos de Usuario*:** son declaraciones en lenguaje natural y en diagramas, de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales debe operar.

***Requerimientos del Sistema*:** Establecen con detalle los servicios y restricciones del sistema. El documento de requerimientos del sistema, algunas veces denominado especificación funcional, debe ser preciso.

Los requerimientos del sistema a su ves se clasifican en:

- ***Requerimientos funcionales*:** Son declaraciones de los servicios que proveerá el sistema, de la manera en que este reaccionará a entradas particulares y de cómo se comportará en situaciones particulares.

- ***Requerimientos no funcionales**:* Son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo, estándares, etcétera.

En el siguiente apartado, se presentan los requerimientos expuestos en el área de administración de personal de la UIE, redactados en forma de un documento de requerimientos:

2.4.3 REQUERIMIENTOS DE LA UV

A) Propósito

Dar una descripción de los requerimientos generales que presenta la UV en algunos de los procesos que se llevan a cabo en su área de personal, para después definir los requerimientos propios del sistema que se desarrollará para automatizar estos procesos.

B) Alcance

El sistema abarcará los procesos de selección y contratación de personal, el manejo de altas, bajas y cambios de puestos y plazas y diseño de estructura organizacional de la UV.

C) Resumen

Este documento describe los requerimientos del área de administración de personal de la UV.

Se comienza describiendo el requerimiento tal como lo expresa el encargado del área de personal (requerimiento de usuario) para después hacer un detalle de dicho requerimiento y de cómo se tendrá que comportar el sistema a desarrollar, los módulos que este deberá contener y las restricciones para que este opere correctamente y cumpla con los requerimientos expresados por el encargado de personal. (requerimientos del sistema).

D) DESCRIPCIÓN GENERAL

1) Perspectiva

El sistema automatizará los procesos de reclutamiento, selección y contratación de personal, diseño de estructura organizacional y manejo de altas, bajas y cambios de puestos y plazas.

.

2) Funciones

Las principales funciones del sistema serán:

- Altas, bajas y cambios de puestos y plazas de nueva creación
- Registro de información de candidatos
- Creación de organigramas para representar la estructura organizacional de la UV
- Impresión de Documentos.

3) Características del Usuario

El usuario que manejará el sistema, debe cumplir con las siguientes características:

- Ser empleado de la UV
- Contar con conocimientos básicos de computación (manejo de Windows, Word y Excel)
- Contar con conocimientos básicos a cerca del manejo de la herramienta VISIO
- Tener conocimientos de administración de personal.

4) Restricciones

- El sistema solo podrá ser utilizado por el personal de administración de personal autorizado.
- Los candidatos, solamente podrán ingresar al módulo de alta de datos generales de candidatos

- Los módulos de puestos y plazas, podrán ser utilizados (modo consulta) por los responsables de las diferentes entidades de la UV.

- Para que el sistema funcione correctamente, el equipo en donde se instale deberá trabajar bajo la plataforma Windows (98 en adelante), y tener instalados los paquetes VISO y EXCEL.
- Se requiere contar con una impresora y con hojas tamaño carta para la impresión de organigramas.

5) Dependencias

- El sistema a desarrollar es un complemento para el sistema de manejo de nómina y empleados con que actualmente cuenta la UV, por lo que se deberá desarrollar con herramientas compatibles a este sistema. El sistema de manejo de nómina y empleados está desarrollado en Visual Basic 6.0 y su administrador de Bases de Datos es SQL 2000.

E) REQUERIMIENTOS ESPECÍFICOS

Se solicitan los siguientes módulos:

1) MODULO PARA MANEJO DE INFORMACIÓN DE PUESTOS Y PLAZAS

Requerimientos de Usuario:

El área de personal necesita optimizar el proceso de altas, bajas y cambios de puestos y plazas de nueva creación existentes en la universidad.

Actualmente cuando se requiere realizar alguna modificación en la información de puestos y plazas, la entidad solicitante debe de llenar , formatos en papel:

Una vez que el encargado de la entidad solicitante llena correctamente el formato, debe de entregar el documento al encargado del área de personal, quien procede a validar la información del formato y, en caso de autorizar el movimiento, pasar a firma el documento con las autoridades correspondientes del área de personal para que finalmente el área de sistemas se encargue de realizar la modificación.

Actualmente no existe ningún control que permita conocer los movimientos de puestos y plazas que han sido realizados por alguna entidad. Tampoco hay forma de saber el estatus de los movimientos de puestos y plazas que se encuentran en proceso de autorización

Se requiere llevar un control de las altas, bajas y cambios de puestos y plazas de nueva creación, así como un control del status en que se encuentran estos movimientos, en caso de que aún no estén finalizados.

Requerimientos de sistema: Los módulos que se proponen para el manejo de puestos y plazas, deben de tener las siguientes características:

- Deben de existir dos pantallas diferentes: una para realizar altas, bajas y modificaciones de nuevos puestos y otra para realizar altas, bajas y modificaciones de nuevas plazas.

- Deben permitir ingresar y modificar puestos y plazas de nueva creación de la UV

- El área solicitante debe de tener acceso al sistema para hacer la solicitud del movimiento

- Se requiere una pantalla para visualizar la información general de los puestos y plazas existentes en la UV (organizada por entidad), cada puesto y cada plaza tendrán un consecutivo que los identifique.

- Los puestos y plazas también deberán de poder presentarse organizados de acuerdo a los siguientes estatus:

    - Solicitud: Alguna entidad solicitó el alta de un puesto ó plaza nuevos

    - Validación: El departamento de administración de personal se encuentra validando la solicitud
    - Autorizado: El puesto ó plaza fueron autorizados y por lo tanto están dados de alta en la base de datos principal de la UV
    - Rechazado: El puesto ó plaza fueron rechazados.

- Los puestos o plazas rechazados se borrarán automáticamente cada semana, o bien según el criterio del encargado de personal

- A este módulo pueden acceder los usuarios del área de personal y los usuarios de las diferentes entidades de la universidad.

2) MÓDULO DE RECLUTAMIENTO SELECCIÓN Y CONTRATACIÓN DE PERSONAL

Requerimientos de usuario: Se requiere un módulo para optimizar el proceso de reclutamiento, selección y contratación del personal de la UV.

Actualmente este proceso se lleva a cabo realizando expedientes de candidatos y archivando la información en papel. Esta información se encuentra organizada por área y dividida a su vez en candidatos potenciales y no potenciales a cubrir la plaza.

El candidato debe proporcionar a la UV una serie de datos requeridos para poder ser evaluado y posteriormente tomar una decisión final.

Requerimientos del Sistema: El sistema debe permitir Ingresar datos de candidatos potenciales a la plaza o plazas que se encuentren libres. El sistema contará con una serie de pantallas para poder llevar el control de la información que proporciona el candidato, estas pantallas tomarán como base la información proporcionada en papel por el área de personal.

Pantalla de Consulta de Candidato

- Se requiere tener una pantalla con el resumen de la información general de los candidatos existentes en la base de datos, organizados de acuerdo a los siguientes status:
  - Registro
  - Examen Conocimiento
  - Examen Psicométrico
  - Estudio Socio-Económico
  - Examen médico
  - Contratado
  - No Contratado

- Los candidatos rechazados, se deben de borrar de acuerdo al criterio del encargado del área de personal.

- El encargado del área de personal debe de poder distinguir si un candidato es contratable o no contratable.

La información que el usuario necesita manejar para el proceso de reclutamiento, selección y contratación de personal, se describe a continuación .La información se organiza en grupos de datos con características comunes y cada uno de estos grupos serán una pantalla para el sistema.

Pantalla de Datos Generales

Esta pantalla debe contener la siguiente información:

- Apellido Paterno
- Apellido Materno
- Nombres
- Nacionalidad
- Lugar de nacimiento
- Estado Civil
- Sexo
- CURP
- IMSS
- Crédito Infonavit
- Religión
- Licencia
- Cartilla
- Domicilio Completo
- Teléfono
- Intereses profesionales
- Área de Interés

Pantalla de Datos familiares

- Nombre de los familiares con quien vive el candidato
- Dependientes económicos
- Tiene familiares trabajando en la UIE?
- En que dependencia?
- Datos del familiar que labora en la UIE (en caso de existir)

- Medio por el cual se enteró del empleo

Pantalla de Escolaridad

- Nivel de Escolaridad
    - Nombre de Escuela
    - Fecha de Inicio
    - Fecha de Finalización
    - ¿Obtuvo certificado?
    - Fecha de titulación
    - ¿Obtuvo certificado?

- Idiomas
    - Nombre del Idioma
    - Habla
    - Escribe
    - Traduce

- Estudios Actuales
    - Tipo de estudios que realiza actualmente
    - Universidad en que realiza sus estudios
    - Horario

Pantalla de Experiencia Laboral

- Nombre de la empresa
- Domicilio Fiscal
- Código postal
- Teléfono

- Nombre del Jefe Inmediato
- Puesto
- Actividades
- Fecha de Ingreso
- Fecha de Salida
- Ultimo Sueldo
- Motivo de separación
- Sindicato
- Cargo

Pantalla de Datos Socioeconómicos

- Ingresos Mensuales y de donde provienen
- Tipo d casa
- A cuanto asciende su valor
- Gastos mensuales
- Especificar si tiene deudas y a cuanto ascienden
- Especificar si tiene auto propio, modelo y marca
- Referencias
    - Nombre
    - Domicilio
    - Teléfono
    - Ocupación y tiempo de conocerlo

Pantalla de Datos de Salud

El candidato debe contestar el siguiente cuestionario:

- ¿A sido satisfactorio su estado de salud en los últimos dos años? (especificar)
- ¿A sido intervenido quirúrgicamente? (especificar)

- ¿Padece alguna enfermedad crónica? (especificar)
- ¿Padece alguna alergia? (especificar)
- ¿Utiliza algún aparato ortopédico?

En caso de emergencia avisar a:

Indicar los siguientes datos del a persona:

- Nombre
- Dirección
- Teléfono
- Parentesco

Pantalla de Beneficiarios

- Concepto
- Tipo de Persona
- Nombre
- Parentesco
- % de participación
- Sexo
- Fecha de nacimiento dirección

Se requieren un par de pantallas para información exclusiva del área de personal, con la siguiente información:

Pantalla de Información para Administración de Personal

- Nombre del puesto
- Número de puesto
- No. De plaza
- Centro de Costos
- Sector
- Categoría académica

- Status
- Sueldo
- Fecha de Ingreso
- Tipo de plaza (eventual o definitiva)

El sistema permitirá especificar las fechas de inicio y de vencimiento para el caso de las plazas indeterminadas.

- Tipo de Contrato
    - Indeterminado Administrativo de confianza
    - Determinado Administrativo de confianza
    - Indeterminado Administrativo Sindicalizado
    - Determinado Administrativo Sindicalizado
    - Indeterminado de Servicio
    - Determinado de Servicio
    - Indeterminado académico
    - Determinado Académico

El sistema permitirá especificar las fechas de inicio y de vencimiento para el caso de los contratos indeterminados.

Pantalla para control de Documentos

- Nombre del documento
- Fecha de Entrega
- Indicar si fue entregado ó no entregado

Las pantallas para reclutamiento, selección y contratación de personal no se imprimen y solo puede ser consultadas por el usuario perteneciente al área de personal.

MODULO DE CONTROL DE VENCIMIENTOS

Requerimientos de Usuario: Se requiere llevar un control de los vencimientos, es decir, contratos que por diferentes circunstancias están por finalizar, se solicita que el sistema de aviso del vencimiento con dos semanas de anticipación a la fecha de vencimiento del contrato. Actualmente, la persona que contrató al candidato es la responsable de estar al pendiente de la fecha de vencimiento de su contrato. Esto lo hace llevando un control en Excel con los nombres de los candidatos que tienen contratos eventuales así como las fechas de vencimiento.

Los motivos para que un contrato finalice, pueden ser los siguientes:

- Por período de prueba
- Por vencimiento de Contrato
- Por vencimiento por nombramiento
- Por vencimiento de horario eventual
- Por vencimiento de licencia
- Por vencimiento de período sabático
- Vencimiento de período sabático

Requerimientos del sistema: En la pantalla de vencimientos, se solicita presentar la siguiente información del contrato que este por vencer.

- Dependencia
- Nomina
- Fecha de vencimiento
- Nombre
- Apellido Paterno
- Apellido Materno
- Puesto Anterior
- Puesto Actual
- Status
-

- En la parte superior de la pantalla, se deberá mostrar el status del contrato, su periodo de inicio y de finalización y el tipo de vencimiento del mismo.

- Los contratos se podrán sortear por tipo de vencimiento y por status

- El sistema deberá permitir ingresar la evaluación del periodo de prueba de los contratos que este próximos a vencer

- Se debe permitir modificar la fecha de vencimiento del contrato, así como las condiciones generales de este y del empleado eventual.

MODULO DE ESTRUCTURA ORGANIZACIONAL

Requerimientos de Usuario: Actualmente los diagramas organizacionales son realizados en Excel por algún empleado del área de personal. El empleado realiza el diagrama agregando las entidades que contendrá el diagrama una por una.

- Se requiere un módulo que realice los diagramas automáticamente, tomando la información directamente de la base de datos

- Se requiere que el personal del área de administración de personal de la UV, pueda llevar el control de su estructura organizacional, esto es, poder conocer la dependencia de entidades, la jerarquía y el número de entidades que le reportan a una entidad superior.

- Se requiere conocer los empleados que laboran en una entidad específica, la dependencia entre ellos, la plaza y el puesto que ocupan y, si el responsable del área

de personal así lo requiere, el sueldo de cada uno de los empleados, así como la cantidad total que invierte un área específica en los sueldos mensuales de su personal.

Requerimientos de Sistema

- El sistema debe presentar la información de entidades y de empleados en forma de organigramas.
- El sistema debe de generar dos tipos diferentes de organigramas: por estructura y por empleado
- El encargado del área de personal debe de poder seleccionar los niveles de dependencia que mostrará el diagrama.
- Los diagramas deben de poderse imprimir, modificar, eliminar y guardar.

- El sistema debe de identificar cada organigrama generado por el encargado del área de personal de forma única

Conclusión

Una vez que se a realizado el estudio de factibilidad, la definición detallada de requerimientos y que se ha elegido el modelo de software que mas se apega a las necesidades del sistema, podemos entrar al diseño del mismo, de esta etapa se hablará en el capítulo 3 (Diseño del sistema).

CAPITULO 3

DISEÑO DEL SISTEMA

Objetivos:

El presente capítulo, tiene por objetivo definir un conjunto de descripciones del sistema como son:

- El Diseño de la Arquitectura del sistema.
- El Diseño de los procesos que cubre el sistema
- El Diseño de la estructura de base de datos y usuarios del sistema.

3.1 Arquitectura del sistema

La primera fase para el diseño arquitectónico del sistema se refiere a la descomposición del sistema en un conjunto de subsistemas

Para comenzar, identificaremos el entorno del sistema, es decir, los sistemas con los que convive y los subsistemas en que se divide.

Como ya se mencionó en el capítulo anterior, la universidad UV actualmente cuenta con dos sistemas que ya se encuentran funcionando. Dichos sistemas son:.

- El **sistema de Administración de empleados de la UV**, el cual se encarga de llevar el control de toda la información referente a los empleados que laboran en la Universidad UV, el puesto y plaza que cubren, sueldos, horario y datos referentes tanto a docentes como administrativos de la Universidad.

- El **Sistema de pago de Nomina**, que se encarga de controlar y administrar todos los pagos y prestaciones que se realizan a los empleados

El sistema que se trata en el presente trabajo (**Sistema para controlar la información que se maneja en el área de personal de la UV**), compartirá una interfáz principal en donde el usuario podrá seleccionar a cual de los tres sistemas desea ingresar, también compartirá información de la base de datos principal del área de Recursos Humanos.

A continuación se muestra un diagrama de la arquitectura general del sistema, los subsistemas con que cuenta, -mismos que se definieron en el capitulo de análisis- y los sistemas con los cuales compartirá información.

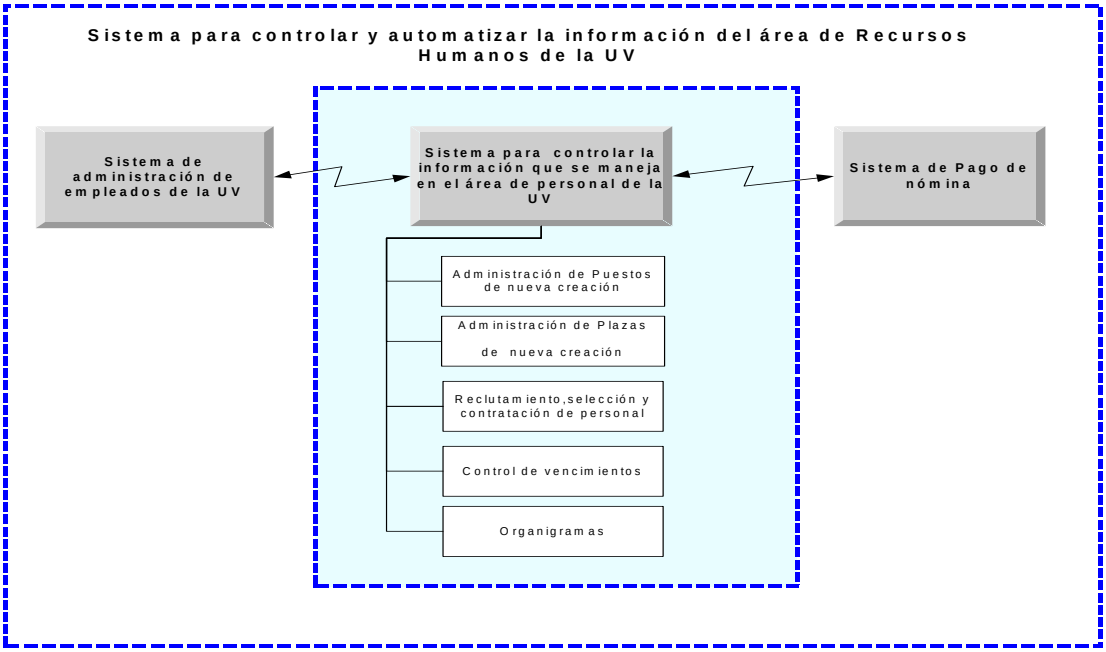


Figura 3.1 *Arquitectura del sistema*

Como se puede ver en la figura 3.1, el sistema estará dividido en 5 subsistemas, mismos que se detallan a continuación:

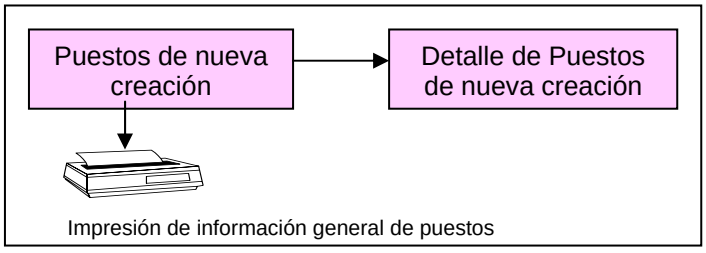


Figura 3.2 *Modulo de Puestos*

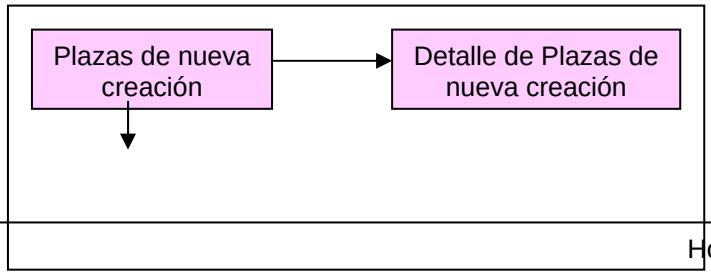




Figura 3.3 *Modulo de Plazas*

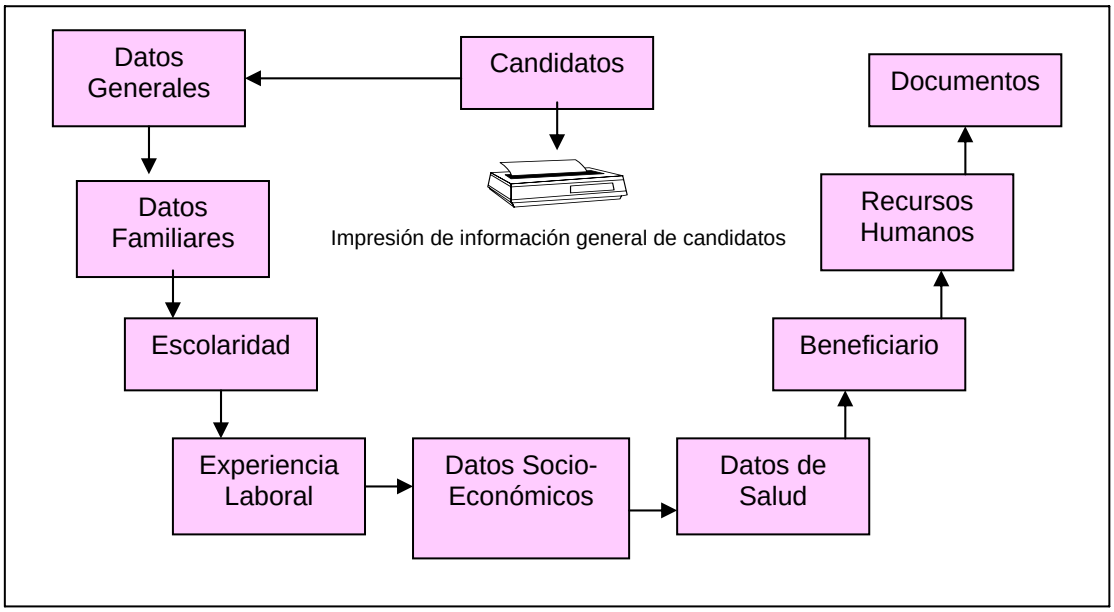


Figura 3.4 *Módulo de Candidatos*

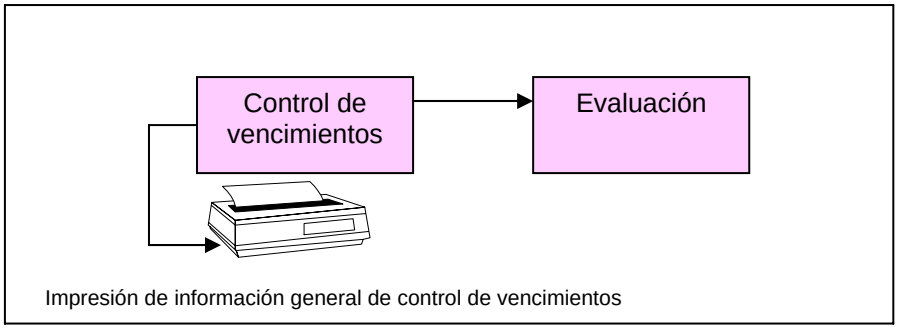


Figura 3.5 *Módulo de Control de Vencimiento*

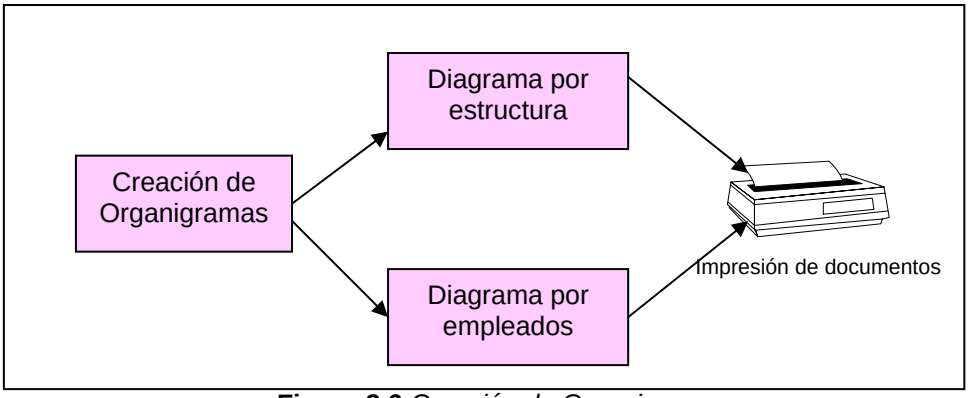


Figura 3.6 *Creación de Organigramas*

El funcionamiento de cada uno de los subsistemas mencionados anteriormente, se detalla a con los siguientes organigramas:

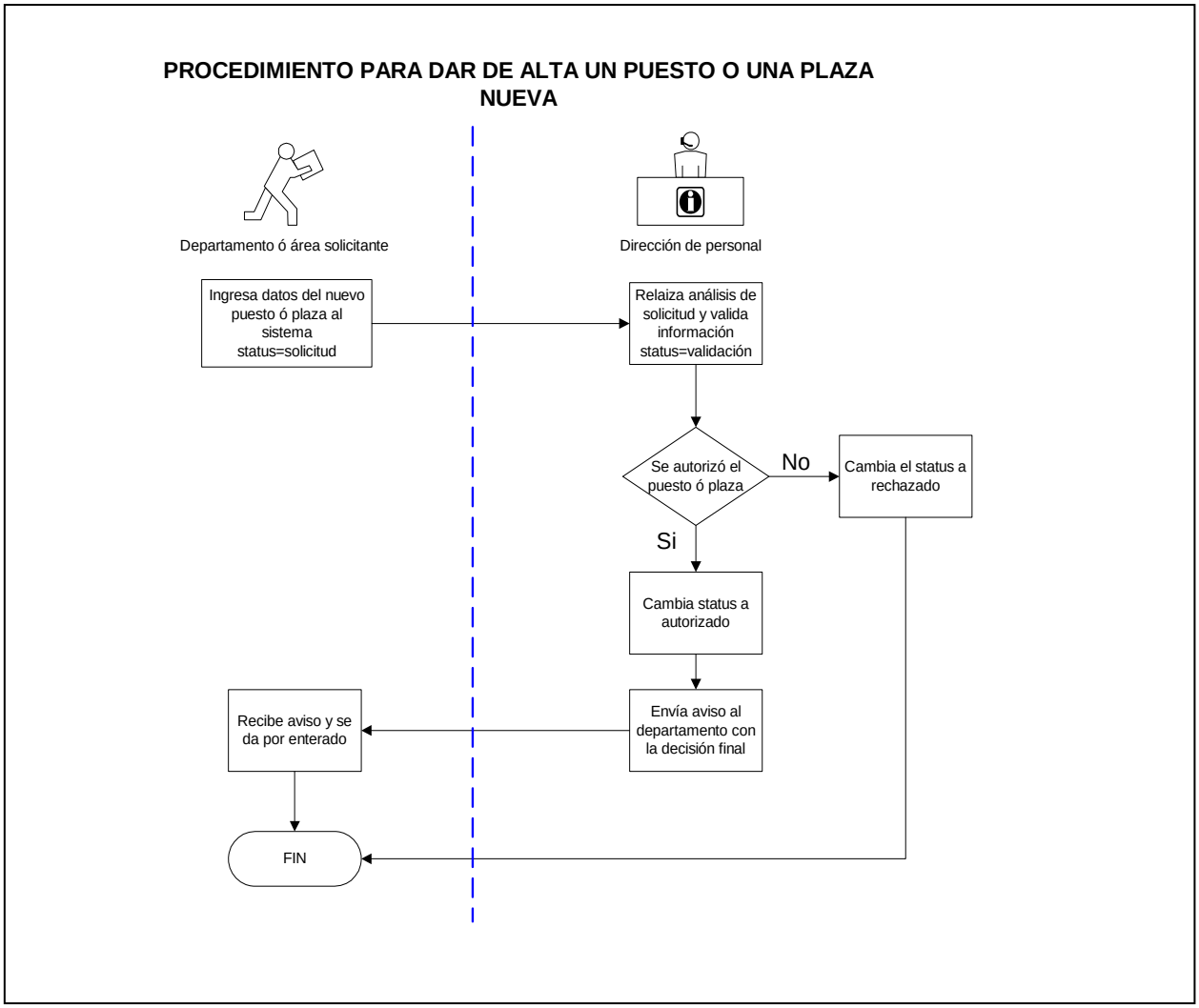


Figura 3.7 *Procedimiento para Alta de puestos y plazas*

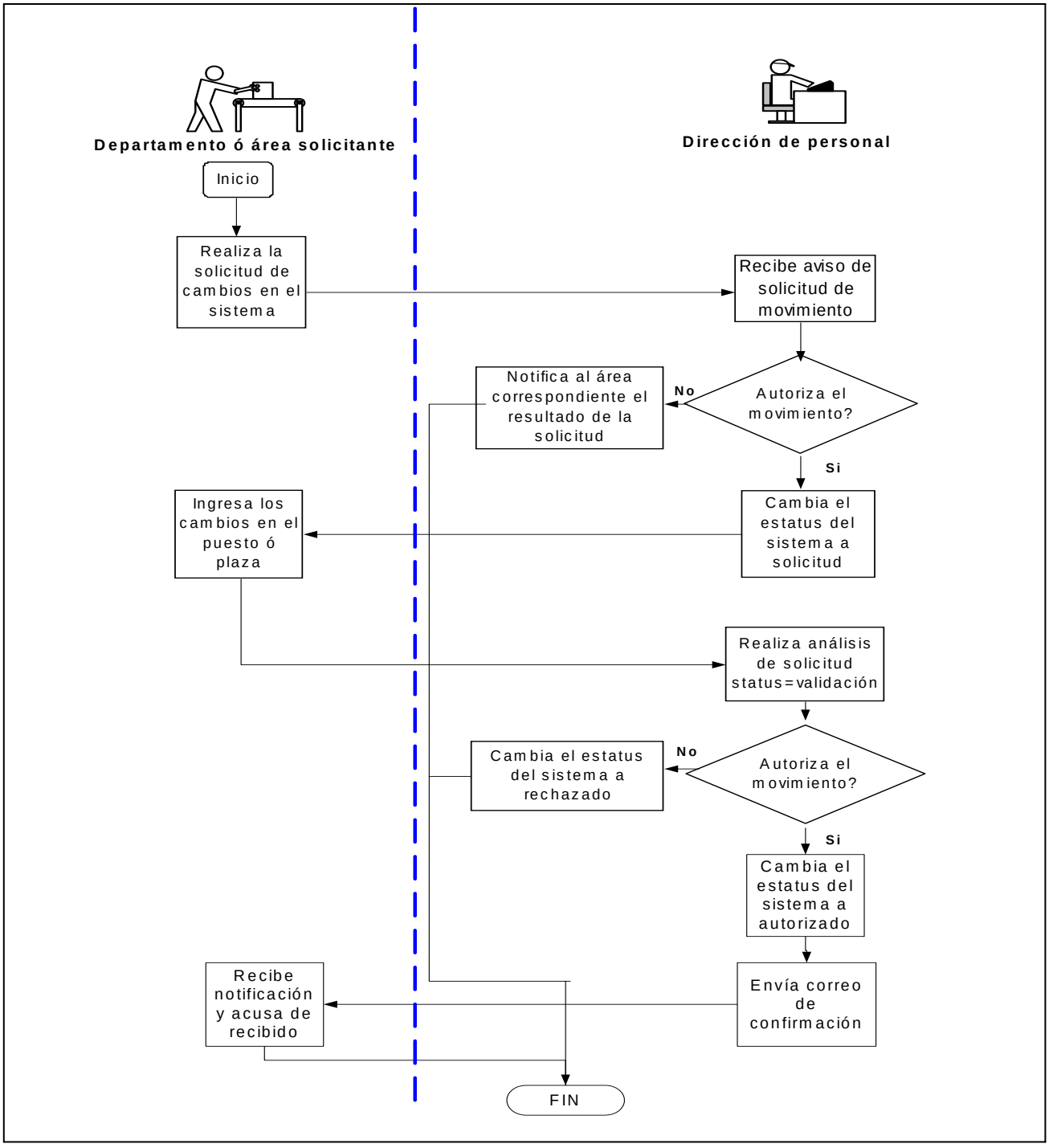


Figura 3.8 *Movimiento de puestos y plazas*

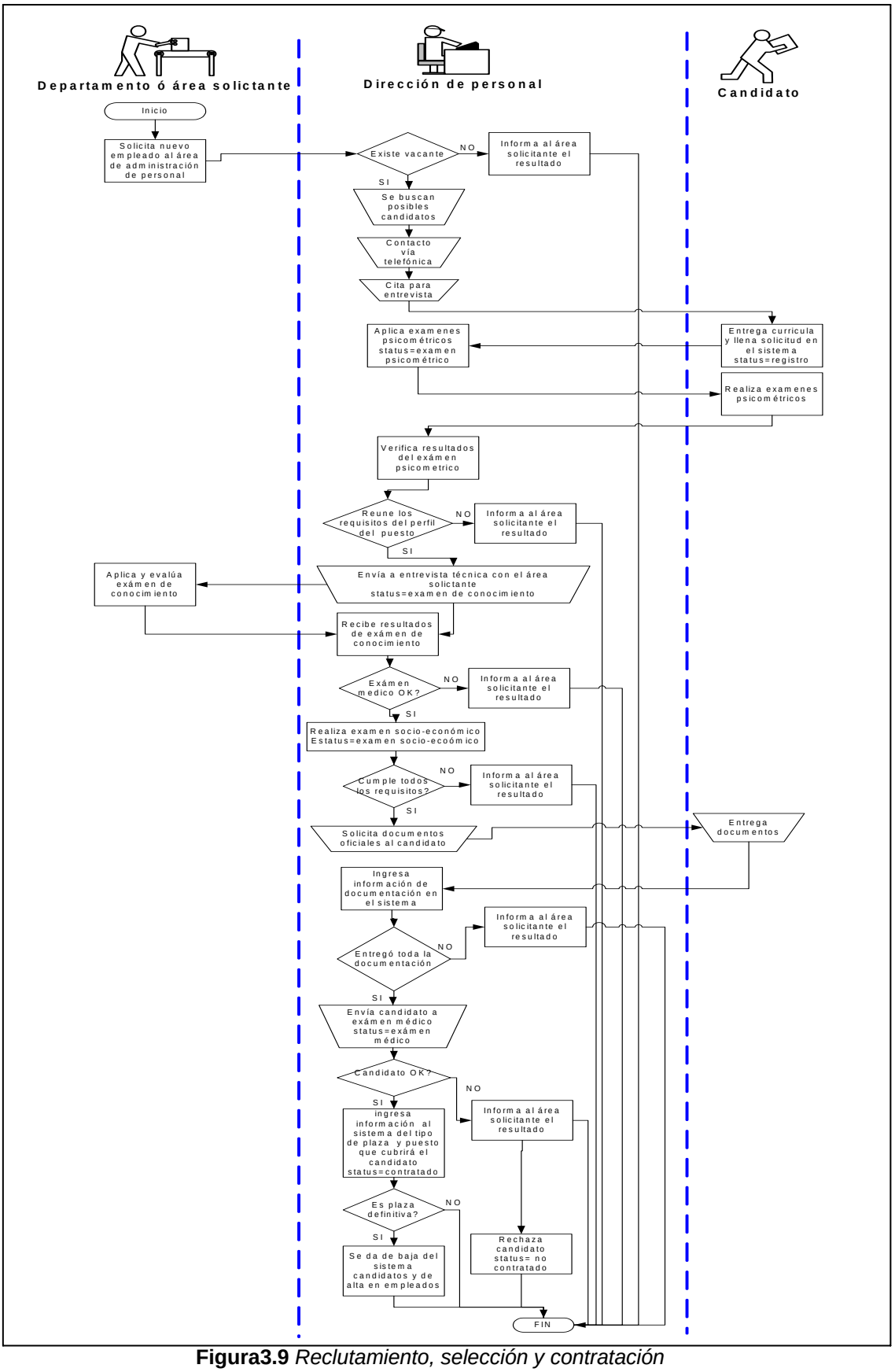


Figura3.9 *Reclutamiento, selección y contratación*

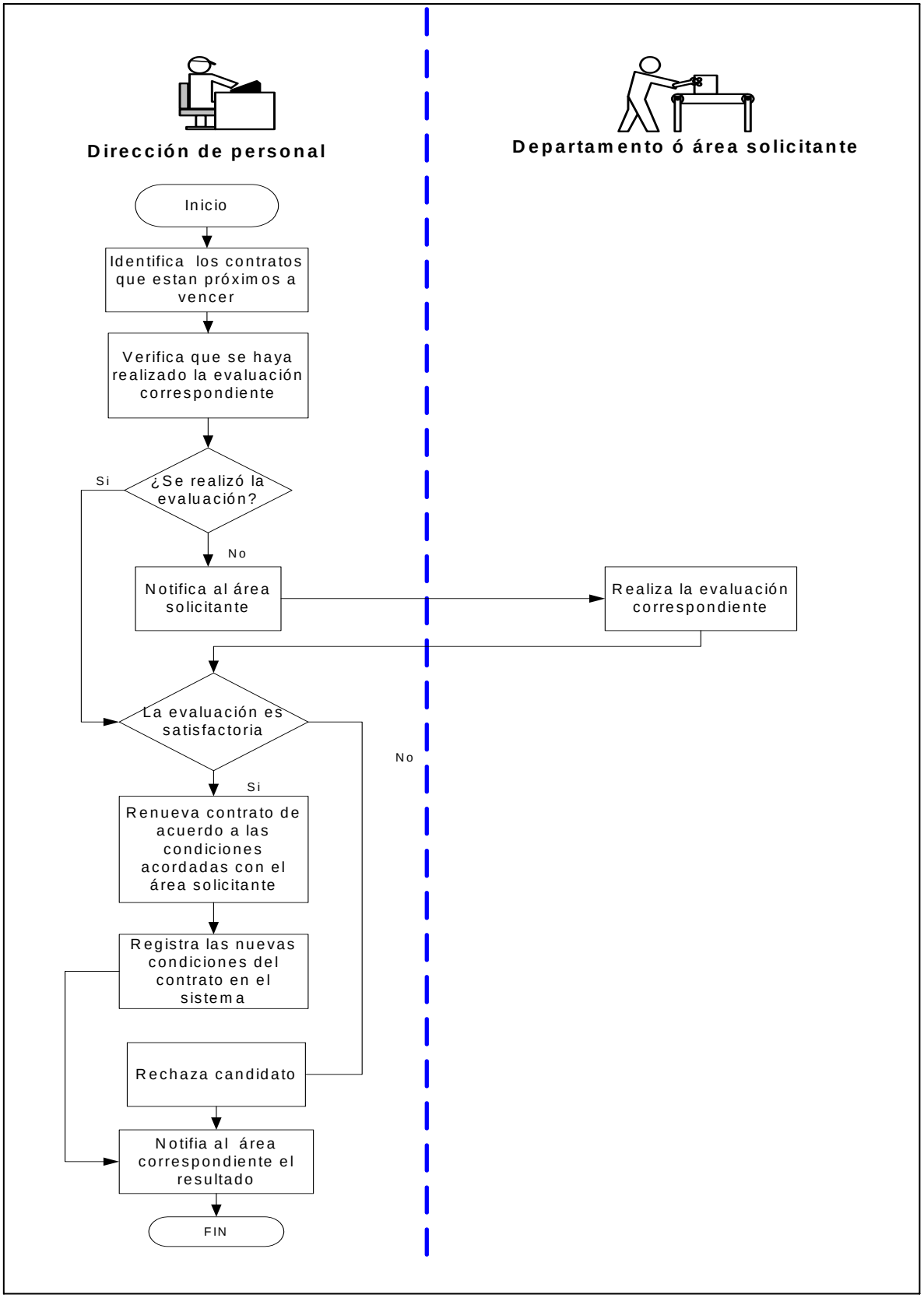


Figura 3.10 *Vencimientos*

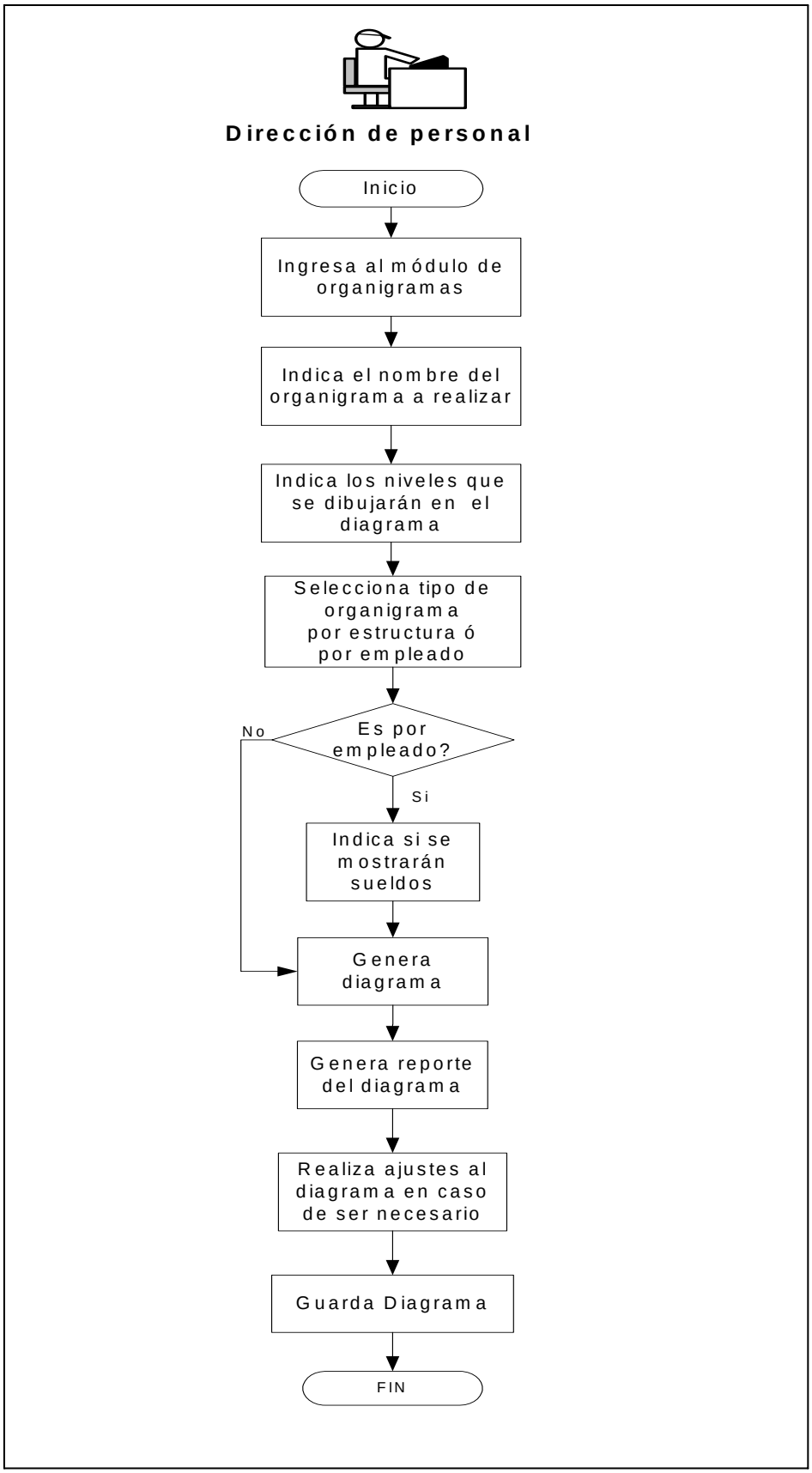


Figura 3.11 *Organigramas*

3.2 Diccionario de Datos

El contenido del diccionario puede considerarse como “datos acerca de los datos” (los cuales comúnmente reciben el nombre de meta-datos), es decir, definiciones de otros objetos de la base de datos.

En particular, todos los diversos esquemas (externo, conceptual e interno), se almacenan físicamente en el diccionario, tanto en forma fuente como en forma objeto. Un diccionario amplio incluirá también las referencias cruzadas que indican, por ejemplo que partes de datos utiliza cada programa, que informes necesita cada departamento, etc. De hecho, el diccionario puede integrarse a la base de datos que describe, y, por tanto, incluir su propia descripción.

Debe ser posible consultar el diccionario de la misma manera que cualquier otra base de datos, de modo que, por ejemplo, el DBA pueda describir con facilidad que programas tienen probabilidad de ser afectados por un cambio propuesto al sistema.

Sus principales funciones son las siguientes:

-Describe todos los elementos en el sistema.
-Los elementos se centran en los datos.
-Comunica los mismos significados para todos los elementos del sistema.
-Documenta las características del sistema.
-Facilita el análisis de los detalles para evaluar las características y determinar cómo deben realizarse los cambios.
-Localiza errores y omisiones del sistema.

A continuación, se presenta el diccionario de datos que se elaboró después de analizar los requerimientos para el sistema de la UV.

Cabe mencionar que las tablas que aquí se detallan, son las que se agregaron a la base de datos de datos de Recursos Humanos de la UV.

DIAGRAMA Tabla para guardar la información de los diagramas creados en la UV

	Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
🔑	Diagrama_Id	int	4	
	Diagrama_Nombre	varchar	100	
	Diagrama_Tipo	int	4	
	fecha_creacion	datetime	8	✓

DIAGRAMA_OBJETO Tabla para guardar la información de los objetos que conforman un diagrama

	Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
🔑	Diagrama_Objeto_Id	int	4	
	Diagrama_Id	int	4	✓
	Diagrama_Objeto_Nombre	varchar	100	
	num_plaza	smallint	2	✓
	folio_entidad	smallint	2	✓
	Reporte_T	varchar	50	✓

OBJETO_CELL Tabla para guardar el valor de las propiedades de un objeto de un idagrama

	Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
🔑	Objeto_Cell_Id	int	4	
	Diagrama_Objeto_Id	int	4	
	Objeto_Cell_Name	varchar	100	
	Objeto_Cell_Value	varchar	100	✓

CANDIDATO Guarda la información general de los candidatos

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
candidato_id	int	4	
Candidato_Fecha_Sol	varchar	10	✓
Candidato_Idioma1	varchar	50	✓
Candidato_Idioma1_Habla	varchar	10	✓
Candidato_Idioma1_Escribe	varchar	10	✓
Candidato_Idioma1_Traduce	varchar	10	✓
Candidato_Idioma2	varchar	10	✓
Candidato_Idioma2_Habla	varchar	10	✓
Candidato_Idioma2_Escribe	varchar	10	✓
Candidato_Idioma2_Traduce	varchar	10	✓
Candidato_Idioma3	varchar	10	✓
Candidato_Idioma3_Habla	varchar	10	✓
Candidato_Idioma3_Escribe	varchar	10	✓
Candidato_Idioma3_Traduce	varchar	10	✓
vCandidato_Puesto_Sol	varchar	255	✓
dlCandidato_Sueldo_Sol	decimal	9	✓
vCandidato_Nombre	varchar	30	✓
vCandidato_Ap_Paterno	varchar	30	✓
vCandidato_Ap_Materno	varchar	30	✓
vNacionalidad	varchar	4	✓
sCandidato_Lugar_Nac	smallint	2	✓
iEstado_Civil_Id	int	4	✓
iReligion_Id	int	4	✓
cCandidato_Sexo	char	1	✓
cCandidato_RFC	char	13	✓
cCandidato_CURP	char	18	✓
vCandidato_IMSS	varchar	20	✓
bCandidato_Infonavit	bit	1	✓
vCandidato_Infonavit_No	varchar	20	✓
vCandidato_Cartilla	varchar	10	✓
cCandidato_Licencia_Tipo	char	1	✓
vCandidato_Licencia_No	varchar	20	✓
vCandidato_Int_Profesionales	varchar	255	✓
vCandidato_Calle	varchar	60	✓
vCandidato_Colonia	varchar	60	✓
vCandidato_Delegacion	varchar	150	✓
sEstado	smallint	2	✓

CAN_DOCTO Guarda la información de los documentos entregados por los candidatos

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
Can_Docto_Id	int	4	
Candidato_Id	int	4	✓
documento	smallint	2	✓
Can_Docto_Dsc	varchar	50	✓
Can_Docto_Entregado	bit	1	✓
Can_Docto_Fec_Entrega	varchar	10	✓

CAN_EXP_LABORAL Guarda información de la experiencia laboral de los candidatos

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
Can_Exp_Laboral_Id	int	4	
Candidato_Id	int	4	✓
Can_Exp_Laboral_Empresa	varchar	100	✓
Can_Exp_Laboral_Puesto	varchar	100	✓
Can_Exp_Laboral_Calle	varchar	60	✓
Can_Exp_Laboral_Colonia	varchar	60	✓
Can_Exp_Laboral_Delegacion	varchar	150	✓
estado	int	4	✓
Can_Exp_Laboral_CP	char	5	✓
Can_Exp_Laboral_Telefono	varchar	20	✓
Can_Exp_Laboral_Jefe	varchar	100	✓
Can_Exp_Laboral_Fec_Inicio	varchar	10	✓
Can_Exp_Laboral_Fec_Fin	varchar	10	✓
Can_Exp_Laboral_Sueldo	decimal	9	✓
Motivo_Separacion_Id	int	4	✓
Can_Exp_Laboral_Sindicato	bit	1	✓
Can_Exp_Laboral_Sindicato_Nombre	varchar	100	✓
Can_Exp_Laboral_Sindicato_Cargo	varchar	100	✓

NIVELES_ESTUDIOS Guarda información de los niveles de estudios de los candidatos

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
nivel_estudios	int	4	
descripcion	varchar	50	✓

CAN_BENEFICIARIO Guarda información de los beneficiarios de un candidato

Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
Can_Beneficiario_Id	int	4	
Candidato_Id	int	4	✓
Concepto_Id	int	4	✓
Tipo_Persona_Id	int	4	✓
Can_Beneficiario_Nombre	varchar	30	✓
Can_Beneficiario_Ap_Paterno	varchar	30	✓
Can_Beneficiario_Ap_Materno	varchar	30	✓
Can_Beneficiario_Sexo	char	1	✓
Can_Beneficiario_Fec_Nacimiento	varchar	10	✓
parentesco	int	4	✓
Can_Beneficiario_Participacion	decimal	9	✓
Can_Beneficiario_Calle	varchar	60	✓
Can_Beneficiario_Colonia	varchar	60	✓
Can_Beneficiario_Delegacion	varchar	150	✓
estado	smallint	2	✓
Can_Beneficiario_CP	char	5	✓

PARENTESCO Descripción del tipo de parentesco de los beneficiarios con el candidato

	Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
🔑	parentesco	int	4	
	nombre	varchar	50	✓

CALIF_EVALUACION Guarda información de la evaluación realizada al candidato

	Nombre de columna	Tipo de datos	Longitud	Permitir valores nulos
🔑	Calif_ID	int	4	
	calificacion	varchar	30	✓
	candidato_id	int	4	✓

3.3 Modelo Entidad-Relación

El modelo entidad-relación es una técnica empleada en la definición de las necesidades de información en una organización. Estos diagramas proveen de una fuente fundamental para garantizar un alto nivel de calidad, asegurando que se cubren las necesidades de la empresa. El modelado entidad relación involucra la identificación de los elementos de importancia en una organización (entidades), las propiedades de estos elementos (atributos) y como están relacionados unos con otros (relaciones).

Los objetivos del modelado entidad-relación son:

- Proporcionar un modelo preciso de las necesidades de información, el cual actúa como una base para el desarrollo de nuevas características al sistema.

- Proporcionar un modelo independiente de cualquier almacén de datos y método de acceso, para permitir que las decisiones objetivas sean hechas de acuerdo a las técnicas de implementación y puedan coexistir con el sistema existente

El diagrama entidad-relación es una parte que modela el negocio o empresa y es producido en el estado de análisis, el cual es parte de todo el ciclo de vida de un sistema. Este diagrama nos muestra gráficamente las entidades y las relaciones vitales entre ellas.

El proceso de crear este diagrama es llamado “modelado de entidades”.Los términos modelo de entidades, modelo entidad-relación o modelo entidad / relación son todos sinónimos del diagrama entidad-relación.

3.3.1 Convenciones Básicas y Definiciones Empleadas

Entidad: Una entidad es un elemento u objeto con significado propio, ya sea real o imaginaria y cuya información se necesita conocer y guardar.
Una entidad es representada gráficamente por una caja rectangular con las orillas redondeadas y con el nombre de la entidad dentro, En singular y con letras mayúsculas, como se muestra en la figura 3.12

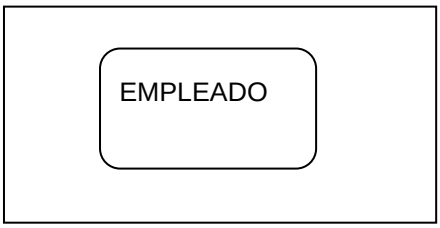


Figura 3.12 *Representación gráfica de una entidad*.

Cada entidad debe tener un identificador único.

Cuando se modelan las entidades no hay ningún significado especial dado el tamaño de una entidad, las cajas pueden ser alargadas, ensanchadas o acortadas simplemente para ayudar a la lectura de los diagramas.

Relación: Es una asociación con un nombre asignado y que representa alguna característica entre dos entidades. Una relación es binaria, en el sentido de que siempre es una asociación entre exactamente dos entidades o entre una entidad y ella misma.

Cada relación tiene dos extremos en cada uno de los cuales hay:

- Un nombre
- Un grado o cardinalidad (cuantos)
- Opcionalidad (Opcional u Obligatoria)

Estas propiedades son empleadas para describir la asociación en un extremo, ambos extremos deben estar definidos.

Una relación es representada por una línea que une dos entidades o bien recursivamente une una entidad consigo misma.

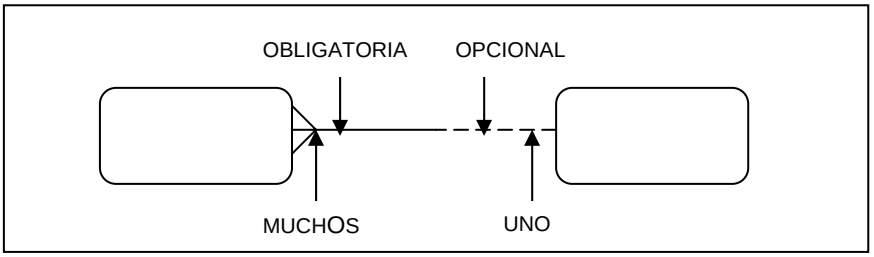


Figura 3.13 *Representación de relaciones entre entidades*

Para representar el grado de muchos la línea de relación se une a la entidad en tres puntos conocido como “pata de gallo”.Para representar el grado uno la línea se une en un solo punto.

Los nombres de las relaciones deben ser colocados en cada extremo de la línea y se colocan cerca de la entidad apropiada y en letras minúsculas.

La obligatoriedad en una relación es representada por una línea continua y la opcionalidad se representa con una línea discontinua.

Cuando la relación es obligatoria se lee “debe ser o estar”.

Cuando la relación es opcional se lee “puede ser o estar”

Si el grado es muchos se lee “uno o más”

Si el grado es uno se lee “uno y solo uno”

El tipo de relación mas común es la que tiene el grado de muchos a uno y es obligatoria del lado de muchos y opcional de lado de uno.

3.3.2 ATRIBUTOS

Un atributo es cualquier detalle que nos sirve para calificar o expresar el estado de una entidad o bien cualquier descripción de una cosa que tenga significado dentro del sistema. Un atributo puede ser texto, números, fechas, una imagen etc.

En un diagrama entidad-relación no es necesario mostrar los atributos, pero esto podría ser útil para distinguir fácilmente las entidades.

Los atributos deben cumplir con las siguientes reglas:

- Un atributo describe una entidad
- No puede haber atributos repetidos en una misma entidad
- Pueden ser un identificador único, útil al sistema

3.3.3 IENTIFICADORES ÚNICOS

Cada entidad debe de ser identificada de forma única, de tal forma que cada elemento de esa entidad es separado de todos los demás elementos de la misma y de otras entidades e indicado de forma única. El identificador único puede ser un atributo ó combinación de atributos.

Llave primaria. El atributo que identifica de manera única a un registro. Esto es, no debe haber dos tuplas que tengan el mismo valor en todos los valores de la llave. Por lo tanto, con sólo conocer el valor de K para un tupla será suficiente para identificarlo de manera única.

Llave candidata. Atributo o conjunto de atributos que podrían servir como llaves primarias.

Llave secundaria. Todas aquellas llaves candidatas que no se eligieron como llave primaria. Son llaves que tiene todas las características para ser primarias, pero que por una razón o por otra no fueron tomadas como tales, ya que hubo otra (s) que cumplían mejor con ese objetivo.

Llave extranjera o foránea. Llave primaria que es la llave primaria en otra relación. Son la materialización de las asociaciones entre las entidades. Son llaves que son compartidas por dos tablas para lograr una relación entre ellas.

A continuación, se muestra el diagrama entidad-relación realizado para el sistema de la UV:

3.3.4 DIAGRAMA ENTIDAD-REALACION DE LA UV

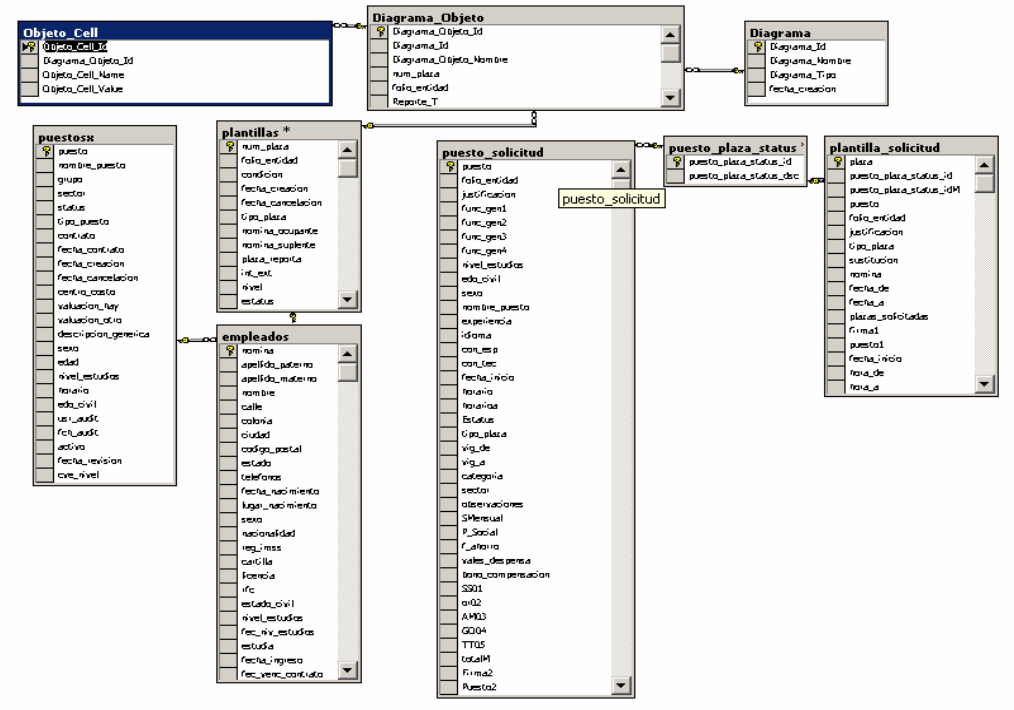


Figura 3.14 *Diagrama Entidad-Relación 1*

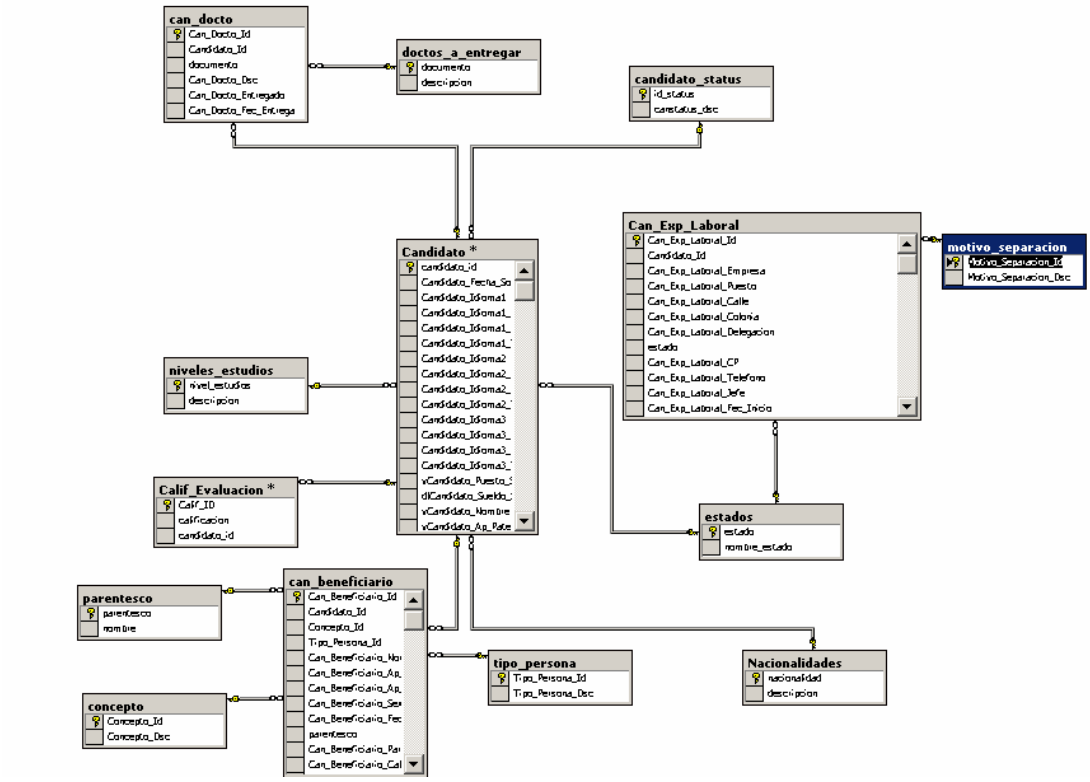


Figura 3.15 *Diagrama Entidad-Relación 2*

3.4 USUARIOS

Para acceder a una base de datos, el usuario debe estar dado de alta en esa base de datos. Cada usuario que se define en la base de datos, debe tener diferentes roles, de manera que solo puedan ver y accesar a la información que deben y necesitan accesar, de esta forma, se protege la información de un mal uso por parte de los usuarios.

A continuación, se definen algunos usuarios necesarios en una base de datos:

Operadores: Los operadores pueden respaldar y cargar todas las bases de datos y logs de transacciones en el servidor.

Propietarios de la Base de Datos:

- Es el dueño de la base de datos
- Es quien la crea y a quien se le ha transferido la propiedad de una base de datos
- Puede agregar o retirar usuarios de la base de datos.
- Otorga y revoca permisos a usuarios para crear objetos en la base de datos
- Tiene todos los privilegios sobre todos los objetos de la base de datos

Oficiales de Seguridad del Sistema:

- Crean Logines en SQL Server
- Cambian passwords
- Crean, otorgan y revocan roles de usuario
- Bloquean y desbloquean logines

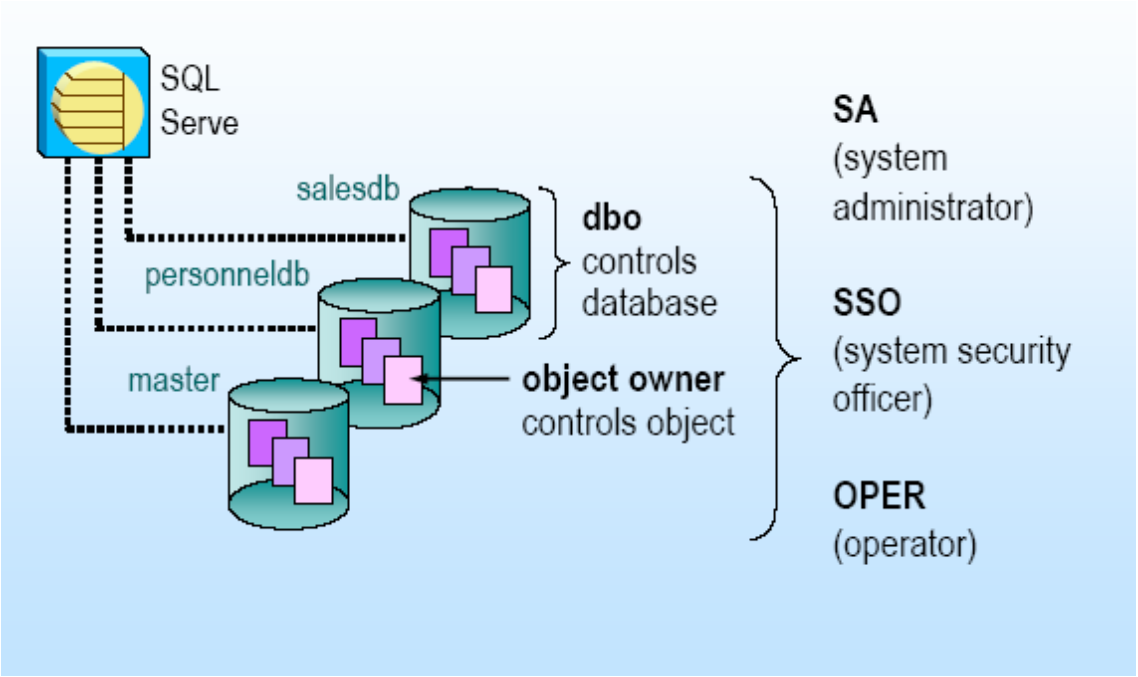


Figura 3.16 *Usuarios de una Base de Datos*

Para la base de datos del sistema que se detalla en el presente trabajo, se definieron diferentes roles de usuarios de acuerdo a los requerimientos del cliente. Cada usuario tendrá los permisos para manipular la información de la base de datos de acuerdo a la tabla de permisos de usuarios que se muestra a continuación:

Tablas/Usuarios	DBO				SSO				OPER				ENCARGADO DE ÁREA				ADMON. DE PERSONAL				CANDIDATOS			
	S	U	I	D	S	U	I	D	S	U	I	D	S	U	I	D	S	U	I	D	S	U	I	D
Diagrama	■	■	■	■	■	■	■	■	■	■	■	■					■	■	■	■				
Diagrama_Objeto	■	■	■	■	■	■	■	■	■	■	■	■					■	■	■	■				
Objeto_Cell	■	■	■	■	■	■	■	■	■	■	■	■					■	■	■	■				
Entidades	■	■	■	■	■	■	■	■	■	■	■	■	■				■							
Plantillas	■	■	■	■	■	■	■	■	■	■	■	■	■				■	■	■					
Puestos	■	■	■	■	■	■	■	■	■	■	■	■	■				■	■	■					
Plantilla_Solicitud	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
Puesto_Solicitud	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
Empleados	■	■	■	■	■	■	■	■	■	■	■	■	■				■	■	■	■				
Candidatos	■	■	■	■	■	■	■	■	■	■	■	■	■				■	■	■	■			■	

SELECT =	S
UPDATE =	U
INSERT =	I
DELETE =	D

Figura 3.16 *permisos a usuarios*

3.5 RESPALDOS DE LA INFORMACIÓN

El tener un esquema de respaldos definidos en la empresa, garantiza que en caso de un evento de perdida de datos en la base de datos principal, se cuente con dispositivos alternos que permitan la recuperación de la mayoría de la información con perdidas permanentes mínimas.

