

29
Zj



**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE INGENIERIA

**C.A.S.E., LA OPCION ACTUAL
EN LA INGENIERIA DE PROGRAMACION**

T E S I S

**QUE PARA OBTNER EL TITULO DE:
INGENIERO EN COMPUTACION
P R E S E N T A N :
CASTAÑEDA PERDOMO MARICELA
GONZALEZ CASTRO VICTOR**

DIRECTOR DE LA TESIS: ING. ADOLFO MILLAN NAJERA

MEXICO, D.F.

1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE.

CAPITULO I

Introducción	2
------------------------	---

CAPITULO II

Antecedentes y definición del problema.

II.1 Antecedentes	5
II.2 La primera generación CASE.	6
II.3 La segunda generación CASE.	7
II.4 La incorporación de la tecnología de bases de datos	8
II.5 El Banco de Horas	11

CAPITULO III

La Metodología CASE.

III.1 La Metodología CASE.	13
III.2 Etapa de estrategia.	16
III.3 Etapa de análisis	17
3.1 Análisis de entidades.	19
3.2 Análisis de datos	26
3.3 Análisis funcional	29
3.4 Recolección de datos	31
3.5 Diagrama de flujo de datos	31
3.6 Técnicas de validación	34
3.7 Limitaciones	35
III.4 Etapa de diseño	38
4.1 Diseño inicial	38
4.2 Estimación del espacio de la base de datos	40
4.3 Disminución de requerimientos de espacio.	44
4.4 Posibles problemas en el desempeño (performance)	44
4.5 Posibles soluciones a los problemas de desempeño	45
4.6 Afinación	48
III.5 Etapa de construcción	52
5.1 Los prototipos	52
5.2 Elementos para la validación de los módulos	52
III.6 La etapa de transición	53
III.7 La etapa de producción	54

CAPITULO IV

Resultados obtenidos del Banco de Horas con la metodología CASE.

IV.1 Etapa de estrategia	56
Plan de Actividades	57

IV.2 Etapa de análisis	59
Términos	59
Identificar a quién y cuándo entrevistar.	61
Otras fuentes de información	61
Estructura del proyecto	62
Determinación de estándares de documentación y convenciones en los nombres	62
Restricciones técnicas	67
Diagrama de entidad-relación para el banco de horas	67
Ejemplos de documentación del análisis de entidades y atributos.	74
Descomposición de funciones	82
Diagrama de flujo de datos. (Diagrama de contexto)	112
Diagrama de flujo de datos. (Diagrama de primer nivel)	114
Matriz de chequeo	115
Identificar funciones complejas	122
Identificar cómo se hará la transición	122
Factores críticos de éxito	123
Definición de los criterios de aceptación	124
Definir requerimientos de respaldo y recuperación.	125
IV.3 Etapa de diseño	126
Cálculo del espacio requerido	126
Revisar y diseñar las llaves primarias, foráneas e índices de las tablas.	128
IV.4 Etapa de construcción	129
Programa para crear la estructura de la base de datos	131
Programa para crear los índices	134
Programa para borrar las tablas.	135
Ejemplo de reporte. Reporte para obtener el catálogo de departamentos de la división.	136
Ejemplo de reporte. Reporte para emitir un listado general.	137
Ejemplo de menú. Menú principal	139
Ejemplo de programa para efectuar movimiento. Alta por nuevo ingreso.	140
 CAPITULO V.	
Conclusiones y recomendaciones.	161
Anexo A Manual de usuario	169
Glosario	181
Bibliografía	184

CAPITULO I.
INTRODUCCION.

I. INTRODUCCION.

La información no debe ser vista tan solo como un valioso activo de las corporaciones, sino también como el punto de partida para construir un sistema de información empresarial.

Es así que muchas organizaciones han encontrado que el entendimiento de cuál es la información corporativa requerida, es el prerequisite para construir sistemas de información bien integrados y de alta calidad.

Por esta razón, uno de los mayores retos para la tecnología de software, los ingenieros en computación y para los departamentos de sistemas, es cambiar de los métodos para desarrollar aplicaciones orientados a procedimientos, hacia aquellos orientados a datos. La ingeniería de la información está llegando a la metodología más popular orientada a datos, y que cubre todo el ciclo de vida de los sistemas.

La ingeniería de la información es frecuentemente descrita como una manera del desarrollo de información centralizada. Esta es una estrategia global que se enfoca a los datos estratégicos planeando y entendiendo las metas del negocio. Se basa en la premisa de que los sistemas de información corporativos pueden estar mejor integrados, si los datos compartidos son controlados centralmente y son parte del mismo modelo lógico de datos.

La ingeniería de información se avoca a buscar a través de todos los sistemas de una organización, para identificar como se utilizan y comparten datos. El descubrir qué información es empleada en una empresa es una tarea difícil, que consume mucho tiempo y dinero. Los ingenieros en computación, son auxiliados por poderosas herramientas y técnicas apropiadas que aseguran que la información corporativa sea correcta y consistentemente definida.

Es por ello que el objetivo de esta tesis es extender el conocimiento de la metodología CASE (Computer Aided Software Engineering), presentar sus fundamentos, analizar sus resultados y ver la forma en que el ingeniero en computación puede aprovecharla, ya que en la actualidad se está dando una revolución a nivel mundial en el campo de la ingeniería de programación.

Se ha dicho que las herramientas CASE, son para la ingeniería de la programación, lo que el CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) es a la manufactura, esta aseveración es válida, dado el símil que se presenta; Durante mucho tiempo la gente de procesamiento de datos se avocó a crear soluciones para las diferentes industrias: Manufactura, Finanzas, Diseño Gráfico, etcétera; Olvidando elaborar aplicaciones que le ayudasen en su trabajo. En los últimos años se han desarrollado diferentes herramientas, que se les ha denominado CASE, para facilitar, auxiliar y dar mayor calidad a los trabajos desarrollados por los ingenieros de software.

Por otra parte se decidió crear un sistema útil, para mostrar como seguir y aplicar la metodología CASE, se llegó a la conclusión que el desarrollo del sistema para el manejo del banco de horas de la División de Ingeniería Eléctrica (DIE), cumplía con las expectativas fijadas, ya que será de gran utilidad para la División, dado que ésta requería de un nuevo sistema que brindara mayores facilidades para el personal.

En el capítulo II, se hará una reseña histórica de cómo ha evolucionado la ingeniería de programación. Iniciando en la fase de la programación sin estructura alguna, pasando por la programación estructurada y los métodos tradicionales de programación; y sobre todo, marcando los inicios de la tecnología CASE, describiendo y explicando cómo y cuándo se han dado las dos generaciones de CASE.

Es en el capítulo III donde se explica amplia y profundamente en que consiste la metodología CASE para el desarrollo de sistemas de información. Se explican todas y cada una de las herramientas de las que se auxilia dicha metodología (diagramas de descomposición funcional, Diagramas de entidad relación, diagramas de flujo de datos, etcétera). Así como cada fase de las que consta el ciclo de vida de los sistemas de información, los derivados a obtener en cada una, así como la manera de resolver algunos problemas que comúnmente se encuentran.

En el capítulo IV se muestran los resultados de aplicar la metodología en la creación de un sistema útil de la vida real: "El Sistema para el manejo del Banco de Horas de la DIE"; se presentan los derivados obtenidos en cada fase.

En el capítulo V se hacen una serie de conclusiones, recomendaciones y expectativas que se tienen como resultado de este trabajo; con lo cual nos permitimos decir que: "CASE es la opción actual en la ingeniería de programación".

Finalmente se tiene un glosario de términos y la bibliografía.

CAPITULO II.
ANTECEDENTES Y DEFINICION DEL PROBLEMA.

II. ANTECEDENTES Y DEFINICION DEL PROBLEMA.

II.1 ANTECEDENTES.

En sus comienzos, la programación de los sistemas de información estuvo determinada por el trabajo individual de los programadores; no existía un grupo que se coordinara en el desarrollo de sistemas, ni una forma estructurada y sistemática de abordar el estudio de los mismos.

Como consecuencia lógica, cuando se entregaban los sistemas, éstos presentaban una cantidad de dificultades para el usuario final, y en gran medida se debía a que la programación del sistema no cumplía con los requerimientos de la empresa.

Además, dado que los analistas no seguían un método específico para estudiar los sistemas y los programadores no tenían metodologías para codificación, éstos trataban de demostrar su ingenio, ver quien podía hacer las rutinas más complejas, lo anterior aunado con la falta de documentación, hacia el mantenimiento a los sistemas difícil o incluso imposible; lo cual redundaba en altos costos de mantenimiento.

Después de esta fase crítica en la programación de sistemas, surgieron los métodos estructurados que solucionaron algunas de dichas deficiencias, pero aún subsisten una serie de problemas, como son, no tener fijos los resultados a alcanzar en las diversas etapas, y en especial en la de análisis de requerimientos, el analista obtenía lo que él consideraba necesario, según su entendimiento.

Posteriormente, cuando el analista había terminado con las entrevistas, este dedicaba todos sus esfuerzos a realizar el sistema "infalible", pero como en muchas ocasiones se alejaba por varios meses en diseñar y programar el sistema, cuando por fin lo tenía, las necesidades del negocio habían cambiado, y los programas resultaban ineficientes para la empresa.

Como a la mayoría de la gente, lo que siempre le apremia es hacer los programas, el análisis era mínimo, y la documentación nula, debido a que "lo importante es el programa, la documentación se hará después" y como consecuencia muchas veces nunca se hacía, o era muy ambigua; es por ello, que en la actualidad gran parte de los sistemas que se encuentran operando, no cuentan con la mínima documentación necesaria para darles mantenimiento. Es así que en este rubro las empresas tienen verdaderos ejércitos de programadores para mantener con vida los viejos sistemas (aproximadamente el 50 ó 60% del personal de sistemas se dedica a dar mantenimiento).

Debido a esta problemática, surgió CASE.

II.2 LA PRIMERA GENERACION C.A.S.E.

El anuncio público del nacimiento de CASE, fue en un artículo titulado "Computer Aided Software Engineering" (Ingeniería de Programación Asistida por Computadora), presentado el 17 de abril de 1984, en la exposición de procesamiento de datos federal en Washington D.C., Estados Unidos. El autor, Dr. John H. Manley, era entonces vicepresidente de ingeniería y tecnología de NASTEC Corporation.

Desde entonces, CASE ha llegado a ser un tema de primera línea en el mundo del software.

Los proveedores han desarrollado muchos "productos CASE" para ayudar en el desarrollo de software, a tal grado que se ha llegado a tener confusión en que es CASE y que no lo es.

Los primeros productos CASE, automatizaron las tareas para determinar los requerimientos de los sistemas, diseño de soluciones, y construcción del sistema. Las herramientas CASE hacen más fácil y rápido el uso de las convenciones y técnicas estructuradas, que son consideradas como parte esencial de la ingeniería de programación.

Con CASE, el desarrollador actualmente puede dibujar, actualizar, editar y almacenar varios diagramas utilizados en las fases iniciales del ciclo de vida del software. La computadora también ha llegado a ser una herramienta, actuando más como un ayudante electrónico, que puede verificar diagramas para tener precisión.

Estos productos CASE se conocen como herramientas UPPER-CASE.

En contraste, las herramientas LOWER-CASE, automatizan tareas de las fases posteriores a la fase de diseño, incluyendo generación de código, pruebas, depuraciones, compilaciones, prototipos y construcción. Es así como proveedores como Pansophic System, Inc; Sage, y Knowledge Ware introdujeron sus herramientas CASE generadoras de código.

En este punto, se llegó a la PRIMERA GENERACION de CASE. Las herramientas estaban disponibles para ayudar en casi todas las fases del ciclo de vida del software; sin embargo, los datos e información no necesariamente fluían automáticamente de una etapa a otra.

II.3 LA SEGUNDA GENERACION CASE.

La segunda generación de herramientas CASE comenzó a aparecer entre 1985 y 1986, con la introducción de productos que soportaban tareas en varias fases del ciclo de vida de los sistemas. Los proveedores usaron información de los requerimientos por fase para desarrollarlos y transformar los ya existentes de la etapa de diseño.

Las compañías proveedoras de CASE, usaron esta aproximación de "integración hacia adelante" para crear productos que soportaban la totalidad del ciclo de vida del software. Las herramientas cumplieron entonces con la integración de requerimientos, especificaciones, diseño, generación de código, pruebas, construcción y mantenimiento. Es así que el REPOSITORIO central de información surge como una ayuda para almacenar todos los datos relacionados con el desarrollo del proyecto.

CASE continuó para crear por sí mismo líneas tácticas y estratégicas. Los proveedores agregaron técnicas de programación en tiempo real y también se incluyó la generación de código directamente de los diagramas, además se descubrió que al utilizar sistemas expertos y bases de conocimiento, del campo de la inteligencia artificial, podía disminuirse el tiempo de desarrollo de los sistemas.

Con la inteligencia artificial, los diagramas pueden ser substituidos por modelos lógicos, de los cuales, los sistemas pueden generarse automáticamente. La inteligencia artificial es capaz de verificar grandes y complejos diagramas, y aún tratar de probar código existente. Este es otro camino que proporcionó CASE para que los usuarios fueran de los requerimientos básicos, a sistemas operacionales, con CASE definido en estos términos, PRE-CASE se refiere a herramientas que ven con la alta estrategia y administración del desarrollo de software. PRE-CASE representa herramientas de planeación para ser usadas en la identificación y asentamiento del alcance de los proyectos de software, antes de iniciar las actividades tradicionales de la ingeniería de programación.

Dentro de estas herramientas, se pueden listar aquellas para desarrollar el modelo corporativo de datos, estructuras de datos dentro de los sistemas de información, y así poder medir los sistemas de acuerdo a su impacto en las misiones corporativas.

Las funciones de administración soportadas por herramientas PRE-CASE, incluyen planeación del proyecto, calendarización, estimación y estudio de factibilidad.

II.4 LA INCORPORACION DE LA TECNOLOGIA DE BASES DE DATOS.

La siguiente adición importante a los productos CASE fue la tecnología de las bases de datos. Los sistemas de bases de datos proporcionaron el lugar donde las herramientas CASE pudieran almacenar datos anteriores sin ninguna organización e información nueva desarrollada por las herramientas UPPER-CASE y PRE-CASE.

La sección de la base de datos de los productos CASE es conocido como REPOSITORIO, muchos de los cuales son sistemas manejadores de bases de datos relacionales (RDBMS), que utilizan lenguajes de consulta como SQL. Sin embargo, los repositorios pueden usar otro tipo de estructura de base de datos, por ejemplo jerárquica.

No sólo los proveedores de bases de datos desarrollan productos CASE, si no que los de CASE, también desarrollan bases de datos. Los repositorios CASE, usados para aplicaciones y desarrollo de sistemas, fueron usualmente separados de las bases de datos corporativas, que almacenaban la información de producción.

Gradualmente entre 1985 y 1986, los proveedores iniciaron la reconstrucción de los repositorios de información incluidos con sus herramientas CASE, utilizando las bases de datos más populares en el momento. La integración de la información corporativa y el repositorio fue fácil, ya que se argumentó que el repositorio no era más que una base de datos para desarrolladores.

Así la tecnología de bases de datos ha llegado a ser parte de toda la estructura CASE.

Algunas herramientas CASE han soportado una aproximación orientada a datos para el desarrollo de las aplicaciones, usando diagramas de entidad-relación, normalizadores de información, y generadores de sentencias SQL. Con esta visión, la estructura de la base de datos de producción es parte integral del proceso para diseñar el software de aplicación.

El tener una base de datos de producción adecuadamente construida, acelera el desarrollo de aplicaciones. Se tienen disponibles herramientas CASE para ayudar con el análisis de la información corporativa y el desarrollo del modelo de datos empresarial.

Los proveedores de bases de datos tomaron ventaja del "traslape" existente con el mercado CASE, desarrollando y comercializando herramientas diseñadas para usar sus bases de datos. Estas compañías agregaron generadores de formas, escritores de reportes y diccionarios de datos para lenguajes de cuarta generación (4GLs), en paquetes de programación, permitiendo así a los desarrolladores incluir los prototipos en el ciclo de vida del software.

Con los prototipos, el usuario puede interactivamente desarrollar la aplicación empleando datos de producción. Cuando finalmente está satisfecho, él o ella pueden dar este prototipo al diseñador, para que lo tome como un modelo y construya una aplicación más eficiente. Las bases de datos también soportan la producción rápida e interactiva de prototipos (Rapid Interactive Production Prototyping RIPP). Las aplicaciones desarrolladas usando RIPP no son muy empleadas actualmente; ya que consiste en dejar una versión ineficiente en producción, y guardar una copia en el repositorio CASE, donde en forma interactiva se puede ir cambiando para mejorar su eficiencia y efectividad. La última versión liberada del prototipo se utiliza entonces en producción.

Con la tecnología de bases de datos establecida como parte del cuerpo de CASE, los científicos y desarrolladores necesitan herramientas para transferir el código existente a un ambiente CASE. El código estructurado puede ser almacenado para su reutilización como módulos de futuros programas, pero el código no estructurado no entra bien en un ambiente CASE.

La ingeniería reversiva es la respuesta, esta es una herramienta POST-CASE, la cual procesa el código no estructurado en módulos estructurados y los almacena en el repositorio, donde están disponibles para mantenimiento, edición o reutilización con las herramientas CASE.

La evolución de los productos CASE y sus relaciones se muestran en la figura 1.

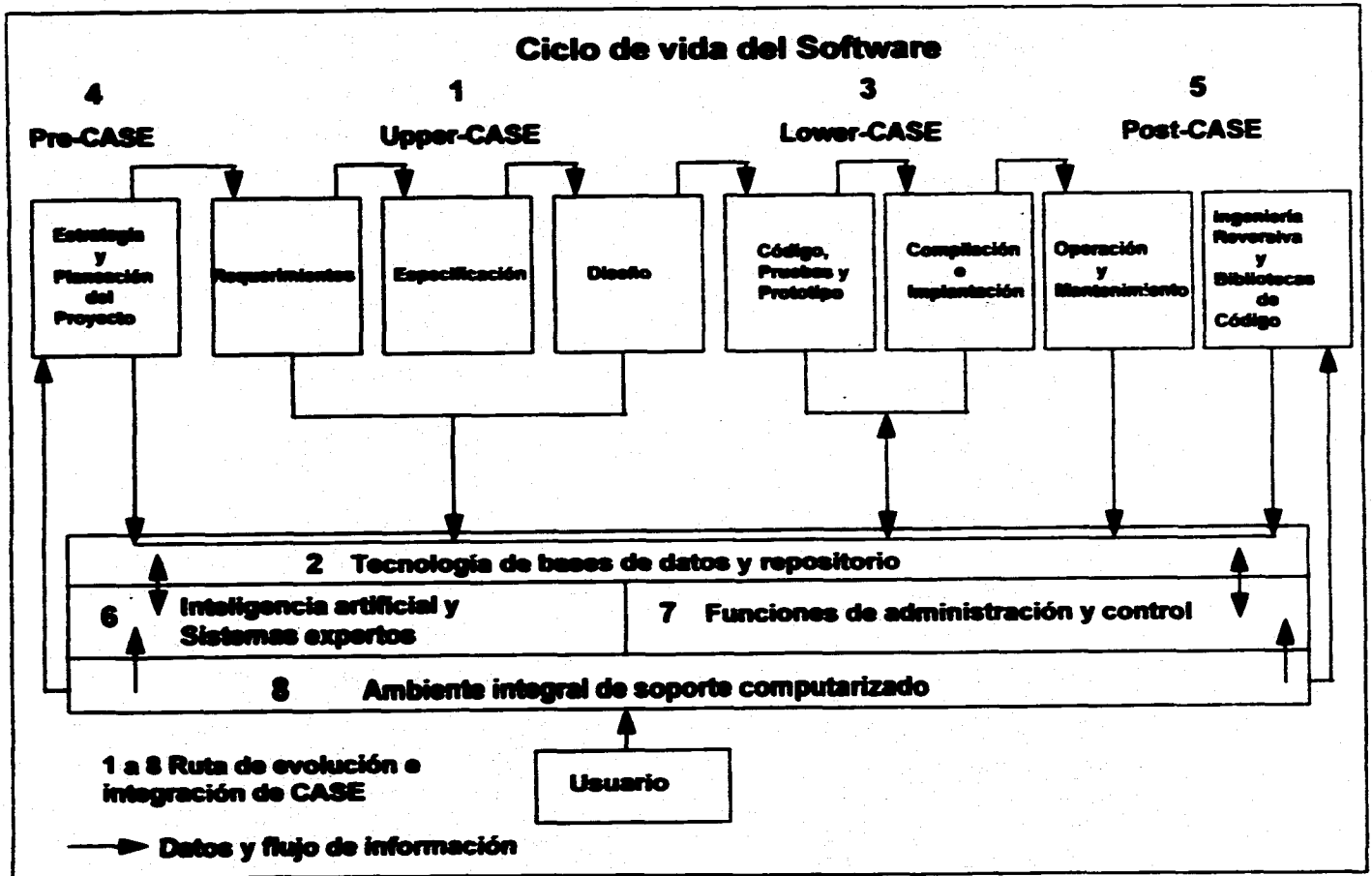


Figura 1. Ciclo de vida del Software.

II.5 EL BANCO DE HORAS

Debido al gran número de profesores que se manejan en la División de Ingeniería Eléctrica, es necesario contar con un sistema que facilite y optimise el banco de horas.

Actualmente existe uno que tiene algunas deficiencias y es por ello que se penso en hacer otro que pudiera manejar los movimientos de manera eficaz.

El objetivo del nuevo sistema es emitir los reportes que requiere la Secretaría Auxiliar, el jefe de la División y los jefes de Departamento para hacer un óptimo manejo del banco de horas, y a la vez facilitar la toma de decisiones.

Este sistema se ha basado en el manejo existente en la División de Ingeniería Eléctrica pero es posible que se pueda emplear en otras divisiones de la facultad, dado que el manejo y la problemática son muy similares.

Los métodos que se emplean para la elaboración del sistema son:

Hacer una interfase para vaciar la información que se requiere del actual sistema. Con esto se logra un aprovechamiento de datos ya existentes, dado que si se capturara nuevamente todo, se retrasaría la productividad y los beneficios del sistema, además se gastarían recursos en la captura de la información.

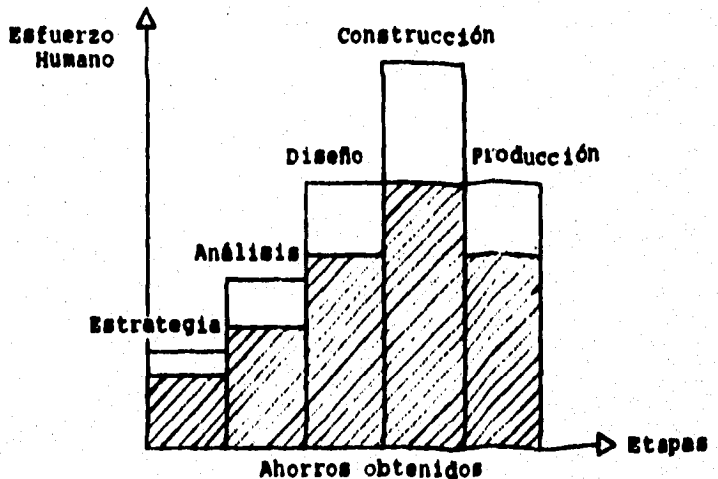
Para la elaboración de los reportes se empleará el lenguaje SQL (Structured Query Language- Lenguaje de Consulta Estructurado), el cual es un lenguaje de cuarta generación que permite escribir reportes no planeados en minutos.

CAPITULO III.

LA METODOLOGIA C.A.S.E.

III.1 LA METODOLOGIA CASE.

En la actualidad, cerca del 55% de los "errores" que se encuentran en los sistemas, se deben a malos entendidos en la etapa de análisis de requerimientos. La metodología CASE no sólo sirve para ahorrar tiempo y dinero, si no que también ha demostrado mejoras significativas en la calidad de los derivados y el grado de aceptación del usuario. Estos beneficios "intangibles" pueden ayudar al negocio a operar efectivamente.



Para que se de el éxito, de las empresas deberán adaptarse a nuevas circunstancias, esto es vitalmente importante para que cualquier método utilizado sea posible para adaptarse a los cambios que tengan en los negocios.

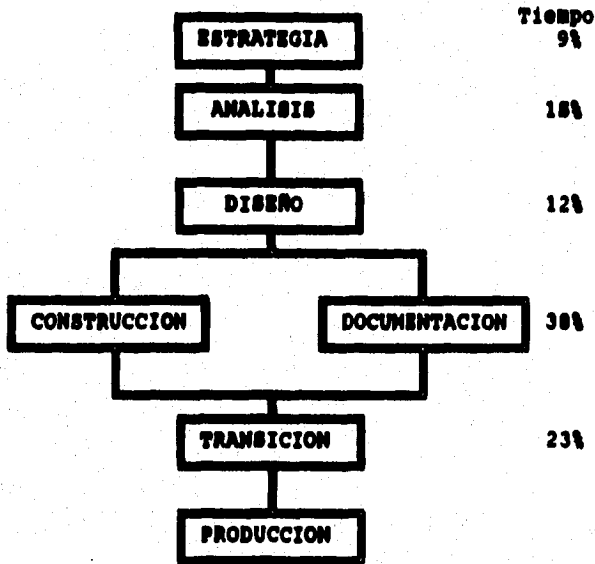
La metodología CASE realiza un análisis descendente, enfatiza en las personas antes que en documentos, se autochecha.

Es independiente de la técnica de programación.

Ha sido diseñado para permitir cambios.

Tiene una metodología para cada una de las fases del desarrollo de sistemas, para lograr sus objetivos, se basa en obtener ciertos derivados preestablecidos.

Es una metodología enfocada a cubrir todo el ciclo de desarrollo de un sistema, el cual se compone de las siguientes etapas:



La metodología se basa en la ejecución de tareas específicas para el desarrollo de cada etapa, con el objetivo de obtener derivados precisos al término de cada una, los cuales son requeridos para la ejecución de la siguiente. Los derivados por fase son:

ESTRATEGIA:

Objetivos del negocio, prioridades y restricciones

Modelo corporativo del negocio:

Descomposición funcional de primer nivel

Modelo global de entidades

Plan para análisis (y etapas siguientes).

ANALISIS:

Modelo detallado del negocio:

Descomposición funcional detallada.

Modelo detallado de entidades/atributos.

Volúmenes/frecuencias

Matriz de uso de entidades/atributos por función.

Especificación de restricciones.

Criterios de aceptación del usuario.

Planes para las etapas subsecuentes.

Indicar el posible grado de automatización.

DISEÑO:

Diseño detallado de la base de datos.

Detalle de los métodos para automatizar las funciones.

Documentación de decisiones de diseño.

Estimaciones de tamaño/espacio.

CONSTRUCCION:

Un sistema trabajando.

Programas documentados.

DOCUMENTACION:

Guía del usuario / notas de referencia.

Material de entrenamiento.

III.2 ETAPA DE ESTRATEGIA.

La etapa de estrategia, inicia el ciclo de vida del sistema. Esta fase sirve para entender y conocer la organización, hay que realizar una serie de entrevistas, normalmente de 6 a 8 para proyectos de tamaño mediano.

Las personas que hay que entrevistar en esta etapa deberán cubrir en conjunto todo el panorama de la empresa. De las visitas que se lleven a cabo, se produce un modelo, el cual consta de dos partes:

A) Un Diagrama de Entidad-Relación.

Dicho diagrama de manera pictórica, representa los elementos relevantes de la organización, acerca de los cuales se requiere conocer o mantener información, las relaciones existentes entre ellos y los atributos utilizados para describirlos.

B) Una Jerarquía de funciones.

La cual define lo que el negocio hace o necesita hacer, de una manera totalmente independiente de su estructura o de algún mecanismo en particular para hacerlo. Lo anterior es importante, dado que las estructuras de las organizaciones y los métodos para ejecutar las funciones cambian constantemente.

Es también en esta etapa donde se identifican y definen restricciones, prioridades, objetivos futuros y puntos de vista de la empresa o la aplicación, los cuales serán presentados en una sesión de retroalimentación al usuario. Durante esta sesión, se confirma la terminología, se corrigen los errores y omisiones, culminando así con un modelo satisfactorio del negocio.

Como se mencionó, durante la etapa de estrategia, la terminología del negocio deberá ser entendida totalmente por los analistas, para así asegurar la interpretación correcta durante todo el proyecto, los términos deberán ser documentados en un glosario.

Al finalizar la etapa de estrategia se ha construido un modelo lógico y conciso del organismo. De aquí, se puede establecer un plan detallado y definir las fases necesarias para automatizar algún requerimiento en particular del negocio.

El tiempo para estas fases puede ir de semanas o incluso años para proyectos muy grandes.

III.3 ETAPA DE ANALISIS.

La etapa de análisis es quizá la más importante de la metodología CASE, tiene un enfoque descendente (Top Down) dirigido por los derivados de la fase de estrategia, además cuenta con una verificación selectiva ascendente (Bottom-up).

Para efectuar el análisis detallado, éste es dirigido por funciones y verificado por datos.

Algunas otras características de la metodología CASE son las siguientes:

- * Interacción continua por parte del usuario final.

Este aspecto es muy importante, ya que si se involucra al usuario del proyecto, al término del mismo, él sentirá que es suyo ya que ha estado colaborando en su construcción, de lo contrario sentirá que es una imposición que se le da.

- * Independiente de la tecnología de programación.

Es conveniente aclarar, que al seguir la metodología CASE, no obliga a utilizar herramientas relacionales o de cuarta generación; aunque bien es cierto que ayudan bastante, pero se podría hacer los programas con lenguajes y herramientas de tercera generación.

- * Metódico, planeado y sensible al tiempo.

Se dice que es metódico y planeado, ya que se sabe que se obtendrá como resultado a cada momento. Es sensible al tiempo, ya que se establecen tiempos y secuencia para obtener cada derivado.

- * Soportado por un diccionario.

Se tiene un diccionario de datos que puede ser compartido por todas las personas involucradas en el proyecto, asegurando con ello la unicidad de la información que se maneja.

- * Uso efectivo del tiempo de los usuarios y de los analistas.

Porque al conocer lo que deben obtener, los esfuerzos realizados se enfocan a conseguirlo, no se tienen desviaciones.

Cuando se llega a la etapa de construcción, el método brinda gran ayuda ya que se presenta la siguiente evolución inmediata:

Se convierte en

ANALISIS	CONSTRUCCION
Entidades	Tablas, Archivos
Relaciones	Tablas binarias, Columnas de llaves foráneas
Atributos	Columnas, campos
Funciones	Transacciones, Programas.
Dominios	Reglas y programas de validación.
Eventos	Transacción recibida.

La metodología CASE basa gran parte de su éxito en tener derivados específicos, como producto de cada fase, así bien, los que se marcan para la etapa de análisis son los siguientes:

OBLIGATORIOS

- * Modelo de entidades validado y aceptado por el usuario.
- * Jerarquía de funciones validada y aceptada por el usuario, incluyendo frecuencias.
- * Matriz que relaciona funciones contra entidades.
- * Lógica de funciones elementales cuando se requiere, incluyendo validaciones y manejo de errores.
- * Especificación de volúmenes estimados, a un buen nivel de detalle.
- * Objetivos, prioridades y restricciones.
- * Notas de entrevistas.
- * Definición del alcance y límites del sistema, aceptada por el usuario.
- * Matriz de relación tarea vs. función, incluyendo la frecuencia de uso.
- * Definición de las reglas del negocio.
- * Criterios de aceptación.

OTROS DERIVADOS ADICIONALES Y OPCIONALES

- * Notas sobre la transición de información (y cambios a procedimientos existentes).
- * Descripción a interfaces externas, en relación a datos, formato, etc.
- * Notas sobre retención de datos.
- * Lista de requerimientos de auditoría.
- * Matriz de relación organización vs. funciones elementales.
- * Diseño informal de reportes, formas, pantallas para la etapa de construcción

Para lograr dichos derivados se auxilia de técnicas específicas que a continuación se detallan.

III.3.1 ANALISIS DE ENTIDADES.

El modelo de Entidad-Relación es una técnica, para definir la información que una empresa necesita, a la vez provee un firme fundamento para poseer una alta calidad en los sistemas.

Su objetivo es proporcionar un modelo de la información que necesita la organización, la cual actuará como marco para el desarrollo de nuevos y mejores sistemas.

El modelo es independiente de cualquier forma de almacenamiento de datos o método de acceso, esto es importante ya que puede coexistir con los sistemas actuales.

En la figura 2, se muestra que al lograr el modelo de entidad-relación, la implementación se puede hacer bajo cualquier esquema de almacenamiento (base de datos jerárquica, reticular o relacional), ya que se han identificado las entidades sobre las cuales habrá de mantenerse información.

Diagrama de Entidad Relación

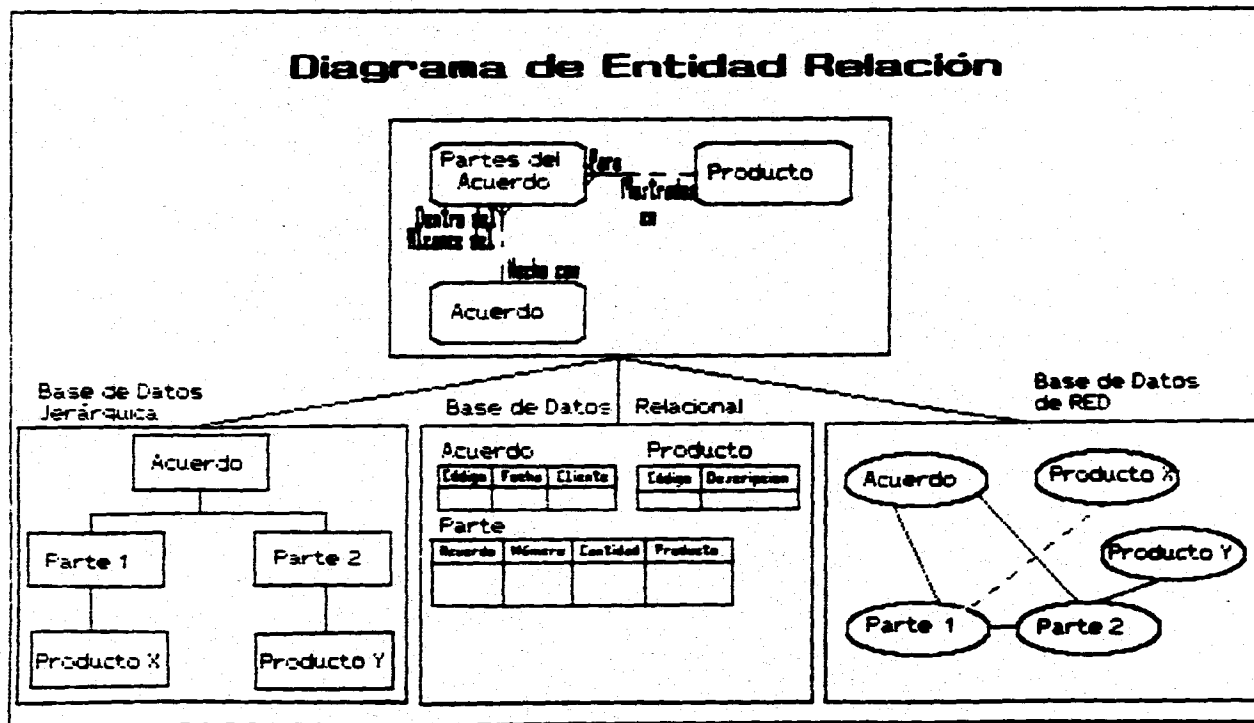
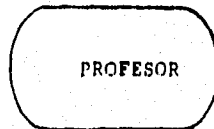


Figura 2. Implementación del diagrama de Entidad-Relación.

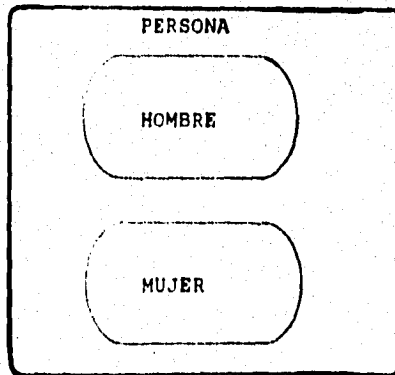
El análisis de entidades, por otro lado sirve para definir y entender las cosas de importancia sobre las que el negocio necesita conocer o mantener información, y las relaciones existentes entre ellas.

Para hacer el análisis de entidades hay que seguir las convenciones que a continuación se marcan.

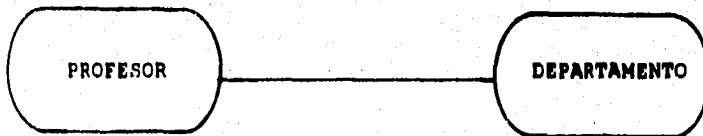
La unidad básica para el modelado es la entidad, la cual se representa esquemáticamente por rectángulos con esquinas redondeadas (llamados softboxes) y con el nombre de la entidad en singular dentro del mismo, en letras mayúsculas. Los "softboxes" pueden ser de cualquier tamaño para poder incluir nombres claros (no usar abreviaciones).



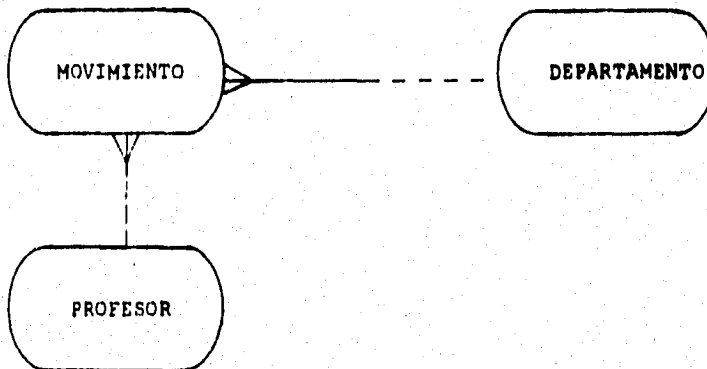
Una entidad puede tener subtipos o subentidades, los cuales a su vez también pueden tener subtipos. Los subtipos de una entidad son mutuamente excluyentes, comparten algunos de los atributos y las relaciones del supertipo, a la vez pueden tener atributos y relaciones propias y pueden tener relaciones entre sí.



Las relaciones muchos a muchos (M:M) son muy comunes durante los inicios de la etapa de estrategia y/o análisis, pero al final de estas fases, todas deberán estar resueltas, a menos que simplemente representen dos maneras de listar la información



Una relación de muchos a muchos (M:M) entre dos entidades (o con una misma) se puede modelar con las mismas entidades, cada una con una relación de uno a muchos (1:M) con una ENTIDAD INTERSECCION.



Los atributos de una entidad pueden ser listados dentro del rectángulo que la representa, con la siguiente convención:

- * indica llave primaria.
- indica obligatorio.
- indica opcional.

PROFESOR

- # RFC
- * Nombre
- * Nacionalidad
- * Dirección
- o Código postal
- o Teléfono
- * Sexo
- o Estado Civil

En un diagrama Entidad-Relación no necesariamente se muestran los atributos, pero el poner uno o dos durante el período inicial es altamente benéfico.

REGLAS DE LECTURA.

" CADA..... (Entidad)

DEBE ser o estar

o

(relación)

PUEDE ser o estar

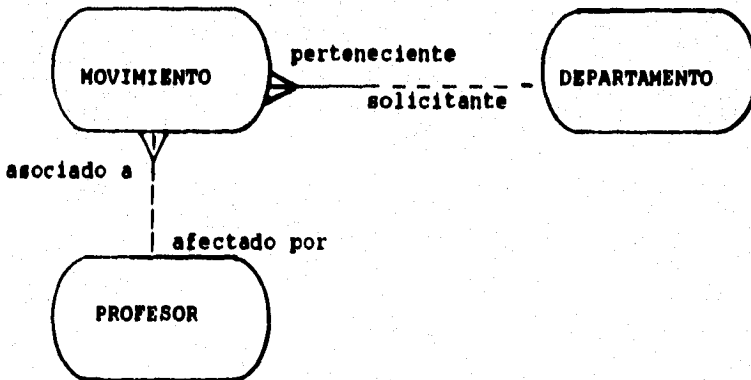
UNO o MUCHOS

o

(Entidad)

UNO Y SOLO UNO

Ejemplo de sintaxis de lectura:



cada PROFESOR puede estar AFECTADO por uno o muchos MOVIMIENTOS, cada MOVIMIENTO debe ser ASOCIADO a uno y sólo un PROFESOR, además, cada MOVIMIENTO debe ser PERTENECIENTE a uno y sólo un DEPARTAMENTO cada DEPARTAMENTO puede ser solicitante de uno o muchos MOVIMIENTOS

III.3.2 ANALISIS DE DATOS.

El objetivo de realizar un análisis de datos es revisar el modelo de entidades a la luz de datos conocidos, que puedan ser estudiados para normalizar el modelo y documentar formalmente los hallazgos.

El efectuar una normalización, permitirá dar la flexibilidad requerida para el sistema, garantizando que los datos estarán en un solo lugar, con los de su misma especie; si la información está una sola vez, se elimina la redundancia y se incrementa la consistencia.

Existen una serie de reglas de normalización, conocidas como formas normales; a continuación se mencionan las básicas.

Cero Forma Normal (0FN).

Toda entidad debe tener un conjunto de atributos y/o relaciones que identifiquen en forma única a cada una de sus ocurrencias.

Primera Forma Normal (1FN).

Cada atributo debe tener un solo valor, para cada ocurrencia de su entidad, en cualquier punto del tiempo.

Para pasar de 0FN a 1FN:

Quitar cualquier atributo multivalor y crear una nueva entidad con éstos.

Copiar el identificador de la entidad original a la nueva.

Segunda Forma Normal (2FN).

Cada atributo debe depender, para su valor, del identificador completo de la entidad.

Para pasar de 1FN a 2FN:

Quitar cualquier atributo parcialmente dependiente y crear una nueva entidad con ellos.

Copiar la parte del identificador de la entidad original de la que depende él o los atributos, como identificador de la nueva entidad.

Tercera Forma Normal (3FN).

Cada atributo solo debe depender, para su valor, del identificador único de su entidad.

Para pasar de 2FN a 3FN:

Quitar los atributos no dependientes y ponerlos en una nueva entidad.

Determinar el identificador para esta nueva entidad.

En la etapa de construcción, las formas normales, pueden ser interpretadas de la siguiente manera:

1NF

Existe una columna o conjunto de ellas, tales que dado un cierto valor de éstas, identifican en forma única a un sólo renglón. A estas columnas se les denomina la llave primaria.

1NF

La relación entre la llave primaria de una tabla y cada una de las otras columnas debe ser uno a uno (en esta dirección).

2NF

La relación entre cualquier parte de la llave primaria de una tabla y cada una de las otras columnas no debe ser uno a uno (en esta dirección). Generalmente aplica en tablas con llaves compuestas.

3NF

La relación entre cualesquiera dos columnas que no sean llave primaria de la tabla, exceptuando las columnas no duplicadas, no debe ser uno a uno (en esta dirección).

Ejemplo:

1NF	
#	EMP_NUM
	EMP_NOMBRE
	DEP_NUM
	DEP_NOMBRE
	GTE_NUM
	GTE_NOMBRE
	PRO_NUM
	PRO_NOMBRE
	FECHA
	TARIFA

1NF

EMPLEADO
#EMP_NUM
EMP_NOMBRE
DEP_NUM
DEP_NOMBRE
GTE_NUM
GTE_NOMBRE

ASIGNACION
#EMP_NUM
#PRO_NUM
PRO_NOMBRE
FECHA
TARIFA

2NF

EMPLEADO
#EMP_NUM
EMP_NOMBRE
DEP_NUM
DEP_NOMBRE
GTE_NUM
GTE_NOMBRE

ASIGNACION
#EMP_NUM
#PRO_NUM
FECHA
TARIFA

PROYECTOS
#PRO_NUM
PRO_NOM

3NF

EMPLEADO
#EMP_NUM
EMP_NOMBRE
DEP_NUM
GTE_NUM

ASIGNACION
#EMP_NUM
#PRO_NUM
FECHA
TARIFA

PROYECTOS
#PRO_NUM
PRO_NOM

DEPARTAMENTOS
#DEP_NUM
DEP_NOMBRE

Para el sistema del banco de horas se llegó a la tercera forma normal, ya que en esta forma normal, se logra que los requerimientos de almacenamiento se reduzcan substancialmente; cualquier dato está almacenado una sola vez, por lo cual las actualizaciones se simplifican y los datos actualizados están disponibles inmediatamente a otros usuarios. No existe la posibilidad de perder datos parciales. Los postulados de SQL para la manipulación de datos son más sencillos en la mayoría de los casos. Los cambios a la base de datos se facilitan substancialmente.

III.3.3 ANALISIS FUNCIONAL.

El propósito del análisis funcional es identificar lo que se hace actualmente en el negocio y lo que necesita hacerse en el futuro, planeando aproximadamente de cuatro a cinco años.

El análisis funcional se hace para poder:

- * Definir y entender lo que el negocio hace.
- * Verificar el análisis de entidades.
- * Dirigir el análisis detallado.
- * Seleccionar las áreas de automatización.
- * Definir el alcance del sistema.

Dicho análisis se hace llevando a cabo entrevistas de recolección de datos para completar el detalle del modelo obtenido en la etapa de estrategia, a la vez se van añadiendo o actualizando funciones para completar la jerarquía hasta llegar a funciones elementales. Hay que hacer cuantas entrevistas se vayan requiriendo.

Un buen método para realizar el análisis funcional es el siguiente:

- * Escribir una oración que describa la totalidad del negocio.
- * Descomponer dicha oración en las 7 u 8 funciones mayores.
- * Desglosar subsecuentemente dichas funciones hasta llegar a funciones elementales, cubriendo todo el alcance.
- * Las oraciones que se obtengan deben ser estructuradas (iniciar con un verbo en infinitivo) describiendo lo que se hace en el negocio, refiriéndose sólo a entidades, atributos, relaciones o sus sinónimos.
- * Convertir todos los mecanismos a funciones (identificar que se hace, más no cómo se hace).
- * Si una misma función ocurre varias veces, registrarlas.
- * Si es necesario, añadir condiciones (ej. la función se ejecuta si ...).
- * Documentar frecuencia de funciones.
- * NUNCA mencionar personas, puestos, máquinas, etc.

Dentro del análisis funcional se tienen que seguir algunas convenciones como son:

- * Las funciones se describen dentro de cajas redondeadas.
- * Numeración estrictamente jerárquica.
- * Las funciones exclusivas se denotan con un arco.
- * Para indicar la descomposición de una función se denota por un asterisco.
- * Las funciones opcionales se denotan con líneas punteadas (incluyendo la condición en la que aplica)
- * Las funciones comunes se referencian en forma cruzada.

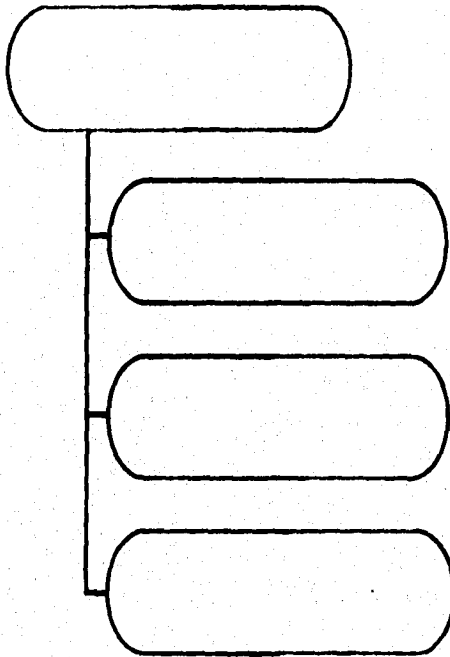


Figura 3. Esquema de descomposición funcional.

III.3.4 RECOLECCION DE DATOS

La recolección de datos es una etapa muy importante dentro del análisis de cualquier sistema, para iniciar se deben revisar y entender los derivados de la etapa de estrategia, detallar el plan a seguir en la etapa de análisis para lo cual hay que identificar las personas que estarán involucradas y presentarse con ellas, determinando las áreas que requieren ser investigadas y que entrevistas realizar. También se debe reunir documentos, reportes, etcétera, que serán usados en la recolección de datos.

El objetivo de llevar a cabo las entrevistas, es entender el área de interés mejor que al usuario final. Para lograr el mayor provecho de los visitas se debe identificar a los ejecutores de las funciones, arreglándolos de tal forma que relacione funciones vs. ejecutor, ya que se tenga dicho arreglo, se puede decidir a quien entrevistar y que preguntarle, reservando a la gente "inaccesible" para verificaciones. Es muy recomendable hacer un calendario/horario de visitas, establecer algunas preguntas, para así cubrir las funciones que se deseen.

Los documentos recolectados servirán para hacer una verificación ascendente (bottom-up) de los modelos de datos y funciones.

También hay que obtener información de volúmenes y frecuencias, concentrándose en aquellas funciones y entidades grandes o importantes ya que son las que tienen los mayores requerimientos de almacenamiento y proceso. Hacer estimaciones y cálculos basándose en sistemas existentes, ya sea en computadora o en procedimientos manuales.

De las entidades hay que obtener el volumen promedio, inicial, máximo y tasa de crecimiento. De los atributos su formato, su longitud promedio, máximo y su porcentaje de opcionalidad.

Esto será de gran utilidad para predecir el tamaño de la base de datos y el desempeño (performance) del sistema.

Para la etapa de transición se debe investigar qué y cuántos datos están disponibles para la conversión, que tanto debe ser llevado al nuevo sistema y en qué forma se hará.

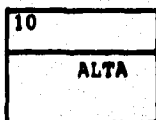
III.3.5 DIAGRAMA DE FLUJO DE DATOS.

Una herramienta de gran utilidad para hacer el análisis de sistemas es el Diagrama de Flujo de Datos (DFD); El cual es un modelo de la forma en que fluye la información dentro y fuera del negocio.

Los diagramas de alto nivel muestran el alcance del sistema y las interfaces entre subsistemas; mientras que los diagramas de bajo nivel sirven para cruzar el modelo de entidades y la descomposición funcional, a la vez que muestran las dependencias funcionales.

Al elaborar los diagramas de flujo de datos se deben tomar en cuenta las siguientes convenciones:

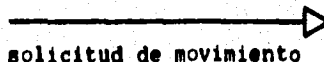
* Los procesos ejecutados por el negocio se representen en rectángulos con la descripción del proceso que ejecute, una identificación de referencia y el nombre del proceso.