Para realizar un esquema de respaldos, se deben de tomar en consideración varios puntos, entre los que están:

- Tamaño de la base de datos
- Grado de modificaciones que se realizan ó bien, si la base de datos es primordialmente de consultas
- Importancia de la información que se almacena en la base de datos

Para el caso de este proyecto, la base de datos se modifica constantemente, por lo que es necesario tener un esquema de respaldos frecuente, la estrategia que se lleva a cabo, es la siguiente:

- Se realiza un respaldo de la información completo una vez por semana
- Se realiza un respaldo diferencial al final del día de trabajo
- El respaldo del log de transacciones se lleva a cabo cada hora durante el día de trabajo de (8:00 a.m. a 6:00 p.m.)

En caso de que la base de datos se dañara, la estrategia de recuperación sería la siguiente:

- Si es posible, en ese momento se respalda el log de transacciones
- Se restaura el último respaldo completo de la base de datos
- Se restaura el respaldo diferencial que se realizó al final del último día de trabajo. Este respaldo contiene todos los cambios que se realizaron, desde el último respaldo completo.
- Se aplican los respaldos del log de transacciones que se llevaron a cabo el día en que la Base de Datos se dañó
- Se restaura el respaldo del log de transacciones que se realizó al principio del proceso de restauración para asegurar la consistencia de la información.

Conclusión:

En el presente capítulo, se diseñaron los requerimientos del usuario de la UV, estableciendo una arquitectura del sistema. Se presentaron diagramas de los módulos principales que conformarán al sistema, así como de los procesos principales. También se realizó el diseño de la base de datos por medio de un diagrama entidad-relación, el diccionario de datos para tener una visión general de la estructura de la base de datos del sistema.

CAPITULO 4

DESARROLLO DEL SISTEMA

Objetivos:

- Mostrar las funciones generales utilizadas para el desarrollo del sistema
- Mostrar el flujo general de las pantallas que conforman el sistema así como la función principal de cada una de estas.

4.1 Funciones Generales.

A) La siguiente función realiza la carga de FlexGrids a partir de un Query que se le pasa como parámetro.

Los encabezados se toman de los nombres de las columnas de los procedimientos almacenados.

Declaración
Public Sub CargaFGrid(ByVal frmGeneral As Form, ByRef grdGeneral As MSFlexGrid, ByVal SQL_Query As String)

Parámetros

ByVal frmGeneral As Form,	Forma donde se encuentra el Grid
ByRef grdGeneral As MSFlexGrid,	FlexGruid a llenar
ByVal SQL_Query As String	Query que llenara el control.

Variables globales
<Ninguna>

Código de Regreso
Los datos que se cargarán en el FlexGrid.

```
Public Sub CargaFGrid(ByVal frmGeneral As Form,
ByRef grdGeneral As MSFlexGrid, ByVal SQL_Query As
String)
Dim iCol As Integer
Dim iRow As Integer
Dim iTmp As Single

Set RsQuery = New ADODB.Recordset

RsQuery.ActiveConnection = bd

RsQuery.Open SQL_Query

 'Se formatea el Grid
 grdGeneral.Visible = False

 FormateaFGrid frmGeneral, grdGeneral

 'Se llena el grid con los valores del Query
 While Not (RsQuery.EOF)
    grdGeneral.Col = FGRID_COL_ENCABEZADO
    grdGeneral.Text = grdGeneral.Row
    For iCol = 1 To RsQuery.Fields.Count
      grdGeneral.Col = iCol
      grdGeneral.Text = NULL_STRING &
RsQuery.Fields(iCol - 1).Value
'Se normaliza porque el Rs empieza en 0
      iTmp =
(frmGeneral.TextWidth(Len(Trim(grdGeneral.Text))) *
FGRID_LEN_DATA_TEXT_FACTOR)
      If (grdGeneral.ColWidth(iCol) < iTmp) Then
        grdGeneral.ColWidth(iCol) =
frmGeneral.TextWidth(Len(Trim(grdGeneral.Text))) *
FGRID_LEN_DATA_TEXT_FACTOR * 2
      End If
    Next iCol
    RsQuery.MoveNext
    If Not (RsQuery.EOF) Then
      grdGeneral.Rows = grdGeneral.Rows + 1
      grdGeneral.Row = grdGeneral.Rows - 1
    End If
  Wend

  grdGeneral.RowSel = FGRID_ROW_ENCABEZADO
  grdGeneral.Row = FGRID_ROW_ENCABEZADO + 1

  grdGeneral.Visible = True

  RsQuery.Close
  Set RsQuery = Nothing
Exit Sub
ManejaErrores:
  MsgBox "Error al cargar los datos", vbCritical, "Mensaje
del Sistema"
  Set RsQuery = Nothing
End Sub
```

B) La siguiente función realiza la carga de combo boxes a partir de un Query que se pasa como parámetro. Opcionalmente se puede incluir un texto extra como “TODOS” o “NINGUNO”, pasándolo por parámetro y lo coloca como primera opción.

Declaración
Public Function CargaCombo(cboGeneral As ComboBox, SQL_Query As String, Optional szExtraItem As String) As Integer

Parámetros

cboGeneral As ComboBox,	Control a llenar
SQL_Query As String	Query que llenará el control.
Optional szExtraItem As String	Parámetro opcional que indica si se agrega un elemento extra en el combo

Variables globales
<Ninguna>

Código de Regreso
El número de términos cargados en el control

```
Public Function CargaCombo(cboGeneral As ComboBox, SQL_Query As String, Optional szExtraItem As String) As Integer
Dim iCounter As Integer

  On Error GoTo ManejaErrores

  Set RsQuery = New ADODB.Recordset

  cboGeneral.Clear

  RsQuery.ActiveConnection = bd

  If (szExtraItem <> "") Then ' En caso de que se requiera agregar un elemento extra (TODOS, NINGUNO)
    cboGeneral.AddItem szExtraItem
    cboGeneral.ItemData(cboGeneral.NewIndex) = 0
  End If

  iCounter = 0
  RsQuery.Open SQL_Query
  'Se llena el combo con el resultado del Query
  While Not (RsQuery.EOF)
    cboGeneral.AddItem RsQuery.Fields(1).Value
    cboGeneral.ItemData(cboGeneral.NewIndex) = RsQuery.Fields(0).Value
    RsQuery.MoveNext
    iCounter = iCounter + 1
  Wend

  RsQuery.Close

  If (cboGeneral.ListCount > 0) Then
    cboGeneral.ListIndex = 0
  End If

  Set RsQuery = Nothing
  CargaCombo = iCounter
Exit Function

ManejaErrores:
  MsgBox "Error al cargar los datos", vbCritical, "Mensaje del Sistema"
  Set RsQuery = Nothing
  CargaCombo = -1
End Function
```

C) Esta función efectúa la separación de un string con delimitador igual al que se pasa como parámetro y lo almacena en un arreglo que también es pasado como parámetro.

Es importante saber que el arreglo no necesita estar dimensionado porque la función realiza esta operación, dejando siempre al arreglo dimensionado al número de elementos que se hayan insertado.

Declaración
Function Parser (ByVal vsWork As String, ByVal vsDelim As String, rasArray() As String) As

Parametros

ByVal vsWork As String	String a descomponer
ByVal vsDelim As String	String utilizado como delimitador
'rasArray() As String	Arreglo para insertar los términos.

Variables globales
<Ninguna>

Código de Regreso
El número de términos cargados al arreglo

```
Const MAX_EsParse_ARRAY_LEN = 10
 Dim iBeginPtr As Integer
 Dim iElement As Integer
 Dim iEndPtr As Integer

 iBeginPtr = FIRST_STR_POS
 iElement = NULL_INTEGER

 For iEndPtr = FIRST_STR_POS To Len(vsWork)
   If InStr(vsDelim, Mid$(vsWork, iEndPtr,
FIRST_STR_POS)) Then
     If ((iElement Mod MAX_EsParse_ARRAY_LEN) =
NULL_INTEGER) Then
       ReDim Preserve rasArray(iElement +
MAX_EsParse_ARRAY_LEN)
     End If
     iElement = iElement + 1
       rasArray(iElement) = Mid$(vsWork, iBeginPtr,
iEndPtr - iBeginPtr)
       iBeginPtr = iEndPtr + 1
     End If
   Next iEndPtr
   'Inserta el último termino en caso de que exista
   If (iBeginPtr <= Len(vsWork)) Then
     iElement = iElement + 1
     ReDim Preserve rasArray(iElement)
     rasArray(iElement) = Mid$(vsWork, iBeginPtr)
   Else
     ReDim Preserve rasArray(iElement)
   End If

   Parser = iElement

End Function
```

D) Selecciona el texto del TextBox pasado por parámetro

Declaración
Public Sub Selecciona (txtTexto As TextBox)

Parámetros
txtTexto As TextBox TextBox a seleccionar

Variables globales
<ninguna>

Código de Regreso
<ninguno>

```
Public Sub Selecciona(txtTexto As TextBox)
  txtTexto.SelStart = NULL_INTEGER
  txtTexto.SelLength = Len(txtTexto.Text)
End Sub
```

E) Verifica que el código pasado por parámetro sea de un carácter alfanumérico válido

Declaración
Public Sub ValidaAlfa(iKey As Integer)

Parámetros
iKey As Integer Código a validar

Variables globales
<ninguna>

Código de Regreso
<ninguno>

```
Public Sub ValidaAlfa(iKey As Integer)
  iKey = Asc(UCase(Chr$(iKey)))
End Sub
```

F) Verifica que el código pasado por parámetro sea de un carácter numérico válido

Declaración
Public Sub ValidaNum(iKey As Integer)

Parámetros
iKey As Integer --> Código a validar

Variables globales
<ninguna>

Código de Regreso
<ninguno>

```
Public Sub ValidaNum(iKey As Integer)
  If (iKey < KEY_0 Or iKey > KEY_9) And iKey <> KEY_PERIOD And iKey <> KEY_BACK Then
    iKey = KEY_NULL
  End If
End Sub
```

G) Calcula una edad, en años, a partir de la fecha de nacimiento

Declaración
Public Function Edad(dFecNac As Date) As Integer

Parámetros
dFecNac As Date --> Fecha de nacimiento

Variables globales
<ninguna>

Código de Regreso
Edad en años

```
Public Function Edad(dFecNac As Date) As Integer
  If CDate(String(DAY_MONTH_SIZE -
Len(Day(dFecNac)), CStr(NULL_INTEGER)) &
CStr(Day(dFecNac)) & Chr$(KEY_SLASH) &
String(DAY_MONTH_SIZE - Len(Month(dFecNac)),
CStr(NULL_INTEGER)) & CStr(Month(dFecNac)) &
Chr$(KEY_SLASH) & CStr(Year(Date))) < Date Then
    Edad = DateDiff(INTERVAL_YEAR, dFecNac, Date)
  Else
    Edad = DateDiff(INTERVAL_YEAR, dFecNac, Date) -
1
  End If
End Function
```

H) Verifica si el primer parámetro es nulo y, en ese caso, regresa el valor alternativo

Declaración
Public Function nvl(vValue As Variant, vIsNullValue As Variant) As Variant

Parámetros
vValue As Variant --> Valor a verificar
vIsNullValue As Variant --> Valor alternativo

Variables globales
<ninguna>

Código de Regreso
Valor alternativo en caso de que el valor original sea nulo. En caso contrario regresa el valor original

```
Public Function nvl(vValue As Variant, vIsNullValue As Variant) As Variant
  If IsNull(vValue) Then
    nvl = vIsNullValue
  Else
    nvl = vValue
  End If
End Function
```

I) La siguiente función formatea la fecha dependiendo de la Base de Datos que se esté utilizando.

Declaración
Function PonFecha(Fecha As Long) As String.
Parámetros
Fecha → variable que contiene la fecha que se va a convertir
Variables globales
<ninguna>
Código de Regreso
Fecha en formato adecuado

```
Function PonFecha(Fecha As Long) As String
Dim aa As Integer, mm As Integer, dd As Integer
Dim Temp$, v$

  If Fecha = 0 Then
    PonFecha = "null"
    Exit Function
  End If

  Temp$ = Trim$(Format$(Fecha))
  aa = Fecha / 10000
  mm = (Fecha Mod 10000) / 100
  dd = Fecha Mod 100

Select Case BD_TIPO

  ' Oracle
  Case "ORACLE"
    v$ = "TO_DATE('" & Temp$ & "','YYYYMMDD')"

  ' Ingress
  Case "INGRES"

  ' Informix
  Case "INFORMIX"
```

```
      v$ = """ & Format$(aa, "0000") & "/" & Format$(mm, "00") & "/" & Format$(dd, "00") & """

    '  Sybase
    Case "SQLServer"
      v$ = """ & Format$(mm, "00") & "/" & Format$(dd, "00") & "/" & Format$(aa, "0000") & """ End Select

  PonFecha = v$ End Function
```

4.2 Conexión a la Base de Datos

Para realizar la conexión a la base de datos, se utiliza un ODBC y se realizan los pasos que se enlistan a continuación:

1. Se declara e instancia un objeto de conexión

```
Dim bd As ADODB.Connection
Set bd = New Connection
```

2. Se declara e instancia un objeto recordset

```
Dim RsVerifica As ADODB.Recordset
Set RsVerifica = New ADODB.Recordset
```

3. Se pasa la cadena de conexión al objeto conexión con la información del ODBC DNS, nombre de la base de datos, así como su usuario y pasword

```
bd.ConnectionString = "ODBC;DATABASE=" & DBName$ & ";DSN=" & DriverName$ & ";UID="
& USER & ";PWD=" & txtPassword.Text
```

4. Se abre la conexión

```
bd.Open
```

5. Se realiza la consulta a la base de datos y se guarda en una variable

```
SQL_Query = "SELECT count(*) FROM usuarios_programas WHERE usuario='" & USER & "'
AND programa='" & Mid$(Help_File_Name, 1, n% - 1) & "'"
```

6. Se abre el recordset y se le pasa la variable que contiene la consulta

```
RsVerifica.Open SQL_Query
```

7. Se cierra el record set

```
RsVerifica.Close
```

8. se cierra la conexión

```
bd.Close
```

4.3 Procedimientos almacenados

Las consultas a la base de datos, se hacen por medio de procedimientos almacenados.

Se manejaron tres diferentes tipos de procedimientos para el desarrollo del sistema, a continuación se ejemplifican cada uno de ellos:

A) Procedimiento de consulta de datos

```
If exists (select * from sysobjects where id = object_id('dbo.sps_RSC_Candidato') )
        drop procedure dbo.sps_RSC_Candidato
go

CREATE PROCEDURE dbo.sps_RSC_Candidato
/*********************************************************************
Nombre: dbo.sps_RSC_Candidato
 Descripcion: Obtiene los beneficiarios de un candidato
Parámetros:
        @Candidato_Id --> Clave del candidato

 Formas de donde se llama: frmCandidatoAlta

 Autor: Gabriela Barajas Ruiz

**********************************************************************
 Modificación:   <correcciones realizadas>
                 <nombre>                        <fecha>
**********************************************************************/
As
Begin
        Select  C.Candidato_Id 'No.',
                C.Candidato_Nombre 'Nombre',
                C.Candidato_Ap_Paterno 'Apellido Paterno',
                C.Candidato_Ap_Materno 'Apellido Materno',
                C.Candidato_Calle + space(1) + C.Candidato_Colonia + space(1) +
C.Candidato_Delegacion + space(1) + E.nombre_estado + space(1) + C.Candidato_CP
'Domicilio',
                C.Candidato_IMSS 'No. de IMSS',
                C.Candidato_Telefono 'Teléfono',
                C.Candidato_Licencia_Tipo + ' - ' + C.Candidato_Licencia_No 'Licencia',
                C.Candidato_Sexo 'Sexo',
                N.descripcion 'Nacionalidad',
                C.Candidato_Cartilla 'Cartilla Militar'
        From    Candidato C, nacionalidades N, estados E
        Where   C.nacionalidad = N.nacionalidad
        And     C.estado = E.estado
End
go

Grant Execute On dbo.sps_RSC_Candidato To [public]
go
```

B) Procedimiento de Inserción de datos

```
If exists (select * from sysobjects where id = object_id('dbo.spi_RSC_Can_Escolaridad') )
        drop procedure dbo.spi_RSC_Can_Escolaridad
go
CREATE PROCEDURE dbo.spi_RSC_Can_Escolaridad (   @iCandidato_Id int,
                                                 @sNivel_estudios smallint,
                                                 @vCan_Escolaridad_Institucion varchar(100),
                                                 @vCan_Escolaridad_Fec_Inicio varchar(10),
                                                 @vCan_Escolaridad_Fec_Fin varchar(10),
                                                 @bCan_Escolaridad_Certificado bit,
                                                 @vCan_Escolaridad_Fec_Titulacion  varchar(10),
                                                 @vCan_Escolaridad_Cedula varchar(20),
                                                 @bCan_Escolaridad_Beca bit )
Nombre: dbo.spi_RSC_Can_Escolaridad

Descripcion: Inserta los datos de un nivel escolar del candidato pasado por parámetro

 Parámetros:
        @iCandidato_Id                  --> Identificador del candidato
        @sNivel_estudios                --> Clave del nivel de estudios
        @vCan_Escolaridad_Institucion   --> Nombre de la institución
        @vCan_Escolaridad_Fec_Inicio    --> Fecha de inicio
        @vCan_Escolaridad_Fec_Fin               --> Fecha de término
        @bCan_Escolaridad_Certificado   --> Indica si se tiene certificado de estudio
        @vCan_Escolaridad_Fec_Titulacion --> Fecha de titulación
        @vCan_Escolaridad_Cedula        --> Número de cédula
        @bCan_Escolaridad_Beca          --> Indica si el estudio se realizó con beca

 Formas de donde se llama: frmCanEscolaridad

 Autor: Gabriela Barajas Ruiz
/*********************************************************************
 Modificación:      <correcciones realizadas>
                    <nombre>                                <fecha>
*********************************************************************/
As
Begin
        Insert Into Can_Escolaridad ( Can_Escolaridad_Id,
                                      Candidato_Id,
                                      nivel_estudios,
                                      Can_Escolaridad_Institucion,
                                      Can_Escolaridad_Fec_Inicio,
                                      Can_Escolaridad_Fec_Fin,
                                      Can_Escolaridad_Certificado,
                                      Can_Escolaridad_Fec_Titulacion,
                                      Can_Escolaridad_Cedula,
                                      Can_Escolaridad_Beca )
        Select  isnull(max(Can_Escolaridad_Id), 0) + 1,
                @iCandidato_Id,
                @sNivel_estudios,
                @vCan_Escolaridad_Institucion,
                convert(datetime, @vCan_Escolaridad_Fec_Inicio, 103),
                convert(datetime, @vCan_Escolaridad_Fec_Fin, 103),
                @bCan_Escolaridad_Certificado,
                convert(datetime, @vCan_Escolaridad_Fec_Titulacion, 103),
                @vCan_Escolaridad_Cedula,
                @bCan_Escolaridad_Beca
        From    Can_Escolaridad
        Where   Candidato_Id = @iCandidato_Id

End
go
Grant Execute On dbo.spi_RSC_Can_Escolaridad To [public]
go
```

C) Procedimiento de actualización de datos

```
If exists (select * from sysobjects where id = object_id('dbo.spu_RSC_Can_Datos_Familiares') )

        drop procedure dbo.spu_RSC_Can_Datos_Familiares
go

CREATE PROCEDURE dbo.spu_RSC_Can_Datos_Familiares (@iCandidato_Id int,
                                        @bCandidato_Dep_Eco_Padres bit,
                                        @bCandidato_Dep_Eco_Hijos bit,
                                        @bCandidato_Dep_Eco_Hermanos bit,
                                        @vCandidato_Dep_Eco_Otros varchar(30),
                                        @bCandidato_Fam_UIA bit,
                                        @vCandidato_Fam_UIA_Dependencia varchar(50),
                                        @vCandidato_Fam_UIA_Nombre varchar(30),
                                        @vCandidato_Fam_UIA_Ap_Paterno varchar(30),
                                        @vCandidato_Fam_UIA_Ap_Materno varchar(30),
                                        @vCandidato_Fam_UIA_Puesto varchar(90),
                                        @vCandidato_Medio varchar(50),
                                        @vCandidato_Area_Interes varchar(100))

 Nombre:  dbo.spu_RSC_Can_Datos_Familiares

 Descripcion: Actualiza los datos familiares de un candidato.

Parámetros:
@iCandidato_Id                  --> Identificador del candidato
@bCandidato_Dep_Eco_Padres --> Indica si los padres dependen económicamente del candidato
@bCandidato_Dep_Eco_Hijos       --> Indica si los hijos dependen económicamente del candidato
@bCandidato_Dep_Eco_Hermanos --> Indica si los hermanos dependen económicamente del candidato
@vCandidato_Dep_Eco_Otros       --> Indica que otras personas dependen económicamente del candidato
@bCandidato_Fam_UV      --> Indica si el candidato tiene algún pariente laborando en UV
@vCandidato_Fam_UV_Dependencia  --> Dependencia en que labora el familiar
@vCandidato_Fam_UV_Nombre       --> Nombre del familiar que labora en UIA
@vCandidato_Fam_UV_Ap_Paterno   --> Apellido paterno del familiar que labora en UIA
@vCandidato_Fam_UV_Ap_Materno   --> Apellido materno del familiar que labora en UIA
@vCandidato_Fam_UV_Puesto       --> Puesto del familiar que labora en UIA
@vCandidato_Medio --> Medio informativo a través del cual el candidato se enteró del empleo
@vCandidato_Area_Interes --> Área de interés del candidato

 Formas de donde se llama:
        frmCandidatoAlta

 Autor:
        Gabriela Barajas Ruiz
*********************************************************************
 Modificación:      <correcciones realizadas>
                    <nombre>                                    <fecha>
*********************************************************************/
As
Begin
        Update  Candidato
        Set     Candidato_Dep_Eco_Padres = @bCandidato_Dep_Eco_Padres,
                Candidato_Dep_Eco_Hijos = @bCandidato_Dep_Eco_Hijos,
                Candidato_Dep_Eco_Hermanos = @bCandidato_Dep_Eco_Hermanos,
                Candidato_Dep_Eco_Otros = @vCandidato_Dep_Eco_Otros,
                Candidato_Fam_UIA = @bCandidato_Fam_UIA,
                Candidato_Fam_UIA_Dependencia = @vCandidato_Fam_UIA_Dependencia,
                Candidato_Fam_UIA_Nombre = @vCandidato_Fam_UIA_Nombre,
                Candidato_Fam_UIA_Ap_Paterno = @vCandidato_Fam_UIA_Ap_Paterno,
                Candidato_Fam_UIA_Ap_Materno = @vCandidato_Fam_UIA_Ap_Materno,
                Candidato_Fam_UIA_Puesto = @vCandidato_Fam_UIA_Puesto,
                Candidato_Medio = @vCandidato_Medio,
                Candidato_Area_Interes = @vCandidato_Area_Interes
        Where   Candidato_Id = @iCandidato_Id
End
Go
```

4.4 Pantallas del sistema

A) Pantalla de autentificación del usuario: Aquí el usuario ingresará su usuario y su password.

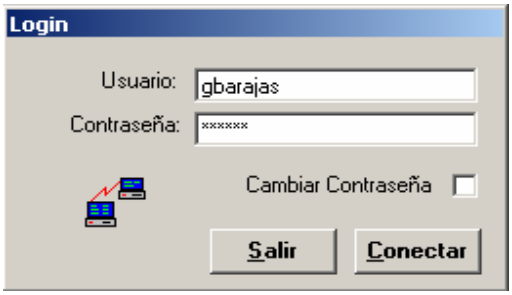


Figura 4.1 *Pantalla de Login*

A.1) Pantalla para que el usuario pueda cambiar su contraseña, en caso de que así lo requiera

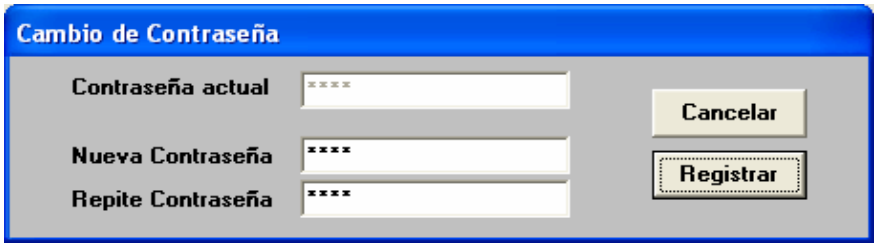


Figura 4.2 *Cambio de Contraseña*

B) Menú principal en donde el usuario seleccionará la opción que desea acceder

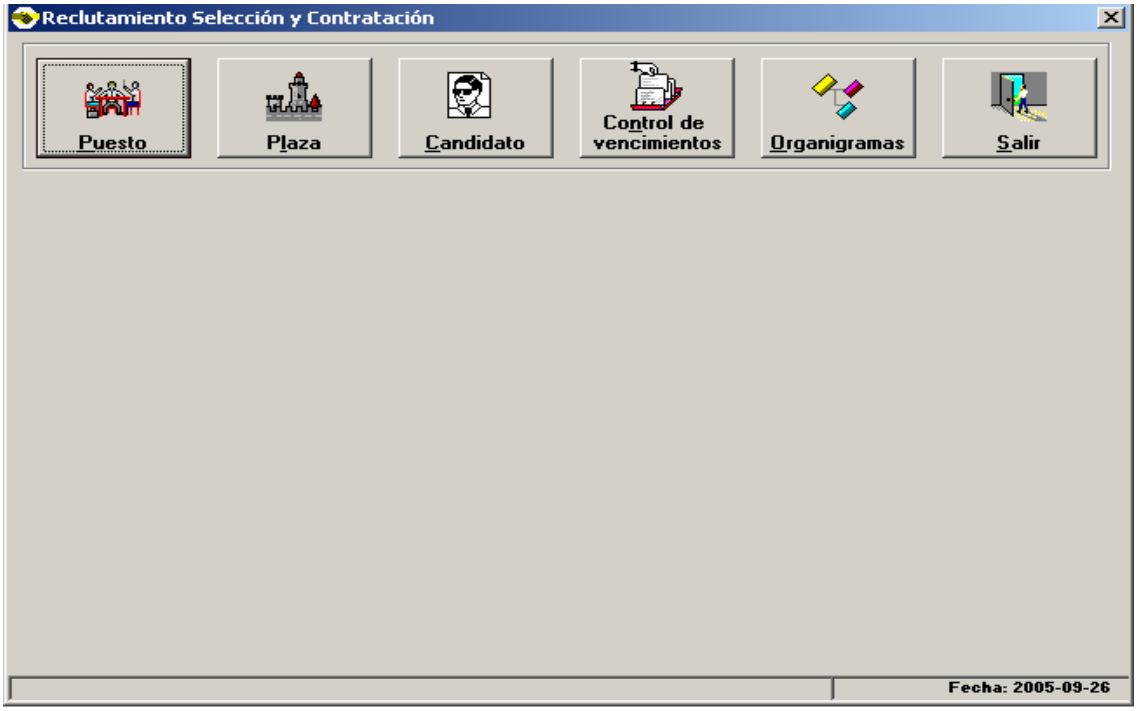


Figura 4.3 *Menú principal*

C) Listado de puestos de nueva creación. Se pueden seleccionar por estatus y por entidad

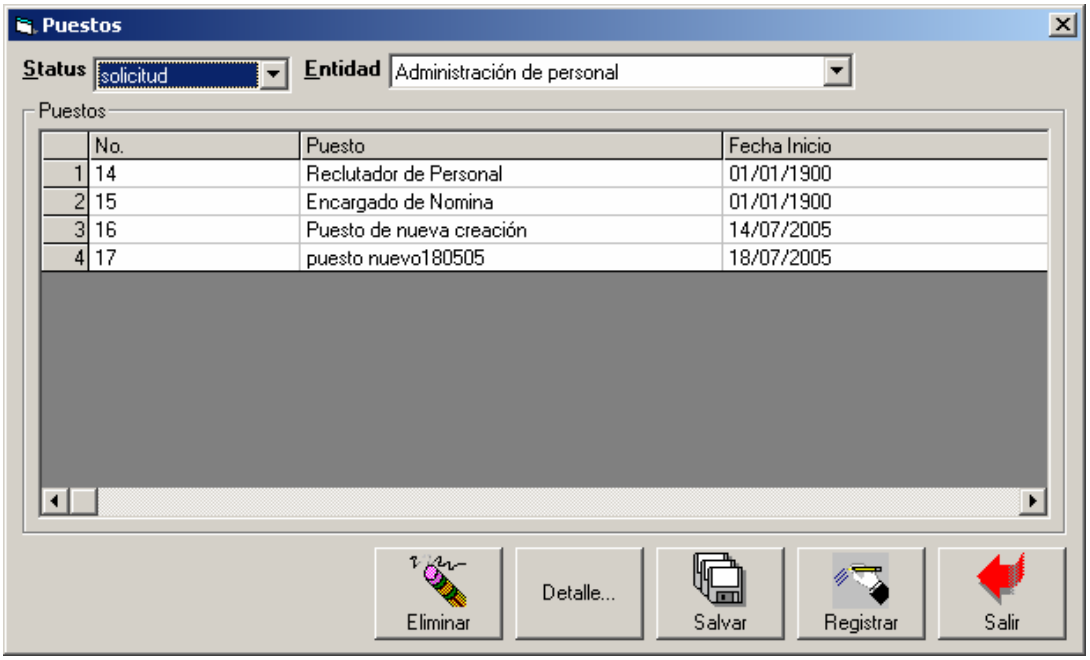


Figura 4.4 *Listado de puestos de nueva creación*

C.1) Detalle de puestos de nueva creación, a esta pantalla ingresará el usuario después de oprimir el botón Detalle y seleccionar el puesto deseado (Figura 4.4)