* Las entidades externas que interactúan con el sistema se representen como óvalos, con el nombre de la entidad en el centro.



* Todos los flujos de datos dentro y fuera del negocio se representan con flechas, sobre las cuales va el nombre del flujo de datos.



* Los datos almacenados y utilizados por el negocio, se representen como sigue.



El uso de las palabras correctas es la clave de esta técnica, y es lo que hace útiles a los diagramas de flujo para el analista y el usuario.

Un método que se puede seguir al elaborar los diagramas de flujo de datos es el siguiente:

- * Definir los límites del sistema.
- * Encontrar las entradas y salidas importantes.
- * Identificar los procesos, acciones o funciones.
- * Utilizar las flechas para indicar datos en movimiento.
- * Emplear las celdas de almacenamiento para los datos en reposo.
- * Etiquetar cada proceso, cada celda de almacenamiento y flujo de datos.
- * Revisar con el usuario una y otra vez.
- * Descomponer sucesivamente los procesos hasta donde sea necesario.

Las conexiones válidas en un diagrama de flujo de datos son las siguientes:

	Externo	Proceso	Almacenamiento
Externo	NO	SI	NO
Proceso	SI	SI	SI
Almacenamiento	NO	SI	NO

III.3.6. TECNICAS DE VALIDACION

El objetivo del chequeo cruzado es asegurar la calidad del modelo de negocio. También identifica inconsistencias u omisiones en el modelo.

La documentación formal de los requerimientos del sistema está incompleta hasta que todas las funciones elementales hayan sido definidas completamente y sin ambigüedades. Esto puede lograrse mediante:

- * La descripción de funciones.
- * La lógica de funciones.

La descripción de funciones son oraciones en formato libre, que deben de ser usadas cuando los requerimientos de información pueden ser satisfechos plenamente mediante el uso de las funciones básicas de manipulación de entidades (create, delete, update, query).

Por su parte la lógica de funciones es utilizada cuando la descripción de la función no es clara (ejemplo: cuando existe más de una ruta a través del modelo de entidades), o bien cuando las reglas para conectar relaciones opcionales no son obvias, también se usan cuando no es evidente que las ocurrencias de entidades serán afectadas por la función. También es muy utilizada para definir que hacer en casos de error complejos.

- * Lógica detallada de funciones.

Con esta herramienta se busca expresar precisamente y sin ambigüedades los requerimientos de la función al referirse al modelo de entidades, describir la función en detalle, usando entidades, relaciones y añadiendo verbos que describan el tipo de proceso.

- * Triggers.

Un "trigger" es un suceso, o rango de eventos que provocan que una función sea ejecutada ya sea refiriéndose al valor del atributo, a una fecha o al término de otra función.

- * Acciones.

Tomar una copia de la parte relevante del diagrama de entidades, relaciones y resaltar las entidades involucradas en la función.

Especificar las reglas para calcular el valor que tomará cada atributo (excepto que haya valores por omisión en la especificación del atributo).

La secuencia en que las acciones sean presentadas ayudará a reducir el contexto en el que los atributos deberán ser considerados.

*** Algoritmos.**

Los algoritmos deben identificarse en la lógica de la función; para elaborarlos se debe tomar la descripción del usuario y expresarla en la terminología del negocio, para así comunicar conceptos de manera precisa y concreta.

III.3.7. LIMITACIONES.

Otra parte a considerar, como ya se ha mencionado anteriormente, es el asegurar que todas las limitaciones importantes, así como su impacto, sean conocidas por los usuarios finales, desarrolladores y los responsables de la toma de decisiones.

Las limitaciones son en gran medida los criterios de aceptación del usuario final.

Dentro de las limitaciones que pueden presentarse en el desarrollo de sistemas se identifican en los siguientes rubros:

*** ALCANCE**

Hay que definir los límites del sistema, de los subsistemas, definidos por dependencias entre funciones y entidades. Después que se hayan identificado, debe existir un mutuo acuerdo con el usuario final responsable, de qué está incluido y qué no.

*** COSTO**

¿Que presupuesto tiene el usuario para el desarrollo?, también hay que estimar el tiempo para desarrollo y transición. A la vez se deben proyectar las pérdidas provocadas por no tener el nuevo sistema en operación a tiempo.

*** GENTE**

Para poder hacer la planeación se debe identificar el personal que aportará la empresa, con lo cual se puede establecer un calendario de entrenamiento, se pueden ver las capacidades, aptitudes y la disponibilidad de los responsables de decisión.

*** RECURSOS**

Hardware / Software

Tecnología disponible

Comunicaciones

Medio ambiente (aire acondicionado, sala de terminales, etc.).

*** DESEMPEÑO (PERFORMANCE)**

Tiempos límites para la ejecución de procesos.

Tiempos de respuesta al usuario.

*** DISPONIBILIDAD**

Impacto por la no disponibilidad de subsistemas.

Mantenimiento.

Sistemas de respaldo.

*** SEGURIDAD**

- Fijar los derechos de acceso por:

+ Persona.

+ Puesto.

+ Localidad.

+ Terminal.

- Restricción en:

+ Entidades.

+ Ocurrencias.

+ Atributos.

+ Funciones.

*** ESTILO**

Formas.

Reportes.

Estándares o convenciones.

Correspondencia con otros sistemas.

Limitaciones del hardware.

*** TIEMPO**

Fechas absolutas.

Compromisos políticos.

Períodos de transición.

Horas pico.

Requerimientos de mantenimiento.

III.4 ETAPA DE DISEÑO

La etapa de diseño, recibe los derivados de la fase de estrategia.

También se toman decisiones iniciales sobre cómo las entidades se convertirán en tablas, se identifican llaves; para así producir un primer prototipo, validar el desempeño, facilidad de uso, calidad, etc.

III.4.1 DISEÑO INICIAL

Los objetivos de efectuar un diseño inicial son:

- * Producir un primer diseño de la base de datos, a partir del modelo de entidades.
- * Aplicar opciones disponibles en el proceso inicial de diseño.
- * Obtener una estimación del requerimiento de la base de datos inicial

El proceso a seguir es:

- a) Cada entidad se convierte en una tabla.
- b) Cada atributo se convierte en una columna en la tabla de la entidad a la que pertenece.
- c) El identificador único de cada entidad se convierte en llave primaria (esta puede ser una combinación de atributos y/o relaciones).
- d) Cada relación 1:M se convierte en la llave primaria de la entidad del lado 1 insertada como llave foránea en la entidad del lado M.
- e) Las subentidades se pueden representar de dos maneras:

1. Tablas separadas

- Ventajas

La lógica para la subentidad es más sencilla, ya que es más específica.

Hay menos renglones para cada tabla.

La opcionalidad de columnas se da siempre que lo sea para la subentidad misma.

- Desventajas

El procesamiento de la superentidad requiere de hacer una UNION.

Los programas no pueden ser genéricos, por lo que son más.

2. Una Sola tabla

- Ventajas

Todo en un solo lugar.

Fácil acceso a la superentidad.

Vistas para las subentidades.

- Desventajas

La lógica tiene que adecuarse a diferentes conjuntos de atributos y relaciones de las subentidades.

Potencial para problemas de bloqueo (locking).

Los atributos propios de las subentidades tienen que ser opcionales todos.

f) Los arcos de exclusividad.

Explícitamente se tendrá una columna para cada una de las relaciones, esto es, para cada una de las llaves foráneas.

Genéricamente se hace con una columna para cada llave foránea y una para indicar la relación a la que pertenece (se requiere que todas las llaves foráneas sean compatibles en formato).

Otro punto importante a considerar es el diseño de INDICES, para lo cual se toma en cuenta lo siguiente:

Las llaves primarias deben tener un índice sin duplicados las llaves foráneas, principalmente aquellas a ser usadas en operaciones 'JOIN', deberán tener índices.

Las columnas muy comúnmente usadas para búsqueda (ver las funciones), también lo tendrán.

g) Las VISTAS son otro objeto ampliamente usado, ya que permiten presentar tablas a los usuarios en cualquier forma, sin afectar la estructura de la base de datos.

Son una forma para restringir el acceso a datos específicos y proveen un buen grado de 'independencia de datos'. Sin embargo presentan algunas limitaciones; como son:

Las columnas y renglones de actualización y borrado (UPDATE y DELETE) sólo pueden ejecutarse sobre vistas que:

- * Estén basadas en una sola tabla
- * No contengan cláusulas 'GROUP BY'
- * No utilicen cláusulas 'DISTINCT'
- * No hagan referencia al 'ROWNUM'

La operación de inserción (INSERT) sólo se puede ejecutar sobre vistas que cumplan con las condiciones previas y no contengan columnas con expresiones.

III.4.2 ESTIMACION DEL ESPACIO DE LA BASE DE DATOS

* Almacenamiento usando Oracle

Dentro de la etapa de diseño también se debe calcular el espacio que ocupará la base de datos, para ello hay que tomar en cuenta lo siguiente:

Los datos se almacenan en un conjunto de bloques o páginas. El tamaño de los bloques depende del sistema operativo (en VMS es de 2048 bytes).

Cada página tiene un encabezado de 76 bytes (en la Versión 5), que contiene apuntadores a otras páginas y además información de control inaccesible al usuario.

Cada tabla requiere 3 páginas para información del RDBMS (2 en datos y 1 en índices).

Cada renglón requiere dos bytes para su longitud total y 2 bytes para el número secuencial de renglón (row sequence number).

Cada columna requiere 1 byte para la identificación de la columna y 1 byte para el número de bytes de datos en ésta (los valores NULL no son almacenados y no requieren estos dos bytes).

CALCULO DE ESPACIO REQUERIDO POR LA BASE DE DATOS

PROMEDIO DE BYTES POR RENGLON

$$BR = \text{SUM}_i ((1-N_i) * (C_i + K + I)) + L + Q$$

Donde:

BR: Promedio de bytes por renglón.

SUM_i: Sumatoria de todos los valores de *i* que va de 1 al número de columnas.

N_i: Fracción en que la *i*ésima columna es NULL.

C_i: Longitud promedio de la *i*ésima columna.

K: Indicador de longitud de columna (1 byte).

I: Indicador de identificación de columna (1 byte).

L: Indicador de longitud de renglón (2 bytes).

Q: Row sequence number (2 bytes).

PROMEDIO DE RENGLONES POR BLOQUE

$$RB = \text{trunc}(((1-F)*(S-H))/BR)$$

Donde:

RB: Renglones por bloque.

F: Porcentaje libre en cada bloque.

S: Tamaño del bloque (VMS 2048 bytes).

H: Tamaño del encabezado del bloque (76 bytes).

BR: Promedio de bytes por renglón.

NUMERO DE BLOQUES

$$B = R / RB$$

Donde:

B: Número de bloques.

R: Número de renglones en la tabla.

RB: Promedio de renglones por bloque.

BLOQUES PARA INDICES

$$BI = 1.1 * (R/(((S-H)*0.75)/KL))$$

Donde:

BI: bloques para índices.

R: Número de renglones en la tabla.

S: Tamaño del bloque (VMS 2048).

H: Tamaño del encabezado del bloque (76 bytes).

KL: Longitud estimada en la llave

= 10 para índices comprimidos

= 16 + número de columnas en la llave

+ suma de longitudes de columnas en la llave.

EL ARCHIVO DE IMAGEN ANTERIOR (BEFORE IMAGE FILE)

Cada operación INSERT, UPDATE o DELETE ocupa un bloque en el BEFORE IMAGE FILE, el cual tiene un tamaño del 25 al 30 % del tamaño de la base de datos.

DEFINICION DEL ESPACIO.

CREATE SPACE [DEFINITION] nombre

DATAPAGES (INITIAL n,

INCREMENT n,

MAXEXTENDS n,

PCTFREE n)

INDEXPAGES (INITIAL n,

INCERMENT n,

MAXEXTENDS n)

PARTITION nombre

Cuando se crea una tabla, se aloja un espacio para el INITIAL EXTENT y en éste se van almacenando los datos, cuando se llena se aloja espacio para un INCREMENT EXTENT y así sucesivamente hasta MAXEXTENTS.

Las páginas de datos se van usando secuencialmente hasta ocuparlas, dejando el espacio libre especificado por PCTFREE (se toma un 20% por omisión), el cual es empleado para actualizaciones. Cuando el espacio se llena, la página se encadena con otra y así sucesivamente.

Mediante el uso de SPACE se puede optimizar el uso del espacio de almacenamiento, minimizando la creación de EXTENTS especificando el parámetro INITIAL en base a las estimaciones de espacio requerido y evitando encadenamientos al emplear PCTFREE.

Para usar un SPACE éste debe estar creado antes de crear la(s) tabla(s) que lo usará(n):

```
CREATE TABLE nombre (columna1,  
                      columna2,
```

.

.

```
                      columna n)
```

```
SPACE nombre del space definition.
```

III.4.3 DISMINUCION DE REQUERIMIENTOS DE ESPACIO

Se pueden reducir los requerimientos de espacio si:

- * Se evita codificar llaves primarias largas que son usadas como llaves foráneas, especialmente en relaciones 1:M.
- * Codificar valores de dominios comunes.
- * Almacenar periodos de tiempo como fecha inicial y número de días en vez de fecha final.
- * Usar el valor nulo con algún significado y preferentemente para el valor de mayor ocurrencia.
- * Quitar aquellas tablas e índices que no se requieren en línea.
- * En tablas de información histórica, considerar tener en línea el periodo más cercano mínimo posible.

III.4.4 POSIBLES PROBLEMAS EN EL DESEMPEÑO (PERFORMANCE)

Se pueden presentar problemas de rendimiento en las siguientes circunstancias:

- * JOIN entre muchas tablas.
- * SELECT en tablas grandes, no especificando el valor completo de la columna.
- * CONNECT BY para recorrido de jerarquías, especialmente costosos en tiempo, inaceptable en un subquery.
- * SORT, cuando se tienen aplicaciones interactivas.
- * LOCKS cuando varias funciones actualizan una misma tabla o renglón concurrentemente.
- * Actualizaciones masivas sobre columnas indexadas.
- * Funciones de acumulación.
- * Fragmentación de datos, a nivel de registro y a nivel de tabla.
- * Recursos del sistema como memoria, canales de entrada/salida, paginación, swapping, colas de impresión, buffers, etc.

III.4.5 POSIBLES SOLUCIONES A LOS PROBLEMAS DE DESEMPEÑO.

* DESNORMALIZACION.

Realizar JOIN's duplicando columnas del lado 1 de la relación en el lado M. Esto siempre implica una sobre carga de proceso para el mantenimiento de los datos duplicados.

Crear CLUSTERS para tablas involucradas en JOINS. Esto implica recalcular los requerimientos de espacio y además un efecto negativo en el procesamiento de operaciones no JOIN.

Asegurar que cualquier dato de búsqueda sea una columna por si mismo y no esté contenido dentro de otro (ejemplo. ciudad en dirección).

Almacenar valores acumulados en alguna tabla del lado 1 de una relación cuando el detalle esté del lado M.

Separar tablas que representen a superentidades, creando una tabla para cada subentidad.

Juntar tablas que representen subentidades en una sola tabla como superentidad.

* INDICES.

Los índices pueden ser empleados para mejorar desempeño y para mantener integridad.

Crear un índice no único sobre la llave primaria de todas las tablas.

Crear un índice no único sobre las llaves foráneas (columnas de JOIN).

Crear índices sobre columnas de búsqueda frecuente.

Un SORT puede ser reemplazado en ocasiones por un índice.

Nunca crear dos índices sobre el mismo conjunto de columnas.

Evitar índices que se traslapen y nunca incluir la llave primaria en ningún otro índice.

Recordar que los índices no son empleados si alguna de las columnas con el índice está contenida en una función.

El costo de un índice no comprimido es alto pero pueda valer la pena si alguna consulta importante puede ser resuelta usándolo.

*** BLOQUEO (LOCKING).**

El mecanismo utilizado para evitar interferencia entre transacciones. La protección en contra de actualizaciones concurrentes sobre los mismos datos.

Los candados DML protegen la consistencia de los datos.

Los candados DDL protegen la integridad de definición de estructuras de datos.

Los candados pueden ser COMPARTIDOS (shared) o EXCLUSIVOS (exclusive).

Pueden existir cualquier número de candados compartidos sobre un objeto al mismo tiempo.

Si existe un candado compartido sobre un objeto, entonces no se puede poner ningún otro candado sobre el mismo objeto, si no hasta que el primero sea liberado.

Si se tiene un candado exclusivo sobre un objeto, entonces no se puede poner ningún otro tipo de candado, si no hasta que el primero sea liberado.

Los candados pueden existir de la siguiente forma:

OBJETO	CANDADO
Una tabla de usuario	C/E
Un renglón en tabla de usuario	E
La definición de una tabla	C/E
Un recurso "dummy" (uno por tabla)	E

y se aplican de la siguiente manera:

1. LOS CANDADOS DDL

OPERACION	DEFINICION DICCIONARIO
CREATE	E
DROP	E
ALTER	E
GRANT	E
SELECT	C
SELECT FOR UPDATE	C
INSERT	C
UPDATE	C
DELETE	C
LOCK TABLE IN EXCLUSIVE	C
LOCK TABLE IN SHARE	C
LOCK TABLE IN SHARE UPDATE	C

2. LOS CANDADOS DML

OPERACION	TABLA	REGLON	DUMMY
SELECT FOR UPDATE	C	E	
INSERT (modo E default)	E		
UPDATE (modo E default)	E		
DELETE (modo E default)	E		
INSERT (modo C/A)	C	E	E
UPDATE (modo C/A)	C	E	E
DELETE (modo C/A)	C	E	E
LOCK TABLE IN EXCLUSIVE	E		
LOCK TABLE IN SHARE	C		E
LOCK TABLE IN SHARE UPDATE	C		

IMPORTANTE

Cualquier operación DDL impide otra operación sobre la tabla.

Los SELECTs nunca esperan a una operación DML.

Cualquier operación DML por omisión pondrá un candado exclusivo sobre toda la tabla.

Nunca codificar una operación DML por omisión en un trigger de SQL*FORMS, siempre deberá ir precedida por un LOCK IN SHARE FOR UPDATE MODE (bloqueo compartido para modo de actualización).

III.4.6 AFINACION

1. EL OPTIMIZADOR DE CONSULTAS (QUERY OPTIMIZER)

Para cada consulta el optimizador elegirá la trayectoria de acceso, la cual estará basada en consideraciones por omisión sobre los datos (no es estadístico). Al momento de la elección, los índices le ofrecen una mejor elección. Por lo anterior hay que conocer los criterios que sigue, para crear índices apropiados y verificar su uso.

2. INDICES

Los índices son el único mecanismo no-serial que el RDBMS puede utilizar para:

SELECT (una sola tabla)

....WHERE column = literal

usará el índice si existe sobre la columna.

Si habiendo un índice concatenado se usa la 'primera' parte, el índice es usado.

Cláusulas AND con índices en ambos argumentos harán un MERGE de la información (ROWID) en los índices que cumplan las condiciones.

Si el MERGE no es conveniente (por las condiciones), se usará el índice de mejor calificación (ver tabla de decisión en JOIN) para evaluar su condición y la otra se evaluará sobre los registros seleccionados.

Si el MERGE no es conveniente y la calificación de ambos índices es igual, se elegirá el primero referido en la cláusula WHERE.

Si uno de los índices es único y el otro no, solo se usará el único.

El máximo número de índices a usar será 5.

Cláusulas OR con índices en ambos argumentos usarán los índices para seleccionar registros independientemente y harán una UNION de los resultados.

Los índices con cláusulas OR no serán utilizados si:

- * Sólo hay uno de los parámetros.
- * La proposición SQL contiene un CONNECT BY.
- * La cláusula SQL contiene un OUTER JOIN.
- * El optimizador decide que no ayudará.

Un índice no será utilizado si el valor de la columna es modificado de alguna forma antes de la comparación.

Los casos más comunes en los que habiendo un índice este no será usado son:

```
... WHERE columna OPERACION literal ...  
... WHERE función (columna) = ...  
... WHERE columna IS [NOT] NULL  
... WHERE columna != ...  
... WHERE columna LIKE '%.....'  
... WHERE columna = literal de otro tipo de dato  
  
DISTINCT  
  
GROUP BY
```

Proposiciones que seleccionan un alto porcentaje de los renglones de una tabla ejecutarán más rápido haciendo un recorrido completo de la tabla que usando un índice.

En el caso de los "JOIN", el RDBMS utilizará los índices disponibles para mejorar el desempeño de un JOIN siempre que sea posible.

Si no existen índices disponibles para el JOIN:

- * Cada una de las tablas se procesa por separado.
- * Se seleccionan los registros que procedan y se ordenan en una tabla temporal.
- * Las tablas ordenadas son intercaladas (MERGE) en base a la llave primaria; esto minimiza el número de comparaciones.

* El MERGE se hará tomando como tabla directiva a la que corresponda a la última cláusula FROM, y buscando la correspondencia de llaves con la otra tabla.

Si sólo un lado del JOIN está indexado; la tabla no indexada se convierte en la tabla directiva.

Si ambos lados del JOIN están indexados; se elige la tabla directiva de acuerdo al siguiente sistema de evaluación:

MEJOR

ROWID = constante

columna con índice único = constante

índice concatenado único entero = constante

índice concatenado entero = constante

columna con índice = constante

índice concatenado entero >= límite inferior

'primera' parte de índice concatenado = constante

columna con índice único BETWEEN 'x' and 'y'

o columna con índice único LIKE 'C%'

columna con índice único o constante

columna con índice o constante

SORT/MERGE (JOIN)

MAX ó MIN de columna con índice

ORDER BY índice entero

recorrido completo de la tabla.

PEOR

- Si hay un empate se elige la última tabla referida en la cláusula FROM.

En el caso del "ORDER BY", el RDBMS usará un índice para producir el resultado de la cláusula ORDER BY, sin necesidad de llevar a cabo un SORT si:

- No hay cláusula WHERE o en la cláusula WHERE no hay referencia a otro índice.

- La columna referida en la cláusula ORDER BY fue definida como NOT NULL.

3. Las aplicaciones hay que diseñar aplicaciones alrededor de funciones, no de tablas; involucrar a los usuarios.

Agrupar funciones en los programas de acuerdo a los requerimientos del negocio sobre funciones a ejecutar juntas, la facilidad para el usuario, la seguridad (quien puede tener acceso e qué función), la propia estructura de la organización, o bien a los requerimientos de proceso (batch, interactivo).

Se puede decidir la herramienta a utilizar conforme a los requerimientos de proceso (batch, interactivo), el volumen de transacciones, los tipos de transacciones (consulta, cerge de datos, actualización, reportes), la propia complejidad de proceso (lógica de procedimientos), el estilo de acceso requerido, la habilidad del usuario.

Diseñar esquemas de reportes y pantallas, especificar las convenciones o estándares e seguir, procurar minimizer el uso del teclado, aplicar las vistas del negocio de las entidades, escribir pseudo-código, especificar reglas de integridad como son:

Tipos de datos, dominios, opcionalidad / obligetoriedad de atributos, reglas de llaves foráneas, la integridad referencial, tablas de codificación.

III.5 ETAPA DE CONSTRUCCION

III.5.1 LOS PROTOTIPOS

Es una técnica iterativa que reemplaza la necesidad de documentar detalladamente la aplicación en la etapa de diseño, se debe aplicar de manera muy controlada y con una validación continua de parte del usuario final; el resultado debe ser un módulo documentado, probado y aceptado por el usuario .

III.5.2 ELEMENTOS PARA LA VALIDACION DE LOS MODULOS.

Algunos elementos que pueden ser útiles para la validación de los módulos son:

- * Funcionalidad para el usuario final.
- * Criterios de aceptación del usuario final.
- * Integridad del negocio.
- * Control de acceso/seguridad.
- * Procedimientos de recuperación.
- * Procedimientos de operación computadora-usuario.
- * Interfases a otros sistemas.
- * Pruebas de alto volumen y larga duración.
- * Ejemplos en la documentación del usuario final.

Estos elementos servirán para crear programas funcionales, satisfactorios y útiles para la empresa.

III.6 LA ETAPA DE TRANSICION

En lo que respecta al entrenamiento, se debe crear un esqueleto derivado del desarrollo, complementario con el usuario final (un grupo selecto).

Llevar a cabo sesiones conjuntas de entrenamiento, realizar el entrenamiento siguiendo la operación del sistema.

Planear la capacitación del resto de los usuarios, o bien puede ser que el grupo selecto entrene al resto de los usuarios.

Se debe asegurar que la conversión o transición a un nuevo sistema se pueda lograr con la mínima afectación al usuario final, con el mínimo costo para el negocio y manteniendo la integridad del mismo.

Investigar que debe hacerse para pasar de donde se está actualmente a donde se quiere estar.

Establecer la definición del sistema actual, a partir de la observación y gente, las formas, documentos y reportes existentes, así como de los procedimientos, o la descripción de funciones.

Documentar de donde vendrá la información en el nuevo sistema.

Crear un organigrama de la empresa, para así identificar a la gente que tiene más experiencia, habilidad, conoce los "trucos", "Hace" lo que nadie sabe, o tiene liderazgo en la opinión de los demás, para que puedan estar en el grupo selecto de entrenamiento.

En lo que respecta a la operación, se deberá corregir los datos en el sistema viejo, usar los sistemas de validación del nuevo sistema, especificar claramente qué viene de donde, probar todos los aspectos del sistema, procedimientos de recuperación, simular intentos de violar la seguridad, probar los procedimientos de respaldo, verificar la integridad de datos y verificar la operación correcta de usuarios y operadores.

III.7 LA ETAPA DE PRODUCCION.

En esta fase, habrá de hacerse la planeación del mantenimiento que se proporcionará, haciendo verificaciones selectivas y verificando la integridad de la información.

Monitoreo constante del desempeño del sistema, medir y ajustar para el crecimiento, para así garantizar la confiabilidad constante.

También habrá de establecerse un mecanismo para realizar modificaciones preventivas y correctivas; ¿cómo se originarán?, ¿hacia quién se dirigirán?, ¿cómo se harán las pruebas?, ¿cómo se implantarán estos cambios en el sistema.

Establecer procedimientos para mantener una educación periódica al personal involucrado en el sistema, y así de esta manera aprovecharán mejor el sistema.

Definir los perfiles de conocimiento para el personal que operará el sistema, como son: el administrador de la base de datos, el administrador de la red de telecomunicaciones, los operadores, los programadores del sistema, etc.

Hacer una calendarización de la distribución de cargas de trabajo (respaldos, pruebas, cargas masivas de datos, depuración de archivos), considerando los demás sistemas que corren en el equipo (nómina, facturación, inventarios, etc.), para que no se interfieran.

Crear procedimientos para realizar migraciones a otros equipos cuando sea necesario.

Llevar un estricto control de los programas fuentes para la evolución a nuevas versiones del sistema.

En general, fijar y organizar todos los procedimientos requeridos para mantener funcionando de manera óptima el sistema.

CAPITULO IV.

**RESULTADOS OBTENIDOS DEL BANCO DE HORAS
CON LA METODOLOGIA CASE.**

IV.1 ETAPA DE ESTRATEGIA.

En este capítulo se presentan los derivados que se obtuvieron de seguir la metodología CASE; primero se tiene un planteamiento general de las características del sistema actual de la Secretaría Auxiliar, posteriormente se tienen los derivados de la etapa de estrategia, comenzando con el plan de actividades que se muestra en las siguientes hojas.

Actualmente la Secretaría Auxiliar de la DIE, cuenta con un sistema que le ayuda en el manejo del banco de horas. Este sistema está elaborado en Dbase III con las características propias del lenguaje.

El sistema ha funcionado por aproximadamente 6 años en la secretaría, pero han cambiado algunos requerimientos en la Secretaría Auxiliar desde ese entonces, y aunado a que no se ha dado ningún tipo de mantenimiento y no se cuenta con ninguna documentación del sistema.

Dentro de un análisis inicial del sistema del banco de horas que se efectuó, se encontró que:

- Se carece de documentación.
- No se tienen los programas fuentes.
- No se sabe como dar mantenimiento, ni como hacer correcciones.
- Cuenta con una serie de reportes preestablecidos.
- Se tiene la capacidad de crear nuevos reportes, pero es muy elaborado y no se tiene información.
- El sistema tiene alguna información redundante.
- Las ayudas en línea son mínimas.
- No se cuenta con un "spool" para guardar la impresión y poder obtener varias copias de un reporte; para obtenerlas, hay que correr nuevamente el proceso.
- Tiene algunas validaciones.

Se cuenta con otro sistema que fue elaborado por la División de Ciencias Básicas, el cual únicamente sirve para imprimir las formas únicas, pero no lleva un control del banco de horas, ni hace todas las validaciones requeridas para efectuar un movimiento.

Estos aspectos fueron la pauta que impulso el desarrollo de un sistema integral para el manejo del banco de horas.

Plan de Actividades.

Número de Actividad	Actividad	Responsable
1000	ETAPA DE ESTRATEGIA	
1002	Armar Carpeta	VGC
1004	Elaborar plan detallado para todas las fases y diagrama de Gantt	VGC
1006	Obtener listados del sistema actual	MCP
2000	ETAPA DE ANALISIS	
2002	Confirmar y escribir términos	MCP
2004	Identificar a quién entrevistar	MCP, VGC
2006	Realizar entrevistas	MCP, VGC
2008	Identificar otras fuentes de información	VGC, MCP
2010	Decidir la estructura de control del proyecto	VGC
2012	Determinar los estandares para documentación y las convenciones en los nombres	VGC
2014	Definir restricciones técnicas	VGC, MCP
2016	Revisar diagrama entidad-relación	VGC
2018	Revisar jerarquía de funciones	MCP
2020	Elaborar diagrama de flujo de datos	MCP
2022	Realizar chequeo cruzado de entidades vs. Funciones	VGC, MCP
2024	Identificar aquellas funciones que por su complejidad requieren de especial atención y discutir las	MCP, VGC
2026	Identificar cómo se hará la transición	VGC, MCP
2028	Verificar volúmenes y frecuencias	MCP
2030	Revisión	AMN, VGC, MCP
2032	Asegurar que todos los atributos han sido identificados	MCP, VGC
2034	Revisar opcionalidad de atributos	MCP, VGC
2036	Modelo de entidades totalmente documentado	VGC
2038	Definir factores críticos de éxito	MCP, AMN, VGC
2040	Definir criterios de aceptación	MCP, AMN, VGC
2042	Definir requerimientos legales	MCP
2044	Identificar requisitos de seguridad del usuario y las pruebas que se harán para satisfacerlas	MCP, AMN, VGC
2046	Definir requerimientos de respaldo y recuperación	MCP, VGC
2048	Establecer requerimientos de almacenamiento	VGC
2050	Hacer predicción del tamaño de la base de datos y el rendimiento	VGC
2052	Revisión	AMN, MCP, VGC

3000 ETAPA DE DISEÑO

3002 Verificar diagrama Entidad-Relación	VGC,MCP
3004 Pasar entidades a tablas	VGC
3006 Someter a normalización las tablas	VGC
3008 Revisar si es necesario desnormalizar tablas	VGC,MCP
3010 Revisar y diseñar las PK's de cada tabla	VGC
3012 Revisar FK's entre tablas	VGC
3014 Diseñar índices necesarios	VGC
3016 Hacer esquemas de formas y reportes	MCP,VGC
3018 Establecer estándares de programación documentación de programas	VGC
3020 Revisión global de diseño	AMN,VGC,MCP

4000 ETAPA DE CONSTRUCCION

4002 Crear tablas e índices de la Base de datos	VGC
4004 Hacer programa para borrar tablas	MCP
4006 Crear programa para borrar índices	MCP
4008 Crear programa para borrar datos	MCP
4010 Elaborar forma prototipo	VGC
4012 Elaborar forma de mantenimiento prototipo	MCP
4014 Revisar prototipos con el usuario responsable	VGC,MCP AMN,MAN
4016 Elaborar formas de mantenimiento	VGC,MCP
4018 Elaborar formas de altas	VGC
4020 Elaborar formas de bajas	VGC
4022 Construir forma de licencias	VGC
4024 Hacer forma de estatus	VGC
4026 Elabora menú de formas	VGC
4028 Revisión de formas	AMN,MCP,VGC
4030 Definir reportes	MCP,VGC,AMN
4032 Elaborar reportes	MCP,VGC
4034 Carga de la base de datos	VGC,MCP
4036 Pruebas del sistema	VGC,MCP,AMN
4038 Elaborar manual de usuario	MCP
4040 Elaborar manual técnico	VGC
4042 Educación del personal	MCP,VGC

VGC: Victor González Castro
MCP: Maricela Castañeda Perdomo
AMN: Adolfo Millán Nájera

IV.2 ETAPA DE ANALISIS.

A continuación, se tienen los derivados obtenidos en la etapa de análisis, están numerados según el número de actividad que representa en base al plan de actividades, y según lo explicado en el capítulo anterior, donde se planteo que se debería obtener, qué significa y cómo interpretarlo.

Documento: ANAL2002

TERMINOS

Artículo 51: Contratación bajo situación especial, la cual es reflejada dentro del artículo 51 del estatuto del personal académico de la UNAM.

Asignatura: Cátedra impartida en la Universidad Nacional.

Banco de horas: Disponibilidad de dinero, valuado en horas, con que cuenta la división, para la contratación de profesores.

Clave de la asignatura: número que identifica una asignatura dentro de la UNAM.

Clave de categoría: Número único asignado, para identificar cada una de las categorías de profesores existentes en la UNAM.

Categoría: Clasificación del profesor, según su nivel en el tabulador.

Código programático: Identificador de la dependencia donde trabaja el profesor.

Colonia: Colonia donde se ubica el domicilio o bien la oficina del profesor.

CP: Código Postal.

Departamento: Entidad académico-administrativa a la cual están asignados los profesores.

Descripción de la Asignatura: Nombre de la asignatura.

DIE: División de Ingeniería Eléctrica.

Dirección: Calle y número de la vivienda del profesor.

Estado Civil: Estado civil que guarda un profesor.

Estatus: Estado administrativo en el que se encuentra un movimiento (ejemplo: pendiente de firma, en consejo, etc.).

Fecha de Alta: Fecha en la cual un movimiento de un profesor empezará a tener vigencia.

Fecha de terminación: Fecha en la que concluye la vigencia de un movimiento.

Grupo: Número asignado al grupo al cual el profesor dará clase.

Histórico de Movimientos: Relación que contiene los movimientos anteriores de los profesores.

Histórico de Profesores: Relación que contiene los datos de los profesores que han sido dados de baja.

Horas ejercidas: Horas de clase u asesoría de un profesor dentro de la UNAM.

Horario: Horas del día que el profesor tiene comprometidas con la dependencia.

Horas de teoría: Número de horas de clase teórica que imparte el profesor.

Horas de práctica: Número de horas de laboratorio impartidas por el profesor.

Jefe de departamento: Encargado administrativo de un departamento de la DIE.

Licencia: Permiso otorgado por la UNAM a un profesor, para separarse temporalmente de sus labores.

Límite de horas: Cantidad máxima de horas de las que puede disponer, un departamento, o la división

Movimiento: Cualquier alteración en la relación del Banco de horas.

Nacionalidad: Nacionalidad del profesor.

Nombre: Nombre completo del profesor.

Total de horas: Suma aritmética de las horas de teoría, más las horas de práctica de un profesor.

Profesor: Personal de la DIE, que desempeña una labor docente.

RFC: Registro Federal de Contribuyentes del Profesor.

IDENTIFICAR A QUIEN ENTREVISTAR.

Persona	Puesto
Miguel Amezcuita	Usuario Final
Adolfo Millán	Secretario de la DIE
Maricela Castañeda	Secretaria Auxiliar
Ricardo Garibay	Jefe de departamento
Maricruz Galindo	Secretaria

OTRAS FUENTES DE INFORMACION.

Manuales: Normatividad para llevar a cabo los movimientos de personal académico.

Contrato colectivo de trabajo AAPAUNAM.

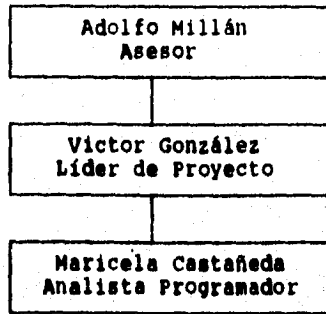
Ley orgánica de la UNAM.

Estatuto del personal académico de la UNAM.

Documentación de sistemas anteriores.

Departamento de personal académico.

Otras divisiones de la Facultad de Ingeniería.



FUNCIONES Y RESPONSABILIDADES.

ASESOR.

- Supervisar el desarrollo del proyecto.
- Sugerir mejoras y modificaciones a los derivados obtenidos.
- Encargado de supervisar la calidad del trabajo.

LIDER DE PROYECTO.

- Coordinar las actividades a realizar en cada fase del proyecto.
- Enseñar y explicar los temas necesarios de la metodología CASE a los involucrados.
- Realizar el análisis de algunos módulos.
- Checar la veracidad y completitud de los derivados.
- Programar algunos módulos.
- Redactar los resultados.
- Estructurar presentaciones.

ANALISTA PROGRAMADOR.

- Efectuar el análisis de los módulos.
- Validar que los procedimientos y funciones sean los adecuados
- Realizar programas.
- Conseguir la información necesaria.
- Asegurar la calidad a los derivados obtenidos.
- Participar en las presentaciones.
- Verificar la funcionalidad de los módulos.

DETERMINACION DE ESTANDARES DE DOCUMENTACION Y CONVENCIONES EN LOS NOMBRES.

La documentación que se genere deberá contener las iniciales correspondientes a la etapa a la que pertenecen:

ESTR - Estrategia
ANAL - Análisis
DISE - Diseño
CONS - Construcción
DOCU - Documentación
TRAN - Transición

Seguido del número de actividad a la que corresponde.

La documentación requerida en el análisis de entidades será la pedida en la forma anexa titulada "ANALISIS DE ENTIDADES".

Para los atributos llenar la forma "ANALISIS DE ATRIBUTOS".

En las relaciones se pondrá grado, opcionalidad, las entidades que enlaza, nombre de la relación de cada lado.

Las matrices de chequeo llenar la forma anexa "MATRIZ DE FUNCIONES VS. ENTIDADES".

Los nombres de la entidades serán en singular, de las tablas será en plural.

En ningún caso emplear abreviaciones.

Los nombres de las llaves primarias de las entidades intersección o tablas asociativas, llevarán las tres iniciales de todas y cada una de las tablas o entidades que relaciona.

Cuando se emplee más de una palabra en una descripción utilizar el guión bajo para separación.

Toda la documentación se escribirá en procesador de textos.

Toda la documentación se irá integrando en la sección correspondiente de la carpeta. Si se toma algún documento poner una hoja en la que se indique el número de documento, nombre y firma de la persona responsable.

Análisis de Entidades

Nombre (plural) Subtipo de

Sinónimos Vol. inicial

..... Vol. Promedio

..... Crecimiento% por año

Descripción: Tiene significado como

Atributos	Opcional	Formato	Longitud Máxima	Notas	Ver definición total	Identificador Unico
Nombre	Dominio					

Relaciones: Cada ocurrencia de esta entidad

debe ser pueden	frase de liga	uno y solo uno uno o muchos	nombre entidad	Arc o

Notas

Análisis de Atributos

Nombre de entidad en dominio			
Descripción/nota			
Obligatorio/Opcional			
.....% inicialmente		en condición	
.....% normalmente		
Formato	Longitud Max.	Longitud Promedio	Unidad de medida
.....
	Usuario	Derechos de Acceso (C.U.D.A.R.All)	Nivel/Autoridad
Disponible para
.....
.....
Responsabilidad de
Reglas de Validación			
.....			
Valor Default (sólo si es obligatorio)		
Valor para nulo (sólo si es opcional)		
Derivación			
Valor	Valor Máximo	Abreviación	Significado

RESTRICCIONES TECNICAS

Se tratará de conseguir CASE*DICTIONARY, para llevar el proyecto, sino se obtiene se hará la documentación manualmente, llenando los formatos correspondientes, siguiendo la metodología CASE.

Si se consigue el CASE*DICTIONARY, se le da la opción de emplear la máquina donde se encuentre para el desarrollo, sino se buscará conseguir los productos ORACLE para PC, sino es posible se empleará la VAX de Centro de Cálculo de la Facultad de Ingeniería (CECAFI).

Se decidió emplear una PC para producción, ya que para la DIE es más fácil obtenerla; aunque se podría utilizar la VAX de CECAFI, no se quiere depender de la operación del centro de cómputo para la utilización del sistema.

Diagrama de Entidad Relación del Banco de Horas.

Después de realizar el análisis minucioso de los reportes, documentos y conocimiento del sistema actual del banco de horas, se llegó al modelo de entidad-relación que se muestra en la figura 5.

En las figuras 6.1 a 6.5 se muestra el detalle de los atributos de cada entidad, así como la opcionalidad de los mismos.

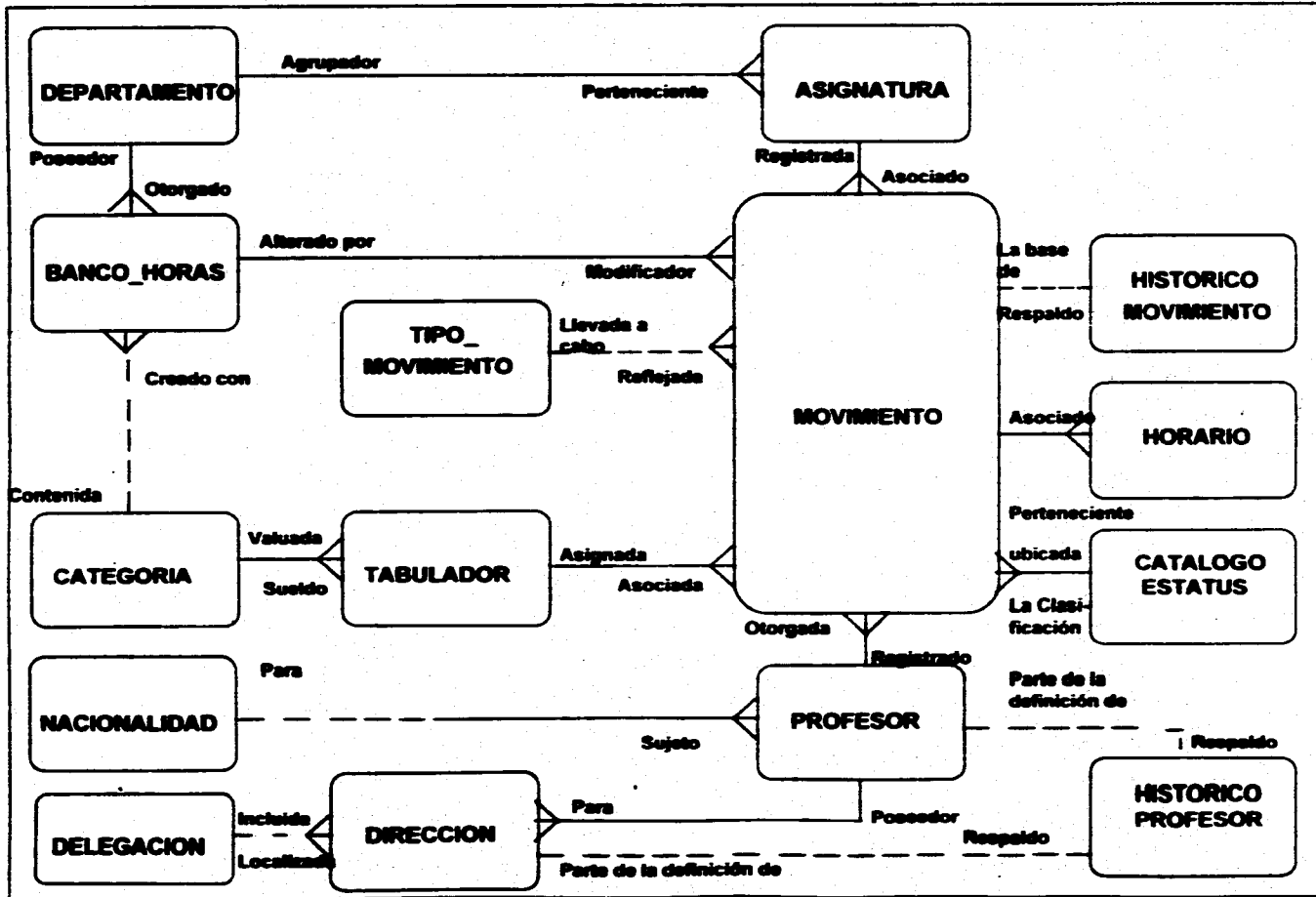


Figura 5. Diagrama de Entidad-Relación del Banco de Horas.

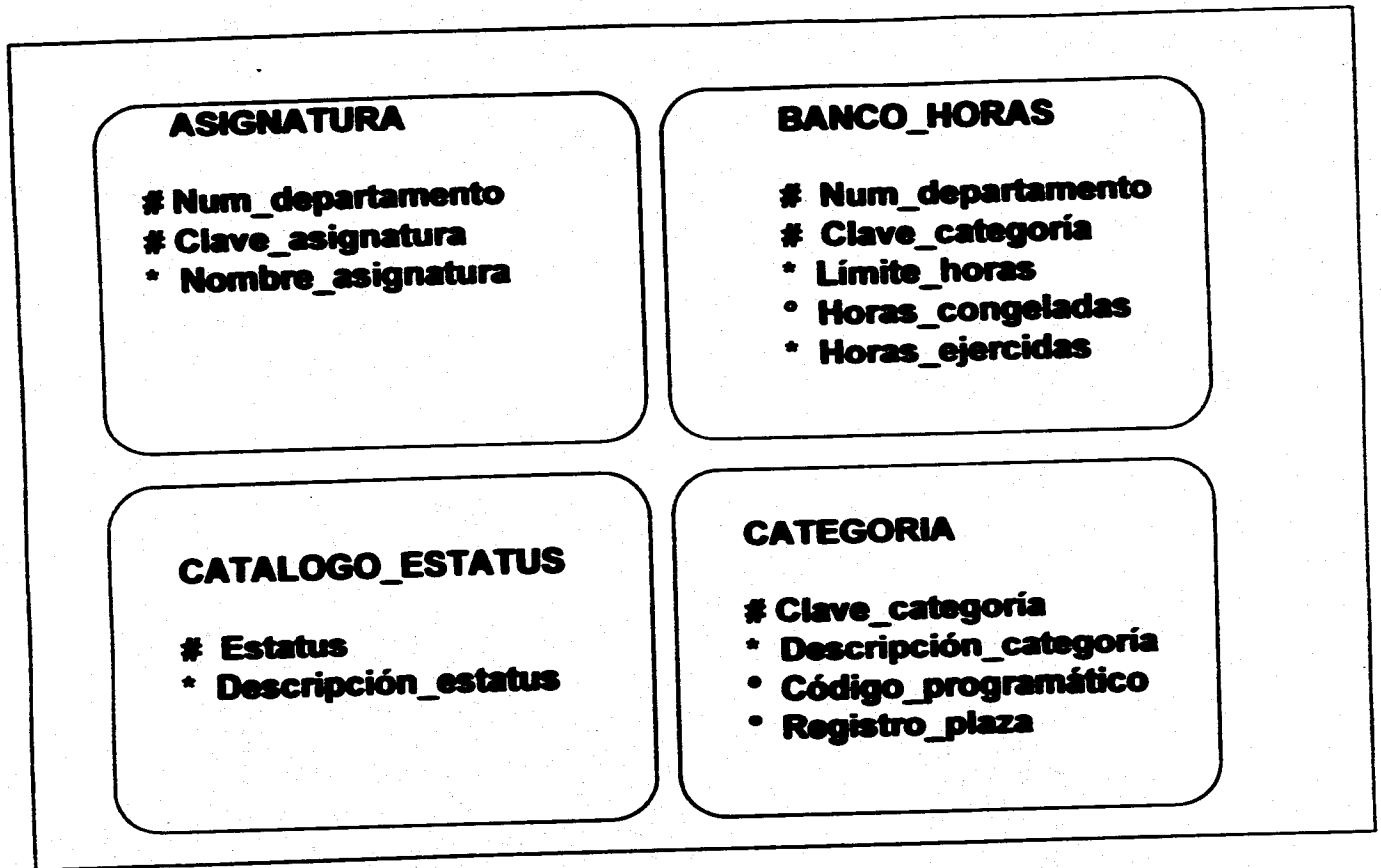


Figura 6.1. Entidades con atributos



Figura 6.2. Entidades con atributos

HISTORICO_PROFESOR

#RFC

Fecha_baja

*** Nombre**

*** Clave_nacionalidad**

*** Sexo**

*** Estado_civil**

*** Antigüedad**

*** Dirección_casa**

◦ Colonia

◦ CP

◦ Teléfono

◦ Clave_delegación

HORARIO

RFC

Clave_asignatura

Num_departamento

Fecha_inicio

Fecha_fin

Tipo_movimiento

Clave_categoria

◦ Grupo

◦ Lunm

◦ Marm

◦ Miem

◦ Juem

◦ Viem

◦ Sabm

◦ Lunt

◦ Mart

◦ Miet

◦ Just

◦ Viet

◦ Sabt

Figura 6.3. Entidades con atributos

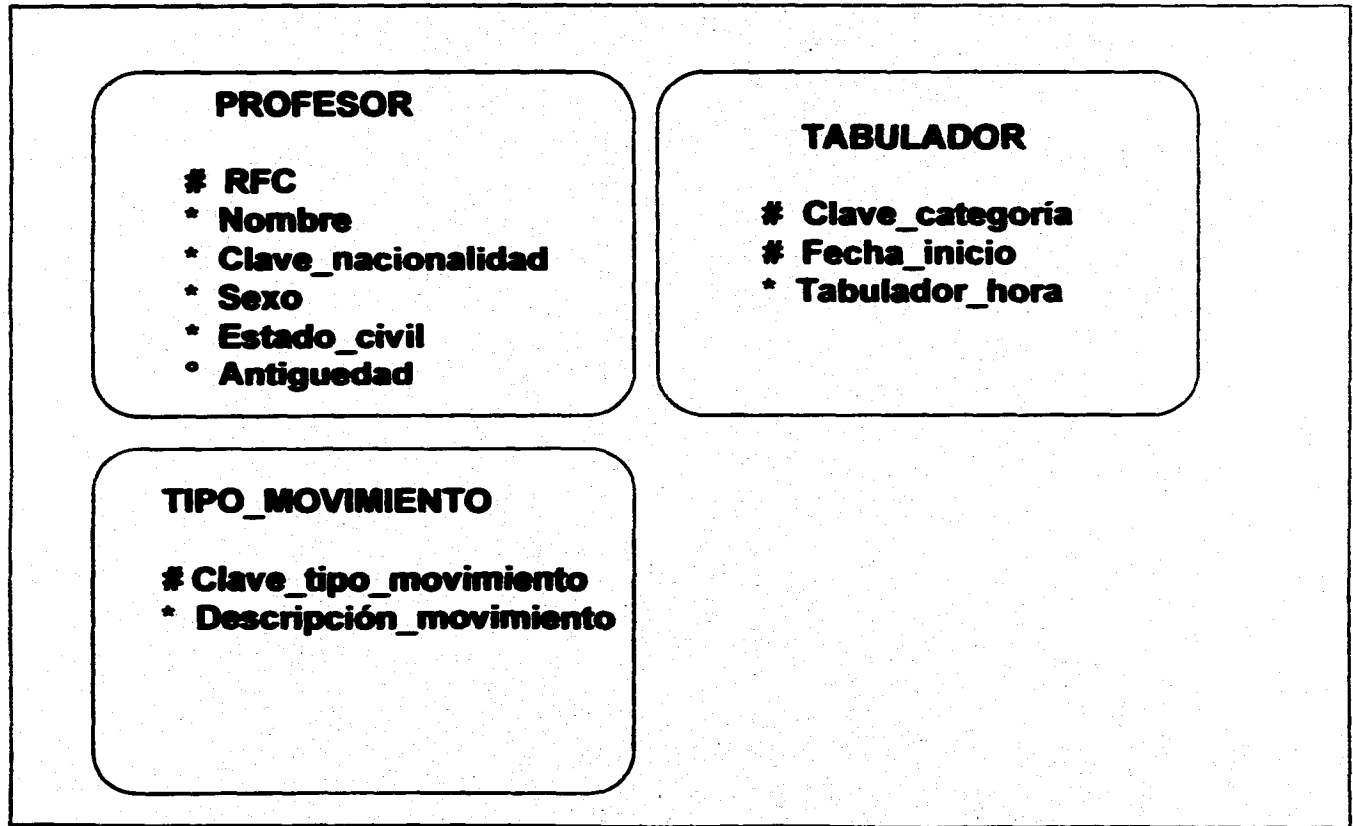


Figura 6.4. Entidades con atributos

MOVIMIENTO

- # RFC
- # Num_departamento
- # Fecha_inicio
- # Fecha_fin
- # Clave_categoria
- # Clave_tipo_movimiento
 - Horas_teoría
 - Horas_práctica
 - Estatus
 - Observaciones
 - Artículo_51
 - Tab_fecha_inicio
 - Sueldo
 - Sueldo_num

NACIONALIDAD

- # Clave_nacionalidad
- * Nombre_nacionalidad

PROFESOR

- # RFC
- * Nombre
- * Clave_nacionalidad
- * Sexo
- * Estado_civil
- Antigüedad

Figura 6.5. Entidades con atributos

Análisis de Entidades

Nombre (plural) ...Profesor(es)..... Subtipo de

Sinónimos ...Maestro..... Vol. inicial400.....