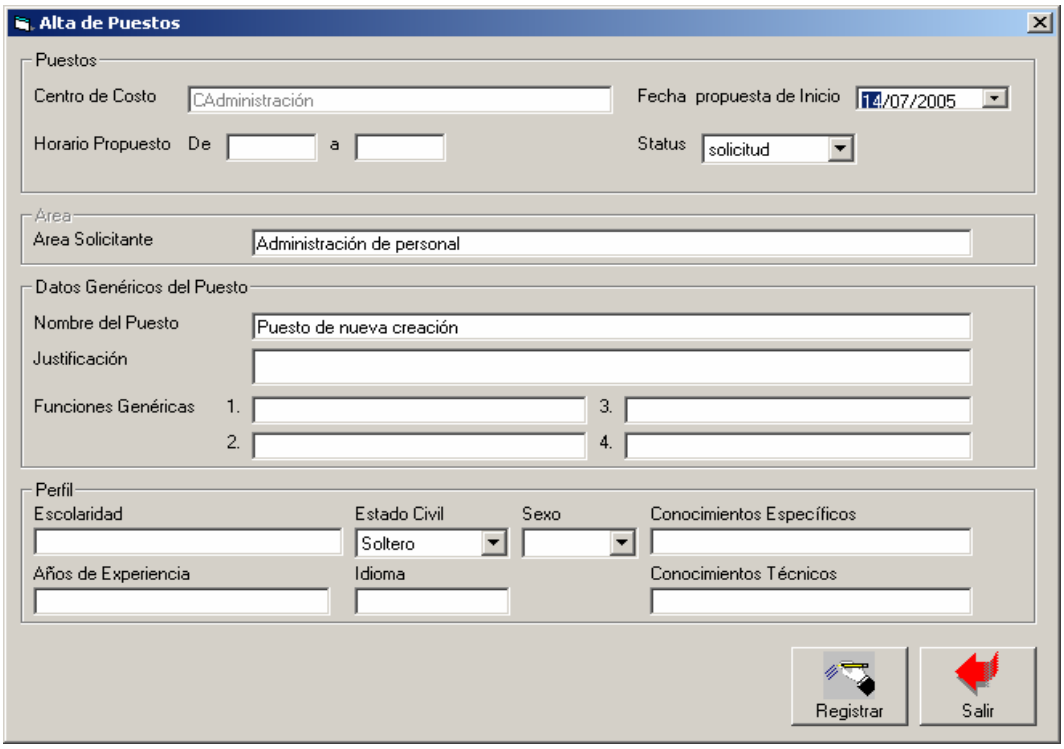


Figura 4.5 *Detalle de puestos de nueva creación*

D) Listado de plazas de nueva creación por Status y solicitud

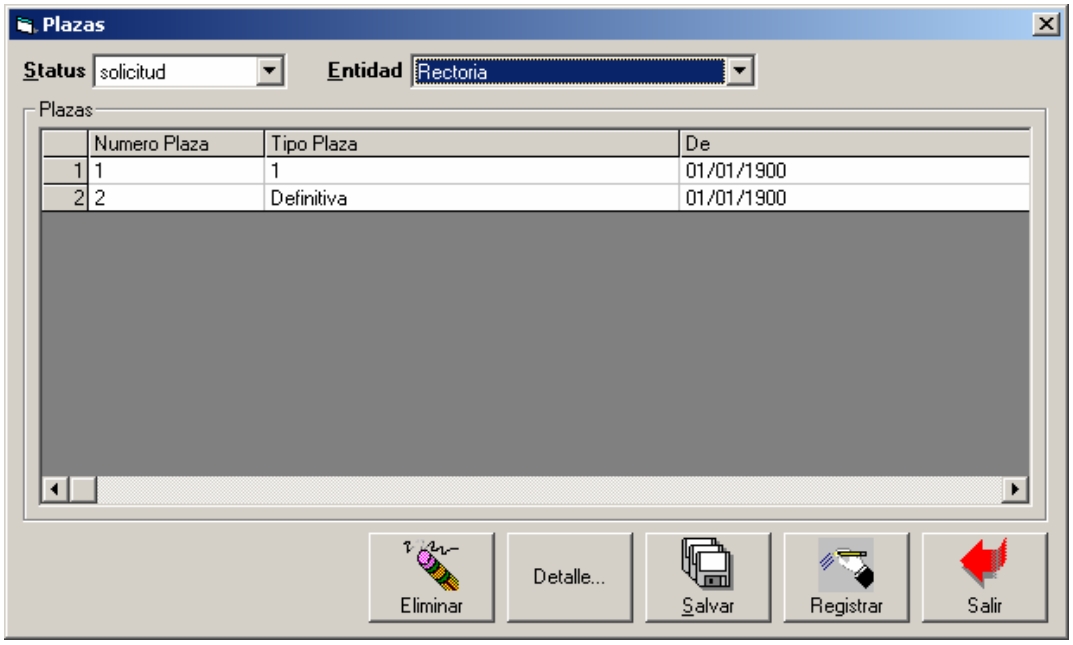


Figura 4.6 *Listado de plazas de nueva creación*

D.1) Detalle de plazas de nueva creación, a esta pantalla ingresará el usuario después de oprimir el botón Detalle y seleccionar la plaza deseada (Figura 4.6)

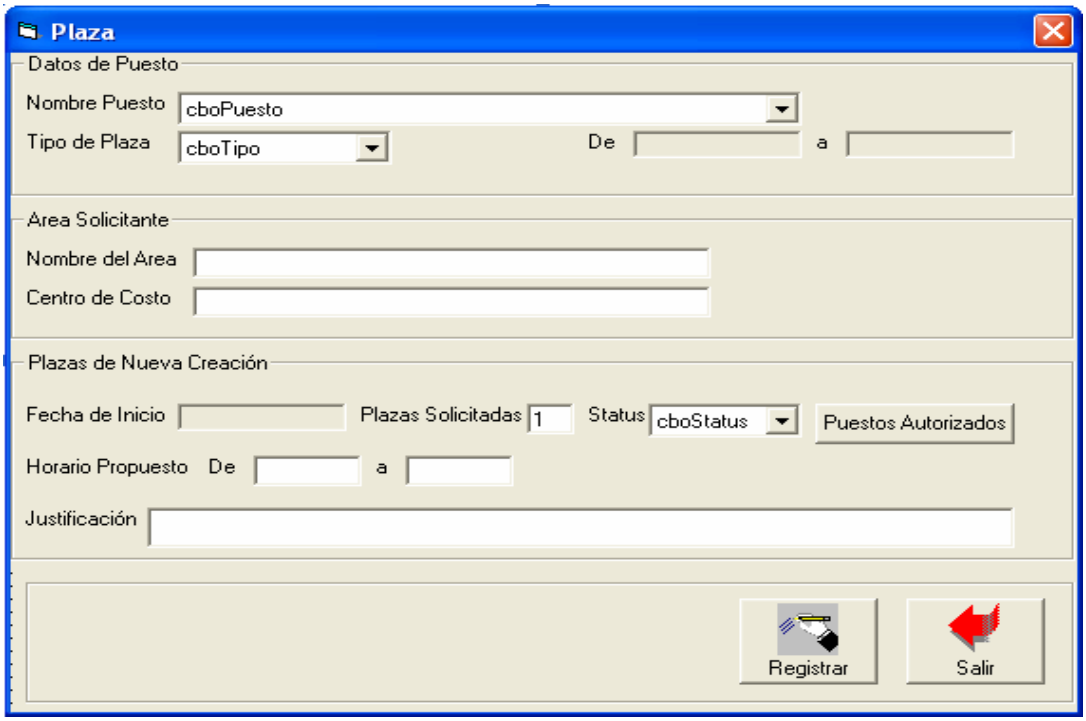


Figura 4.7 *Detalle de plazas de nueva creación*

E) Pantalla general de candidatos por Status

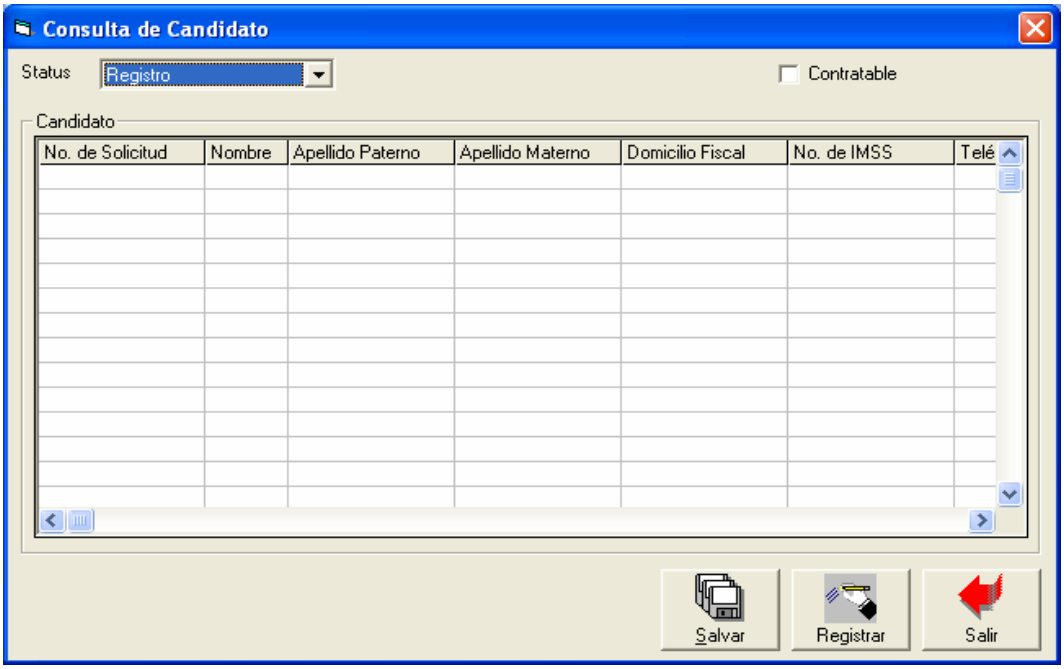


Figura 4.8 *Listado de Candidatos*

E.1) Registro de Candidatos -Datos generales

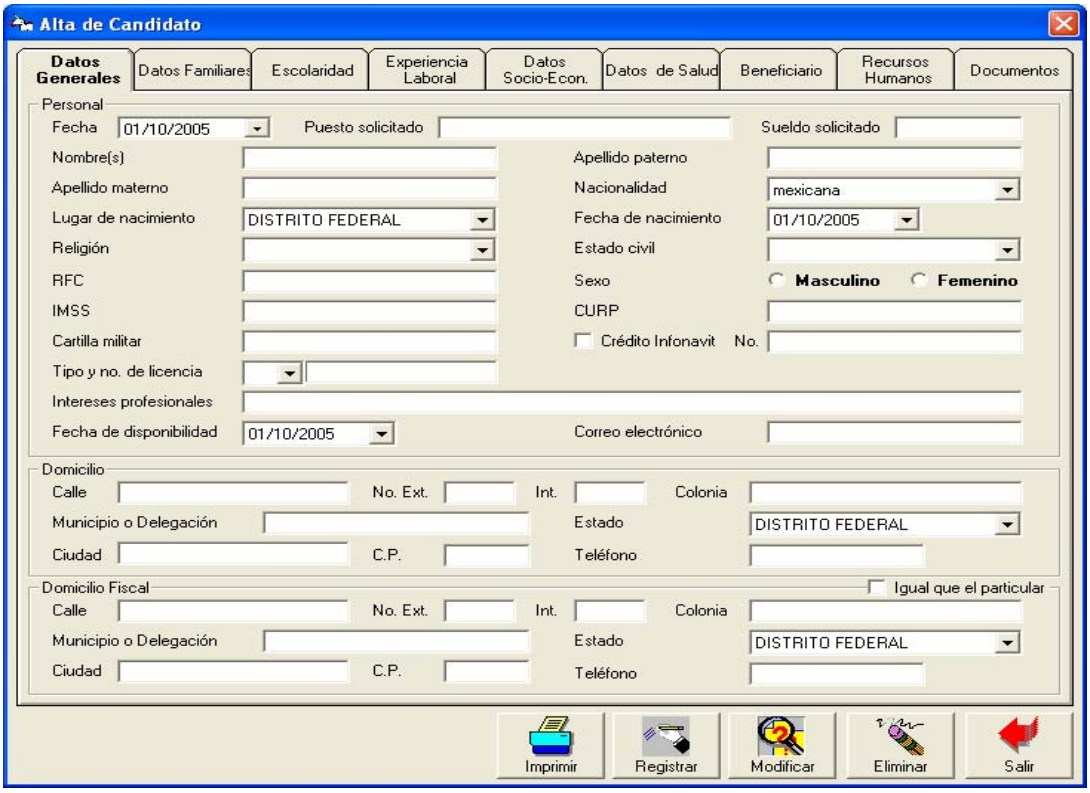


Figura 4.9 *Datos Generales*

E.2) Registro de Candidatos – Datos Familiares

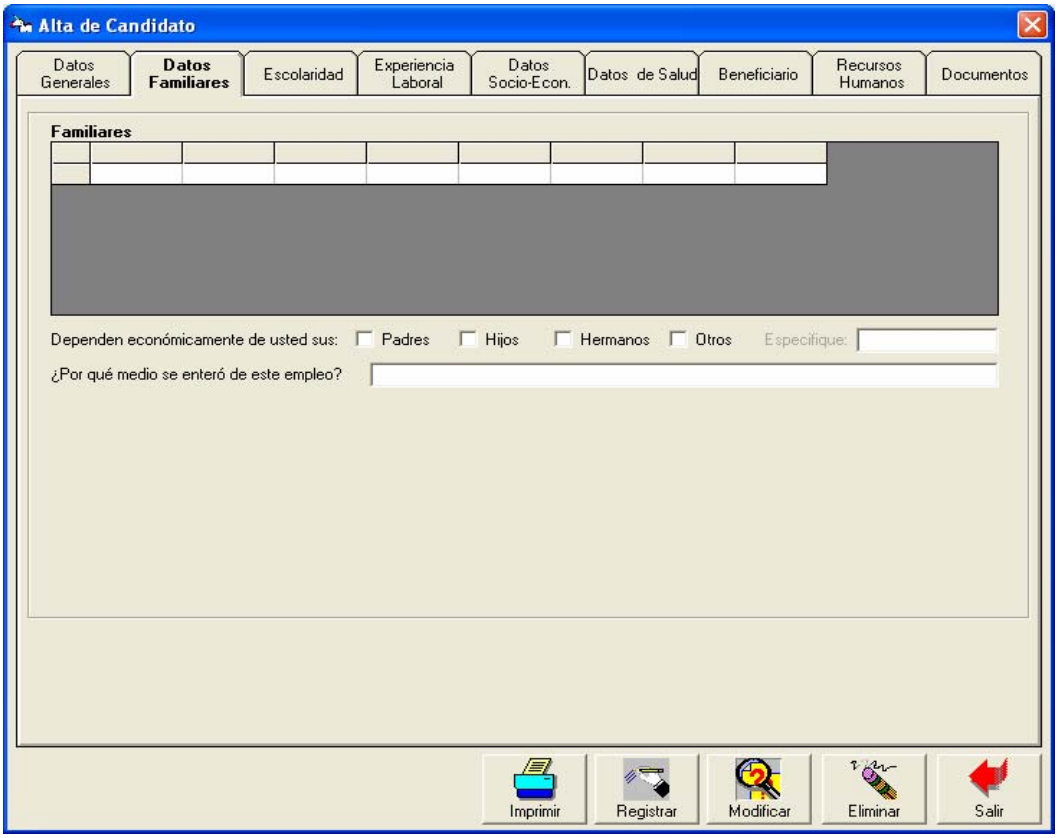


Figura 4.10 *Datos Familiares*

E.2.1) Registro de Familiares. A esta pantalla se ingresa dando un clic al listado de familiares (Figura 4.10)

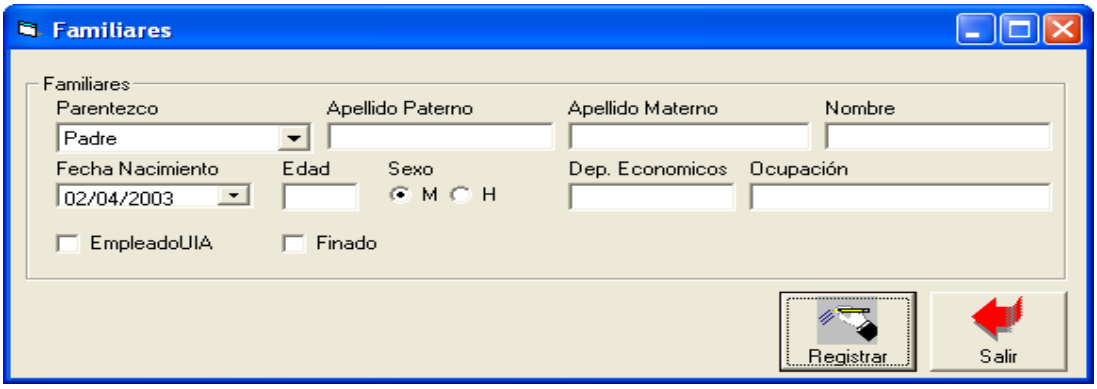


Figura 4.11 *Registro de Familiares*

E.3) Datos Generales-Escolaridad

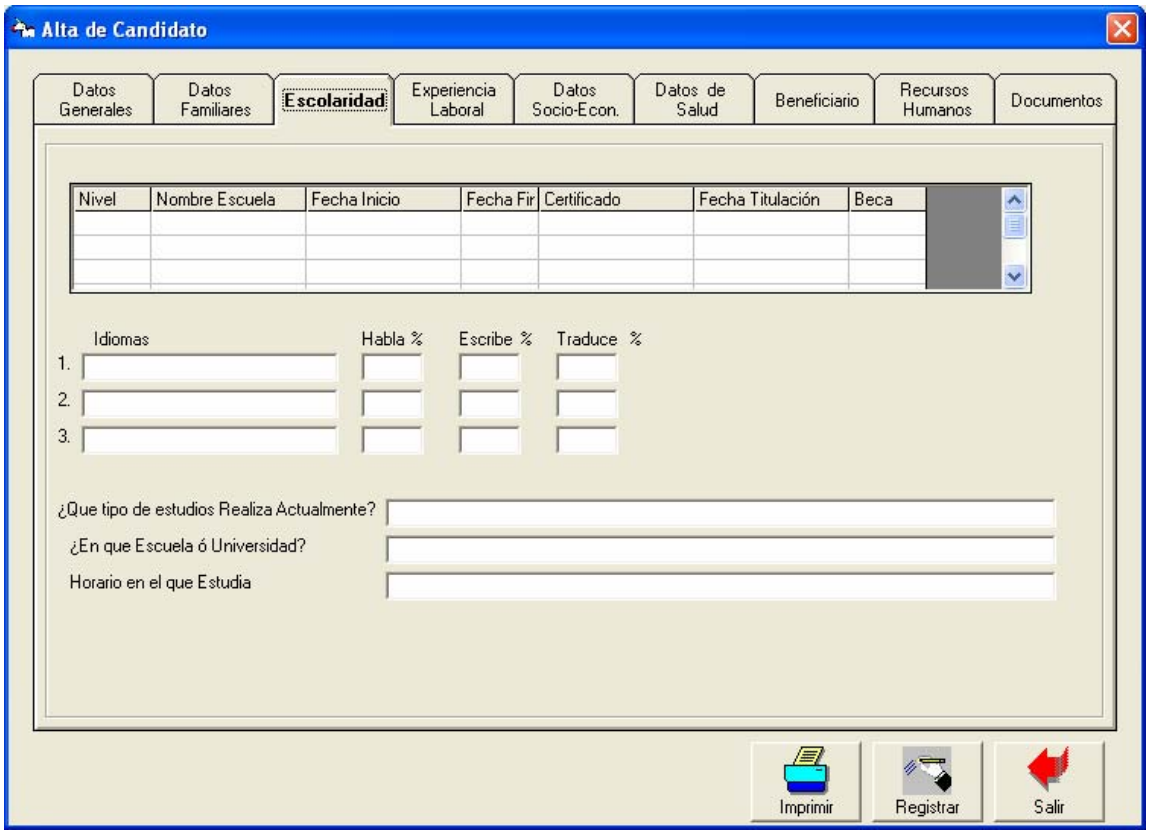


Figura 4.12 *Escolaridad*

E.3.1) Registro de la escolaridad del candidato. A esta pantalla se accede dando clic al listado de escolaridad (Figura 4.12)

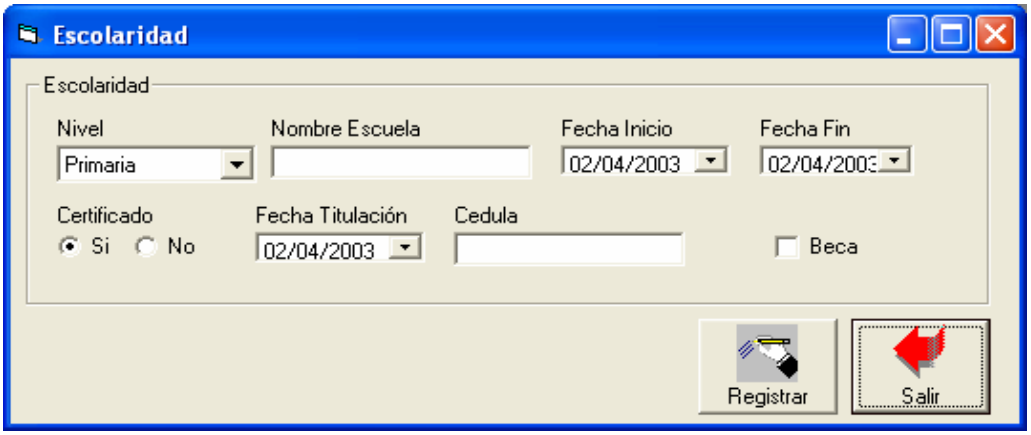


Figura 4.13 *Registro Escolaridad*

E.4) Datos Generales – Experiencia Laboral

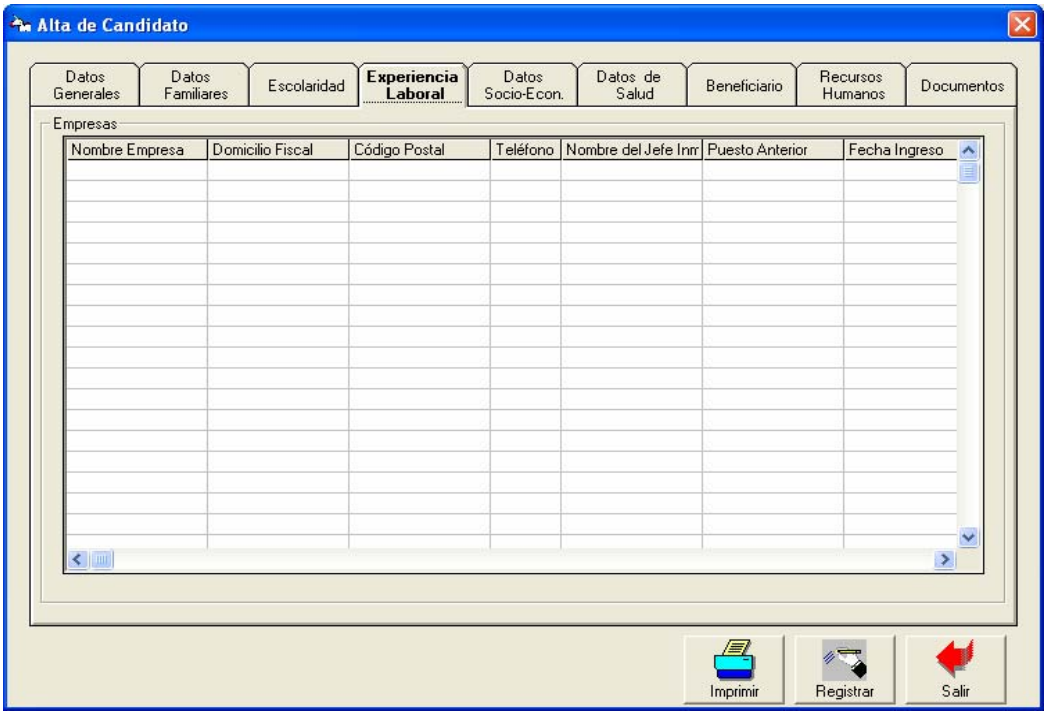


Figura 4.14 *Experiencia Laboral*

E.4.1) Registro de experiencia laboral. A esta pantalla se ingresa oprimiendo el botón registrar de la figura 4.14

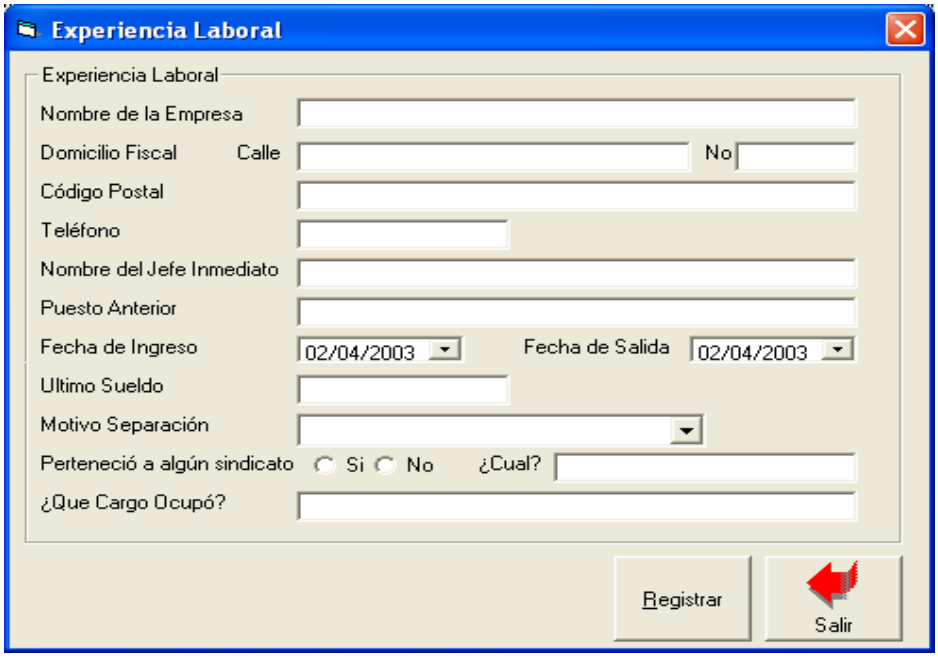


Figura 4.15 *Registro de experiencia Laboral*.

E.5) Registro de Candidatos - Datos Socio-Económicos

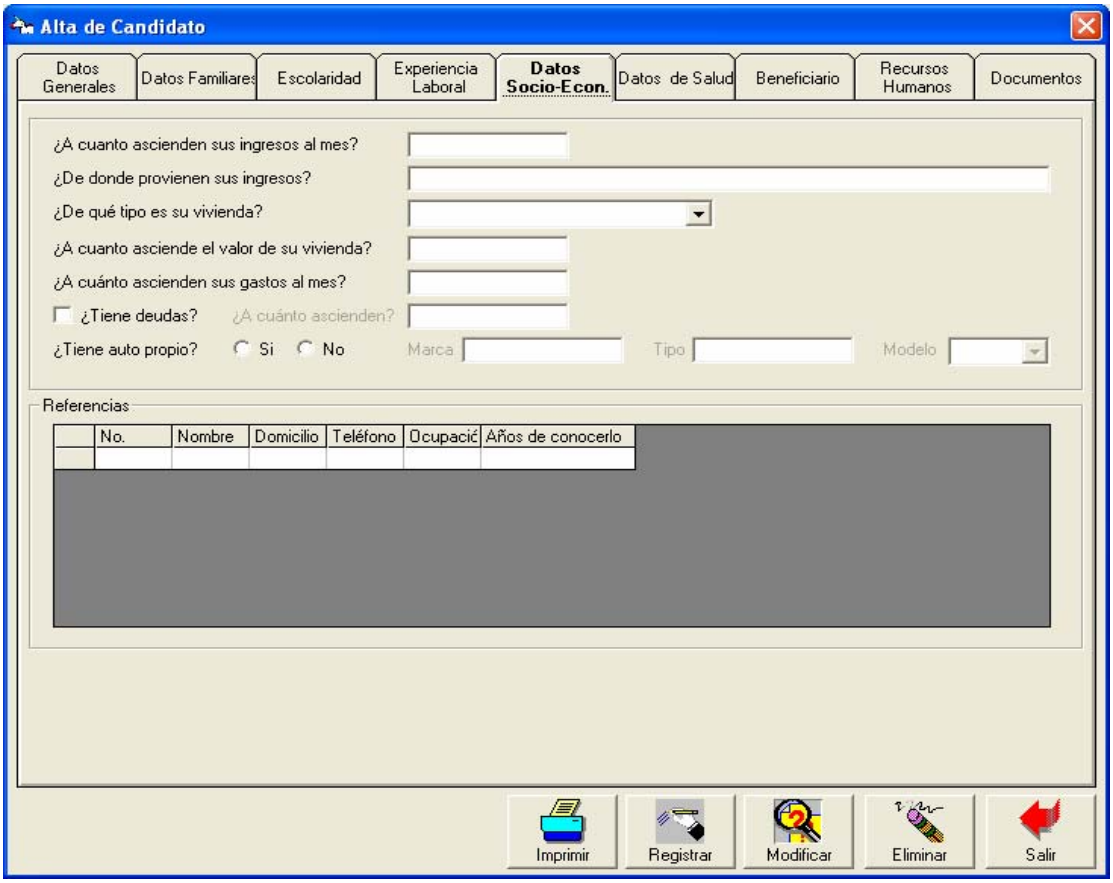


Figura 4.16 *Datos Socioeconómicos*

E.5.1) Referencias del candidato A esta pantalla se ingresa dando un clic en el listado de referencias (Figura 4.16)