.....Catedrático..... Vol. Promedio400.....Max. probable600.....

.....Instructor..... Crecimiento2.....% por año

Descripción: Tiene significado como Personal de la U.N.A.M. que imparte una o más asignaturas.

Atributos	Opcional	Dominio	Formato	Longitud Máxima	Notas	Ver definición total	Identificador Unico
Nombre	N		caracter	15			/
RFC	N		caracter	40			
Sexo	N		caracter	1			
Estado civil	N		caracter	1			
Antigüedad	S		numérico	3			

Relaciones: Cada ocurrencia de esta entidad

debe \ ser	frase de liga	uno y solo uno uno o muchos	nombre entidad	Arco
Debe ser	Poseedor	una o muchas	direccion(es)	
Puede ser	Parte de definición de	una y solo una (ocurrencia)	histórico_profesores	
Debe estar	registrado	uno o muchos	movimientos	
Debe estar	sujeto	una y solo una	nacionalidad	

Notas

Análisis de Entidades

Nombre (plural) ...Banco Horas..... Subtipo de

Sinónimos Vol. inicial90.....

..... Vol. Promedio90.....Max. Probable110...

..... Crecimiento5.....% por año

Descripción: Tiene significado como
 Límite de horas que tiene un departamento, para una categoría específica.

Atributos	Opcional	Formato	Longitud Máxima	Notas	Ver definición total	Identificador Único
Nombre		Dominio				
Límite horas	N	Numérico	8,2			
Horas congeladas	S	Numérico	6,2			
Horas ejercidas	N	Numérico	8,2		Derivado	

Relaciones: Cada ocurrencia de esta entidad

debe ser	frase de liga	uno y solo uno uno o muchos	nombre entidad	Arco
puede ser				
Debe ser	alterado por	uno ó muchos	movimiento (s)	
Debe ser	otorgado a	uno y solo un	departamento	
Puede ser	creada con	una y solo una	categoría	

Notas
 Las horas ejercidas es un atributo derivado, que se obtiene de la suma de horas de los movimientos que tienen los profesores en cada departamento, pero se puso para mejorar el rendimiento del sistema.

Análisis de Atributos

Nombre Limite_Horas..... de entidad Banco_Horas..... en dominio			
Descripción/Nota: Número máximo de horas que posee un departamento en cada categoría.			
Obligatorio/Opcional			
.....% inicialmente en condición			
.....% normalmente			
Formato	Longitud Max.	Longitud Promedio	Unidad de medida
..... Numérico.....8,2.....
	Usuario	Derechos de Acceso (C,U,D,A,R,All)	Nivel/Autoridad
Disponibles para

Responsabilidad de
Reglas de Validación			
Valor Default (sólo si es obligatorio)			
Valor para nulo (sólo si es opcional)			
Derivación: La suma de todas las ocurrencias de una categoría en todos los departamentos, no debe exceder el límite de la División.			
Valor	Valor Máximo	Abreviación	Significado

Análisis de Atributos

Nombre Horas Congeladas..... de entidad ...Banco Horas..... en dominio			
Descripción nota Número de horas que un departamento posee, pero no puede emplear, por haber hecho una mala planeación de movimientos.			
Obligatorio Opcional			
.....% inicialmente en condición			
.....% normalmente			
Formato	Longitud Max.	Longitud Promedio	Unidad de medida
Número.....	6,2.....
	Usuario	Derechos de Acceso (C,U,D,A,R,All)	Nivel/Autoridad
Disponible para			
.....			
.....			
Responsabilidad de			
.....			
Reglas de Validación			
Se da por mal uso, el total de horas congeladas no debe exceder el límite de horas del departamento.			
.....			
Valor Default	(sólo si es obligatorio)	
Valor para nulo	(sólo si es opcional)	
Derivación			
Valor	Valor Máximo	Abreviación	Significado

Análisis de Atributos

Nombre ...Horas ejercidas..... de entidad ...Banco_Horas..... en dominio

Descripción nota:

Total de horas que han sido otorgadas al personal académico, es un atributo derivado de sumar los movimientos de todos los profesores.

Obligatorio Opcional

.....% inicialmente en condición

.....% normalmente

Formato Longitud Max. Longitud Promedio Unidad de medida

.....Número..... 8.2.....

Usuario Derechos de Acceso Nivel/Autoridad
(C,U,D,A,R,All)

Disponble para

Responsabilidad de

Reglas de Validación

Se suman las horas de todos y cada uno de los académicos de una categoría y ésta suma no debe exceder el límite de horas de esa categoría en ese departamento.

Valor Default (sólo si es obligatorio)

Valor para nulo (sólo si es opcional)

Derivación

Valor	Valor Máximo	Abreviación	Significado

Nombre (plural)Movimiento(s)..... Subtipo de

Sinónimos Vol. inicial600.....

..... Vol. Promedio600...Max. probable...1000..

..... Crecimiento10.....% por año

Descripción: Tiene significado como
Cualquier solicitud para afectar a un profesor, resultando en una alteración al banco de horas.

Atributos	Opcional	Formato	Longitud Máxima	Notas	Ver definición total	Identificador Único
Nombre		Nombre				
Fecha inicio	N	Fecha				
Fecha fin	N	Fecha				
horas teoría	S	Número	4,1			
horas práctica	S	Número	4,2			
observaciones	S	Carácter	40			
artículo sl	S	Carácter	1			
sueldo min	N	Número	8,2			
sueldo	N	Carácter	7,5			

Relaciones: Cada ocurrencia de esta entidad

debe / ser / puede/	frase de liga	uno y solo uno / uno o muchos	nombre entidad	Arco
debe estar	modificador	una y solo una (ocurrencia)	banco_horas	
debe estar	reflejado	uno y solo un	tipo_movimiento	
debe estar	asociado	uno y solo un	laborador	
puede ser	la base de	uno y solo un	histórico_movimiento	
debe ser	ubicado	uno y solo una (ocurrencia) en el	catálogo_status	
debe estar	asociado	uno o muchos	horarios	

Notas

Análisis de Entidades

Nombre (plural) Asignatura(s) Subtipo de

Sinónimos Materia Vol. inicial 80

..... Cátedra Vol. Promedio 80

..... Crecimiento 1 % por año

Descripción: Tiene significado como
Clase impartida por un profesor en la U.N.A.M.

Atributos	Opcional	Formato	Longitud Máxima	Notas	Ver definición total	Identificador Unico
Nombre		Dominio				
Clave asignatura	N	Númerico	4			/
Nombre asignatura	N	Caracter	30			

Relaciones: Cada ocurrencia de esta entidad

debe \ ser puede/	frase de liga	uno y solo uno uno ó muchos	nombre entidad	Arco
debe ser	pertenciente	uno y solo un	departamento	
puede estar	registrada	uno ó muchos	movimientos	

Notas

Análisis de Atributos

Nombre casa/oficina..... de entidad ..Dirección..... en dominio			
Descripción/Nota Identificador de si la dirección es de la casa o de la oficina del profesor.			
Obligatorio/Opcional			
.....ºn inicialmente en condición			
.....ºn normalmente			
Formato	Longitud Max.	Longitud Promedio	Unidad de medida
.....caracter.....
	Usuario	Derechos de Acceso (C,U,D,A,R,All)	Nivel/Autoridad
Disponibles para
.....
.....
Responsabilidad de
Reglas de Validación			
C casa			
O oficina			
Valor DefaultC.....	(sólo si es obligatorio)	
Valor para nulo	(sólo si es opcional)	
Derivación			
Valor	Valor Máximo	Abreviación	Significado

Ahora se presenta en la figura 7 el análisis de funciones que se realizó para el banco de horas.

Hay que recordar que en el análisis funcional, se muestra lo que se hace o se requiere hacer en la Secretaría Auxiliar, no importando la manera en que se hace (los llamados mecanismos).

También es importante aclarar que la descomposición funcional, no representa los pasos secuencialmente, si no que pueden ser intercambiadas, sólo habrá de mencionarse que normalmente, al construir el diagrama de descomposición funcional, se hace pensando secuencialmente como se realiza, más no es una regla el hacerlo.

Descomposición de Funciones

Aplicación: Banhoras

Referencia: Banhoras

Banhoras

SISTEMA INTEGRAL DEL BANCO DE HORAS.- Controlar el banco de horas de la DIE.

ALTAS

Descompuesto

Permitir altas de profesores.

BAJAS

Descompuesto

Eliminar o respaldar datos generales y nombramientos de los profesores.

REPORTES

Descompuesto

Emitir reportes.

MODIFICA

Descompuesto

Permitir modificaciones a los catálogos

SEPARACION

Clasificación por tipo de movimiento (manual).

Fin

Figura 7. Análisis de funciones

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTAS

ALTAS

Ingresar datos generales, movimientos y asignaturas que imparte un profesor en el departamento.

ALTA1

Descompuesto

Dar de alta por nuevo ingreso.

ALTA2

Descompuesto

Crear alta por reingreso

ALTA3

Descompuesto

Emitir una prorroga

ALTA4

Descompuesto

Dar de alta por otro nombramiento

ALTA5

Descompuesto

Permitir aumentos de horas

Continua

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTAS

ALTA6

Descompuesto

Efectuar una promoción.

ALTA7

Descompuesto

Licencias

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA1

ALTA1

Dar alta por nuevo ingreso

ALTA1.1

Checar que efectivamente no exista anteriormente el profesor.

ALTA1.2

Captar datos generales del profesor y de la asignatura

ALTA1.3

Validar que el número de horas ejercidas no exceda el límite de horas permitido.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA2

ALTA2

Dar alta por reingreso

ALTA2.1

Buscar datos generales del profesor en el histórico, si existen traerlos.

ALTA2.2

Ver alta 1.3

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA3

ALTA3

Dar una prorroga

ALTA3.1

Verificar que el profesor existe y tiene la categoria indicada.

ALTA3.2

Validar fecha de término menor en un día al nuevo contrato.

ALTA3.3

Ver alta 1.3.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA4

ALTA4

Dar de alta por otro nombramiento.

ALTA4.1

Verificar si tiene nombramiento en la División, si es afirmativo, traer los datos, si no obtenerlos. (Puede tener nombramiento en otra dependencia de la UNAM).

ALTA4.2

Ver alta 1.3.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA5

ALTA5

Permitir aumento de horas.

ALTA5.1

Verificar si tiene nombramiento y categoría en la DIE, si es afirmativo traer los datos. (Puede tener nombramiento en otra dependencia de la UNAM).

ALTA5.2

Ver alta 1.3.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA6

ALTA6

Dar una promoción.

ALTA6.1

Verificar que el profesor tenga nombramiento en la división, el cual debe ser menor a la categoría que se pide y solo son válidas dentro del mismo nivel (Ej. Ayudante "A" a ayudante "B", pero no de ayudante "B" a profesor "A").

ALTA6.2

Ver alta 1.3.

Fin

Descomposición de Funciones

Aplicación: **Banhoras**

Referencia: **ALTA8**

ALTA7

Generar licencias.

ALTA7.1

Descompuesto

Tener registro de una licencia por año sabático.

ALTA7.2

Descompuesto

Registrar una licencia por comisión oficial.

ALTA7.3

Descompuesto

Dar una licencia por estudios en el país.

ALTA7.4

Descompuesto

Otorgar una licencia por estudios en el extranjero.

ALTA7.5

Descompuesto

Otorgar una licencia por gravidez.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA8.1

ALTA7.1

Tener registro de una licencia por año sabático.

ALTA7.1.1

Verificar que sea una categoría con derecho al año sabático

ALTA7.1.2

Verificar que tenga seis años continuos de trabajar en la UNAM como profesor de carrera (durante este tiempo debe obtener la definitividad)

ALTA7.3

Afectar el nombramiento y ponerlo como año sabático.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA8.2

ALTA7.2

Registrar una licencia por comisión oficial.

ALTA7.2.1

Validar la existencia del profesor.

ALTA7.2.2

Tener registro de la licencia.

ALTA7.3

Registrar las observaciones del motivo de la licencia.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA8.3

ALTA7.3

Dar una licencia por estudios en el país.

ALTA7.3.1

Ver ALTA7.2.1

ALTA7.3.2

Ver ALTA7.3.2

ALTA7.3.3

Ver ALTA7.2.3

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA8.4

ALTA7.4

Dar una licencia por estudios en el extranjero.

ALTA7.4.1

Ver ALTA7.2.1

ALTA7.4.2

Ver ALTA7.3.2

ALTA7.4.3

Ver ALTA7.2.3

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:ALTA8.5

ALTA7.5

Dar una licencia por gravedad.

ALTA7.5.1

Ver ALTA7.2.1

ALTA7.5.2

Ver ALTA7.3.2

ALTA7.5.3

Ver ALTA7.2.3

ALTA7.5.4

Verificar que el sexo sea femenino.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:BAJAS

BAJAS

Eliminar o respaldar datos generales y nombramientos del profesor.

BAJA1

Descompuesto

Efectuar una baja por un interinato.

BAJA2

Descompuesto

Dar disminución de horas.

BAJA3

Descompuesto

Otorgar una baja por renuncia.

BAJA4

Descompuesto

Permitir una baja por rescisión.

BAJA5

Descompuesto

Efectuar la baja por defunción.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:BAJA1

BAJA1

Dar de baja por un interinato.

BAJA1.1

Eliminar datos personales y nombramiento, siempre y cuando no tenga otro nombramiento, si lo tiene, sólo eliminar el movimiento, pero conservar los datos personales.

BAJA1.2

Actualizar históricos de movimientos y de profesores.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:BAJA2

BAJA2

Dar de baja por disminución de horas.

BAJA2.1

Validar que las horas que se van a dar de baja sean menos de las que tiene ejercidas.

BAJA2.2

Validar la categoría en la que se desea dar la disminución.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:BAJA3

BAJA3

Dar de baja por renuncia.

BAJA3.1

Ver BAJA1.1

BAJA3.2

Ver BAJA1.2.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:BAJA4

BAJA4

Dar de baja por rescisión.

BAJA4.1

Ver BAJA1.1

BAJA4.2

Ver BAJA1.2.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:BAJA5

BAJA5

Dar de baja por defunción.

BAJA5.1

Dar de baja los datos generales y contrato. No respaldar en históricos.

Fin

Descomposición de Funciones

Aplicación: **Banhoras**

Referencia: **REPORTES**

REPORTES

Emitir reportes.

REP1

Descompuesto

Generales.

REP2.

Descompuesto

Resúmenes.

REP3

Descompuesto

Catálogos.

REP4

Descompuesto

Procesos.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia: REP1

REP1

Generales.

REP1.1

General.

REP1.2

General por categorías.

REP1.3

General por categoría por departamento.

REP1.4

Profesores y técnicos académicos.

REP1.5

Profesores de carrera.

REP1.6

Técnicos académicos.

Continua

Descomposición de Funciones

Aplicación: Banhoras

Referencia: REP1

REP1.7

Profesores de Carrera por departamento.

REP1.8

Técnicos académicos.

REP1.9

Profesores de carrera por departamento.

REP1.10

Técnicos académicos

REP1.11

Licencias y comisiones

REP1.12

Listado de prórrogas.

Continua

Descomposición de Funciones

Aplicación: Banhoras

Referencia: REP2

REP2

Resúmenes.

REP2.1

Resumen General por categorías.

REP2.2

Resumen por categoría y departamento.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia: REP3

REP3

Catálogos.

REP3.1

Posibles Estatus.

REP3.2

Movimientos Válidos.

REP3.3

Categorías.

REP3.4

Departamentos.

REP3.5

Delegaciones.

REP3.6

Nacionalidades.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia: REP4

REP4

Procesos.

REP4.1

Movimientos en bitácora.

REP4.2

Generar consejo técnico.

REP4.3

Impresión de formas únicas.

REP4.4

Movimientos pendientes

REP4.5

Cambiar estatus de movimiento.

Fin

Descomposición de Funciones

Aplicación: Banhoras

Referencia:MODIFICA

MODIFICA

Permitir modificaciones a los catálogos.

MOD1.

Asignaturas.

MOD2

Banco de horas.

MOD3

Categorías.

MOD4

Delegaciones.

MOD5

Departamentos.

MOD6

Estatus.

Continua

Descomposición de Funciones

Aplicación: Banhores

Referencia:MODIFICA

MODIFICA

MOD7.

Nacionalidades.

MOD8

Tabuladores.

MOD9

Tipos de movimientos.

Fin

DIAGRAMA DE FLUJO DE DATOS.

En la figura 8, se muestra el diagrama de flujo de datos del sistema banco de horas, el diagrama 8 muestra el denominado diagrama de contexto, el cual presenta las interfases externas del sistema, las entidades con las que se interrelaciona y como fluyen los datos desde y hacia el sistema, el cual es operado por la Secretaría Auxiliar.

En la figura 8.1, se muestra el diagrama detallado o de primer nivel.

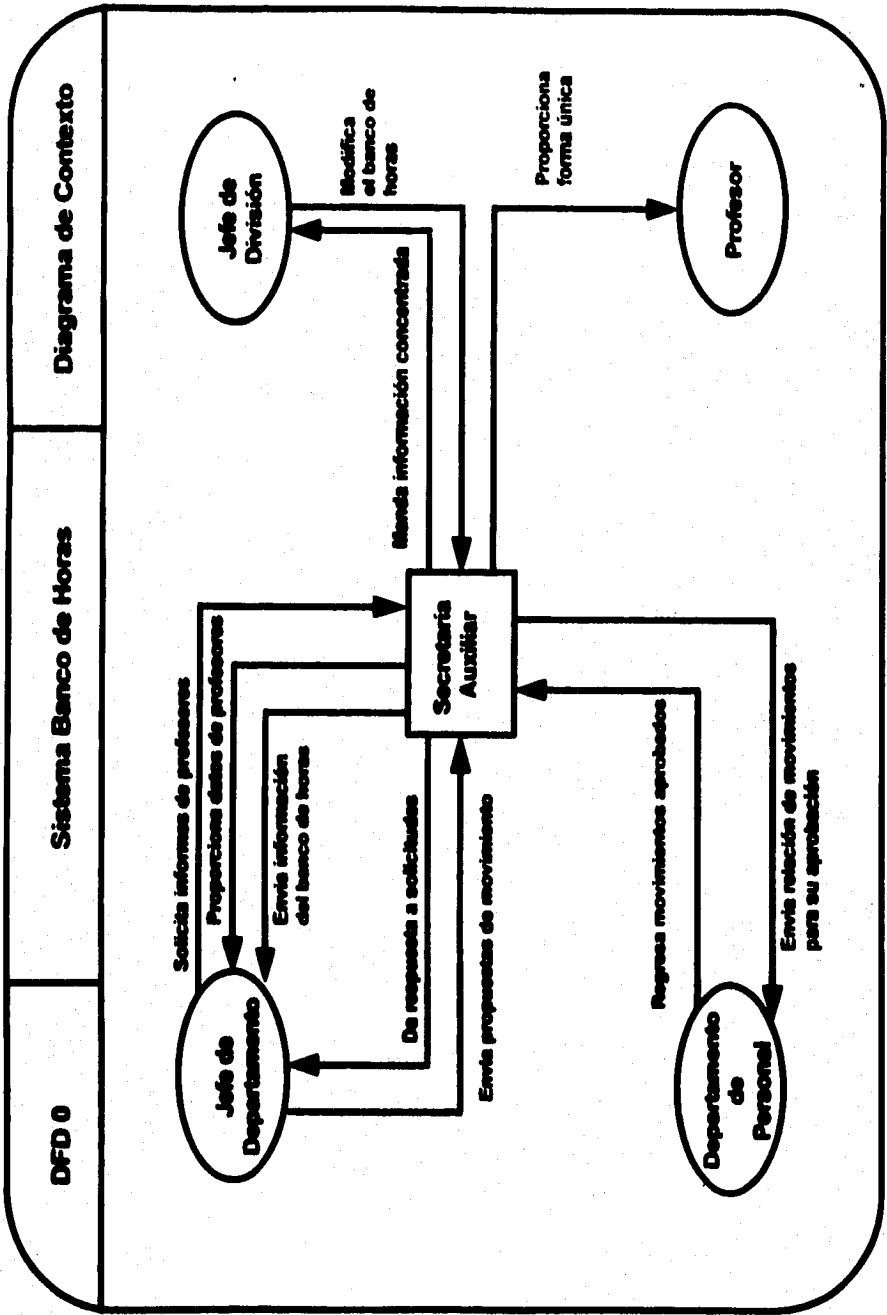


Figura 8.0. Diagrama de flujo de datos. (Diagrama de contexto).

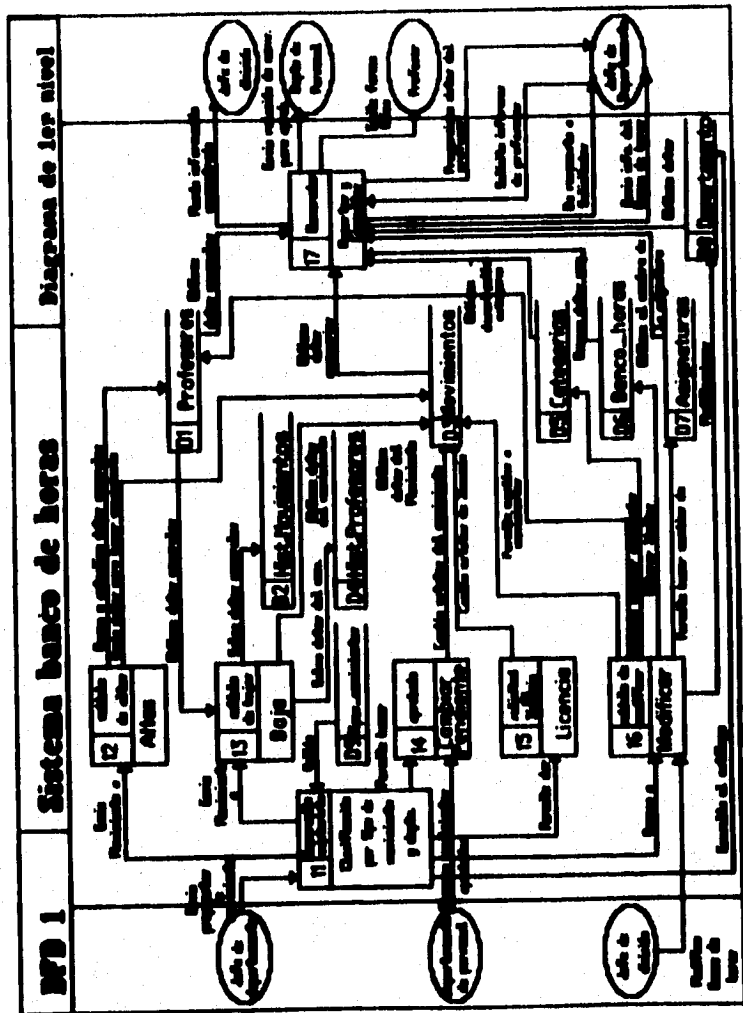


Figura 8.1. Diagrama de flujo de datos. (Diagrama de primer nivel).