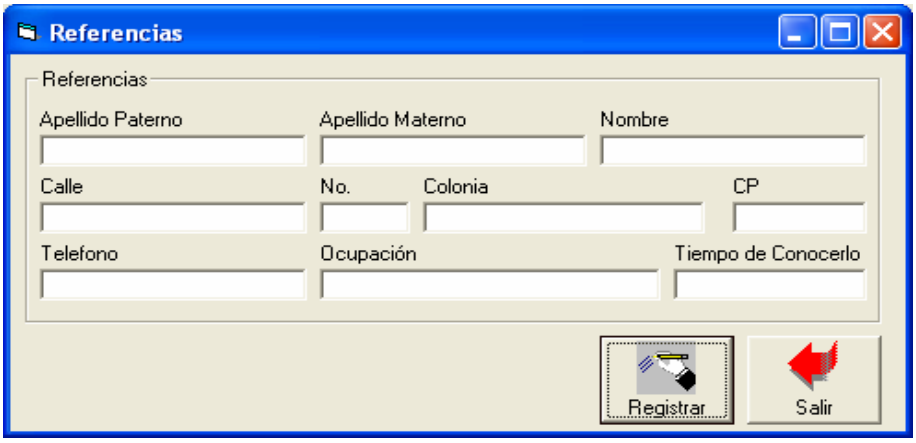


Figura 4.17 *Referencias*

E.6) Registro de candidatos Datos de Salud

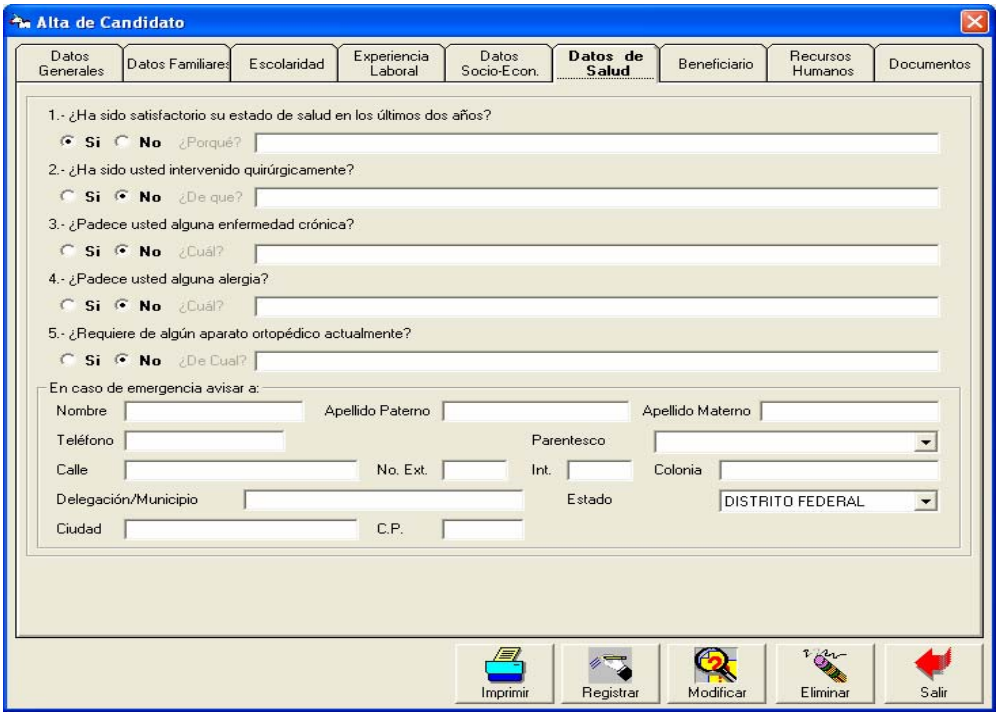


Figura 4.18 *Datos de Salud*

E.7) *Registro de Candidatos Beneficiarios*

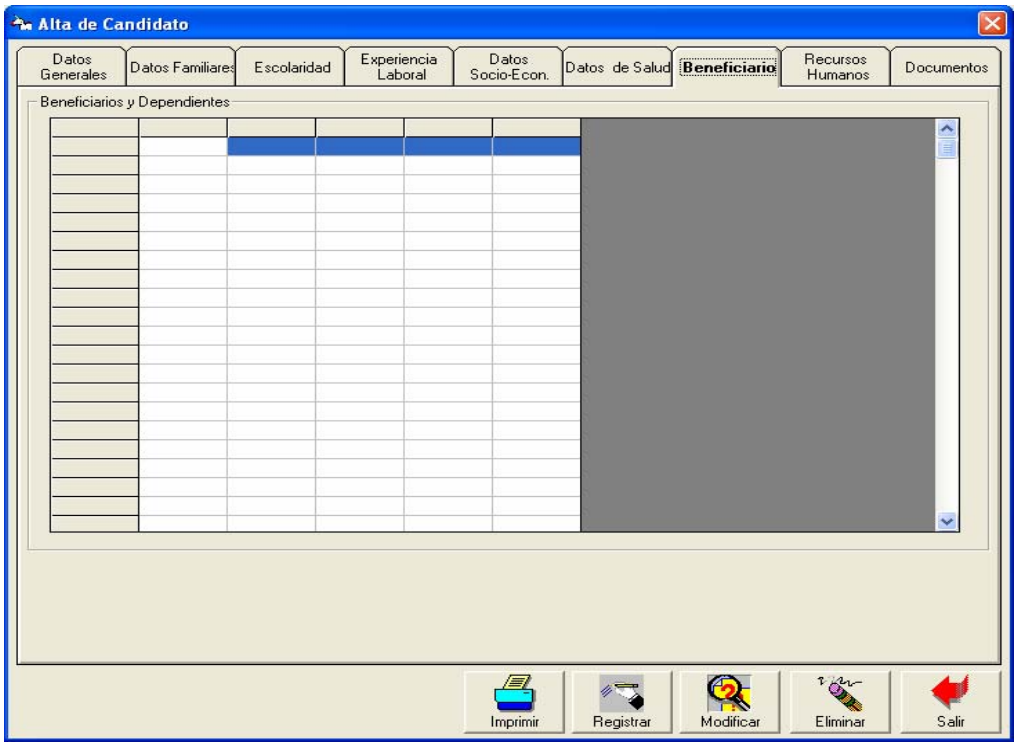


Figura 4.19 *Beneficiarios*

E.7.1) Datos Generales Alta de Beneficiarios

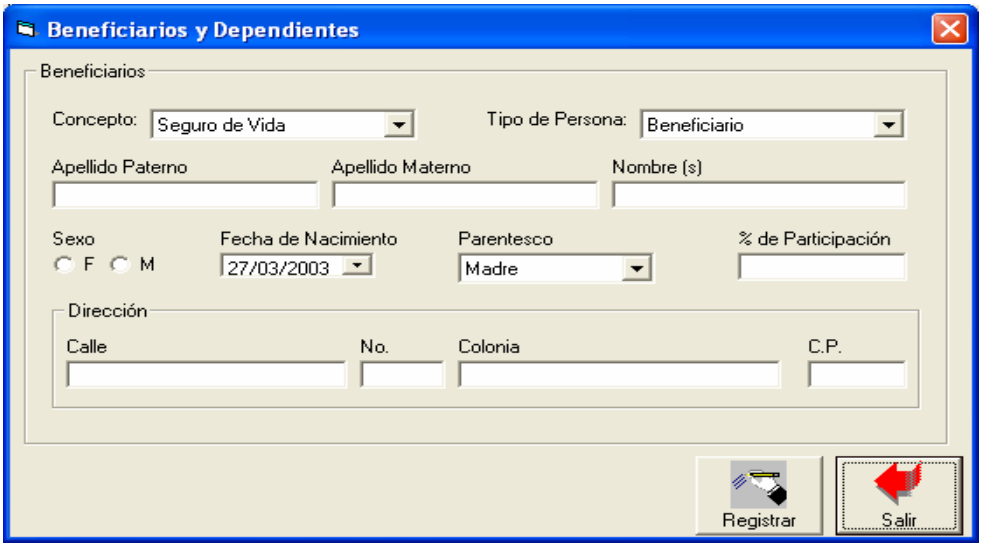


Figura 4.20 *Detalle de Beneficiarios*

E.8) Registro de Candidatos-Recursos Humanos

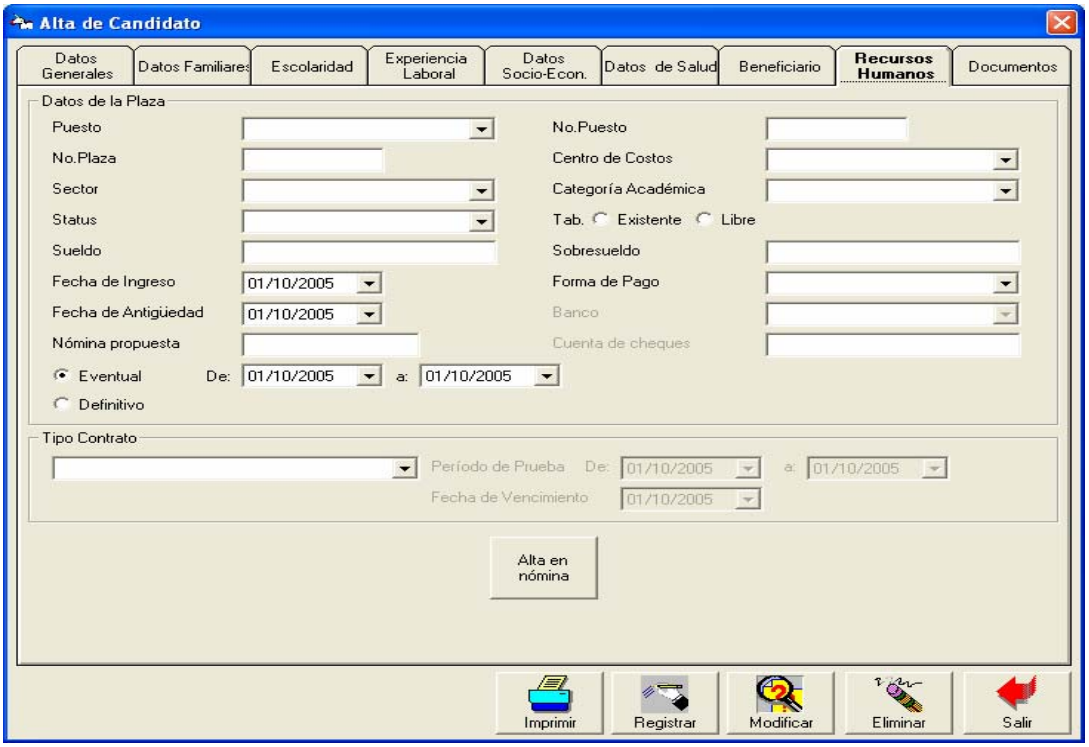


Figura 4.21 *Recursos Humanos*

E.9) Registro de Candidatos- Documentos

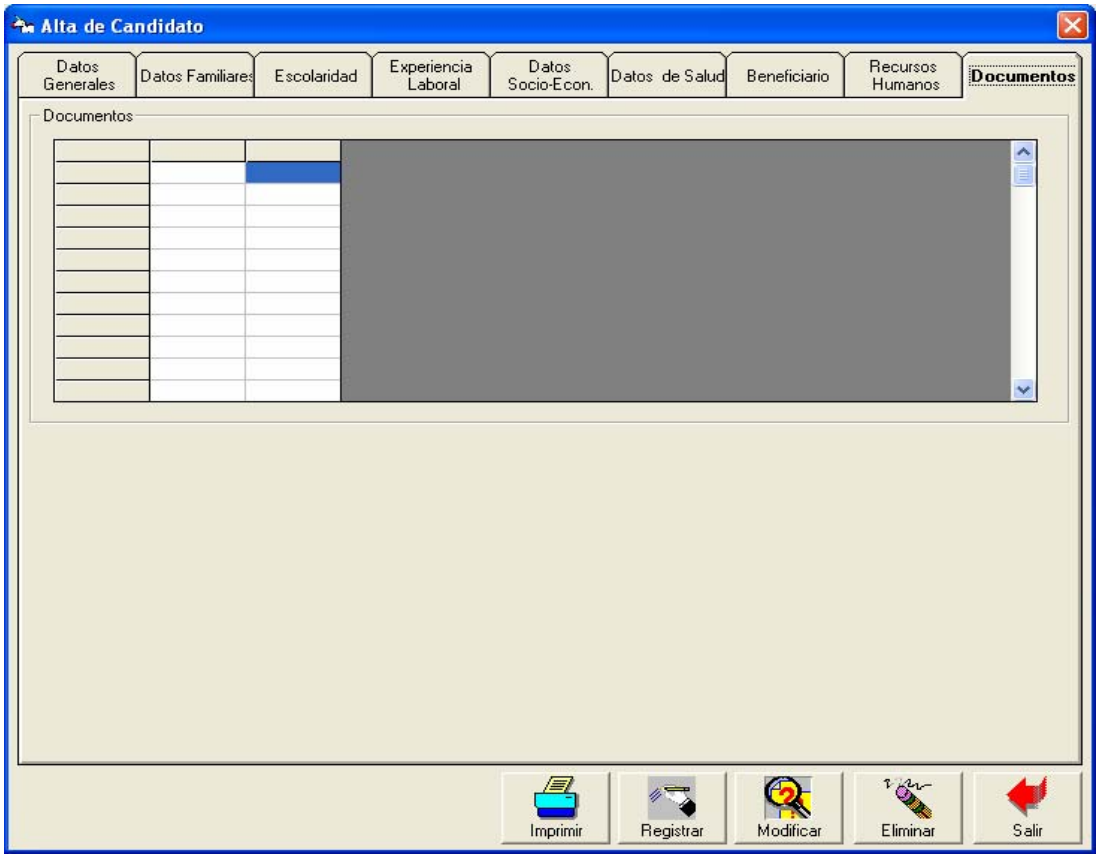


Figura 4.22 *Documentos*

E.9.1) Alta de Documentos



Figura 4.23 *Alta de Documentos*

F) Control de Vencimientos

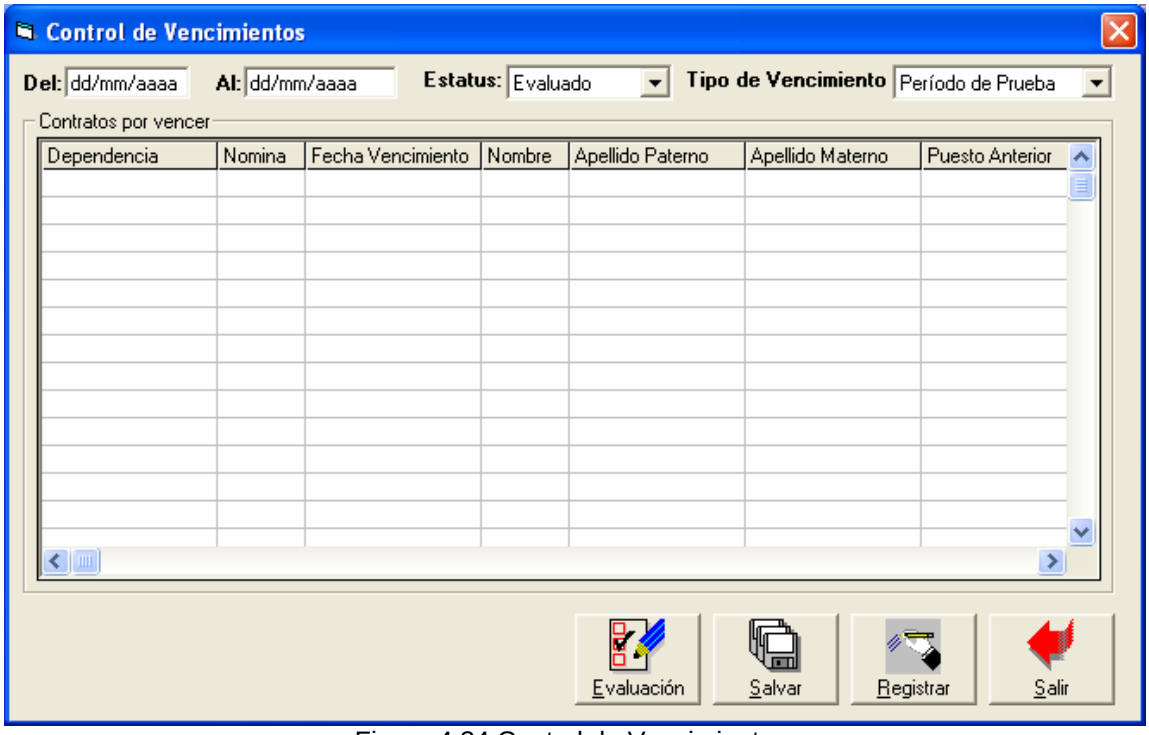


Figura 4.24 Control de Vencimientos

G) *Estructura Organizacional* Creación de un nuevo organigrama. En esta sección, se podrán generar organigramas por estructura y por empleado.

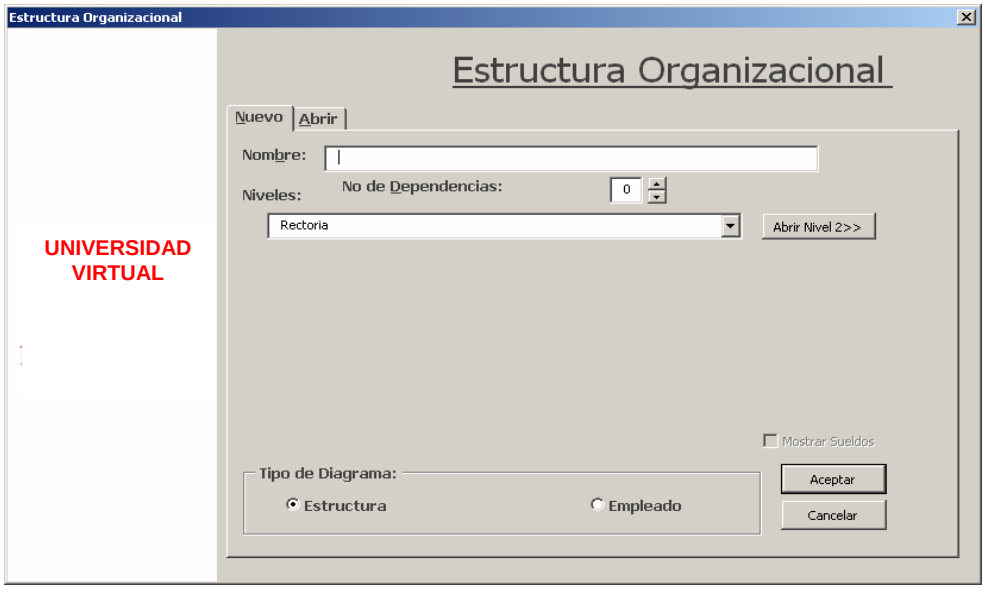


Figura 4.25 *Creación de un organigrama*

H) *Estructura Organizacional* Abrir un organigrama existente. En esta sección, se podrán abrir organigramas existentes por estructura y por empleado.

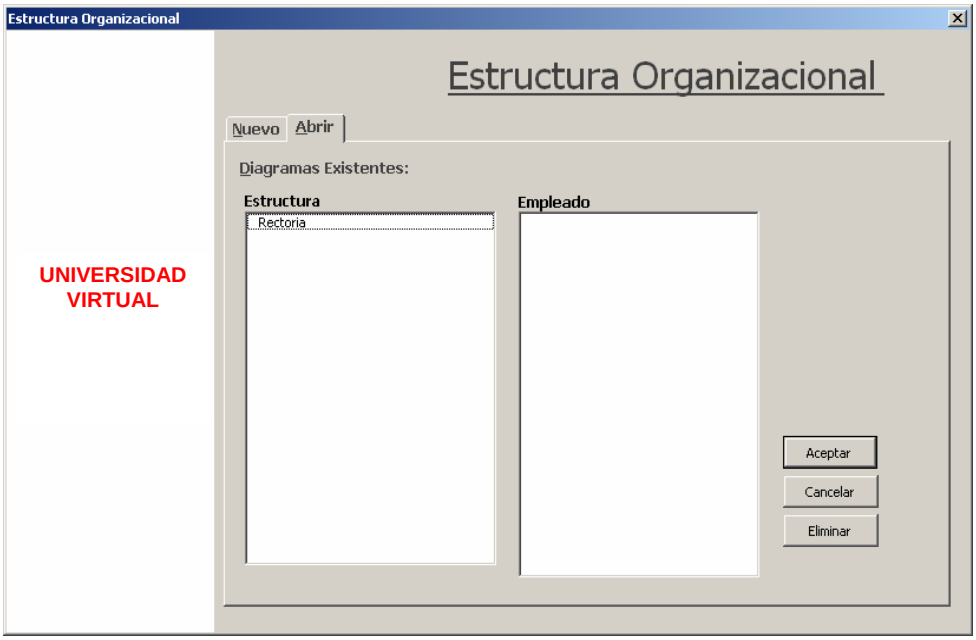


Figura 4.26 *Apertura de un Organigrama Existente*

I) *Estructura Organizacional* Generador de reportes. Desde esta ventana se podrán generar reportes de empleados por plaza o bien, total de empleados por plaza.

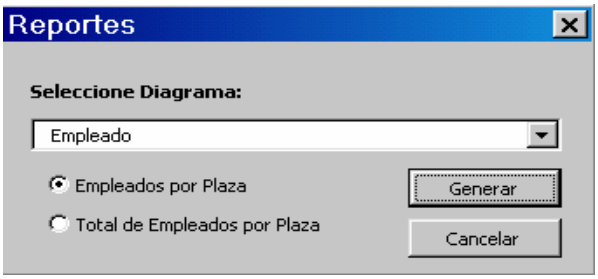


Figura 4.27 *Generación de reportes*

Conclusión

En este capítulo, se presentaron las pantallas finales que conforman el sistema que se trata en la presente tesis. También se presenta el código principal con el cual fueron desarrolladas dichas pantallas. Una vez desarrollado el sistema, se debe de seguir una estrategia de pruebas para garantizar el buen funcionamiento del mismo, de esto se hablará en el capítulo 5 (Pruebas y conclusiones).

CAPITULO 5

PRUEBAS Y CONCLUSIONES

Objetivos:

- Mostrar un panorama general de las técnicas de prueba que se utilizan para descubrir fallas en los programas
- Describir la técnica y el programa utilizados para realizar las pruebas del sistema para la UV

En el capítulo 4, se presentaron las pantallas de los módulos que conforman el sistema realizado para la UV, así como las principales funciones utilizadas para el desarrollo del mismo. En este capítulo, se mostrarán las principales técnicas de prueba y la técnica seleccionada para realizar las pruebas del sistema que aquí se discute.

La etapa de pruebas de componentes comprende las pruebas de funcionamiento de los módulos o en objetos. Durante las pruebas de integración, estos componentes se integran para formar subsistemas o el sistema completo. En esta etapa, las pruebas se enfocan en las interacciones entre los componentes y la funcionalidad y el desempeño del sistema como un todo.

Las pruebas de integración, generalmente se basan en una especificación escrita del sistema. Esta puede ser una especificación detallada de los requerimientos del sistema, o puede ser una especificación orientada al usuario de las características a implementar en el sistema

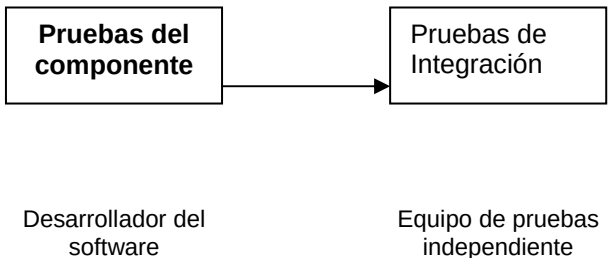


Figura 5.1 *Fase de las pruebas*

5.1 PRUEBAS DE DEFECTOS

La meta de las pruebas de defectos es exponer los defectos latentes en un sistema de software antes de entregar el sistema. Una prueba de defectos exitosa es una prueba que provoca que el sistema se desempeñe incorrectamente y así exponer un defecto.

Las pruebas exhaustivas, donde se prueba cada posible secuencia de ejecuciones del programa, no son prácticas. Por lo tanto, llevar a cabo las pruebas se basa en un subconjunto de posibles casos de prueba. Las organizaciones deberían desarrollar políticas para elegir este subconjunto en lugar de dejar esto al equipo de desarrollo.

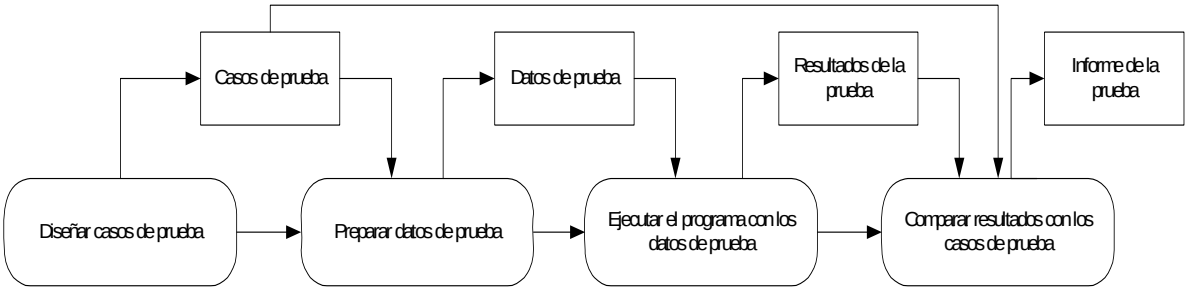


Figura 5.2 *Proceso de pruebas de defectos*

5.1.1 PRUEBAS DE CAJA NEGRA

Las pruebas funcionales o de caja negra son un enfoque para llevar a cabo pruebas donde éstas se derivan de la especificación del programa ó componente. El sistema es una “caja negra” cuyo comportamiento solo se puede determinar estudiando las entradas y salidas relacionadas. También se conocen como pruebas funcionales debido a que al probador solo le interesa la funcionalidad y no la implementación del software.

El probador introduce las entradas el los componentes del sistema y examina las salidas correspondientes. Si las salidas no son las previstas, entonces la prueba ha detectado exitosamente un problema con el software.

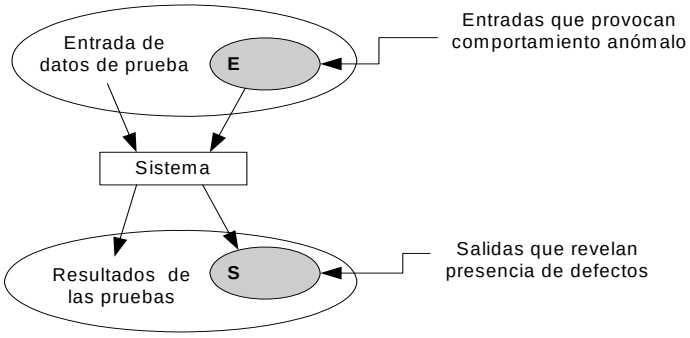


Figura 5.3 *Pruebas de caja negra*

5.1.2 PARTICIONES DE EQUIVALENCIA

Los datos de entrada a un programa por lo regular son de varias clases diferentes. Éstos tiene características comunes, por ejemplo, son números positivos, número negativos, cadenas sin espacios en blanco, etcétera. Normalmente, los programas se comportan de la misma forma para todos los miembros de una clase. Debido a este comportamiento equivalente, algunas veces estas clases se denominan particiones de equivalencia o dominios. Un enfoque sistemático para las pruebas de defectos se basa en identificar todas las particiones de equivalencia que debe de manejar un programa. Los casos de prueba se diseñan de tal forma que las entradas o salidas pertenezcan a estas particiones.

Una vez identificado un conjunto de particiones, se eligen los casos de prueba de cada una de estas particiones. Un buen lineamiento para la selección de los casos de prueba es elegir dichos casos de los límites de las particiones junto con los casos cercanos al punto medio de partición.

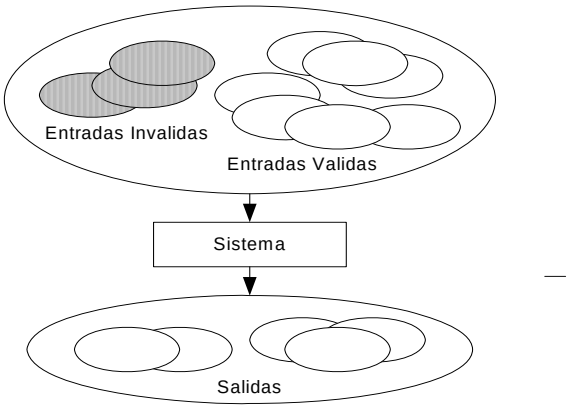


Figura 5.4 *Particiones de equivalencia*

5.1.3 PRUEBAS ESTRUCTURALES

Las pruebas estructurales son el enfoque de prueba donde éstas se generan a partir de la estructura e implementación del software. Algunas veces, a este enfoque se le denomina prueba de “caja blanca, de ”caja de cristal” o de “caja transparente” para distinguirlo del de la prueba de caja negra.

Por lo general, las pruebas estructurales se aplican a unidades del programa relativamente pequeñas como las rutinas o las operaciones asociadas a un objeto. Como su nombre lo indica, el probador puede analizar el código y utilizar el conocimiento de la estructura de un componente para derivar los datos de prueba. El análisis del código se puede utilizar para descubrir cuántos casos de prueba son necesarios para garantizar que todas las instrucciones del programa o componente se ejecuten al menos una vez durante el proceso de prueba.

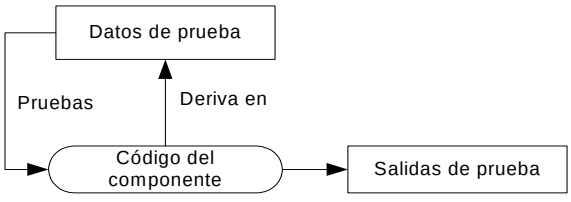


Figura 5.5 *Pruebas estructurales*

5.1.4 PRUEBAS DE TRAYECTORIAS

Las pruebas de trayectorias son una estrategia de prueba estructural cuyo objetivo es probar la trayectoria de ejecución independiente en un componente o programa. Si se ejecuta cada trayectoria independiente, todas las instrucciones en el componente se ejecutan al menos una vez.

Por lo general, el número de trayectorias es un programa proporcional a su tamaño. Puesto que los módulos están integrados en los sistemas, no es factible usar técnicas de prueba estructurales. Por lo tanto las técnicas de pruebas de trayectorias son las que más de utilizan en las pruebas de unidades y en las etapas de prueba de módulos del proceso de pruebas.

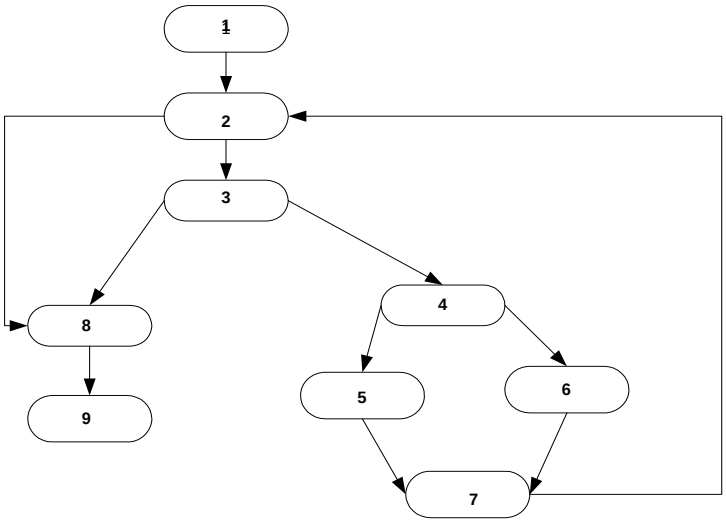


Figura 5.6 *Pruebas de trayectoria*

El número de trayectorias independientes en un programa, se puede descubrir calculando la complejidad ciclomática (McCabe, 1976) del diagrama de flujo del programa. La complejidad ciclomática, se tratará mas adelante en este capítulo (Sección 5.5).

5.2 PRUEBAS DE INTEGRACIÓN

Una vez que se probaron los componentes individuales del programa, deben integrarse para crear un sistema parcial, o completo. Este proceso de integración comprende la construcción del sistema y probar el sistema resultante con respecto a los problemas que surjan de las interacciones de los componentes. Las pruebas de integración se desarrollan a partir de la especificación del sistema y dan inicio tan pronto como estén disponibles versiones utilizables de algunos componentes del sistema.

La principal dificultad que surge en las pruebas de integración es localizar los errores que se descubren durante el proceso. Existen interacciones complejas entre los componentes del sistema y cuando se descubre una salida anómala, es difícil encontrar la fuente del error. Para hacer mas fácil la localización de errores, siempre se utiliza un enfoque incremental para la integración y prueba del sistema. De forma inicial, se debe integrar una configuración mínima del sistema y probar dicho sistema. Luego se agregan componentes a esta configuración mínima y se prueba después de que se agrega cada incremento.

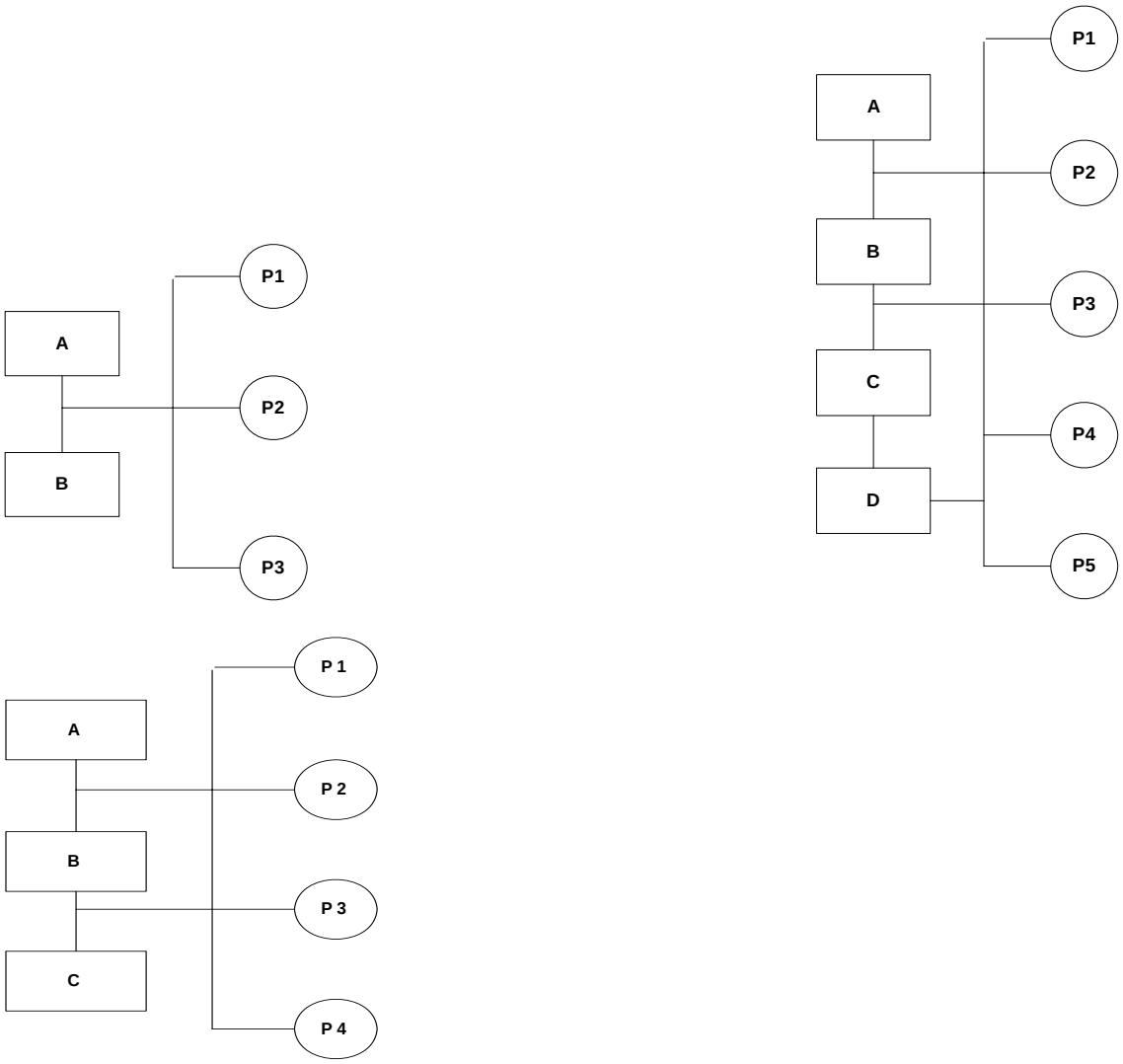


Figura 5.7 *Pruebas de integración*

5.2.1 PRUEBAS DESCENDENTES Y ASCENDENTES

La estrategias de pruebas descendentes y ascendentes reflejan diferentes enfoques de la integración del sistema. En la integración ascendente, los componentes de los niveles altos de un sistema se integran y prueban antes de que se complete su diseño e implementación. En la integración ascendente, los componentes de los niveles bajos se integran y prueban antes de que se desarrollen los componentes de los niveles altos.

Las pruebas de integración, descendentes y ascendentes se comparan bajo cuatro encabezados:

Validación arquitectónica Las pruebas descendentes son susceptibles a descubrir errores en la arquitectura del sistema y en el diseño de alto nivel en las etapas iniciales del proceso de desarrollo. Puesto que éstas son por lo regular errores estructurales, las primeras detecciones implican que se pueden corregir sin costos extra. En las pruebas ascendentes, el diseño de alto nivel no se valida hasta etapas posteriores del proceso.