MATRIZ DE CHEQUEO.

En la figura 9, se presenta la matriz de verificación cruzada de funciones contra entidades, en dicha matriz, se presentan unas letras que significan la manipulación de los datos que hace la función elemental correspondiente, por ejemplo Actualización, Borrado, etc.).

Función Elemental	Matriz de Funciones vs Entidades				
	Nombres de Entidades				
Referencia	Asignatura	Banco horas	Categoría	Departamento	Delegación
Alta 1	R	U,R	R	R	R
Alta 2	R	U,R	R	R	R
Alta 3	R	R	R	R	R
Alta 4	R	U,R	R	R	R
Alta 5	R	U	R	R	R
Alta 6	R	U	R	R	R
Alta 7.1	R	R	R	R	R
Alta 7.2	R	R	R	R	R
Alta 7.3	R	R	R	R	R
Alta 7.4	R	R	R	R	R
Alta 7.5	R	R	R	R	R
Baja 1	R	U,R	R	R	R
Baja 2	R	U	R	R	R
Baja 3	R	U	R	R	R
Baja 4	R	U	R	R	R
Baja 5	R	U	R	R	R
Rep 1.1			R	R	
Rep 1.2			R	R	
Rep 1.3			R	R	
Rep 1.4			R	R	
Rep 1.5			R	R	
Rep 1.6			R	R	
Rep 1.7			R	R	
Rep 1.8			R	R	
Rep 1.9			R	R	
Rep 1.10			R	R	
Rep 1.11			R	R	
Rep 1.12			R	R	
Rep 1.13			R	R	
Rep 1.14			R	R	
Rep 2.1			R	R	
Rep 2.2			R	R	
Rep 3.1					
Rep 3.2					
Rep 3.3			R		
Rep 3.4				R	
Rep 3.5					R
Rep 3.6					
Rep 4.1				R	
Rep 4.2	R	A	R	R	
Rep 4.3	R		R	R	R
Rep 4.4			R	R	
Rep 4.5	R	R	R	R	R
Mod 1	A			R	

U update
C create
R readonly
D delete
A all

Función Elemental	Dirección	Matriz de Funciones vs Entidades			
		Nombres de Entidades			
Referencia		Histórico Movimientos	Histórico Profesores	Horarios	Nacionalidad
Alta 1	A	R	R	A	R
Alta 2	A	A	A	A	R
Alta 3	A			A	R
Alta 4	A			A	R
Alta 5	A			A	R
Alta 6	A			A	R
Alta 7.1	R			A	R
Alta 7.2	R			A	R
Alta 7.3	R			A	R
Alta 7.4	R			A	R
Alta 7.5	R			A	R
Baja 1	U,D	C	C	D	R
Baja 2	U,R	C	C	U,D	R
Baja 3	D	C	C	D	R
Baja 4	D	C	C	D	R
Baja 5	D			D	R
Rep 1.1					
Rep 1.2					
Rep 1.3					
Rep 1.4					
Rep 1.5					
Rep 1.6					
Rep 1.7					
Rep 1.8					
Rep 1.9					
Rep 1.10					
Rep 1.11					
Rep 1.12					
Rep 1.13					
Rep 1.14					
Rep 2.1					
Rep 2.2					
Rep 3.1					
Rep 3.2					
Rep 3.3					
Rep 3.4					
Rep 3.5					
Rep 3.6					R
Rep 4.1					
Rep 4.2				R	R
Rep 4.3	R			R	R
Rep 4.4					
Rep 4.5	R			R	R
Mod 1					

U update
C create
R readonly
D delete
A all

Función Elemental	Matriz de Funciones vs Entidades				
	Nombres de Entidades				
Referencia	Profesor	Estatus	Movimiento	Tabulador	Tipos
Alta 1	C	R	C	R	R
Alta 2	U,C	R	C	R	R
Alta 3	U,R	R	U	R	R
Alta 4	U,R	R	U,C	R	R
Alta 5	U,R	R	U,C	R	R
Alta 6	U,R	R	U,C	R	R
Alta 7.1	R	R	U	R	R
Alta 7.2	R	R	U	R	R
Alta 7.3	R	R	U	R	R
Alta 7.4	R	R	U	R	R
Alta 7.5	U,D	R	U	R	R
Baja 1	U,R	R	D	R	R
Baja 2	D	R	U,D	R	R
Baja 3	D	R	D	R	R
Baja 4	D	R	D	R	R
Baja 5	R	R	D	R	R
Rep 1.1	R	R	R	R	R
Rep 1.2	R	R	R	R	R
Rep 1.3	R	R	R	R	R
Rep 1.4	R	R	R	R	R
Rep 1.5	R	R	R	R	R
Rep 1.6	R	R	R	R	R
Rep 1.7	R	R	R	R	R
Rep 1.8	R	R	R	R	R
Rep 1.9	R	R	R	R	R
Rep 1.10	R	R	R	R	R
Rep 1.11	R	R	R	R	R
Rep 1.12	R	R	R	R	R
Rep 1.13	R	R	R	R	R
Rep 1.14	R	R	R	R	R
Rep 2.1	R	R	R	R	R
Rep 2.2	R	R	R	R	R
Rep 3.1	R			R	
Rep 3.2				R	R
Rep 3.3				R	
Rep 3.4				R	
Rep 3.5				R	
Rep 3.6		R	R	R	
Rep 4.1	R	R	R	R	R
Rep 4.2	R	R	R	R	R
Rep 4.3	R	R	R		R
Rep 4.4	R	R	R		R
Rep 4.5	R	U	R		R
Mod 1			R	R	

U update
C create
R readonly
D delete
A all

Función Elemental	Matriz de Funciones vs Entidades				
	Nombres de Entidades				
Referencia	Asignatura	Banco_horas	Categoría	Departamento	Delegación
Mod 2		A	R	R	
Mod 3			A		
Mod 4					A
Mod 5				A	
Mod 6					
Mod 7					
Mod 8			R		
Mod 9					

U update
 C create
 R readonly
 D delete
 A all

Función Elemental	Matriz de Funciones vs Entidades				
		Nombres de Entidades			
Referencia	Dirección	Histórico Movimientos	Histórico Profesores	Horarios	Nacionalidad
Mod 2					
Mod 3					
Mod 4					
Mod 5					
Mod 6					
Mod 7					A
Mod 8					
Mod 9					

U update
 C create
 R readonly
 D delete
 A all

Función Elemental	Matriz de Funciones vs Entidades				Tipos Movimiento
		Nombres de Entidades			
Referencia	Profesor	Estatus	Movimiento	Tabulador	
Mod 2					
Mod 3					
Mod 4					
Mod 5					
Mod 6		A			
Mod 7					
Mod 8				A	
Mod 9					A

U update
 C create
 R readonly
 D delete
 A all

IDENTIFICAR FUNCIONES COMPLEJAS

- ¿Cómo se van a manejar los tipos de licencia?, Es posible sólo ponerles un atributo al movimiento, o crear otro movimiento por licencia?
- ¿Cómo y cuándo se congelan horas?
- ¿El tabulador sólo se genera para los interinos, si no es así, para qué categorías se tiene?
- ¿Las altas por reingreso y reanudación podrían manejarse igual en el sistema?
- ¿Qué implica una reclasificación?
- ¿Cuál es el límite máximo de horas, 40 ó 48 y cómo se validan?
- ¿Cómo manejar la renuncia, rescisión y defunción?, ya que desde el punto de vista del sistema, podrían eliminarse los datos de manera similar

IDENTIFICAR COMO SE HARA LA TRANSICION

- La tabla de departamentos se hará usando el SQL*LOADER y cargando el archivo DEPTON.dbf.
- Para la carga de la tabla de CATEGORIAS se usará el archivo CATEGORI.dbf y capturar las horas del tabulador, así como el registro de plaza y el código programático.
- En la tabla de ASIGNATURAS, habrá de hacerse una carga de los nombres de las esignaturas del archivo ESTRUC.dbf el campo MATER_MOV, en una tabla temporal, eliminar las repetidas con un select distinct y las que no sea posible quitar a mano. Después habrán de capturarse las claves de las asignaturas y departamentos a los que corresponden. Una vez que se tenga esta tabla temporal terminada, cargarla hacia la definitiva.
- En las tablas de históricos, no se cargará nada.
- La tabla de MOVIMIENTOS temporal, se cargará con el archivo MOVIM.dbf, las horas actuales se cargarán en la columna de horas teoría y ver como obtener la clave_asig en vez de mater_mov (tal vez con un select de la tabla de asignaturas). Posteriormente cargar la tabla definitiva.

- Cargar la tabla TIPO_MOVIMIENTO con el archivo tipos.dbf y capturar las columnas faltantes.

- Crear una tabla de NACIONALIDAD temporal, a la cual se le cargará el campo NACIONAL del archivo GENERALE.dbf, se le hará un select distinc hacia la otra tabla y se les generarán las claves secuencialmente, después se cargará hacia la tabla definitiva.

- Se creará una tabla PROFESOR temporal, a la que se le cargarán datos del archivo GENERALE.dbf, la nacionalidad se cargará tal cual, después con un select, cargar la clave de la nacionalidad.

- Crear una tabla de DIRECCIONES temporal para direcciones de CASA y cargar las columnas necesarias del archivo GENERALE.dbf, en la clave_casa_oficina meter en todos 'C'. Crear otra tabla similar para direcciones de oficina y poner en clave_casa_oficina todas con 'O'. Hacer un select con UNION de las dos tablas temporales, hacia la tabla definitiva de DIRECCIONES.

Documento: ANAL2038

FACTORES CRITICOS DE EXITO

- No conseguir el ORACLE para PC.
- No aprobación del tema de tesis.
- Algún miembro del proyecto salga de viaje.
- El personal no quiera emplear el sistema.
- Si se tienen retardos, es posible no tener el sistema a tiempo.
- Si los retardos se van incrementando, se podría ver la manera de involucrar más personas en la etapa de programación.

DEFINICION DE LOS CRITERIOS DE ACEPTACION

Para la aceptación del sistema del Banco de horas, habrán de satisfacerse los siguientes aspectos:

- Se deberá poder obtener los reportes especificados en la actividad "definición de reportes".
- Debe ser capaz de manejar los tipos de movimientos existentes.
- Un profesor podrá tener varios movimientos en varios departamentos o categorías, así como varios nombramientos de la misma categoría en un departamento.
- Se podrá imprimir la forma única.
- Habrá de ser capaz de emitir las prórrogas de contratos, dando un rango de fechas.
- Deberá poder obtener el listado de ayudantes que tienen 5 años como tales.
- La interfase con el usuario deberá ser amigable.
- Se mantendrá un "spool" del último listado impreso, por si se requieren mandar varias copias.
- Permitirá hacer cambios en línea al banco de horas, para hacer los balances.
- Deberán poder aprovecharse los datos ya existentes.
- Las ayudas en línea deberán ser suficientes.

DEFINIR REQUERIMIENTOS DE RESPALDO Y RECUPERACION

Por seguridad de la información, habrán de hacerse respaldos de la base de datos como tal, usando las utilerías del manejador de la base de datos(import/export), esto se hará antes de cada consejo técnico, por si se presenta alguna falla se pueda recuperar la base de datos.

Por otro lado se harán respaldos del disco del manejador y la base de datos con las utilerías del sistema operativo (backup), para así tener respaldo del ambiente operativo, se harán cada seis meses.

Se podrá hacer respaldos de la base de datos y/o el ambiente operativo cuando se juzgue conveniente.

Para hacer una recuperación de información, ya sea de la base de datos o del ambiente operativo, se habrán de seguir y establecer las condiciones que indican los manuales, ya sea el de administrador de la base de datos o el del administrador del sistema, según sea al caso que se presente.

Todos los respaldos se deberán hacer por duplicado, dejando una copia en las oficinas de la DIE y la segunda en alguna otra localidad segura.

Se deberá responsabilizar por escrito a una persona de llevar a cabo los respaldos de información.

Se recomienda tener un equipo de fuerza ininterrumpible (UPS o no Break), para evitar perdidas de datos por fallas eléctricas; si no es posible contar con dicho equipo, lo que sucederá en caso de estar capturando movimientos, solo se perderá el último movimiento que se graban los cambios a las tablas de la base de datos.

**IV. 3 ETAPA DE DISEÑO
CALCULO DEL ESPACIO REQUERIDO**

UTILIZANDO LAS FORMULAS :

$$BR = \sum_i ((1-N_i) * (C_i + K + I)) + L + Q \quad RB = \text{trunc}(((1-F) * (S-H)) / BR)$$

$$B = R / RB$$

TOTAL DE BLOQUES (B) = 268.8842

SEGUN LOS SIGUIENTES CALCULOS

TABLA DEPARTAMENTO

i=1 a 2

N1=0 BR=29
C1=2 RB=54
N2=0 R=5
C2=15 B=0.0926

TABLA ASIGNATURA

i=1 a 3

N1=0 BR=46
C1=2 RB=34
N2=0 R=80
C2=4 B=2.3529
N3=0
C3=30

TABLA BANCO_HORAS

i=1 a 5

N1=0 N4=0.8
C1=2 C4=8 BR=39
N2=0 N5=0 RB=40
C2=5 C5=8 R=90
N3=0 B=2.250
C3=10

TABLA CATEGORIAS

i=1 a 5

N1=0 N4=0.7
C1=5 C4=12 BR=43.7
N2=0 N5=0.50 RB=36
C2=15 C5=4 R=84
N3=0.50 B=2.3333
C3=15

TABLA TIPO_MOVIMIENTO

i=1 a 2

N1=0 BR=34
C1=1 RB=34
N2=0 R=80
C2=25 B=2.3529

TABLA NACIONALIDAD

i=1 a 2

N1=0 BR=25
C1=2 RB=63
N2=0 R=10
C2=15 B=0.1587

TABLA DIRECCIONES

i=1 a 7

N1=0 N6=0.60
 C1=11 C6=10
 N2=0 N7=0.20
 C2=1 C7=30
 N3=0
 C3=40
 N4=0
 C4=2
 N5=0.50
 C5=5

BR=99.9
 RB=15
 R=700
 B=46.6667

TABLA PROFESORES

i=1 A 8

N1=0 N6=0
 C1=11 C6=1
 N2=0.30 N7=0
 C2=12 C7=1
 N3=0.50 N8=0.80
 C3=12 C8=8
 N4=0
 C4=12
 N5=0
 C5=1

BR=68.88
 RB=22
 R=400
 B=181.8182

TABLA MOVIMIENTOS

i=1 A 15

N1=0 N6=0 N11=0.70
 C1=11 C6=1 C11=2
 N2=0 N7=0.20 N12=0.97
 C2=2 C7=4 C12=40
 N3=0 N8=0.50 N13=0.98
 C3=8 C8=4 C13=1
 N4=0 N9=0.50 N14=0.50
 C4=8 C9=4 C14=5
 N5=0 N10=0 N15=0.50
 C5=5 C10=1 C15=5

BR=74.32
 RB=21
 R=600
 B=28.5714

TABLA HISTORICO_MOVIMIENTOS

i=1 a 6

N1=0 N4=0
 C1=11 C4=5
 N2=0 N5=0
 C2=8 C5=4
 N3=0 N6=0.30
 C3=1 C6=5

BR=47.9
 RB=32
 R=30
 B=0.9375

TABLA HISTORICO_PROFESORES

i=1 a 14

N1=0 N4=0.50 N7=0 N10=0 N13=0.20 BR=169.4
 C1=11 C4=12 C7=1 C10=40 C13=5 RB=9
 N2=0 N5=0 N8=0 N11=0.20 N14=0.30 R=30
 C2=8 C5=15 C8=1 C11=30 C14=10 B=3.3333
 N3=0.05 N6=0 N9=0.50 N12=0.20
 C3=12 C6=1 C9=8 C12=2

REVISAR Y DISEÑAR LAS LLAVES PRIMARIAS, FORANEAS E INDICES DE LAS TABLAS

TABLA	PRIMARIA	INDICE UNICO
ASIGNATURAS	Clave_asignatura	ASIGNATURAS_PK
BANCO_HORAS	Núm_departamento + Clave_Categoria	BANCO_HORAS_PK
CATALOGO_STATUS	Status	STATUS_PK
CATEGORIAS	Clave_categoria	CATEGORIAS_PK
DEPARTAMENTOS	Núm_departamento	DEPARTAMENTO_PK
DELEGACIONES	Clave_delegación	DELEGACIONES_PK
DIRECCIONES	RFC + Casa_oficina	DIRECCIONES_PK
HISTORICO_MOVIMIENTOS	RFC + Fecha_baja	HISTORICO_MOVIMIENTOS_PK
HISTORICO_PROFESORES	RFC + Fecha_baja	HISTORICO_PROFESORES_PK
HORARIOS	RFC+Clave_asignatura+ Num_departamento+ Fecha_inicio+Fecha_fin+ Tipo_movimiento+Categoria	HORARIOS_PK
MOVIMIENTOS	RFC + Núm_departamento + Fecha_inicio + Fecha_fin + Clave_categoria + Clave_ Tipo_Movimiento + Status	MOVIMIENTOS_PK
NACIONALIDADES	Clave_nacionalidad	NACIONALIDAD_PK
PROFESORES	RFC	PROFESORES_PK
TABULADORES	Clave_categoria + Fecha_inicio	TABULADORES_PK
TIPO_MOVIMIENTOS	Clave_Tipo_Movimiento	TIPO_MOVIM_PK

IV.4 ETAPA DE CONSTRUCCION.

A continuación se iniciará la etapa de construcción, en la cuál se presentan ejemplos de diferentes programas que se hicieron, no se presentan todos, dado que serían demasiados, y con el entendimiento de como se realiza cada tipo de programa (reportes, formas, etc.) se transmite la idea de las ventajas que representa el elaborarlos con herramientas y lenguajes de cuarta generación. Tal vez donde mejor se puede apreciar es en los reportes.

Los programas para crear la estructura de la base de datos, los índices, así como para borrar las tablas están elaborados en SQL estandard.

Los reportes están elaborados empleando además del SQL estandard, las extensiones que brinda el SQL*Plus de Oracle.

Los programas de los menús, se elaboraron utilizando SQL*MENU.

Por otra parte, el sistema tiene ayudas en línea, cada campo tiene una ayuda automática, cuando se entra a cada campo se despliega un mensaje donde aclara cual es la información que se espera recibir dicho campo.

Existen otros campos que cuentan con ayudas más amplias, como son: el número de departamento, las claves de las categorías, las claves de las asignaturas, en las que en la parte inferior derecha, aparece el nombre de la tecla de función que activa dicha ayuda, esta tecla es la F8 para activarlo, una vez dentro de las pantallas de ayuda, después de seleccionar la opción deseada, se regresa al programa principal con otra tecla de función, en este caso la F7.

Además de brindar la ayuda en línea, estas pantallas sirven para validar que sólo se introduzcan las claves validas, para mejorar la consistencia de la información.

Otro mecanismo para checar la consistencia de la información es que en cada campo también se valida que se introduzcan el tipo de dato requerido, así como las claves o valores permitidos.

En lo que respecta a la seguridad de la información, el sistema tiene passwords que se deben proporcionar para poder ejecutar diversos movimientos o reportes.

También se pueden tener usuarios con diferentes tipos de privilegios, con lo cual se restringen los accesos a ciertas funciones que sólo puede efectuar la persona responsable del Banco de Horas.

El sistema cuenta con una diversidad de reportes, que son los más utilizados en la división, entre los jefes de departamento, la Secretaría auxiliar y la jefatura de la división, sin embargo si se presenta la necesidad de crear otro reporte, o explotar la información de otra manera, es posible, ya que la estructura de la base de datos se tiene, y los reportes se hacen en SQL, que es un lenguaje de consulta de datos, con una sintaxis muy similar a la manera en la que se quiere realizar las cosas, como es de cuarta generación, basta "decir" que se quiere, y no como se hace, el manejador de la base de datos se encarga de ejecutar la petición, en este sistema, cualquier reporte no planeado, se podrá realizar, refiriéndose al manual de usuario que se encuentra en el anexo A.

Es así que se puede afirmar que se realizó un sistema de calidad, y sobre todo funcional, ya que se satisfacen las expectativas de la Secretaría Auxiliar de la DIE.

PROGRAMA PARA CREAR LA ESTRUCTURA DE LA BASE DE DATOS.

```

CREATE TABLE DEPARTAMENTOS (
    NUM_DEPARTAMENTO          NUMBER (2) NOT NULL,
    NOMBRE_DEPARTAMENTO       CHAR   (20) NOT NULL)
/
CREATE TABLE ASIGNATURAS (
    CLAVE_ASIGNATURA         NUMBER (4) NOT NULL,
    NOMBRE_ASIGNATURA        CHAR   (30) NOT NULL,
    ASI_NUM_DEPARTAMENTO      NUMBER (2) NOT NULL)
/
CREATE TABLE CATEGORIAS (
    CLAVE_CATEGORIA          CHAR   (5) NOT NULL,
    DESCRIPCION_CATEGORIA    CHAR   (20) NOT NULL,
    CODIGO_PROGRAMATICO      CHAR   (23),
    REGISTRO_PLAZA           CHAR   (6))
/
CREATE TABLE NACIONALIDADES (
    CLAVE_NACIONALIDAD       NUMBER (2) NOT NULL,
    NOMBRE_NACIONALIDAD     CHAR   (15) NOT NULL)
/
CREATE TABLE TIPOS_MOVIMIENTOS (
    CLAVE_TIPO_MOVIMIENTO    NUMBER (2) NOT NULL,
    DESCRIPCION_MOVIMIENTO   CHAR   (25) NOT NULL)
/
CREATE TABLE CATALOGO_STATUS (
    STATUS                   CHAR   (1) NOT NULL,
    DESCRIPCION_STATUS       CHAR   (10) NOT NULL)
/
CREATE TABLE PROFESORES (
    RFC                      CHAR   (15) NOT NULL,
    NOMBRE                   CHAR   (40) NOT NULL,
    SEXO                     CHAR   (1) NOT NULL,
    ESTADO_CIVIL             CHAR   (1) NOT NULL,
    ANTIGUEDAD               NUMBER (3),
    PRO_CLAVE_NACIONALIDAD  NUMBER (2))
/
CREATE TABLE DIRECCIONES (
    DIR_RFC                  CHAR   (15) NOT NULL,
    CASA_OFICINA             CHAR   (1) NOT NULL,
    DIRECCION                CHAR   (40),
    CP                       NUMBER (5),
    TELEFONO                 CHAR   (10),
    COLONIA                  CHAR   (20),
    DIR_CLAVE_DELEGACION     NUMBER (2))
/
CREATE TABLE DELEGACIONES (
    CLAVE_DELEGACION         NUMBER (2) NOT NULL,
    DELEGACION               CHAR   (12) NOT NULL)
/

```

```

CREATE TABLE MOVIMIENTOS (
    MOV_RFC CHAR (15) NOT NULL,
    MOV_NUM_DEPARTAMENTO NUMBER (2) NOT NULL,
    FECHA_INICIO DATE NOT NULL,
    FECHA_FIN DATE NOT NULL,
    MOV_CLAVE_TIPO_MOVIMIENTO NUMBER (2) NOT NULL,
    MOV_CLAVE_CATEGORIA CHAR (5) NOT NULL,
    HORAS_TEORIA NUMBER (4,2),
    HORAS_PRACTICA NUMBER (4,2),
    MOV_STATUS CHAR (1) NOT NULL,
    OBSERVACIONES CHAR (40),
    ARTICULO_51 CHAR (1),
    MOV_TAB_FECHA_INICIO DATE,
    SUELDO_NUM NUMBER (8,2),
    SUELDO CHAR (75))

```

```

/
CREATE TABLE HORARIOS (
    HOR_RFC CHAR (15) NOT NULL,
    HOR_CLAVE_ASIGNATURA NUMBER (4) NOT NULL,
    HOR_NUM_DEPARTAMENTO NUMBER (2) NOT NULL,
    HOR_FECHA_INICIO DATE NOT NULL,
    HOR_FECHA_FIN DATE NOT NULL,
    HOR_CLAVE_TIPO_MOVIMIENTO NUMBER (2) NOT NULL,
    HOR_CLAVE_CATEGORIA CHAR (5) NOT NULL,
    GRUPO NUMBER (2),
    LUNM CHAR (11),
    LUNT CHAR (11),
    MARM CHAR (11),
    MART CHAR (11),
    MIEM CHAR (11),
    MIET CHAR (11),
    JUEM CHAR (11),
    JUT CHAR (11),
    VIEM CHAR (11),
    VIET CHAR (11),
    SABM CHAR (11),
    SABT CHAR (11))

```

```

/
CREATE TABLE BANCO_HORAS(
    BAN_NUM_DEPARTAMENTO NUMBER (2) NOT NULL,
    BAN_CLAVE_CATEGORIA CHAR (5) NOT NULL,
    LIMITE_HORAS NUMBER (8,2) NOT NULL,
    HORAS_CONGELADAS NUMBER (6,2),
    HORAS_EJERCIDAS NUMBER (8,2) NOT NULL)

```

```

/
CREATE TABLE TABULADORES(TAB_CLAVE_CATEGORIA CHAR (5) NOT
NULL,
    TAB_FECHA_INICIO DATE NOT NULL,
    TABULADOR_HORA NUMBER (10,2))
/

```



```

CREATE TABLE HISTORICO_PROFESORES(
    HIS_PRO_RFC                CHAR (15) NOT NULL,
    HIS_PRO_FECHA_BAJA        DATE      NOT NULL,
    HIS_PRO_NOMBRE            CHAR (40) NOT NULL,
    HIS_PRO_CLAVE_NACIONALIDAD NUMBER (2) NOT NULL,
    HIS_PRO_SEXO              CHAR (1) NOT NULL,
    HIS_PRO_ESTADO_CIVIL     CHAR (1) NOT NULL,
    HIS_PRO_ANTIGUEDAD       NUMBER (3) NOT NULL,
    DIRECCION_CASA            CHAR (40) NOT NULL,
    HIS_PRO_COLONIA          CHAR (20),
    HIS_PRO_CP                NUMBER (5),
    HIS_PRO_TELEFONO         CHAR (11),
    HIS_PRO_CLAVE_DELEGACION  NUMBER (2))
/

CREATE TABLE HISTORICO_MOVIMIENTOS(
    HIS_MOV_RFC                CHAR (15) NOT NULL,
    HIS_MOV_FECHA_BAJA        DATE      NOT NULL,
    HIS_MOV_NUM_DEPARTAMENTO   NUMBER (2) NOT NULL,
    HIS_MOV_CLAVE_CATEGORIA   CHAR (5) NOT NULL,
    HIS_MOV_HORAS             NUMBER (4,2))
/

REM
REM TABLAS AUXILIARES
REM
CREATE TABLE CAMBIOS ( CAMBIOS CHAR (1))
/

CREATE TABLE FECHAS ( FECHA_1 DATE, FECHA_2 DATE)
/

CREATE TABLE DEPTO (
    NUM_DEPARTAMENTO          NUMBER (2) NOT NULL,
    NOMBRE_DEPARTAMENTO      CHAR (20) NOT NULL)
/

CREATE TABLE TMPHORAS(MOV_CLAVE_TIPO_MOVIMIENTO NUMBER(2) NOT NULL,
    MOV_CLAVE_CATEGORIA     CHAR (5) NOT NULL,
    HORAS_TOTAL             NUMBER(4,2))
/

CREATE TABLE TMPHRSMOV (TMP_RFC CHAR(15) NOT NULL,
    TMP_CLAVE_CATEGORIA     CHAR(5) NOT NULL,
    TMP_HORAS_ACTUALES     NUMBER(4,2) NOT NULL)
/

```

PROGRAMA PARA CREAR LOS INDICES.

```
CREATE UNIQUE INDEX DEPARTAMENTOS_PK ON
DEPARTAMENTOS(NUM_DEPARTAMENTO)
/
CREATE UNIQUE INDEX ASIGNATURAS_PK ON ASIGNATURAS(CLAVE_ASIGNATURA)
/
CREATE UNIQUE INDEX BANCO_HORAS_PK
ON BANCO_HORAS(BAN_NUM_DEPARTAMENTO, BAN_CLAVE_CATEGORIA)
/
CREATE UNIQUE INDEX CATEGORIAS_PK ON CATEGORIAS(CLAVE_CATEGORIA)
/
CREATE UNIQUE INDEX TIPOS_MOVIMIENTO_PK
ON TIPOS_MOVIMIENTOS(CLAVE_TIPO_MOVIMIENTO)
/
CREATE UNIQUE INDEX NACIONALIDADES_PK ON
NACIONALIDADES(CLAVE_NACIONALIDAD)
/
CREATE UNIQUE INDEX DIRECCIONES_PK ON
DIRECCIONES(DIR_RFC, CASA_OFICINA)
/
CREATE UNIQUE INDEX ESTATUS_PK ON CATALOGO_STATUS(STATUS)
/
CREATE INDEX HISTORICO_MOVIMIENTOS_PK
ON HISTORICO_MOVIMIENTOS(HIS_MOV_RFC, HIS_MOV_FECHA_BAJA)
/
CREATE INDEX HISTORICO_PROFESORES_PK
ON HISTORICO_PROFESORES(HIS_PRO_RFC, HIS_PRO_FECHA_BAJA)
/
CREATE UNIQUE INDEX MOVIMIENTOS_PK ON MOVIMIENTOS
(MOV_RFC, MOV_NUM_DEPARTAMENTO, FECHA_INICIO, FECHA_FIN, MOV_CLAVE_TIPO_M
OVIMIENTO,
MOV_CLAVE_CATEGORIA, MOV_STATUS)
/
CREATE UNIQUE INDEX TABULADORES_PK ON TABULADORES
(TAB_CLAVE_CATEGORIA, TAB_FECHA_INICIO)
/
CREATE UNIQUE INDEX PROFESORES_PK ON PROFESORES (RFC)
/
CREATE UNIQUE INDEX DELEGACIONES_PK ON DELEGACIONES
(CLAVE_DELEGACION)
/
CREATE UNIQUE INDEX HORARIOS_PK ON HORARIOS (HOR_RFC,
HOR_CLAVE_ASIGNATURA, HOR_NUM_DEPARTAMENTO, HOR_FECHA_INICIO,
HOR_FECHA_FIN, HOR_CLAVE_TIPO_MOVIMIENTO,
HOR_CLAVE_CATEGORIA)
/
```

PROGRAMA PARA BORRAR LAS TABLAS

```
drop table ASIGNATURAS ;
drop table BANCO_HORAS ;
drop table CATALOGO_STATUS ;
drop table CATEGORIAS ;
drop table CAMBIOS;
drop table DEPARTAMENTOS ;
drop table DELEGACIONES;
drop table HORARIOS;
drop table DIRECCIONES ;
drop table HISTORICO_MOVIMIENTOS ;
drop table HISTORICO_PROFESORES ;
drop table MOVIMIENTOS ;
drop table NACIONALIDADES ;
drop table PROFESORES ;
drop table TABULADORES ;
drop table TIPOS_MOVIMIENTOS ;
drop table FECHAS;
drop table DEPTO;
```

**EJEMPLO DE REPORTE
REPORTE PARA OBTENER EL CATALOGO DE DEPARTAMENTOS DE LA DIVISION.**

```
REM REPORTE DEL CATALOGO DE DEPARTAMENTOS DE LA DIVISION.
REM
REM ESTE REPORTE NOS MUESTRA LOS DEPARTAMENTOS QUE EXISTEN EN LA
REM DIVISION CON SU RESPECTIVA CLAVE.
REM
REM SE DECLARAN LOS ENCABEZADOS Y VARIABLES A UTILIZAR EN EL REPORTE
COLUMN NEWDATE NEW_VALUE FECHAH
COLUMN TIEMPO NEW_VALUE HORAH
COLUMN LPAD(TO_CHAR(NUM_DEPARTAMENTO),18) HEADING "CLAVE
DEPARTAMENTO" FORMAT A40 JUSTIFY CENTER
COLUMN NOMBRE_DEPARTAMENTO HEADING DEPARTAMENTO FORMAT A40
REM
REM SE DECLARA EL TAMAÑO DE PAGINA Y DE LA LINEA
REM
SET PAGESIZE 66
SET LINESIZE 80
SET SPACE 0
REM
REM SE ASIGNAN LOS VALORES A LAS VARIABLES AUXILIARES
REM
SET ECHO OFF
SET FEED OFF
SET HEAD OFF
SELECT TO_CHAR(SYSDATE, 'DD/MON/YY')
      HOY
FROM DUAL;
REM
SELECT TO_CHAR(SYSDATE, 'HH24:MI')
      TIEMPO
FROM DUAL;
REM
REM SE DA EL NOMBRE AL ARCHIVO DEL REPORTE
SPOOL REPORTE.LIS
REM SE DA EL FORMATEO NECESARIO PARA EL REPORTE
REM
SET HEAD ON
TTITLE LEFT 'HOJA:' FORMAT 99 SQL.PNO CENTER 'U.N.A.M.' RIGHT FORMAT
A8 fechah -
SKIP 1 CENTER 'FACULTAD DE INGENIERIA' RIGHT FORMAT A5 horah -
SKIP 1 CENTER 'DIVISION DE INGENIERIA ELECTRICA' -
CENTER 'CATALOGO DE DEPARTAMENTOS' SKIP 3
REM
SELECT LPAD(TO_CHAR(NUM_DEPARTAMENTO),18), NOMBRE_DEPARTAMENTO
FROM DEPARTAMENTOS
ORDER BY NUM_DEPARTAMENTO;
SPOOL OFF
TTITLE OFF
CLEAR BREAKS;
SET HEAD ON;
SET FEED ON;
SET ECHO ON;
EXIT
```

```

EJEMPLO DE REPORTE
  REPORTE PARA EMITIR UN LISTADO GENERAL.
REM REPORTE GENERAL DE PROFESORES DE LA DIVISION.
REM ELABORO MARICELA CASTAÑEDA PERDOMO      27/JULIO/91
REM MODIFICO VICTOR GONZALEZ CASTRO        01/OCTUBRE/93
REM
REM ESTE REPORTE NOS MUESTRA POR ORDEN ALFABETICO A TODOS LOS
REM PROFESORES
REM QUE LABORAN EN LA DIVISION. A LA DERECHA APARECERA LAS HORAS QUE
REM TIENE, LA CATEGORIA Y EL DEPARTAMENTO AL CUAL PERTENECE.
REM
REM SE DECLARAN LOS ENCABEZADOS Y VARIABLES A UTILIZAR EN EL REPORTE
REM
COLUMN hoy new_value fechah
COLUMN tiempo new_value horah
COLUMN DESCRIPCION_CATEGORIA HEADING CATEGORIA   FORMAT A20
COLUMN NVL(HORAS_TEORIA+HORAS_PRACTICA,0) HEADING HORAS FORMAT 99.99
COLUMN MOV_NUM_DEPARTAMENTO HEADING DEPTO        FORMAT 99
REM
REM SE DECLARA EL TAMAÑO DE PAGINA Y DE LA LINEA
REM
SET PAGESIZE 66;
SET LINESIZE 80;
SET SPACE 2;
REM
REM SE ASIGNAN LOS VALORES A LAS VARIABLES AUXILIARES
REM
SELECT TO_CHAR(sysdate,'DD/MM/YY') hoy,
       TO_CHAR(sysdate,'hh24:mi') tiempo
FROM DUAL;
SET ECHO OFF
REM SET FEED OFF
SET HEAD OFF
REM
TTITLE LEFT 'HOJA:' FORMAT 99 SQL.PNO CENTER 'U.N.A.M.' RIGHT FORMAT
AS fechah -
SKIP 1 CENTER 'FACULTAD DE INGENIERIA' RIGHT FORMAT A5 horah -
SKIP 1 CENTER 'DIVISION DE INGENIERIA ELECTRICA' -
SKIP 1 CENTER 'REPORTE GENERAL DE PROFESORES' SKIP 3
REM
REM SE DA EL NOMBRE AL ARCHIVO DEL REPORTE
SPOOL REPORTE.LIS
REM
REM SE DA EL FORMATEO NECESARIO PARA EL REPORTE
REM
SET HEAD ON
REM

```

```
SET PAGESIZE 66;
SET LINESIZE 80;
BREAK ON NOMBRE ON RFC
SELECT RFC,NOMBRE,CLAVE_CATEGORIA,
NVL(HORAS_TEORIA+HORAS_PRACTICA,0),MOV_NUM_DEPARTAMENTO
FROM PROFESORES,MOVIMIENTOS,CATEGORIAS
WHERE RFC=MOV_RFC
AND CLAVE_CATEGORIA=MOV_CLAVE_CATEGORIA
AND (MOV_STATUS='E' OR MOV_STATUS='P')
ORDER BY NOMBRE,CLAVE_CATEGORIA,MOV_NUM_DEPARTAMENTO;
SPOOL OFF
TTITLE OFF
CLEAR BREAKS;
SET HEAD ON
SET FEED ON
SET ECHO ON
EXIT
```

Reportes Generales

División de Ingeniería Eléctrica

- 1 General (80)
- 2 General por Categorías (80)
- 3 General por Categorías por Departamento (132)
- 4 Profesores y Tecnicos Academicos (80)
- 5 Profesores de Carrera (80)
- 6 Tecnicos Academicos (132)
- 7 Profesores de Carrera por Departamento (132)
- 8 Tecnicos Academicos (132)
- 9 Siguiente Menú
- 10 Menú Anterior
- 11 Salir

Make your choice: 1

Escoja su opción:

v Fri Dec 05 13:27:03 1995 OSC DBG Replace BANCO (REPGEN)

Ejemplo de programa para efectuar un movimiento. Alta por Nuevo Ingreso.

En las siguientes hojas se muestra un programa típico, el cual sirve para dar una alta por nuevo ingreso.

Este programa está elaborado en SQL*Forms, que es la herramienta de programación de cuarta generación utilizado.

Como se puede observar, sólo se muestran los llamados "triggers" de la aplicación que conforman el programa.

ORACLE-MYENT
Fecha:

Aplicacion: myingre
Etiquet: ASIGNATURAS

Triggers que conforman a la aplicacion
Horas:

Clave	Nombre	Nº caso	Etiqueta	AS	Etio. éxito	Etio. falla	Texto del Trigger
CLAVE_ASIGNA	KEY-MYFLE	1		Y	N	SIGUE	FIN SELECT NOMBRE ASIGNATURA INTO :ASIGNATURAS,NOMBRE ASIGNATURA FROM ASIGNATURAS WHERE CLAVE ASIGNATURA = :ASIGNATURAS.CLAVE ASIGNA TURA AND ASI_NUM DEPARTAMENTO = :NUM DEPARTAMENTO Mensaje err: La asignatura no corresponde al departamento !! FIN
		2	SIGUE	Y	N		DEBERACRO GOFLD HORARIOS,GRUPO;
	KEY-PRVFLO	1		Y	N		DEBERACRO GOFLD MOVIMIENTOS,SUELDO;
Nombre Etiqueta	KEY-EXEOPY	1		Y	N		OCOPY :DEPARTAMENTOS.NUM DEPARTAMENTO GLOBAL,NUM_D EPARTAMENTO OCOPY '1' GLOBAL,TIPO_CALL DEBERACRO CALLORY ASIGNATURAS; OCOPY GLOBAL,CLAVE ASIGNATURA :ASIGNATURAS.CLAVE_A SIGNATURA
		3		Y	N	
		4		Y	N	
	KEY-MYTRC	1		Y	N		DEBERACRO MYTRC:GOBLR HORARIOS;MYTRC:GOBLR ASIGNA TURAS;
						
	KEY-PRVREC	1		Y	N		DEBERACRO PRVREC: GOBLR HORARIOS; PRVREC: GOBLR AS IGNATURAS;
						
	PRE-QUERY	1		Y	N		OCOPY :NOV CLAVE ASIGNATURA INTO :CLAVE ASIGNATURA

UNALCZ-MEXICO
Fecha:

Aplicacion: nvoingre
Bloque: CATEGORIAS

Triggers que conforman a la aplicacion
Notas:

Ca d o n	Nombre	# paso	Etiqueta	A' A'	Etig. exito	Etig. falla	Texto del Trigger
CLAVE_CATEG	KEY-PRVFLD	1		Y	N		<pre> EXECRACRO GOFLD PROFESORES,ESTADO_CIVIL; </pre>
	POST-CHANGE	1		Y	N		<pre> SELECT DESCRIPCION_CATEGORIA,SUBSTRICLAVE_CATEGOR1 4,1,2) INTO :DESCRIPCION_CATEGORIA,CLAVE_CATEGOR1 FROM CATEGORIAS WHERE CLAVE_CATEGORIA = :ICLAVE_CATEGORIA </pre>
Nivel Bloque	KEY-EKREQY	1		Y	N		<pre> EXECRACRO CALLORY CATEGORIAS; </pre>
		2		Y	N		<pre> OCOPY GLOBAL.CLAVE_CATEGORIA :ICLAVE_CATEGORIA </pre>
	KEY-NXTFLD	1		Y	N		<pre> SELECT '1' FROM BANCO HORAS WHERE BAN.CLAVE_CATEGORIA = :CLAVE_CATEGORIA AND BAN.MUN_DEPARTAMENTO = :MUN_DEPARTAMENTO Mensaje err: El departamento no cuenta con esta categoria. !!! EXECRACRO GOBLK MOVIMIENTOS;CLAREC;PRVBLK;NXTBLK;E XEQRY;GOFLD HORAS_TEGRIAC; </pre>
		2		Y	N		<pre> SELECT CLAVE_ASIGNATURA,NOMBRE_ASIGNATURA INTO :ICLAVE_ASIGNATURA,:NOMBRE_ASIGNATURA FROM ASIGNATURAS WHERE :HOR_CLAVE_ASIGNATURA = CLAVE_ASIGNATURA EXECRACRO GOFLD HORAS_TEGRIAC; </pre>
		3		Y	N		<pre> SELECT CLAVE_ASIGNATURA,NOMBRE_ASIGNATURA INTO :ICLAVE_ASIGNATURA,:NOMBRE_ASIGNATURA FROM ASIGNATURAS WHERE :HOR_CLAVE_ASIGNATURA = CLAVE_ASIGNATURA EXECRACRO GOFLD HORAS_TEGRIAC; </pre>
		4		Y	N		<pre> EXECRACRO GOFLD HORAS_TEGRIAC; </pre>
=RFC	KEY-NXTFLD	1		N	N	TRAE_DIRECCI TERMINAR	<pre> SELECT RFC,NOMBRE,PRO_CLAVE_NACIONALIDAD,SEYO,ESTA DO_CIVIL,:ANTIGUEDAD INTO :RFC,:NOMBRE,:PRO_CLAVE_NACIONALIDAD,:SEYO,:E STADO_CIVIL,:ANTIGUEDAD FROM PROFESORES WHERE RFC = :RFC Mensaje err: El profesor no esta registrado. TERMINAR EXECRACRO NXTBLK;EXEQRY; Mensaje err: El profesor no tiene direccion domicilio EXECRACRO NXTBLK;EXEQRY; Mensaje err: El profesor no tiene direccion de oficina. EXECRACRO NXTBLK;EXEQRY; EXECRACRO GOFLD PROFESORES,NOMBRE; </pre>
		2	TRAE_DIRECCI	N	N		<pre> EXECRACRO NXTBLK;EXEQRY; EXECRACRO NXTBLK;EXEQRY; EXECRACRO GOFLD PROFESORES,NOMBRE; </pre>
		3		N	N		<pre> EXECRACRO NXTBLK;EXEQRY; EXECRACRO GOFLD PROFESORES,NOMBRE; </pre>
		4		N	N		<pre> EXECRACRO NXTBLK;EXEQRY; EXECRACRO GOFLD PROFESORES,NOMBRE; </pre>
		5	TERMINAR	Y	N		<pre> EXECRACRO GOFLD PROFESORES,NOMBRE; </pre>
	KEY-PRVFLD	1		Y	N		<pre> EXECRACRO GOFLD DEPARTAMENTOS,MUN_DEPARTAMENTO; </pre>

Columna	Nombre	N caso	Etiqueta	A/R	Etio. éxito	Etio. falla	Texto del Trigger
CLAVE DELEGA	EJ-EXEOP	1		Y	N		<pre> @COPY '1' GLOBAL.TIPO_CALL ===== @EXEMACRO CALLORY DELEGACIONES; ===== @COPY :GLOBAL,CLAVE DELEGACION ;CLAVE DELEGACION @COPY GLOBAL.CLAVE DELEGACION ;CLAVE DELEGACION ===== @COPY GLOBAL.DELEGACION ;DELEGACION @COPY GLOBAL.DELEGACION ;DELEGACION ===== </pre>
		2		Y	N		
		3		Y	N		
		4		Y	N		
KEY-NVTFLD		1		Y	N		<pre> @EXEMACRO CASE ;DIRECCIONES, DIRECCION IS SELECT CLAVE DELEGACION, DELEGACION INTO :CLAVE DELEGACION, ;DELEGACION WHEN THEN GOSTEP AVANZA; FROM DELEGACIONES; WHEN OTHERS THEN ENDSTEP; END CASE; WHERE CLAVE_DELEGACION = :CLAVE_DELEGACION Mensaje err: Delegation no valida. ===== @COPY :CLAVE_DELEGACION ;DIRECCIONES,DIR_CLAVE_DE ELEGACION ===== SELECT CLAVE DELEGACION, DELEGACION INTO :CLAVE_DELEGACION, ;DELEGACION FROM DELEGACIONES; WHERE CLAVE_DELEGACION = :CLAVE_DELEGACION Mensaje err: Delegation no valida. ===== @EXEMACRO MATBLN; @COPY :CLAVE_DELEGACION ;DIRECCIONES,DIR_CLAVE_D ELEGACION ===== @EXEMACRO GOFLD NACIONALIDADES, NOMBRE_NACIONALIDAD ; ===== </pre>
		2		Y	N		
		3	AVANZA	Y	N		
		4	AVANZA	Y	N		
KEY-PRVFLD		1		Y	N		<pre> @EXEMACRO GOFLD DIRECCIONES, TELEFONO; @EXEMACRO GOFLD DIRECCIONES, TELEFONO; ===== </pre>

Fecha:

Aplicación: hvoingre
Bloque: DEPARTAMENTO

Horas:

C a m p o	Nombre	# paso	Etiqueta	A? R?	Etio. éxito	Etio. falla	Texto del Trigger
NUM_DEPARTAM	KEY-EVESQRY	1		Y	N		@COPY '1' GLOBAL.TIPO_CALL; ***** @EXENACRO CALLOPY DEPARTAMENTOS; ***** @COPY GLOBAL.NUM_DEPARTAMENTO :NUM_DEPARTAMENTO *****
		2		Y	N		
		3		Y	N		
	KEY-NXTFLD	1		Y	N		@EXENACRO NXTBLK; *****
	POST-CHANGE	1		Y	N		SELECT NUM_DEPARTAMENTO, NOMBRE_DEPARTAMENTO INTO :DEPARTAMENTOS, NUM_DEPARTAMENTO, :DEPARTAMENTO S, NOMBRE_DEPARTAMENTO FROM DEPARTAMENTOS WHERE NUM_DEPARTAMENTO = :DEPARTAMENTOS, NUM_DEPART AMENTO
							Mensaje err: Numero de departamento invalido. FB --) AYUDA
		2		Y	N		***** SELECT :NUM_DEPARTAMENTO, :NOMBRE_DEPARTAMENTO INTO :NUM_DEPARTAMENTO2, :NOMBRE_DEPARTAMENTO2 FROM SYSTEM.DUAL *****

C a m p o	Nombre	# pasn	Etiqueta	A ^m R ⁿ	Etio. éxito	Etio. falla	Texto del Trigger
DIRECCION	KEY-PRVFLD	1		Y	N		@EXENACRO GOFLD PROFESORES.ANTIGUEDAD;
Nivel Bloque	KEY-NXTFLD	1		Y	N		@EXENACRO NXTFLD;
	KEY-PRVFLD	1		Y	N		@EXENACRO PRVFLD;
	PRE-QUERY	1		Y	N		@COPY iPROFESORES.RFC iDIRECCIONES.DIR RFC
TELEFONO	KEY-NXTFLD	1		Y	N		@EXENACRO NXTBLK;

UNACLE-MAILU
Fecha:

Aplicación: hvoingre
Bloque: DIRECCIONES1

Triggers que conforman a la aplicación:
Hora:

Cam po	Nombre	Nº paso	Etiqueta	A' R'	Etia. éxito	Etia. falla	Texto del Trigger
DIRECCION	KEY-PRVFLD	1		Y	N		WEXENACRO GOFLD DELEGACIONES.CLAVE DELEGACION;
Nivel Bloque	AEY-NXTFLD	1		Y	N		WEXENACRO NXTFLD;
	KEY-PRVFLD	1		Y	N		WEXENACRO PRVFLD;
	PRE-QUERY	1		Y	N		WCOPY :PROFESORES.RFC :DIRECCIONES1.DIA.RFC
TELEFONO	KEY-NXTFLD	1		Y	N		WEXENACRO GOFLE DELEGACIONES.CLAVE DELEGACION;