Demostración del sistema: En el desarrollo descendente se dispone de un sistema funcional limitado en las primeras etapas de desarrollo. Esto es una importante ayuda psicológica para aquellos que están involucrados en el desarrollo del sistema. La validación puede comenzar al inicio del proceso de prueba puesto que los usuarios pueden disponer de un sistema que tiene cierta funcionalidad.

Implementación de las pruebas Las pruebas descendentes estrictas son difíciles de implementar debido a que se deben producir las versiones simplificadas de los componentes requeridos del programa que simulen los niveles inferiores del sistema

Observación de la prueba Tanto las pruebas ascendentes como descendentes pueden tener problemas con la observación de la prueba. En muchos sistemas, los niveles más altos del sistema que se implementan primero en un proceso descendente no genera salidas; sin embargo, para probar estos niveles, se les debe forzar a hacerlos. El probador debe crear un entorno artificial para generar los resultados de la prueba. Para las pruebas ascendentes, también es necesario crear un entorno artificial con el fin de que se pueda observar la ejecución de los componentes inferiores.

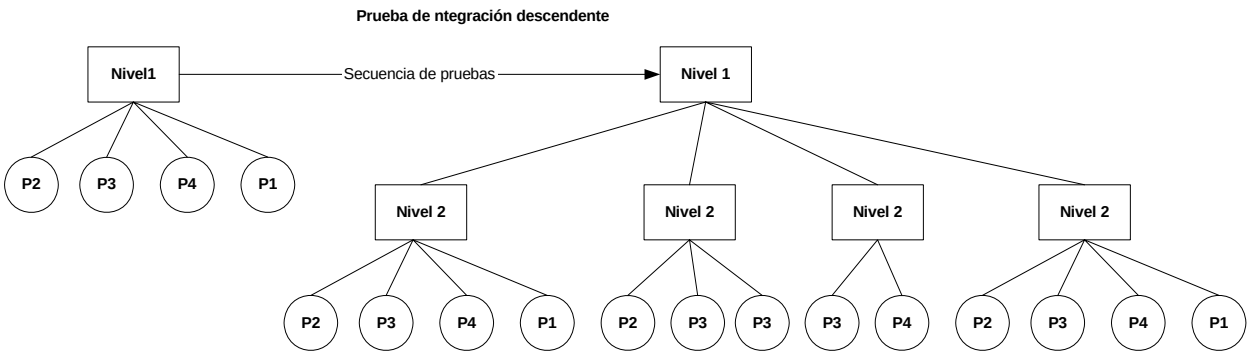


Figura 5.8 *Pruebas descendentes*

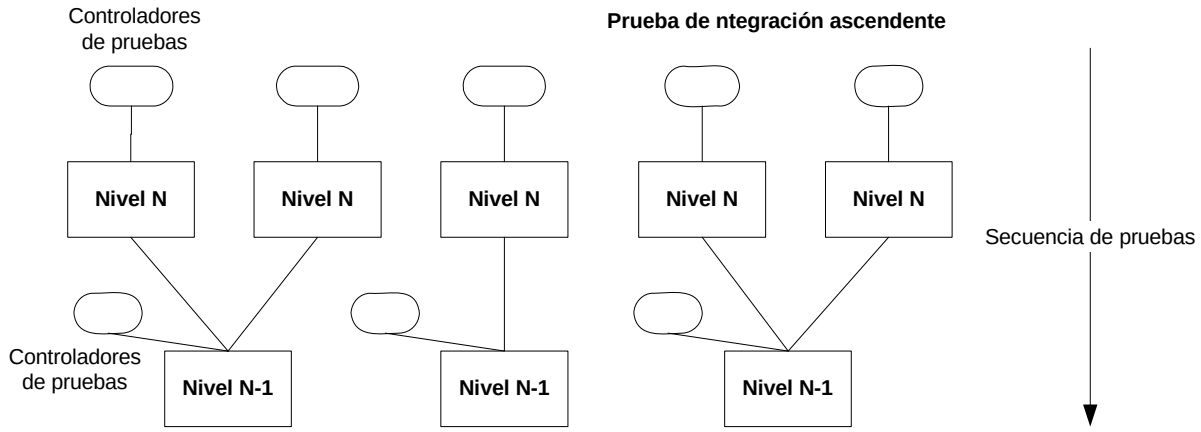


Figura 5.9 *Pruebas Ascendentes*

5.2.2 PRUEBAS DE INTERFACES

Las pruebas de interfaces toman lugar cuando los módulos o subsistemas se integran para crear sistemas más grandes. Cada módulo o subsistema tiene una interfaz definida que es llamada por otros componentes del programa. El objetivo de las pruebas de interfaces es detectar las fallas que se introdujeron en el sistema debido a los errores de las interfaces o suposiciones no válidas a cerca de las interfaces.

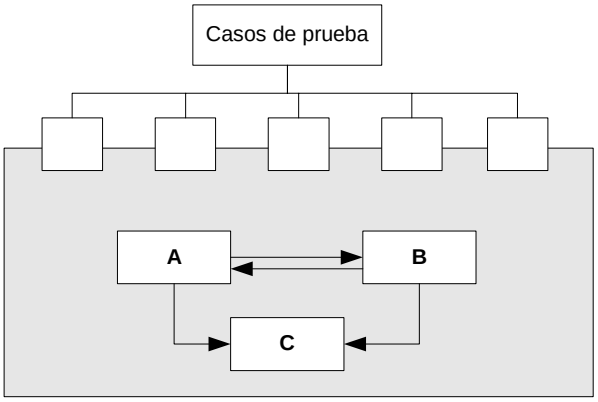


Figura 5.10 *Pruebas de interfaces*

5.2.3 PRUEBAS DE ESFUERZO

Una vez que el sistema se integra completamente es posible probar el sistema de acuerdo a su desempeño y fiabilidad. Se tienen que diseñar pruebas de desempeño para asegurar que el sistema pueda procesar la carga permitida. Por lo general, esto comprende planear una serie de pruebas donde la carga se incrementa de forma constante hasta que el desempeño del sistema sea inaceptable.

Las pruebas de esfuerzo son relevantes de forma particular para sistemas distribuidos basados en una red de procesadores. A menudo estos sistemas exhiben degradación severa cuando se sobrecargan.

5.3 HERRAMIENTA McCabe

En 1970, los sistemas comenzaron a ser muy complejos. Esta complejidad los hizo susceptibles a generar errores y cada vez más difíciles de probar y mantener. Teniendo como antecedente este problema, en 1976, Thomas McCabe desarrolló una técnica matemática, la métrica de complejidad ciclomática. Esta métrica define el software en términos de medidas numéricas de complejidad y permite a los Ingenieros identificar los módulos de software que podrían resultar difíciles de probar o mantener.

Las herramientas de aseguramiento de calidad de McCabe generan métricas que se pueden usar para analizar código que podría causar problemas debido a su alta complejidad o a su difícil mantenimiento.

5.3.1 COMPLEJIDAD CICLOMÁTICA

La complejidad ciclomática es una medida de la complejidad de los módulos que contienen estructuras de decisión. Es el número de partes del programa ligeramente independientes y

por lo tanto el número mínimo de partes de un programa que deben de ser probadas. Una complejidad ciclomática alta indica que el código puede ser de baja calidad y difícil de probar y mantener.

- ***Metodología de cálculo***

Un nodo es la unidad mínima de código en un programa. Las líneas representan la transferencia de control de un nodo a otro

Dado un módulo que contiene *e* líneas y *n* nodos, su complejidad ciclomática es:

V (G) = e – n + 2

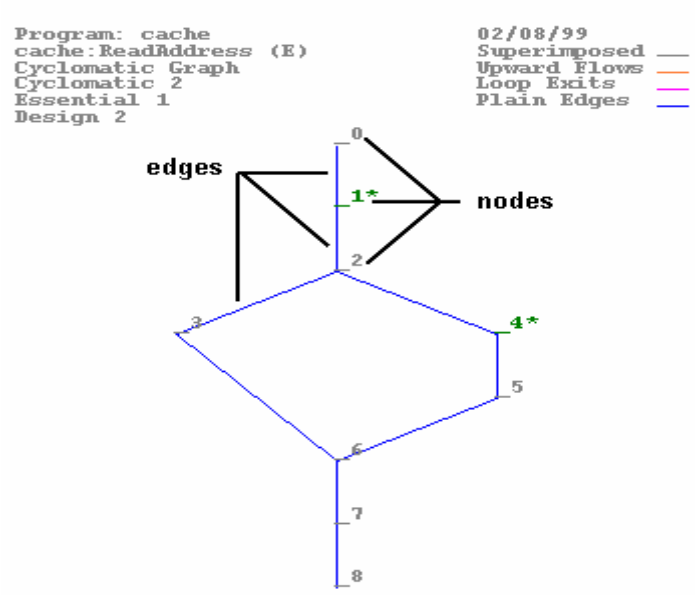


Figura 5.11 *Complejidad Ciclomática*

Ventajas

- Identifica los módulos complejos que pueden requerir subdivisión.
- Mide el mínimo de pruebas que se le deben hacer a un programa
- Aplica a programas desarrollados con cualquier lenguaje de programación y es fácil de probar y mantener.

- Ayuda a estimar el personal necesario para pruebas y mantenimiento

5.3.2 COMPLEJIDAD ESENCIAL Ev (G)

La complejidad esencial mide la cantidad de construcciones sin estructura de un programa.

Las construcciones sin estructura decrementan la calidad del código e incrementan el esfuerzo requerido para el mantenimiento del mismo. Por lo mismo, una complejidad esencial alta implica un número alto de construcciones sin estructura, por lo que el mantenimiento y la modularización son difíciles. De hecho, durante el mantenimiento de este tipo de programas, arreglar un error en alguna sección generalmente genera un error en otra parte del código.

Métodología de Cálculo

La complejidad esencial se calcula removiendo todas las construcciones estructuradas de un módulo y enseguida midiendo la complejidad ciclomática del módulo reducido.

El siguiente esquema contiene una complejidad esencial de 1 y una complejidad ciclomática de 5

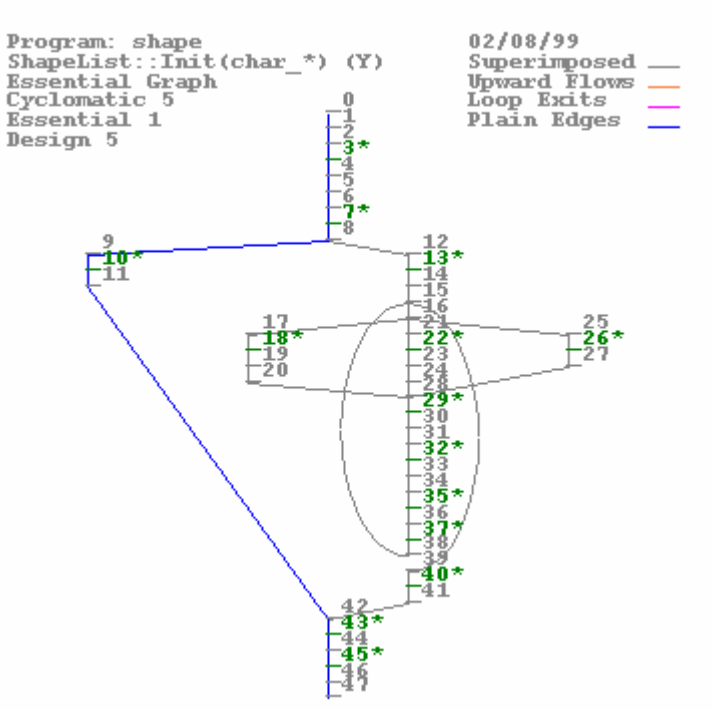


Figura 5.12 *Complejidad Esencial*

La siguiente tabla, muestra la evaluación realizada por la herramienta McCabe, para cada una de las funciones del módulo de organigramas.

Module Name	Mod #	v(G)	ev(G)	iv(G)	# Lines	Line #
FRMLOGON.FRM:GETPRIVATEPROFILESTRING	91	6	1	5	28	82
FRMDIAGRAMA.FRM:CMDOKOPEN_CLICK	19	6	1	4	70	251
FRMDIAGRAMA.FRM:LLENALISTA	81	6	1	4	44	453
FRMDIAGRAMA.FRM:CMDELIMINA_CLICK	12	6	1	4	31	85
FRMLOGON.FRM:BTN_OK_CLICK	34	5	1	5	47	22
FUNCIONES.BAS:RESUMENMOVIMIENTOS	99	5	1	5	43	276
FUNCIONES.BAS:ABREREPORTEESTADISTICO	103	5	1	4	152	359
FUNCIONES.BAS:PLAZASCANCELADAS	101	5	1	4	42	578
FRMMOVIMIENTOS.FRM:CMDOK_CLICK	41	5	1	4	38	59
FUNCIONES.BAS:PERSONALHONORARIOS	196	4	1	3	41	685
FUNCIONES.BAS:CUENTAHONORARIO	179	4	1	3	36	729
FRMDIAGRAMA.FRM:CMDNIVEL5_CLICK	17	4	1	3	20	201
FUNCIONES.BAS:GUARDACELL	61	4	1	2	30	30
FRMDIAGRAMA.FRM:LLENACOMBO	70	4	1	2	27	425
FRMDIAGRAMA.FRM:CMDNIVEL4_CLICK	16	4	1	2	20	180
FRMDIAGRAMA.FRM:CMDNIVEL3_CLICK	15	4	1	2	20	159
FRMDIAGRAMA.FRM:CMDNIVEL2_CLICK	14	4	1	2	20	138
FRMDIAGRAMA.FRM:CMDNIVEL1_CLICK	13	4	1	2	20	117
FRMDIAGRAMA.FRM:CBONIVEL_CLICK	8	4	1	1	13	19
FUNCIONES.BAS:DELETEACCELITEM_EXAMPLE	58	3	3	2	39	189
FRMDIAGRAMA.FRM:USERFORM_INITIALIZE	32	3	1	3	41	383
FUNCIONES.BAS:CREAARBOLPLAZA	57	3	1	3	29	241
FRMREPORTES.FRM:USERFORM_INITIALIZE	49	3	1	3	21	35
FRMMOVIMIENTOS.FRM:USERFORM_INITIALIZE	46	3	1	3	12	114
FRMREPORTES.FRM:CMDOK_CLICK	48	3	1	3	10	22
FUNCIONES.BAS:RECTANGLE1	78	3	1	2	31	653
FRMDIAGRAMA.FRM:CMDNIVEL6_CLICK	18	3	1	2	19	222
FUNCIONES.BAS:REDIMPAGEY	66	3	1	2	10	522
FUNCIONES.BAS:REACOMODAARBOL	63	3	1	1	15	9
FUNCIONES.BAS:REDIMPAGE	65	3	1	1	10	512
FUNCIONES.BAS:OBJETOSCANCELADOS	100	2	1	2	26	551
FRMSAVEAS.FRM:CMDOK_CLICK	50	2	1	2	26	17
FRMMOVIMIENTOS.FRM:CMDMINUSALL_CLICK	39	2	1	2	12	39
FRMMOVIMIENTOS.FRM:CMDADDALL_CLICK	37	2	1	2	11	18
FRMMOVIMIENTOS.FRM:CMDADDONE_CLICK	38	2	1	2	8	30
FRMDIAGRAMA.FRM:CBONIVEL6_CLICK	7	2	1	2	6	66
FRMDIAGRAMA.FRM:CBONIVEL5_CLICK	6	2	1	2	6	60
FRMDIAGRAMA.FRM:CBONIVEL4_CLICK	5	2	1	2	6	53
FRMMOVIMIENTOS.FRM:CMDMINUSONE_CLICK	40	2	1	2	6	52
FRMDIAGRAMA.FRM:CBONIVEL3_CLIC	4	2	1	2	6	47

```
K
FRMDIAGRAMA.FRM:CBONIVEL2_CLIC     3      2     1      2       6     40
K
FRMDIAGRAMA.FRM:CBONIVEL1_CLIC     2      2     1      2       6     33
K
FUNCIONES.BAS:RECTANGLE           64      2     1      1      30    622
FUNCIONES.BAS:SUP                 69      2     1      1      22    766
FUNCIONES.BAS:REDIMPAGEY1         67      2     1      1      10    532
FRMDIAGRAMA.FRM:SPNNO_SPINUP      30      2     1      1       5    372
FRMDIAGRAMA.FRM:SPNNO_SPINDOWN    29      2     1      1       5    365
FUNCIONES.BAS:ADDMENUITEM         55      1     1      1      85    103
FUNCIONES.BAS:LOADMENUSTOFILE     62      1     1      1      23     79
FUNCIONES.BAS:GLUE                59      1     1      1      18    322
FUNCIONES.BAS:SAVEMENUSTOFILE     68      1     1      1      15     63
FUNCIONES.BAS:GLUEPLAZAS          60      1     1      1      14    343
FUNCIONES.BAS:CALCULA_SUELDOS     56      1     1      1      13    799
FRMSAVEAS.FRM:USERFORM_INITIAL    53      1     1      1      13     53
IZE
FRMDIAGRAMA.FRM:COMMANDBUTTON1    22      1     1      1      12    329
_CLICK
FUNCIONES.BAS:CUENTA             159      1     1      1      11    230
FUNCIONES.BAS:PLAZASVACANTES     178      1     1      1      10    789
FUNCIONES.BAS:ABRE                54      1     1      1       7    542
FRMLOGON.FRM:USERFORM_ACTIVATE    36      1     1      1       7     74
FRMDIAGRAMA.FRM:MULTIPAGE1_CHA    26      1     1      1       4    352
NGE
FRMDIAGRAMA.FRM:TXTNOMBRE_CHAN    31      1     1      1       3    379
GE
FRMDIAGRAMA.FRM:OPTESTRUCTURA_    28      1     1      1       3    361
CLICK
FRMDIAGRAMA.FRM:OPTEMPLEADO_CL    27      1     1      1       3    357
ICK
FRMDIAGRAMA.FRM:LSTEMPLEADO_CL    25      1     1      1       3    348
ICK
FRMDIAGRAMA.FRM:LSTDIAGRAMA_CL    24      1     1      1       3    344
ICK
FRMDIAGRAMA.FRM:COMMONDIALOG1_    23      1     1      1       3    325
ENTER
FRMDIAGRAMA.FRM:CMDOK_CLICK       21      1     1      1       3    246
FRMDIAGRAMA.FRM:CMDOK_BEFOREDR    20      1     1      1       3    242
OPORPASTE
FRMMOVIMIENTOS.FRM:LSTMOVIMIEN    45      1     1      1       3    110
TOS_CLICK
FRMMOVIMIENTOS.FRM:LSTELIMINAD    44      1     1      1       3    106
OS_KEYDOWN
FRMMOVIMIENTOS.FRM:LSTELIMINAD    43      1     1      1       3    102
OS_CLICK
FRMMOVIMIENTOS.FRM:LABEL1_CLIC    42      1     1      1       3     98
K
FRMDIAGRAMA.FRM:CMDCANCELOPEN_    10      1     1      1       3     81
CLICK
FRMDIAGRAMA.FRM:CMDCANCEL_CLIC    11      1     1      1       3     77
K
FRMDIAGRAMA.FRM:CHKSUELDOS_CLI     9      1     1      1       3     73
CK
FRMLOGON.FRM:TXTPASSWORD_CHANG    35      1     1      1       3     70
E
FRMSAVEAS.FRM:USERFORM_CLICK      52      1     1      1       3     49
FRMSAVEAS.FRM:TXTGUARDA_CHANGE    51      1     1      1       3     45
FRMLOGON.FRM:BTM_CANCEL_CLICK     33      1     1      1       3     18
FRMREPORTES.FRM:CMDCANCEL_CLIC    47      1     1      1       3     18
K
------------------------------ ----- ------ ----- ------ ------- ------
Total:                                189    82    148    1468
Average:                             2.36  1.03   1.85   18.35
Rows in Report: 80
```

A continuación, se muestra el diagrama generado por la herramienta McCabe, para el módulo de organigramas del sistema que aquí se explica.

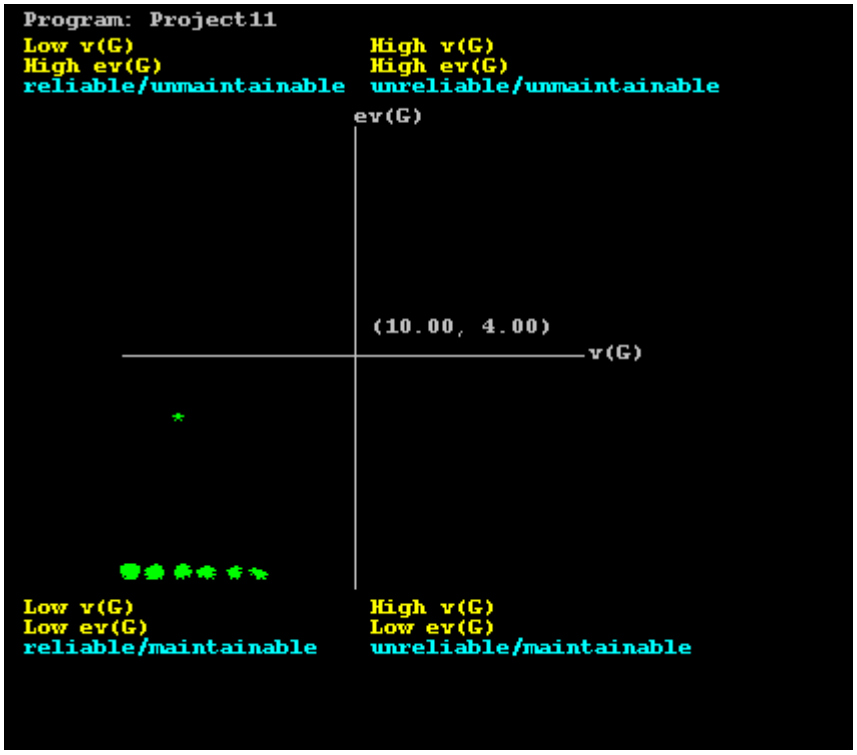


Figura 5.13 *Organigramas*

Como se puede ver en la figura 5.13, todas las funciones caen en el tercer cuadrante, que indica que las funciones son mantenibles y confiables.

5.4 Conclusiones Finales

Como se mencionó al principio de esta tesis, una de las principales tareas de reclutamiento de personal, es proporcionar las capacidades humanas requeridas por la organización. Este elemento se pueden garantizar en gran medida con el diseño de un sistema de selección de personal que provea a la organización del material humano que necesita. Un proceso de selección eficiente, que permita cubrir los puestos de trabajo de tal forma que la organización pueda ser operada de manera competente tanto en el presente, como en el futuro, puede ser la clave para el cumplimiento de la misión organizacional.

Con el desarrollo del sistema que aquí se presenta, se ha optimizado la selección de personal de la UV, ya que se tiene un mayor control de la información de los candidatos para poder identificar aquellos que cumplen todas las características para ocupar una plaza en la Universidad.

También se ha logrado una disminución considerable en los tiempos de desarrollo de organigramas, garantizando que la información de empleados y entidades que en estos se presenta es confiable, ya que, al extraer los datos directamente de la base de datos, se garantiza que la información empata completamente, de acuerdo con la fecha de elaboración del organigrama.

El sistema ha pasado satisfactoriamente por una fase de pruebas, por lo tanto también se puede concluir que el sistema es estable.

Anexo A Manual del Usuario

Con el fin de Ingresar al sistema, hay que dar un doble clic al icono que se encuentra en la pantalla principal de la computadora:



Figura 1 *Ingresar al sistema*

La primer pantalla que se desplegará es la pantalla de login, en la cual se debe de especificar el usuario (en caso de que no aparezca) y la contraseña. Seguido a esto hay que oprimir el botón de conectar.

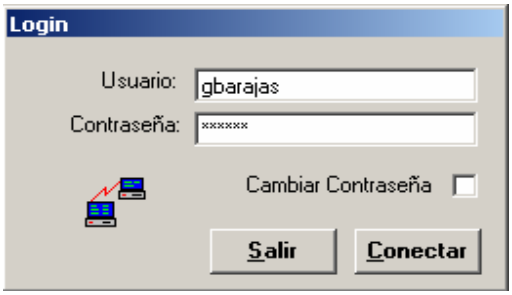


Figura 2 *Pantalla de Login*

Si hubiera un error en los datos, o bien, el usuario no tuviera autorización para usar el sistema, el sistema desplegará un aviso parecido al siguiente:

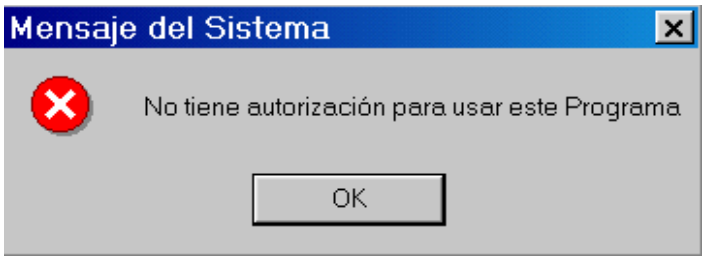


Figura 3 *Mensaje de error*

Si el usuario selecciona la opción de cambiar contraseña, el sistema desplegará una pantalla parecida a la siguiente:

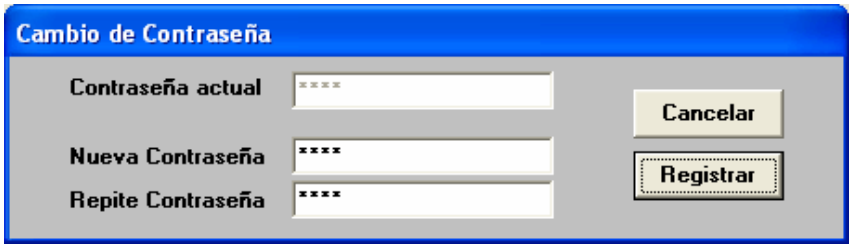


Figura 4 *Cambio de contraseña*

En la figura 4, el usuario deberá seleccionar y confirmar su nueva contraseña, si todo es correcto, el sistema desplegará un mensaje de cambio exitoso de contraseña, de lo contrario, se desplegará el mensaje de error y el usuario deberá volver a introducir su nueva contraseña.

Una vez que el usuario se autentificó correctamente el sistema desplegará la pantalla de menú principal, en donde se muestran botones con las diferentes opciones que el usuario puede seleccionar dependiendo de sus necesidades

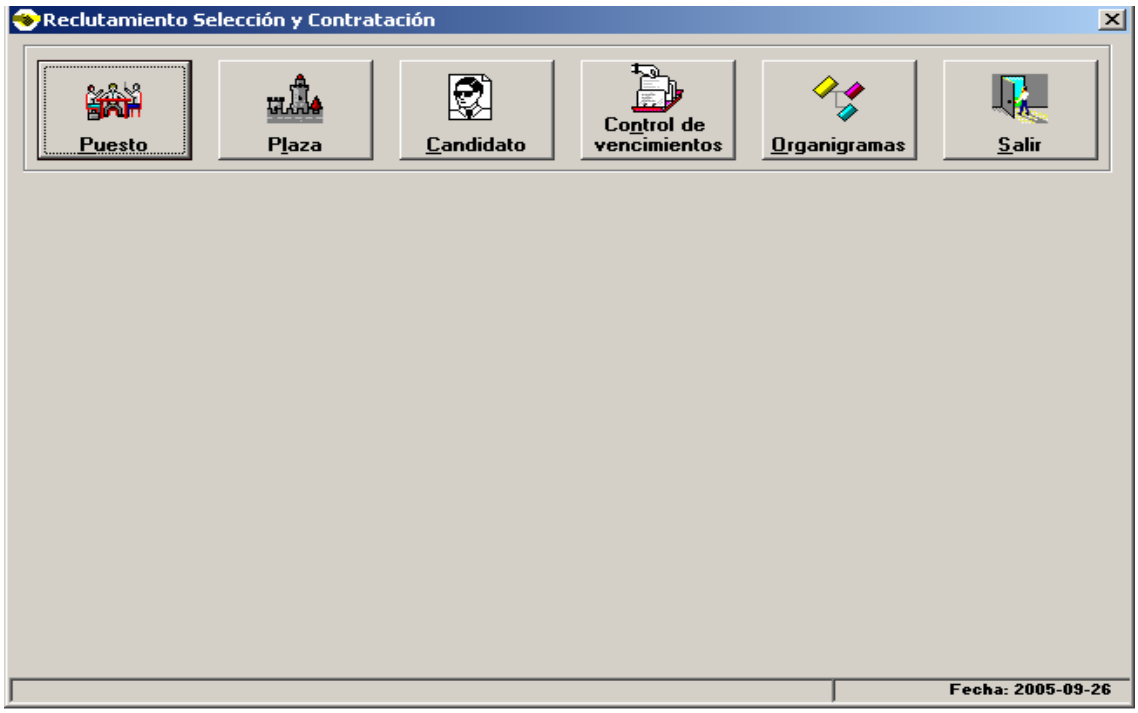


Figura 5 *Menú Principal*

Módulo de Puestos

Si el usuario selecciona el sub-menú puesto que se muestra en la figura 5, el sistema desplegará una pantalla similar a la que se muestra en la figura 6.

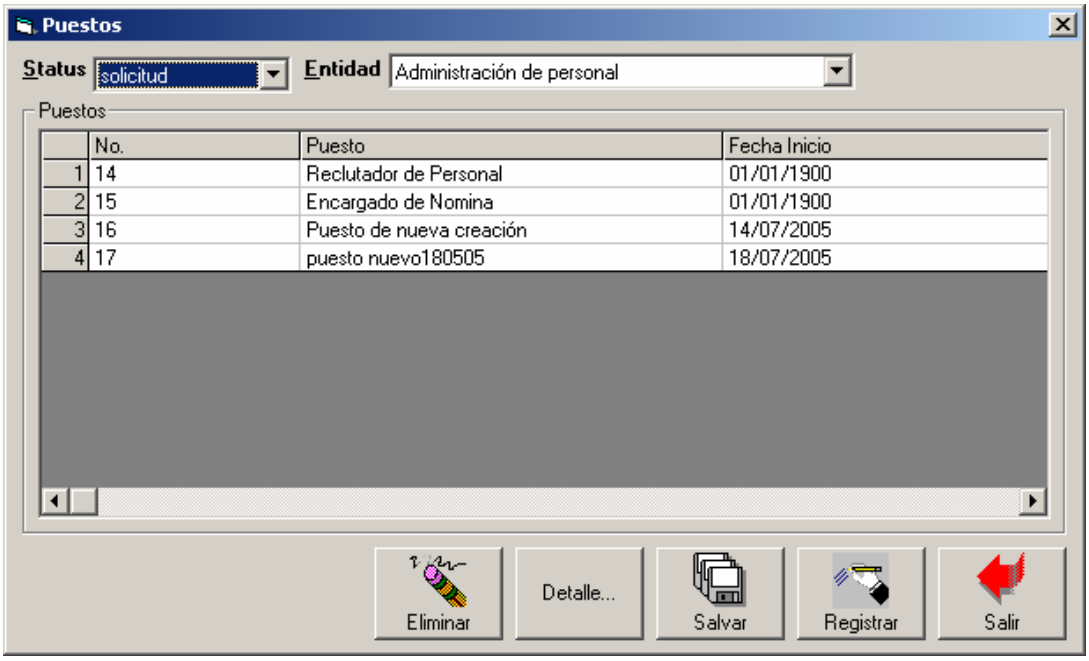


Figura 6 *listado de puestos*

En la pantalla de la figura 6, el sistema mostrará todos los puestos de nueva creación que se encuentran en proceso de aceptación. El usuario podrá seleccionar los puestos por el estatus en que se encuentran o bien por entidad. Cabe mencionar que si el usuario no pertenece a administración de personal, únicamente podrá ver los puestos de la entidad a que el pertenece.

Al seleccionar el botón de detalle, el sistema desplegará una pantalla parecida a la que se muestra en la figura 7, en la que se detalla la información del puesto seleccionado. Si el usuario necesita hacer alguna modificación y tiene los permisos adecuados, podrá hacerlo desde esta pantalla, al terminar de realizar las modificaciones, se deberá de oprimir el botón de registrar. Si el usuario no tiene los permisos adecuados, únicamente podrá ver esta pantalla en modo de consulta.

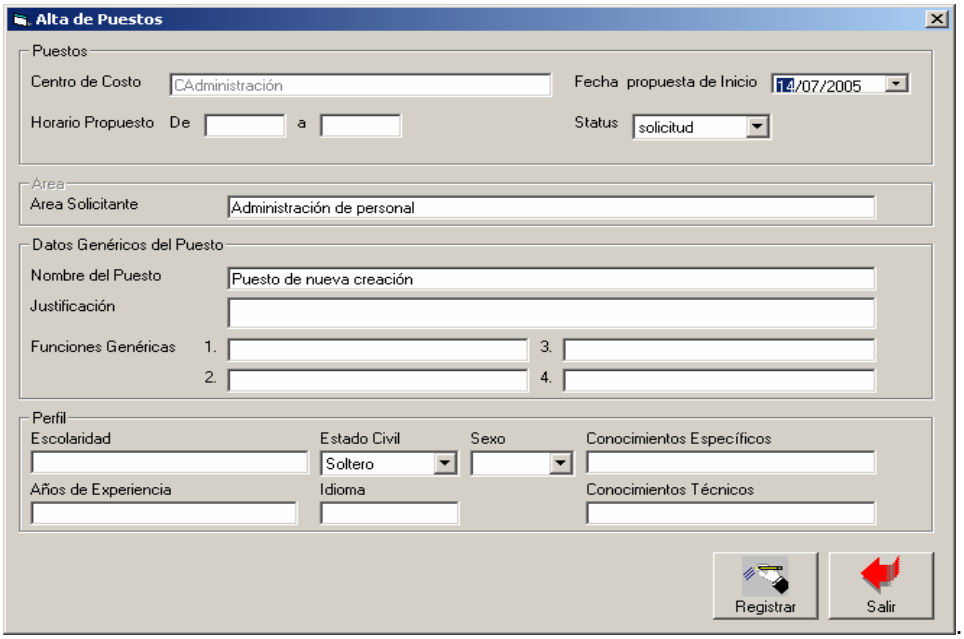


Figura 7 *Detalle de puestos*

En la pantalla de la figura 8, el sistema mostrará todas las plazas de nueva creación que se encuentran en proceso de aceptación. El usuario podrá seleccionar las plazas por el estatus en que se encuentran o bien por entidad. Cabe mencionar que si el usuario no pertenece a administración de personal, únicamente podrá ver las plazas de la entidad a que el pertenece.

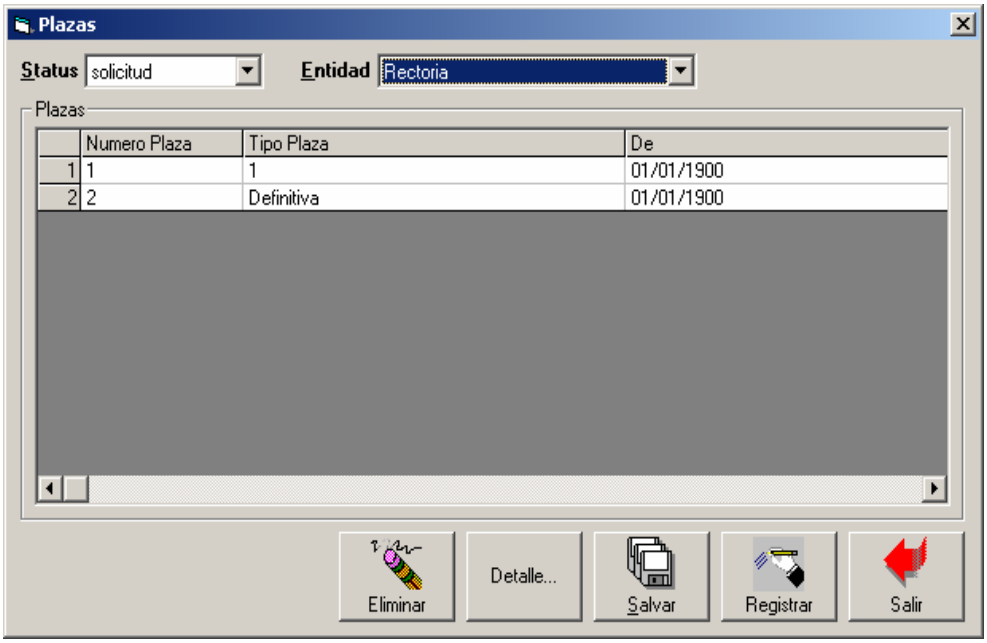


Figura 8 *Listado de plazas*

Al seleccionar el botón de detalle, el sistema desplegará una pantalla parecida a la que se muestra en la figura 9, en la que se detalla la información de la plaza seleccionada. Si el usuario necesita hacer alguna modificación y tiene los permisos adecuados, podrá hacerlo desde esta pantalla, al terminar de realizar las modificaciones, se deberá de oprimir el botón de registrar. Si el usuario no tiene los permisos adecuados, únicamente podrá ver esta pantalla en modo de consulta.

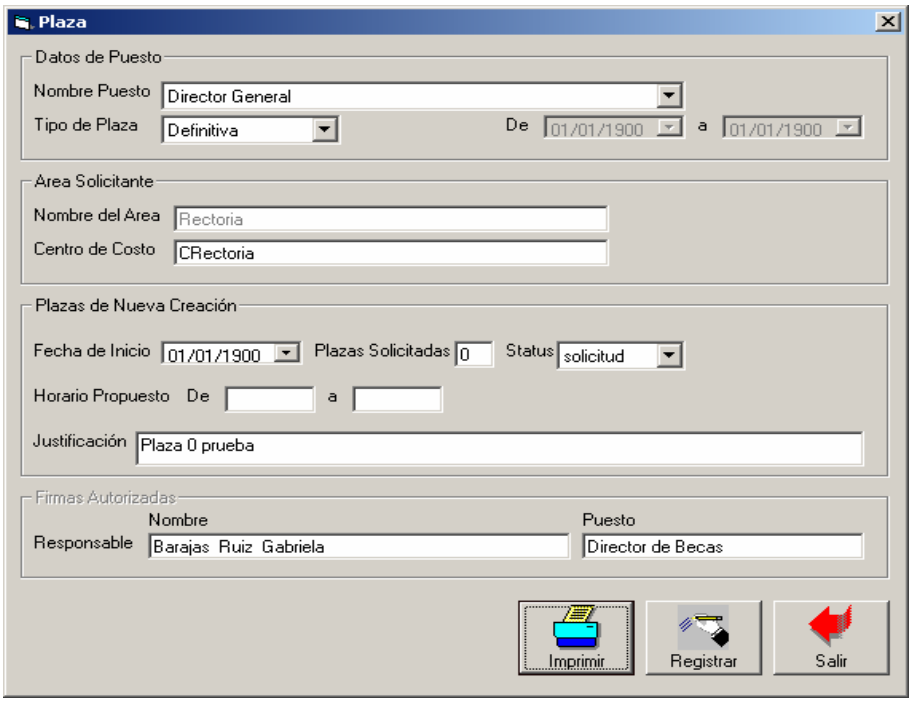


Figura 9 *Detalle de plazas*

Si el usuario selecciona desde el menú principal el botón de candidato, la primer pantalla con la que se encontrará es una parecida a la que se muestra en la figura 10.

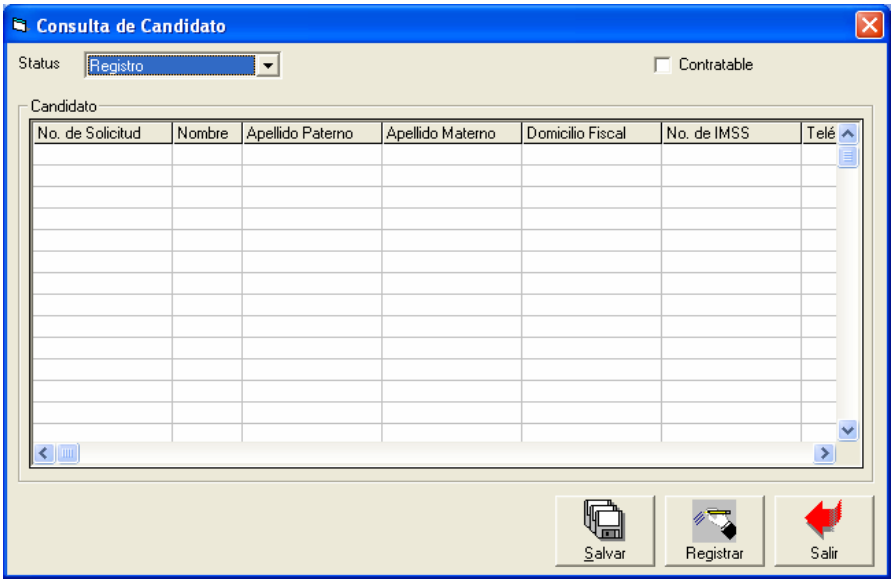


Figura 10 *Listado de candidatos*

Aquí, el usuario podrá ver los datos principales de todos los posibles candidatos a ingresar a la Universidad UV, los usuarios se pueden ver por el estatus en que se encuentran.

Al seleccionar un usuario de la pantalla que se muestra en la figura 6, o bien, al oprimir el botón registrar, se desplegará una pantalla con una serie de sub-menús que se describen a continuación:

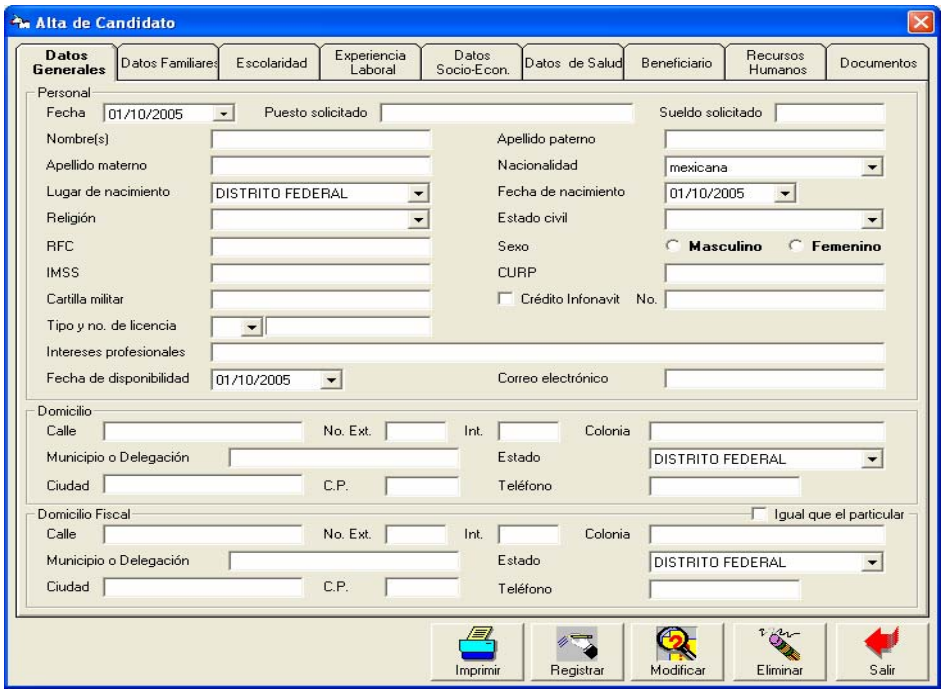


Figura 11 *Datos Generales*

En la pantalla de datos generales, se podrá ingresar información referente al candidato como lo son el nombre, dirección, lugar y fecha de nacimiento, sexo y domicilio fiscal.

En caso de que el usuario esté consultando la información de un candidato en específico, al oprimir el botón modificar podrá guardar los datos que haya modificado del candidato seleccionado.

Si el usuario oprime el botón de imprimir, podrá tener impresa la información del usuario en formato Excel.

Al oprimir el botón de eliminar, el usuario podrá borrar del sistema el candidato seleccionado.

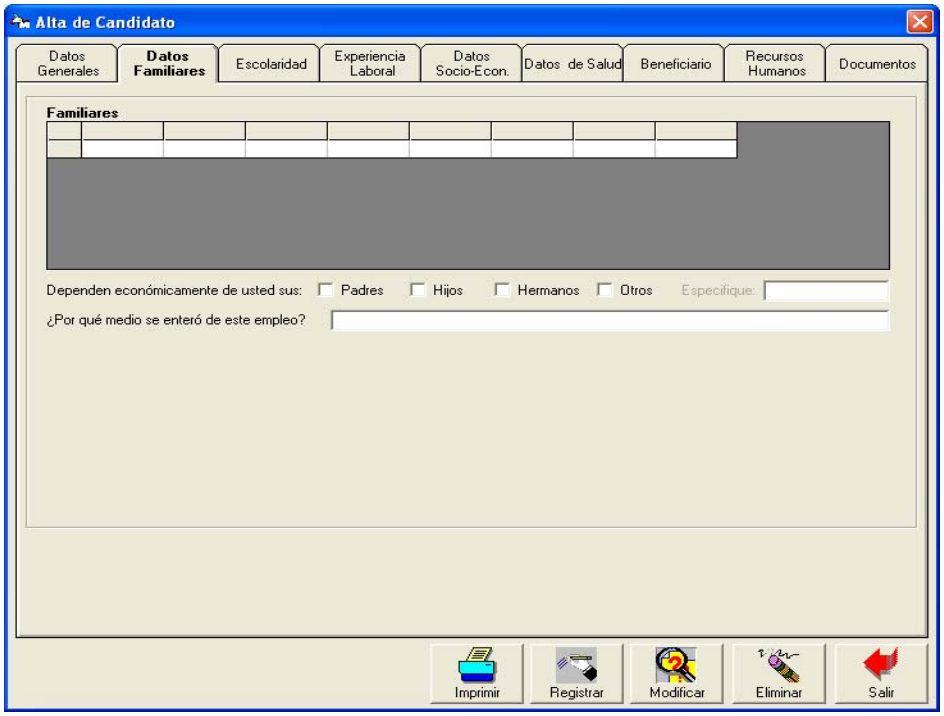


Figura 12 *Datos Familiares*

En la pantalla de la figura 12, el usuario podrá ingresar la información de los familiares del candidato, para lo cual, al seleccionar la opción de registrar, se desplegará una pantalla similar a la que se muestra en la figura 13

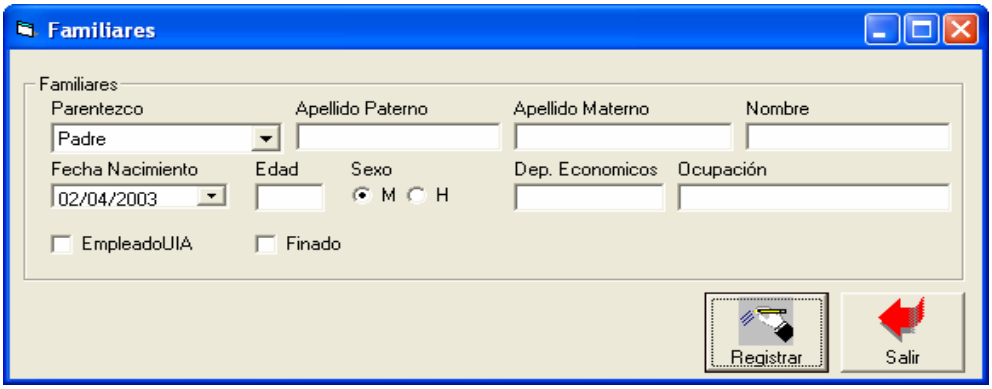


Figura 13 *Familiares*

En el siguiente sub-menú de información de alta de candidatos (figura 14) se ingresa la información referente a la escolaridad.

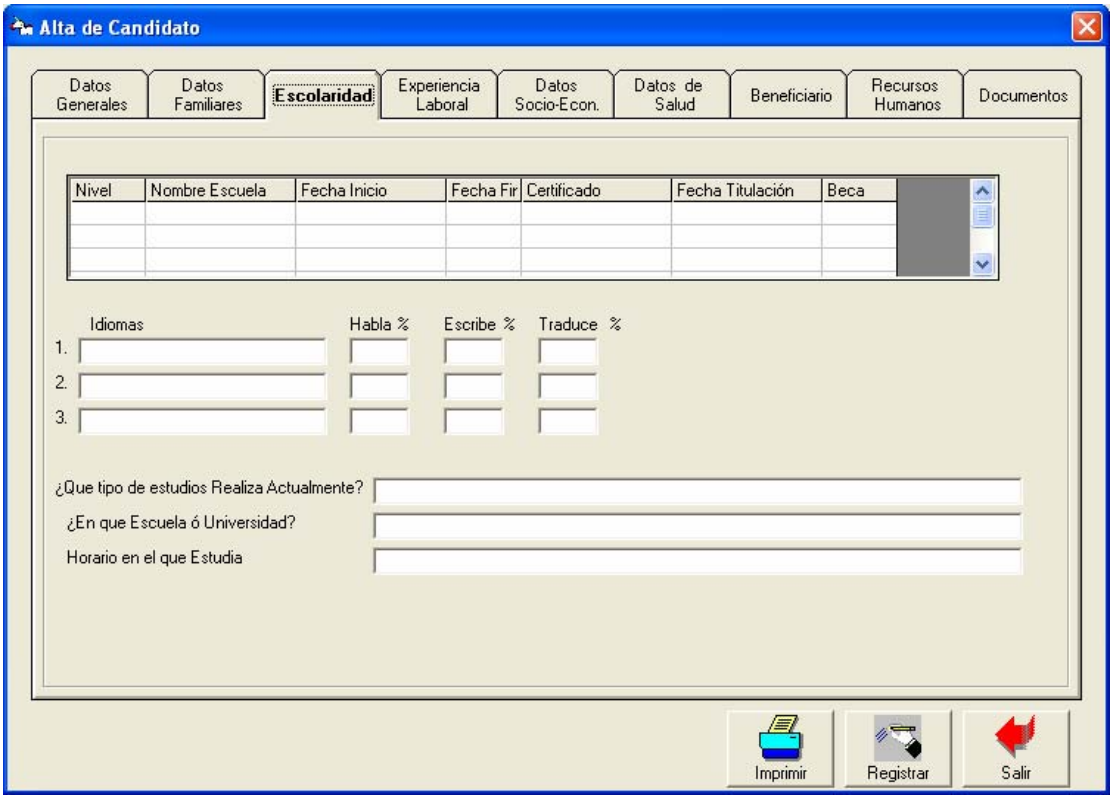


Figura 14 *Datos de Escolaridad*

Al oprimir el botón registrar, el sistema desplegará una pantalla similar a la que se muestra en la figura 15 para ingresar los datos del candidato por escuela de precedencia.



Figura 15 *Escolaridad*

En la pantalla de la figura 16, se lista la experiencia laboral del candidato, con datos como son el nombre de la empresa, domicilio fiscal, código postal, teléfono, nombre del jefe inmediato, puesto anterior y fecha de ingreso

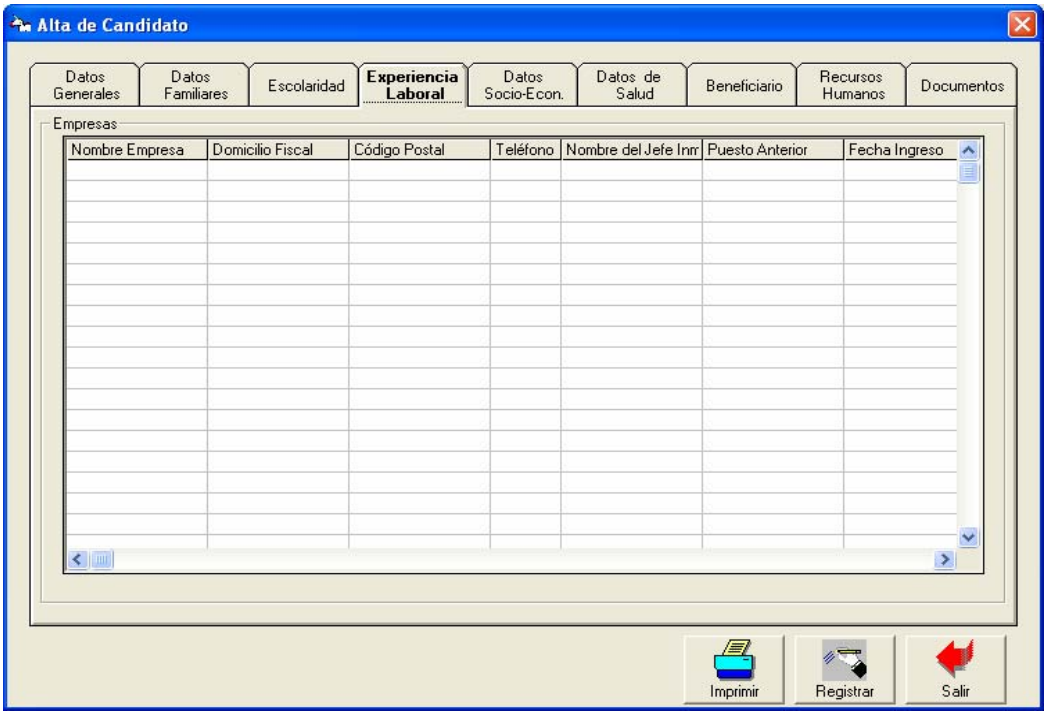


Figura 16 *Datos de Experiencia Laboral*

Al oprimir el botón registrar, el sistema desplegará una pantalla similar a la que se muestra en la figura 17 para ingresar los datos de la experiencia laboral del candidato.

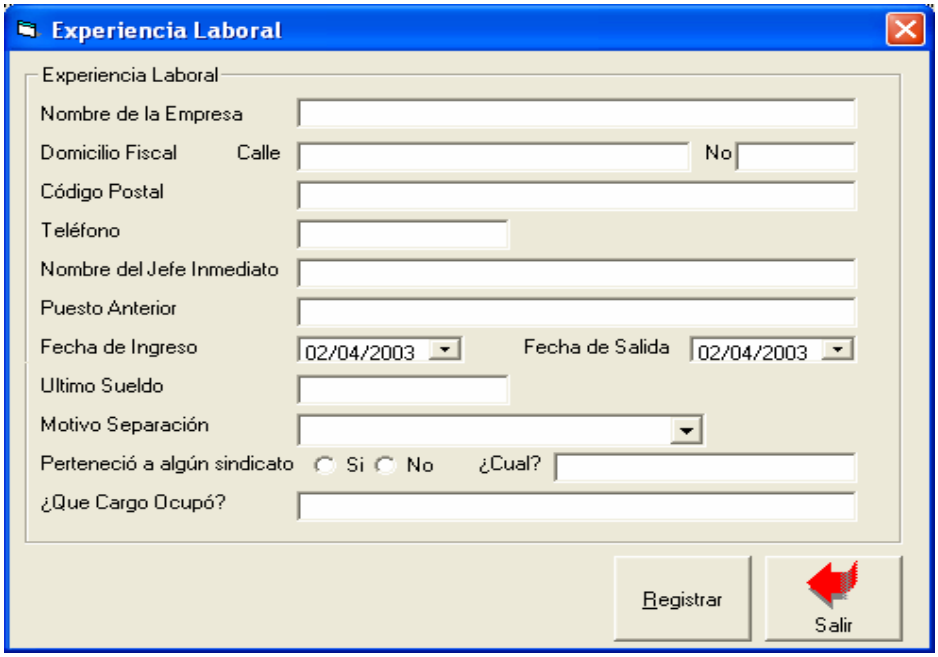


Figura 17 *Experiencia Laboral*

En la figura 18, se muestra la información que el usuario deberá ingresar con respecto a los datos socio-económicos del candidato. Para ingresar las referencias personales, el usuario deberá oprimir el botón de registrar y enseguida se desplegará una pantalla similar a la que se muestra en la figura 19.

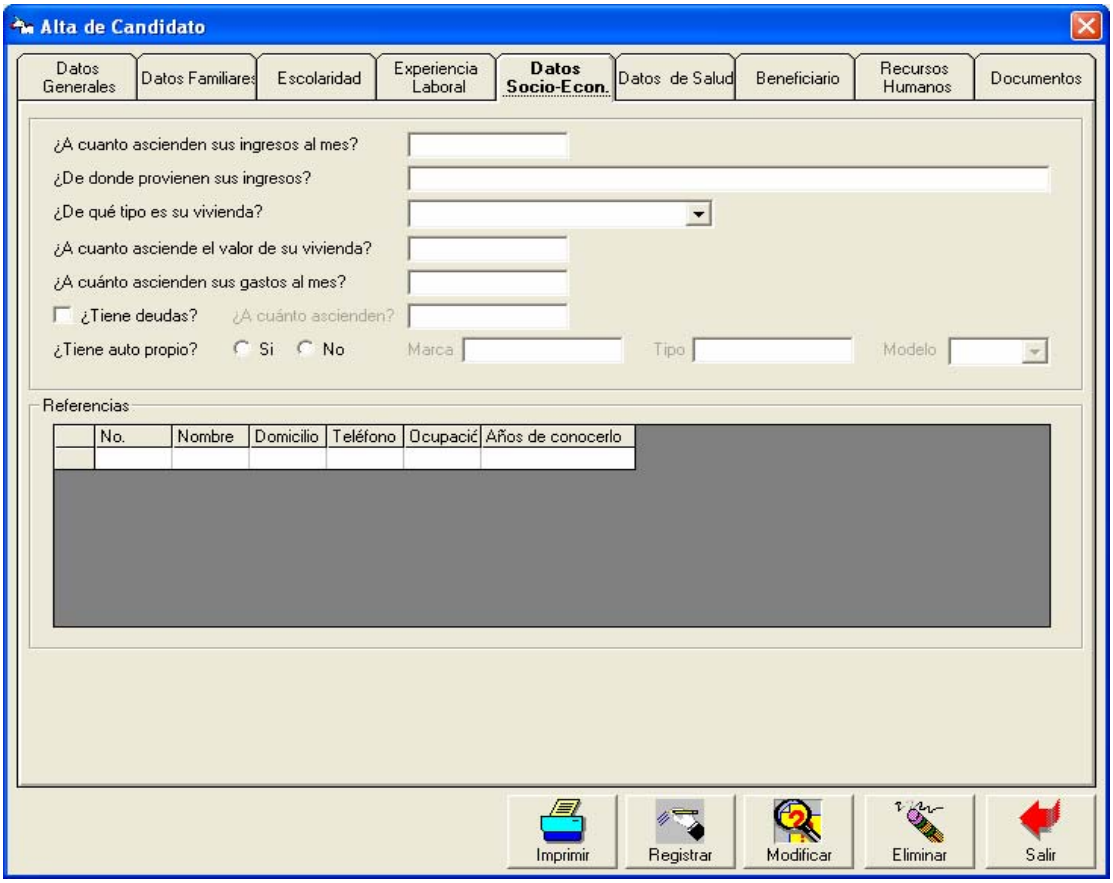


Figura 18 *Datos Socio-Económicos*

En la pantalla que a continuación se muestra, el usuario ingresará la información de las referencias del candidato como lo son el nombre, dirección teléfono, ocupación y tiempo de conocerlo.

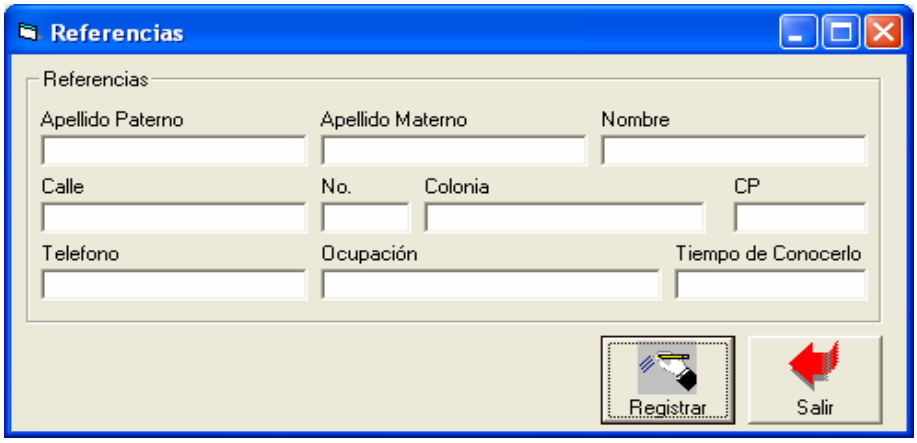


Figura 19 *Referencias*

En la figura 20, se el usuario ingresará la información referente a los datos de salud del candidato.

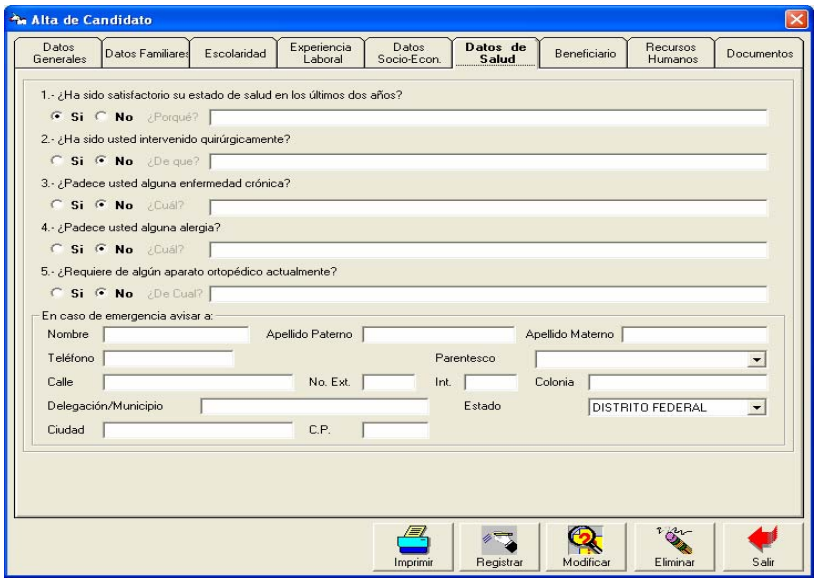


Figura 20 *Datos de salud*

En la pantalla de la figura 21, el usuario podrá ver una lista de los beneficiarios ó dependientes del candidato, para modificar o registrar a un beneficiario, el sistema desplegará una pantalla similar a la que se muestra en la figura 22.

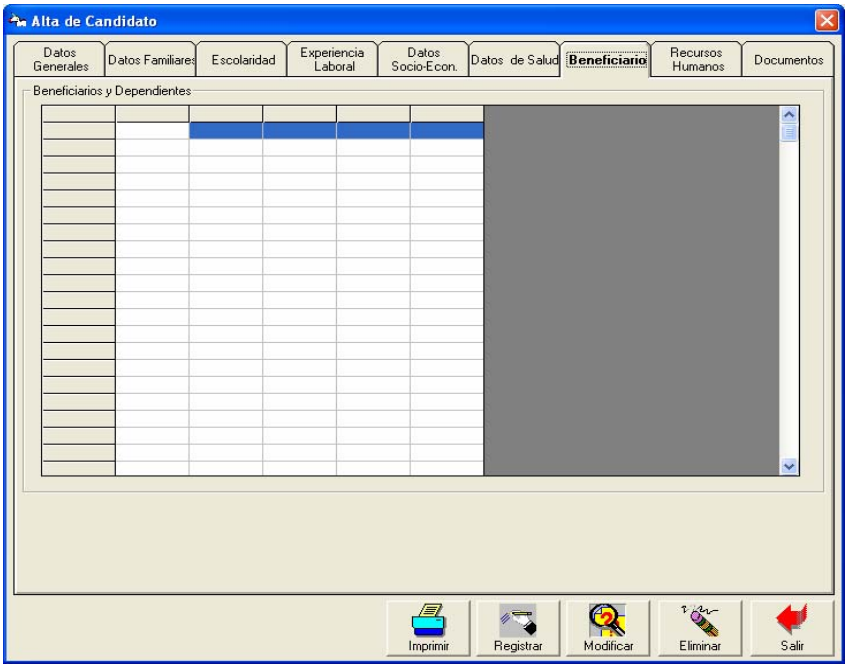


Figura 21 *Beneficiarios*

En esta pantalla, el usuario ingresará datos del beneficiario como nombre, sexo, fecha de nacimiento, parentesco, porcentaje de participación y dirección.

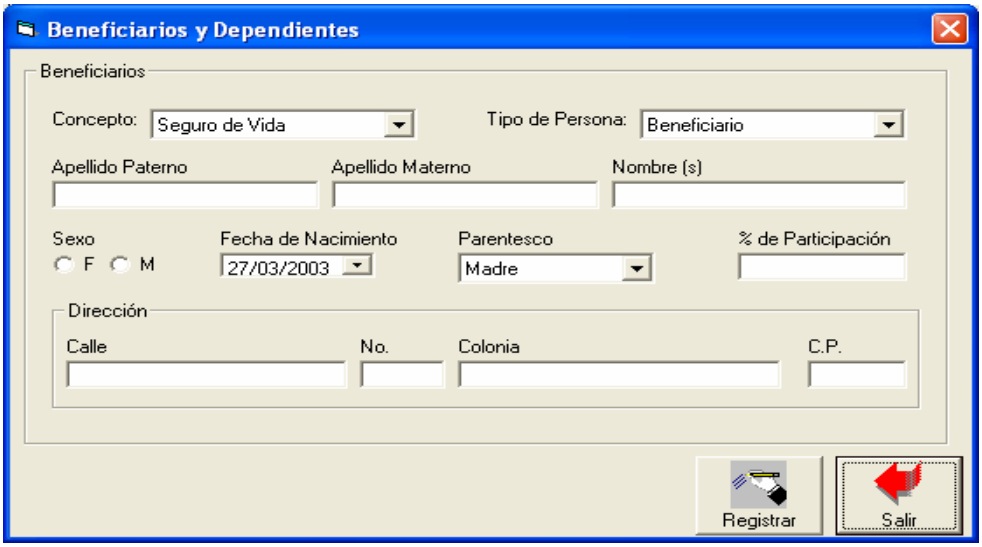


Figura 22 *Datos de beneficiario*

En la sección de recursos humanos, el usuario deberá ingresar la información referente al puesto que el candidato desempeñará, sueldo, fecha de ingreso y tipo de contrato.

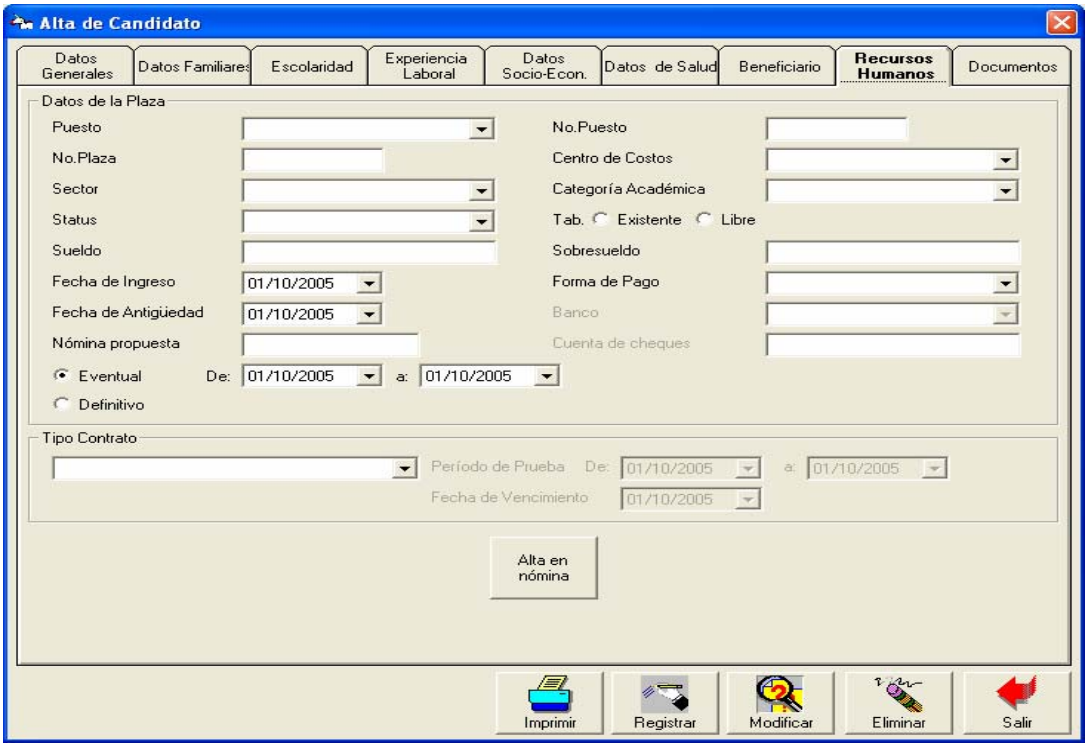


Figura 23 *Recursos Humanos*

En la figura 24, se enlistan los documentos que ha entregado el candidato, para agregar o modificar los datos de un nuevo documento, el sistema desplegará una pantalla similar a la que se muestra en la figura 25.

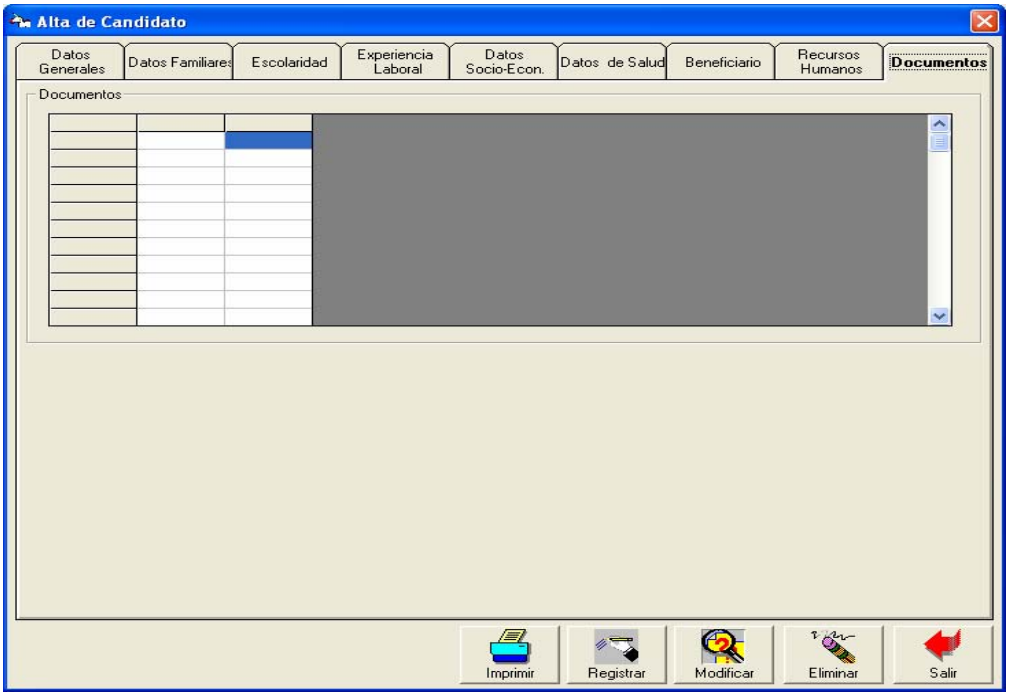


Figura 24 *Documentos*

En esta pantalla, se darán de alta los documentos que el candidato entregue a recursos humanos.



Figura 25 *Alta de Documentos*

Menú de control de vencimientos

Al ingresar al menú de control de vencimientos, la primer pantalla que encontrará el usuario es una similar a la que se muestra en la figura 26

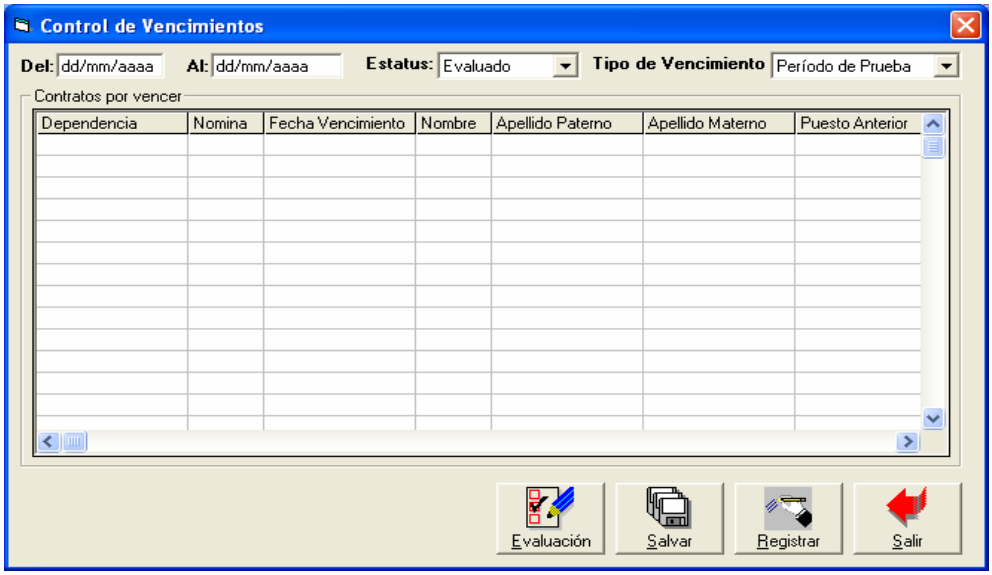


Figura 26 *Control de vencimientos*

En esta pantalla se llevará el control de todos los contratos temporales que se encuentren por vencer, para renovar o evaluar algún contrato eventual, basta con oprimir el botón designado para tal caso.

Modulo de Organigramas.

Al ingresar a este menú, la primer pantalla que aparecerá, es una similar a la que se muestra en la figura 27

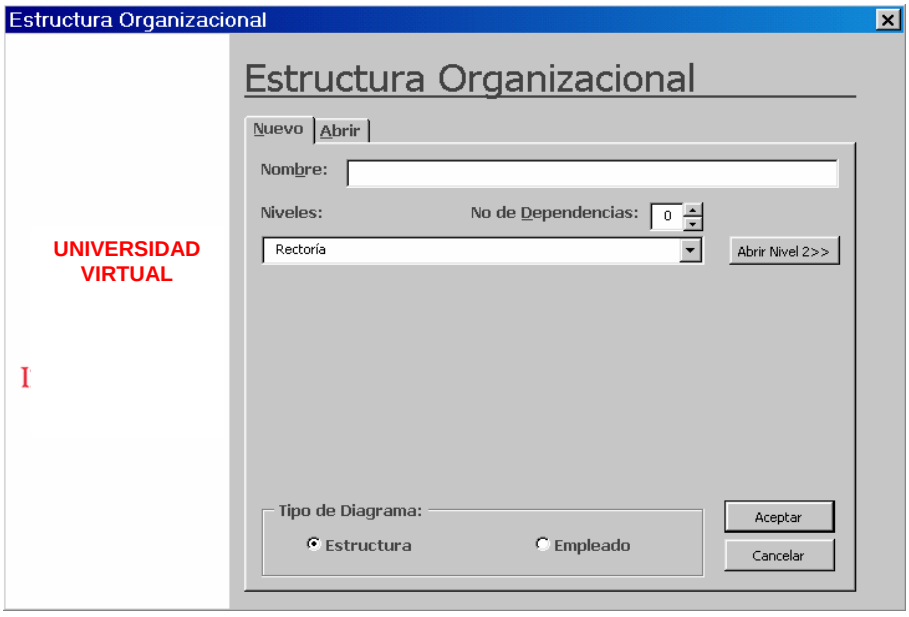


Figura 27 *Creación de organigramas*

1. GENERAR DIAGRAMAS

Los Diagramas se generan de la siguiente forma:

- Dar un nombre al Diagrama:

  Aquí, se debe tener cuidado de dar un nombre diferente a cada diagrama, de lo contrario, se genera un mensaje parecido al de la figura 28:

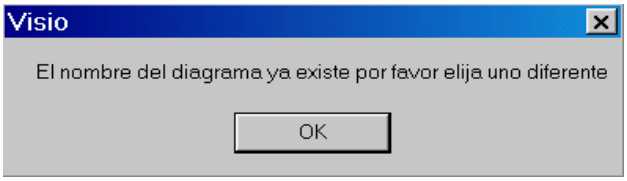


Figura 28 *Mensaje de error de nombre*

- Especificar que tipo de diagrama deseas generar. Existen dos tipos de diagrama, por estructura y por empleado.



Figura 29 *Tipos de organigrama*

En el Diagrama por estructura, se muestran las diferentes entidades dependiendo del nivel seleccionado, y en el diagrama por empleado se muestran las plazas, así como los empleados que ocupan dichas plazas, esto también es según el nivel seleccionado.

- Seleccionar el Nivel

El nivel se refiere a la entidad de la cual se quieren conocer todas sus entidades dependientes, en caso de estructura, y en el caso de empleados a las plazas que pertenecen a la entidad seleccionada y todas sus dependencias. Esto se ilustra en la figura 30

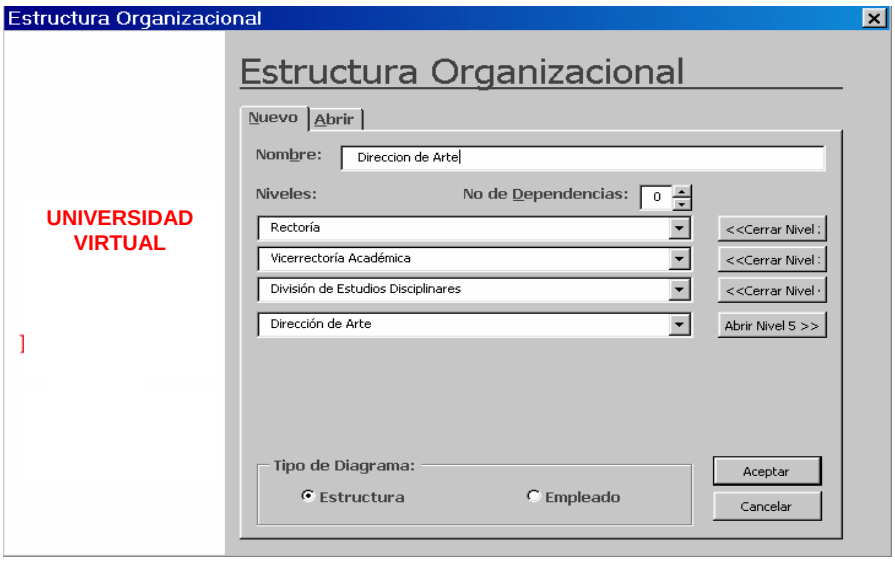


Figura 30 *Niveles*

- Indicar el número de Dependencias

Las dependencias restringen el diagrama, es decir, si se quisiera generar el diagrama de rectoría, pero únicamente con las entidades que dependen directamente de esta, en la parte de dependencias se tendría que seleccionar un uno, como se muestra en la figura 31:



Figura 31 *Dependencias*

El diagrama quedaría parecido al siguiente:

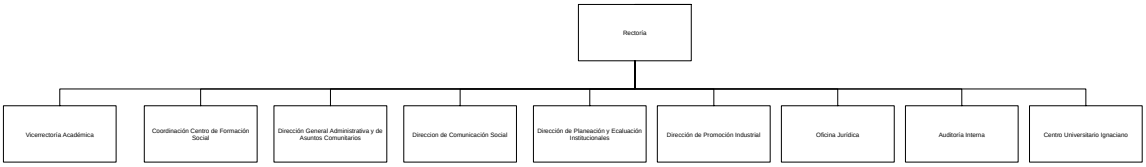


Figura 32 *Organigrama por Entidades*

- Oprimir el Botón de Aceptar

Para los Diagramas de empleados es la misma funcionalidad, solo que -como ya se mencionó anteriormente- se generan las plantillas pertenecientes a las entidades, con sus respectivos empleados.

Nota: *El cero es el valor por default y si este se encuentra seleccionado, el diagrama se genera completo*

2. IMPRIMIR DIAGRAMAS

Para imprimir, solo hay que dar un clic en el icono de impresión, parecido al siguiente:



Figura 33 *Impresión*

En la presentación preliminar se podrá ver la forma final en la que quedará impreso el diagrama y el número de hojas que ocupará

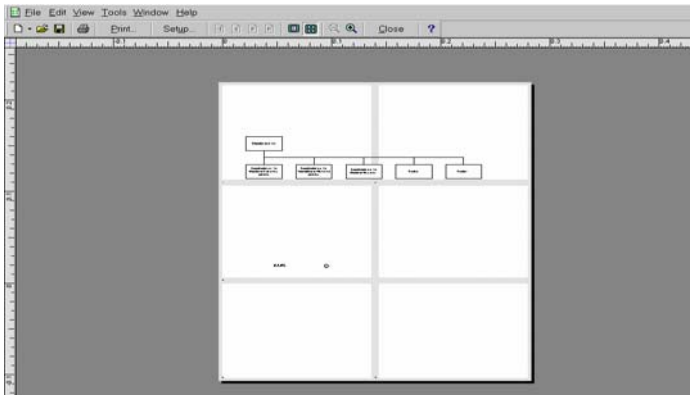


Figura 34 *Vista Preliminar*

En caso de querer modificar el número de hojas en que quedará impreso, se deberá dar un clic al botón de Setup, dentro de la presentación preliminar, lo que llevará a una pantalla parecida a la que se muestra en la figura 35:

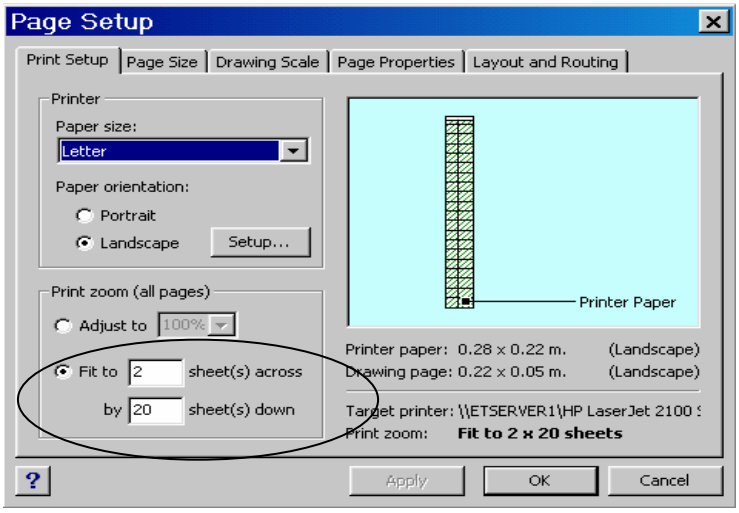


Figura 35 *Page Setup*

Desde esta pantalla se podrá modificar el número de hojas en que aparecerá el diagrama.

3. ABRIR DIAGRAMA

Al momento de generar los diagramas, estos se guardan automáticamente con el nombre que previamente se les asignó, al hacer clic a la pestaña de abrir(que se encuentra en la pantalla

principal) se mostrará un listado de todos los Diagramas por Estructura y por Empleado que se encuentran almacenados hasta el momento.

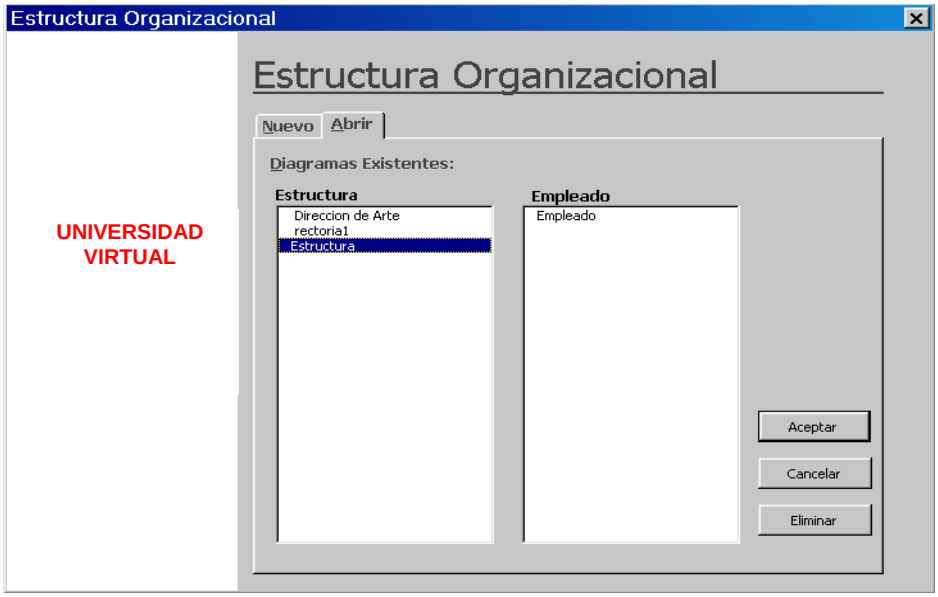


Figura 36 *Abrir Organigrama*

En la parte derecha de esta pantalla se muestran tres botones los cuales tienen la siguiente funcionalidad:

- **Aceptar** Al oprimir este botón, se abre el diagrama que se encuentre seleccionado. En el caso particular del ejemplo, se desplegaría el diagrama de nombre *Estructura*.
- **Cancelar** Con este botón se cancela la operación.

- **Eliminar** Elimina el diagrama seleccionado**.**

La pantalla en la cual se generan los diagramas es parecida a la siguiente:

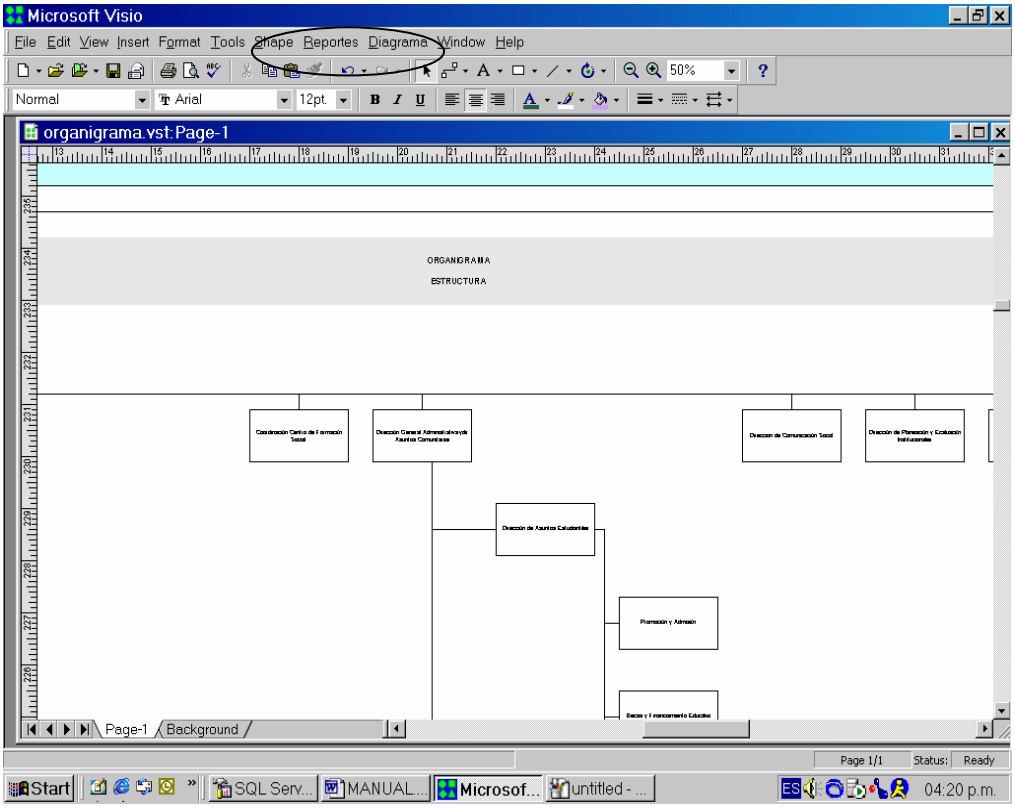


Figura 37 *Presentación de Organigrama*

En la parte superior de esta pantalla, se muestra un menú, con diferentes funcionalidades, las únicas funciones que serán utilizadas por el usuario son los menús de reportes y Diagrama -indicados por el circulo en la pantalla anterior-.

- **Diagrama**

Este menú, contiene los siguientes submenús:

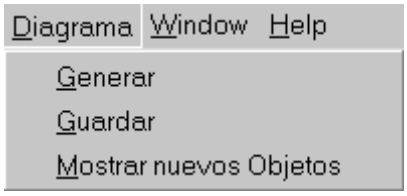


Figura 38 *Menú*

- **Generar**, despliega la pantalla principal
- **Guardar**: En caso de hacerle alguna modificación al Diagrama y querer guardar los cambios, se le dará clic a esta parte del menú.

Es importante mencionar que se pueden modificar los diagramas, siempre y cuando no se afecte la lógica de estos, es decir, no se pueden eliminar las ligas ya que estas son creadas dependiendo de la información que se encuentra almacenada en la base de datos, tampoco se pueden cambiar hijos hacia diferentes padres por la razón explicada anteriormente.

- **Mostrar Nuevos Objetos**

Al dar clic en esta opción, se muestra una pantalla similar a la de la figura 39:

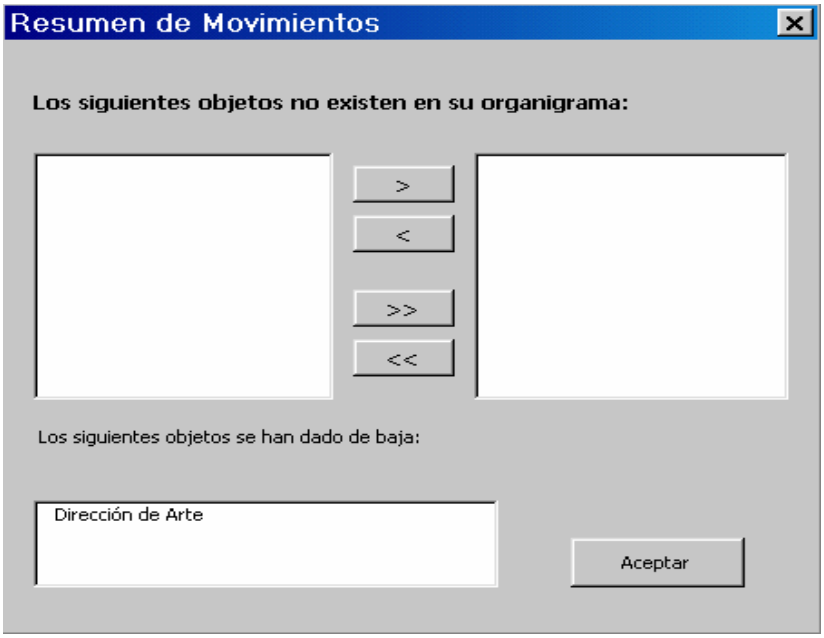


Figura 39 *Nuevos Objetos*

En el recuadro superior izquierdo, se muestran las plazas o entidades (según sea el caso) que se hayan dado de alta en la Base de Datos después de que se generó el diagrama y por lo tanto aún no existan en este, con los botones centrales se podrán pasar estas entidades o

plazas del lado derecho y al oprimir el botón de aceptar, estas serán agregadas al diagrama en la parte inferior izquierda, ligadas a sus respectivos padres, únicamente restará acomodarlas y guardar los cambios realizados a este diagrama para que se encuentre completo.

El recuadro inferior, lista las entidades o plazas que hayan sido dadas de baja. Después de oprimir el botón de aceptar, estas entidades aparecerán en el diagrama de color gris, lo que las diferenciará de las demás plazas.



Figura 40 *Baja de plazas*

5. REPORTES

Al dar clic en esta parte del menú, se desplegará la siguiente pantalla**:**

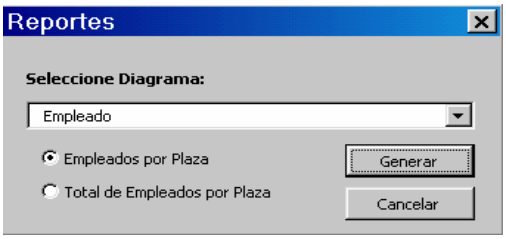


Figura 41 *Selección de reporte*

Se pueden generar dos tipos de reportes, empleados por plaza y total de empleados por plaza, para seleccionar el reporte deseado, se dará clic en el ovalo que aparece en la parte izquierda de cada uno de los dos tipos de reporte, una vez hecho lo anterior se dará clic al botón de generar, lo que conduce a la siguiente pantalla:

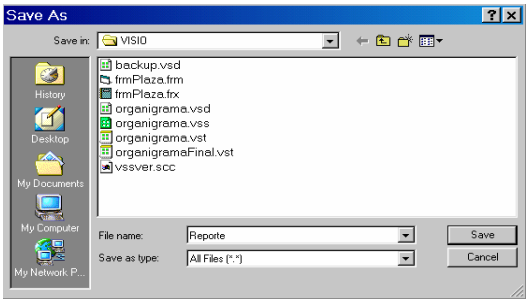


Figura 42 *Salvar reporte*

Aquí se selecciona la carpeta en la que se guardará el reporte, (como en cualquier aplicación común de Windows), y una vez hecho lo anterior, se dará clic el botón de aceptar. Es probable que después de hacer esto, el sistema se quede en espera durante algunos segundos, esto es por que se esta generando el reporte.

Para ver los reportes, vasta con ir a la carpeta en la que previamente fue almacenado, y el reporte se presentará parecido al siguiente:

UNIVERSIDAD VIRTUAL

PLANTILLA DE PUESTOS Y PLAZAS

DATOS DEL PUESTO ACTUAL

NOMINA	FECHA DE INGRESO	S/C/L	NOMBRE EMPLEADO
	ANALISTA DE SISTEMAS (Igual al CCT)		
5307	02/05/2001	S	ALBOR VELAZQUEZ JULIO CESAR
1307	06/11/1978	S	CHAVEZ PEREZ JOSE ANTONIO MAYOLIN
2728	13/10/1987	S	MORENO NOLASCO GLORIA
4280	03/08/2000	S	TORRES MARTINEZ HECTOR HUGO
5306	02/05/2001	S	ZAMACONA CABRA GERARDO
	AUXILIAR DE CONMUTADOR (Igual al CCT)		
4170	05/06/2000	S	PETROVICH SUAREZ ALBORADA
	JEFE DE ANÁLISIS Y PROGRAMACIÓN (Igual al CCT)		
4321	02/01/2001	C	BADO SERRANO CARLOS ENRIQUE
4323	02/01/2001	C	CAMACHO ESCOBAR ADRIANA DEL CARMEN
4295	25/09/2000	C	CHACHA SANCHEZ ADOLFO
	JEFE DE OPERACIÓN (SOPORTE TÉCNICO) (Igual al CCT)		
4220	16/05/2000	C	PONTES GONZALEZ EDUARDO
	JEFE DE PROYECTO (Igual al CCT)		
2656	16/02/1987	S	CAMPOS SANCHEZ JUAN MIGUEL
2275	01/08/1984	S	DE LA PIEDRA GONZALEZ MARIA DEL CARMEN
4319	06/11/2000	S	HERNANDEZ MUÑOZ JOSE LUIS
	OPERADOR DE COMPUTADORA (Igual al CCT)		
1240	16/08/1978	S	VARGAS TREJO ANDRES

Figura 43 *Reporte por Puesto*

Nota: Preferentemente, al generar los reportes, debe de encontrarse cerrada la aplicación de Excel, esto es por que algunas veces el sistema detecta que se esta usando la aplicación y no genera los reportes correctamente.

Para salir del sistema, vasta con ir al menú file que se encuentra en la parte superior izquierda de la aplicación y dar clic en la opción de Exit. Probablemente el sistema preguntará si se quieren guardar los cambios, a lo cual se responderá que no, por que los cambios ya fueron almacenados previamente.

Anexo B Glosario

Base de Datos: Es una colección organizada de datos almacenados la cual es definida independientemente de los procesos que la usarán y soporta la definición y mantenimiento de las relaciones entre datos.

Entidad: Algo que tenga existencia distinta y separada y acerca de lo cual la empresa necesita conocer o mantener información ya sea real ó imaginaria

ERD: Entity-Relationship Diagram (Diagrama entidad-relación)

Estandarización: Proceso que permite ajustar los datos de acuerdo a ciertas normas de ingeniería ú organizacionales

Evento: Cuando ocurre algo (borrar, agregar, conectar) bajo cierta circunstancia

Fallo: La incapacidad de un sistema o de alguno de sus componentes para realizar las funciones requeridas dentro de los requisitos de rendimiento especificados.

Nodo: Computadora conectada a una red de área local por un medio físico (cable).

Plaza: Una Plaza se refiere a la asignación de un puesto a un individuo, esta asignación se realiza dependiendo del lugar físico y de las vacantes de la empresa. Dicho de otra forma, la plaza es el lugar físico con que cuenta la empresa para que el individuo pueda desempeñar su puesto. Así por ejemplo, el puesto de “secretaria” puede contar con 5 plazas dentro de la Empresa X, estas plazas siempre deberán estar cubiertas por “Secretarias”.

Protocolo: Es la definición de cómo deben de comunicarse dos computadoras y sus reglas de comportamiento

Puesto: Un puesto se refiere a un conjunto de actividades y habilidades específicas que un individuo deberá desempeñar dentro de la empresa. Estas actividades conjuntas, le dan al

puesto un perfil y un nombre único. Así por ejemplo, el término “Secretaria” se refiere a un puesto con actividades y habilidades específicas como: sintetizar, organizar, resumir, calendarizar, etc.

Reglas: Estándares para describir y documentar el sistema en forma correcta y completa

Símbolos gráficos: Iconos y convenciones para identificar y describir los componentes de un sistema junto con las relaciones entre estos componentes

Sistema: Es un conjunto de componentes que interaccionan entre si para lograr un objetivo común.

SQL: (Structured Query Languajes) Lenguaje de consulta estructurado.

.

BIBLIOGRAFÍA

Apuntes del diplomado en Bases de Datos impartido por la UNAM (DGSCA 2005)

Granados, Jaime A (2004). Reclutamiento, selección, contratación de personal. México, El manual moderno.

Ian Sommerville (2002). Ingeniería de software .México, Pearson Educación.

Javier Ceballos Sierra (2004). Enciclopedia de Microsoft Visual Basic 6 .México Mc Graw Hill.

Kendall & Kendall (1997). Técnicas de análisis y diseño de sistemas. México, Mc Graw Hill.

Levine Guillermo (1989). Introducción a la computación y a la programación estructurada. México Mc Graw Hill.

Ralph Grabowski (2000).Visio 2000.México Mc Graw Hill.

Sánchez Ambriz, Gerardo (2002). Tesis Profesional, un problema, una hipótesis, una solución. México, UNAM.

Senn, James A.(2004). Análisis y diseño de sistemas de información. México, Mc Graw Hill.

Silberschatz Galván (1997). Sistemas Operativos. México Addison Wesley Longman.