Código	Nombre	Nº paso	Etiqueta	A: R:	Etq. éxito	Etq. falla	Texto del Trigger
GRUPO	KEY-NXTFLD	1		Y	N		SELECT :NFC,:CLAVE_ASIGNATURA,:NUM_DEPARTAMENTO, :FECHA_INICIO, :FECHA_FIN,:CLAVE_CATEGORIA INTO :HOR_RFC,:HOR_CLAVE_ASIGNATURA,:HOR_NUM_DEPAR TAMENTO,:HOR_FECHA_INICIO, :HOR_FECHA_FIN,:HOR_CLAVE_CATEGORIA FROM SYSTEM.DUAL ***** @COPY GLOBAL.TIPO_MOVIMIENTO :HOR_CLAVE_TIPO_MOVIM IENTO ***** @EXENACRO GOFLD HORARIOS,LUNH; *****
		2		Y	N		***** @EXENACRO GOFLD ASIGNATURAS,CLAVE_ASIGNATURA; *****
		3		Y	N		***** @EXENACRO GOFLD HORARIOS,LUNH; *****
LUNH	KEY-PRVFLD	1		Y	N		***** @COPY GLOBAL.TIPO_MOVIMIENTO :HOR_CLAVE_TIPO_MOVIM IENTO ***** @EXENACRO GOFLD HORARIOS,GRUPO; *****
		2		Y	N		***** @EXENACRO NXTFLD; *****
Nivel Bloque	KEY-NXTFLD	1		Y	N		***** @EXENACRO NXTREC;@DBLK ASIGNATURAS; NXTREC; *****
	KEY-PRVFLD	1		Y	N		***** @EXENACRO PRVFLD; *****
	KEY-PRVREC	1		Y	N		***** @EXENACRO PRVREC;@DBLK ASIGNATURAS;PRVREC; Mensaje en: FALLO EL PRVREC *****
	KEY-NXTFLD	1		Y	N		***** @COPY GLOBAL.TIPO_MOVIMIENTO INTO :HOR_CLAVE_TIPO_ MOVIMIENTO ***** @EXENACRO DOFLD MOVIMIENTOS.OBSERVACIONES; *****
SABT	KEY-NXTFLD	1		Y	N		***** @EXENACRO DOFLD MOVIMIENTOS.OBSERVACIONES; *****
		2		Y	N		***** @EXENACRO DOFLD MOVIMIENTOS.OBSERVACIONES; *****

Cam po	Nombre	Nº paso	Etiqueta	A? R?	Etig. éxito	Etig. falla	Texto del Trigger		
ARTICULO_51	KEY-NXTFLD	1		N	N		<pre> DETERMINAR CASE :ARTICULO_51 IS WHEN 'S' THEN GOSTEP 'CHECA'; WHEN 'M' THEN GOSTEP 'AVANZA'; WHEN OTHERS THEN ENDTRIG MESSAGE 'Los valores val idos son S o M.'; END CASE; ===== SELECT 'M' INTO :ARTICULO_51 FROM SYSTEM.DUAL WHERE (:CATEGORIAS.CLAVE_CATEGORIA LIKE '01' OR :CATEGORIAS.CLAVE_CATEGORIA LIKE '02' OR :CATEGORIAS.CLAVE_CATEGORIA LIKE '03' OR :CATEGORIAS.CLAVE_CATEGORIA LIKE '04' OR :CATEGORIAS.CLAVE_CATEGORIA LIKE '05' OR :CATEGORIAS.CLAVE_CATEGORIA LIKE '06'); ===== SELECT TABULADOR_HORA INTO :SUELDO_HUR FROM TABULADORES WHERE TAB_FECHA_INICIO=:TAB_FECHA_INICIO AND TAB_CLAVE_CATEGORIA=:CLAVE_CATEGORIA AND (:CLAVE_CATEGORIA LIKE '07' OR :CLAVE_CATEGORIA LIKE '08'); ===== SELECT (NVL(:HORAS_TEORIA2,0)+NVL(:HORAS_PRACTICA2 ,0))*TABULADOR_HORA INTO :SUELDO_HUR FROM TABULADORES WHERE TAB_FECHA_INICIO=:TAB_FECHA_INICIO AND TAB_CLAVE_CATEGORIA=:CLAVE_CATEGORIA ===== DETERMINAR GOFLO SUELDO; ===== </pre>		
		2	CHECA	N	N	AVANZA			
		3	AVANZA	N	N	FIN			
		4				Y	N		
		5	FIN			Y	N		
CAMBIOS	KEY-NXTFLD	1		Y	N		<pre> DETERMINAR CASE (:DIRECCIONES1.DIRECCION IS WHEN 'S' THEN GOSTEP 'GRABA'; WHEN OTHERS THEN GOSTEP 'DIRECCION'; END CASE; ===== BCOPY :RFC INTO :DIRECCIONES1.DIR_RFC ===== BCOPY '0' INTO :DIRECCIONES1.CASA_OFICINA ===== DETERMINAR CASE (:CAMBIOS IS WHEN 'S' THEN COMMIT;CLRFRR;GOFLO DEPARTAMENTOS.N UN_DEPARTAMENTO; WHEN 'M' THEN CLRFRR;GOFLO DEPARTAMENTOS.NUN_DEPA RTAMENTO; END CASE; ===== </pre>		
		2	DIRECCION	Y	N				
		3				Y	N		
		4	GRABA			Y	N		
FECHA_FIN	KEY-NXTFLD	1		Y	N		<pre> SELECT '1' FROM SYSTEM.DUAL WHERE MONTHS_BETWEEN (:FECHA_FIN,:FECHA_INICIO-1) <= 12 AND (:FECHA_FIN > :FECHA_INICIO Mensaje err: Fecha final no valida. !!) ===== DETERMINAR NXTFLD; ===== </pre>		
		2			Y	N			
FECHA_INICIO	KEY-NXTFLD	1		Y	N		<pre> SELECT ADD_MONTHS(:FECHA_INICIO-1,12) INTO :FECHA_FIN FROM SYSTEM.DUAL ===== SELECT MAX(:TAB_FECHA_INICIO) INTO :TAB_FECHA_INICIO </pre>		
		2		Y	N				

C a d e	Nombre	N caso	Etiqueta	A? R?	Etio. exito	Etio. falla	Texto del Trigger	
FECHA_INICIO	KEY-NXTFLD	2		Y	N		FROM TABULADORES WHERE TAB_CLAVE CATEGORIA=ICLAVE CATEGORIA AND TAB_FECHA_INICIO (<= FECHA_INICIO ***** DEVENACRO GOBLK BANCO_HORAS:EGREGY:GDFLD FECHA_FIN !	
		3		Y	N		***** DEVENACRO NXTFLD; *****	
		4		Y	N		***** DEVENACRO NXTFLD; *****	
		POST-CHANGE	1		Y	N		SELECT ADD MONTHS(1:FECHA_INICIO-1,12) INTO 1:FECHA_FIN FROM SYSTEM.DUAL *****
HORA_PRACTI	KEY-NXTFLD	1		Y	N		DEVENACRO CASE 1:CLAVE CATEGORIA1 IS WHEN '01' THEN GOSTEP CHECA0; WHEN '02' THEN GOSTEP CHECA0; WHEN '03' THEN GOSTEP CHECA15; WHEN '04' THEN GOSTEP CHECA15; WHEN '05' THEN GOSTEP CHECA15; WHEN '06' THEN GOSTEP CHECA15; WHEN '07' THEN GOSTEP CHECA15; WHEN '08' THEN GOSTEP CHECA15; WHEN OTHERS THEN ENDTRIG MESSAGE 'Categoria inval ida'; END CASE; ***** SELECT NVL(SUM(HORAS TEORIA),0)+:HORAS_TEGRIA2 INTO :TOTAL HORAS_TEGRIA FROM MOVIMIENTOS WHERE MOV_RFC = :RFC ***** Mensaje err: Para esta categoria no se permiten horas de teori	
		2	CHECA0	Y	N		***** SELECT NVL(SUM(HORAS TEORIA),0)+:HORAS_TEGRIA2 INTO :TOTAL HORAS_TEGRIA FROM MOVIMIENTOS WHERE MOV_RFC = :RFC *****	
		3		Y	N	CHECA40		SELECT '1' FROM SYSTEM.DUAL WHERE :TOTAL HORAS_TEGRIA = 0 Mensaje err: Para esta categoria no se permiten horas de teori
		4	CHECA15	Y	N			***** SELECT NVL(SUM(HORAS TEORIA),0)+:HORAS_TEGRIA2 INTO :TOTAL HORAS_TEGRIA FROM MOVIMIENTOS WHERE MOV_RFC = :RFC *****
		5		Y	N	CHECA_PRACTI		SELECT '1' FROM SYSTEM.DUAL WHERE :TOTAL HORAS_TEGRIA (<= 15 Mensaje err: Se exceden las horas de teoria MAXIMO 15 !!! *****
		6	CHECA40	Y	N			SELECT NVL(SUM(HORAS PRACTICA),0)+:HORAS_PRACTICA2 INTO :TOTAL HORAS_PRACTICA FROM MOVIMIENTOS WHERE MOV_RFC = :RFC *****
		7		Y	N	CHECA_LIMITE		SELECT '1' FROM SYSTEM.DUAL WHERE :TOTAL HORAS_PRACTICA (<= 40 Mensaje err: Se excede el numero de horas practicas MAXIMO 40 / /) *****
		8	CHECA_PRACTI	N	N			DEVENACRO CASE 1:CLAVE CATEGORIA1 IS WHEN '03' THEN GOSTEP CHECA25; WHEN '04' THEN GOSTEP CHECA25; WHEN '05' THEN GOSTEP CHECA25; WHEN '06' THEN GOSTEP CHECA25; WHEN '07' THEN GOSTEP CHECA40; WHEN '08' THEN GOSTEP CHECA40; WHEN OTHERS THEN ENDTRIG MESSAGE 'Categoria inval ida'; END CASE; *****
		9	CHECA25	Y	N			SELECT NVL(SUM(HORAS PRACTICA),0)+:HORAS_PRACTICA2 *****

Fecha:

Aplicación: novingro
Bloque: MOVIMIENTOS

Triggers que conforman a la aplicación
Horas

Cam po	Nombre	Caso	Etiqueta	A? R?	Etig. éxito	Etig. falla	Texto del Trigger	
HORAS_PRACTI	KEY-NXTFLD	9	CHECA25	Y	N		<pre> INTO :TOTAL HORAS PRACTICA FROM MOVIMIENTOS WHERE NOV RFC = :RFC ===== SELECT '1' FROM SYSTEM.DUAL WHERE :TOTAL HORAS PRACTICA <=40 Mensaje spp: Se excede el número de horas de PRACTICA !!! ===== 11 CHECA48 N N CHECA_LIMITE MAL_PRACTICA SELECT NVL(SUM(HORAS PRACTICA),0)+:HORAS_PRACTICA2 INTO :TOTAL HORAS PRACTICA FROM MOVIMIENTOS WHERE NOV RFC = :RFC AND I (= 3) AND ((:TOTAL HORAS TEORIA = 20) OR(:TOTAL HORAS TEORIA = 40) OR(:TOTAL HORAS TEORIA =48)) Mensaje spp: FALLO EL CHECA 48 MAL_PRACTICA SELECT NVL(:HORAS TEORIA2,0)+NVL(:HORAS_PRACTICA2, 0)+:HORAS EJERCIDAS+ NVL(:HORAS CONGELADAS,0) INTO :LIMITE FROM BANCO HORAS WHERE :CATEGORIAS.CLAVE_CATEGORIA = BAN_CLAVE_CATE GORIA AND :DEPARTAMENTOS.NUM_DEPARTAMENTO = BAN_NUM_DEPA RTAMENTO Mensaje spp: FALLO EL CHECA LIMITE ===== 13 Y N SELECT '1' FROM BANCO HORAS WHERE :LIMITE <= LIMITE HORAS AND :CLAVE_CATEGORIA = BAN_CLAVE_CATEGORIA AND :NUM_DEPARTAMENTO = BAN_NUM_DEPARTAMENTO Mensaje spp: Se excede el límite de horas para esa categoría en el departamento. ===== 14 Y N SELECT NVL(:TOTAL_HORAS_TEDRIA,0)+NVL(:TOTAL_HORAS PRACTICA,0) INTO :HORAS TOTALES FROM SYSTEM.DUAL Mensaje spp: FALLO LA SUMA ===== 15 Y N DEVENACRO GOFLD CATEGORIAS.CLAVE_CATEGORIA; DEVENACRO GOFLD NOV STATUS; ===== </pre>	
		10		Y	N	CHECA_LIMITE		
		11		N	N	CHECA_LIMITE	MAL_PRACTICA	
		12		Y	N			
		13		Y	N			
HORAS_TEORIA	KEY-PRVFLD	1		Y	N		<pre> DEVENACRO GOFLD CATEGORIAS.CLAVE_CATEGORIA; DEVENACRO GOFLD NOV STATUS; ===== </pre>	
		2		Y	N		<pre> BCOPY '1' GLOBAL.TIPO_CALL ===== DEVENACRO CALLQAY ESTATUS; ===== BCOPY GLOBAL.STATUS :NOV_STATUS ===== </pre>	
		3		Y	N		<pre> DEVENACRO GOFLD CATEGORIAS.CLAVE_CATEGORIA; ===== </pre>	
Novel Bloque	KEY-NXTREC	1		Y	N		<pre> DEVENACRO NXTREC; ===== DEVENACRO CASE :NOV RFC IS WHEN '' THEN PRVREC;ENDTRIG MESSAGE 'E: el último movimiento del profesor'; ===== WHEN OTHERS THEN NULL; END CASE; ===== </pre>	
		2		Y	N			

Bloque: MOVIMIENTOS

Clave	Nombre	# caso	Etiqueta	A' P'	Etio. éxito	Etio. falla	Texto del Trigger
Movil. Bloque	KEY-MVTREC	2		Y	N	
	KEY-PRVFLD	1		Y	N		GENERACRO PRVFLD;
	KEY-PRVREC	1		Y	N		GENERACRO PRVREC;
	PRE-QUERY	1		Y	N		BCOPY :CLAVE_CATEGORIA INTO :NOV_CLAVE_CATEGORIA BCOPY :RFC INTO :NOV_RFC BCOPY :NUM_DEPARTAMENTO INTO :NOV_NUM_DEPARTAMENTO
OBSERVACIONE	ACT_FECHA	1		Y	N		UPDATE MOVIMIENTOS SET FECHA_INICIO = :FECHA_INICI O,FECHA_FIN = :FECHA_FIN WHERE NOV_RFC = :RFC AND NOV_CLAVE_CATEGORIA = :CLAVE_CATEGORIA AND NOV_NUM_DEPARTAMENTO = :NUM_DEPARTAMENTO
	KEY-MVTREC	1		Y	N		GENERACRO MVTREC;
	KEY-PRVFLD	1		Y	N		GENERACRO GOFLD HORARIOS.SMT;
	PRE-FIELD	1		N	N		SELECT 'ARTICULO 31' INTO :OBSERVACIONES FROM SYSTEM.DUAL WHERE :ARTICULO_31 = 'S'
sueldo	KEY-MVTFLD	1		Y	N		GENERACRO CASE :SUELDO IS WHEN '' THEN ENDTAB; WHEN OTHERS THEN GOFLD ASIGNATURAS.CLAVE_ASIGNATU RA; END CASE;
	KEY-PRVFLD	1		Y	N		GENERACRO GOFLD MOVIMIENTOS.ARTICULO_31;
	PRE-FIELD	1		Y	N		select decode(substr(substritto_char:sueldo_num,'00000000 0.00'),2,3),1,1) '1',decode(substr(itto_char:sueldo_num,'00000000. 00'),2,3),100,'CIEN' '1','CIENTO '), '2','DOSCIENTOS ', '3','TRESCIENTOS ', '4','CUATRO CIENTOS ', '5','QUINIENTOS ', '6','SEISCIENTOS ', '7','SETECI ENTOS ', '8','OCHOCIENTOS ', '9','NOVECIENTOS ', null) into sueldo from system.dual 2 sel2 Y N select sueldo ' ' decode(substr(substritto_char:sueldo_num,'00000000 0.00'),3,2),1,1) '3',decode(substr(itto_char:sueldo_num,'00000000.0 0'),3,2),30,'TREINTA' 'TREINTA Y ') '4',decode(substr(itto_char:sueldo_num,'00000000. 00'),3,2),40,'CUARENTA'

- llsatechooff

Campos Nombre # paso Etiqueta A? R? Etig. exito Etig. falla Texto del Trigger

sueldo PRE-FIELD 2 sel2 Y N

```

.'CUARENTA Y ')
'5',decode(substr(to_char(sueldo_num,'00000000.00'),3,2), '50')
.'CINCUENTA','CINCUENTA Y ')
'6',decode(substr(to_char(sueldo_num,'00000000.00'),3,2), '60','SESENTA')
.'SESENTA Y ')
'7',decode(substr(to_char(sueldo_num,'00000000.00'),3,2), '70','SETENTA')
.'SETENTA Y ')
'8',decode(substr(to_char(sueldo_num,'00000000.00'),3,2), '80','OCENTA')
.'OCENTA Y ')
'9',decode(substr(to_char(sueldo_num,'00000000.00'),3,2), '90','NOVENTA')
.'NOVENTA Y ')
'1',decode(substr(to_char(sueldo_num,'00000000.00'),3,2), '10','DIEZ')
.'11','ONCE', '12','DOCE', '13','TRECE', '14','CATORCE', '15','QUINCE'
.'16','DIECISEIS', '17','DIECISIETE', '18','DIECIOCHO', '19','DIECINUEVE')
.'2',decode(substr(to_char(sueldo_num,'00000000.00'),3,2), '20','VEINTE')
.'21','VEINTIUN', '22','VEINTIDOS', '23','VEINTITRES'
.'24','VEINTICUATRO', '25','VEINTICINCO', '26','VEINTISEIS'
.'27','VEINTISIETE', '28','VEINTIOCHO', '29','VEINTINUEVE', null)
into sueldo
from
system.dual
-----
select sueldo || ' ' ||
decode(substr(substr(to_char(sueldo_num,'00000000.00'),3,2),1,1),
'1',null, '2',null,
decode(substr(to_char(sueldo_num,'00000000.00'),4,1), '1','UN')
,'2','DOS', '3','TRES', '4','CUATRO', '5','CINCO',
'6','SEIS', '7','SIETE', '8','OCHO', '9','NUEVE',
null))
//
decode(substr(to_char(sueldo_num,'00000000.00'),2,3), '000',null,'001')
,' MILLON ', ' MILLONES ')
into sueldo
from
system.dual
-----
select sueldo || ' ' ||
decode(substr(substr(to_char(sueldo_num,'00000000.00'),5,3),1,1)
,'1',decode(substr(to_char(sueldo_num,'00000000.00'),5,3), '100','CIE')
,'2','CIENTO',
'3','DOSCIENTOS', '3','TRESCIENTOS', '4','CUATROCIENTOS',
'5','QUINIENTOS', '6','SEISCIENTOS', '7','SETECI

```

3 sel3 Y N

4 sel4 Y N

- 12atechoaff

ORACLE-MEXICO
Fecha:

Aplicacion: avoibago
Bloque: MOVIMIENTOS

Triggers que conforman a la aplicacion
Hera:

Id	Nombre	# caso	Etiqueta	A' F	Etiqu. éxito	Etiqu. falla	Texto del Trigger
sueld	PRE-FIELD	4	sel4		Y	N	<pre> ENTOS '0','OCOCIENTOS','9','NOVECIENTOS',null) into sueldo from system.dual ===== select sueldo decode(substr(substrito_char(sueldo_num,'000000000 0.00'),6,2),1,1), '1',decode(substrito_char(sueldo_num,'000000000,0 0'),6,2),30,'TREINTA' ,'TREINTA Y ' ,'4',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),40,'CUARENTA' ,'CUARENTA Y ' ,'5',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),50' ,'CINCUENTA','CINCUENTA Y ' ,'6',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),60,'SESENTA' ,'SESENTA Y ' ,'7',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),70,'SETENTA' ,'SETENTA Y ' ,'8',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),80,'OCHENTA' ,'OCHENTA Y ' ,'9',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),90,'NOVENTA' ,'NOVENTA Y ' ,'1',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),10,'DIEZ' ,'11','ONCE','12','DOCE','13','TRECE','14','CAT GRCE','15','QUINCE' ,'16','DIECISEIS','17','DIECISIETE','18','DIECIOCH O','19','DIECINUEVE') ,'2',decode(substrito_char(sueldo_num,'000000000, 00'),6,2),20,'VEINTE' ,'21','VEINTIUN','22','VEINTIDOS','23','VEINTITR ES' ,'24','VEINTICUATRO','25','VEINTICINCO','26','VE INTISEIS' ,'27','VEINTISIETE','28','VEINTIOCHO','29','VEIN TINUEVE'),null) into sueldo from system.dual ===== select sueldo decode(substr(substrito_char(sueldo_num,'0000 0000,00'),6,2),1,1), '1',null,'2',null, decode(substrito_char(sueldo_num,'000000000,00'), 7,1),1,'UN' ,'2','DOS','3','TRES','4','CUATRO','5','CINCO' ,'6','SEIS','7','SIETE','8','OCHO','9','NOVE', null) into sueldo from system.dual </pre>
		5	sel5		Y	N	
		6	sel6		Y	N	

C a p o	Nombre	# paso	Etiqueta	A ^o R ^o	Etio. éxito	Etio. falla	Texto del Trigger
sueldo	PRE-FIELD	6 sel6		Y	N		select issueldo !! ' ' !! decode(substr(to_char(issueldo_num,'00000000.00'), 5,3),000,null,'NIL')
		7 sel7		Y	N		into issueldo from system.dual
		8 sel8		Y	N		select issueldo !! ' ' !! decode(substr(substr(to_char(issueldo_num,'00000000 0.00'),8,3),1,1) '1',decode(substr(to_char(issueldo_num,'00000000. 00'),8,3),100,'CIEN' 'CIENTO '), '2','DOSCIENTOS ', '3','TRESCIENTOS ', '4','CUATRO CIENTOS ', '5','QUINIENTOS ', '6','SEISCIENTOS ', '7','SETECI ENTOS ', '8','OCHOCIENTOS ', '9','NOVECIENTOS ', null) into issueldo from system.dual
		9 sel9		Y	N		select issueldo !! ' ' !! decode(substr(substr(to_char(issueldo_num,'00000000 0.00'),9,2),1,1) '3',decode(substr(to_char(issueldo_num,'00000000.0 0'),9,2),30,'TREINTA' 'TREINTA Y ') '4',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),40,'CUARENTA' 'CUARENTA Y ') '5',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),50' 'CINCUENTA','CINCUENTA Y ') '6',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),60,'SESENTA' 'SESENTA Y ') '7',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),70,'SETENTA' 'SETENTA Y ') '8',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),80,'OCHENTA' 'OCHENTA Y ') '9',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),90,'NOVENTA' 'NOVENTA Y ') '1',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),10,'DIEZ' '11','ONCE', '12','DOCE', '13','TRECE', '14','CAT ORCE', '15','QUINCE' '16','DIECISEIS', '17','DIECISIETE', '18','DIECIOCN O', '19','DIECINUEVE') '2',decode(substr(to_char(issueldo_num,'00000000. 00'),9,2),20,'VEINTE' '21','VEINTIUN', '22','VEINTIDOS', '23','VEINTITR ES' '24','VEINTICUATRO', '25','VEINTICINCO', '26','VE INTISEIS'

- 14setechooff

ORACLE-MEXICO
Fecha:

Aplicacion: nvaingre

Triggers que conforman a la aplicacion
Nora1

Cond.	Nombre	Nº paso	Etiqueta	A. R.	Etig. evita	Etig. falla	Texto del Trigger
sueldo	PRE-FIELD	9	sol19	Y	N		<pre> '27','VEINTISIETE' , '28','VEINTIOCHO' , '29','VEIN TINUEVE'), null) into sueldo from system.dual ===== select sueldo ' ' !! decode(substr(substr(to_char(sueldo_num,'00000000 0.00'),9,2),1,1), '1',null,'2',null, decode(substr(to_char(sueldo_num,'00000000.00'), 10,1),'1','UN' '2','DOS' , '3','TRES' , '4','CUATRO' , '5','CINC O' '6','SEIS' , '7','SIETE' , '8','OCHO' , '9','NUEV E' , null)) into sueldo from system.dual ===== select sueldo ' NUEVOS PESOS' into sueldo from system.dual ===== select sueldo '//substr(to_char(sueldo_num,'00000000.0000') ,12,2)//' /100 N.N.' into sueldo from system.dual ===== </pre>
		10	sol10	Y	N		
		11	sol11	Y	N		
		12	sol12	Y	N		

Fecha:

Aplicación: moingre
Bloque: Nivel forma

Triggers que conforman a la aplicación

Campo	Nombre	# paso	Etiqueta	A? P?	Etig. éxito	Etig. falla	Texto del Trigger
KEY-EXIT		1		Y	N		OCOPY ' ' INTO GLOBAL.NUM_DEPARTAMENTO;
		2		Y	N		OCOPY ' ' INTO GLOBAL.TIPO_CALL;
		3		Y	N		GENERACRO EXIT;
KEY-NXTFLD		1		Y	N	GENERACRO NXTFLD;	
KEY-OTHERS		1		Y	N	GENERACRO NULL;	
KEY-PRINT		1		Y	N	GENERACRO PRINT;	
KEY-PRVFLD		1		Y	N	GENERACRO PRVFLD;	
PRE-FORM		1		Y	N	OCOPY '1' INTO GLOBAL.TIPO_MOVIMIENTO	
PRE-INSERT		1		Y	N		OCOPY :RFC INTO :DIRECCIONES.DIA_RFC Mensaje err: FALLO EL RFC
		2		Y	N		OCOPY 'C' INTO :DIRECCIONES.CASA_OPTICINA Mensaje err: FALLO LA C
		3		Y	N		OCOPY :CLAVE_ASIGNATURA INTO :MOV_CLAVE_ASIGNATURA Mensaje err: FALLO LA CLAVE_ASIGNATURA
		4		Y	N		OCOPY :CLAVE_CATEGORIA INTO :MOV_CLAVE_CATEGORIA Mensaje err: FALLO LA CLAVE_CATEGORIA
		5		Y	N		OCOPY :CLAVE_NACIONALIDAD INTO :MOV_CLAVE_NACIONALIDAD Mensaje err: FALLO LA CLAVE_NACIONALIDAD
		6		Y	N		OCOPY 'B' INTO :MOV_STATUS Mensaje err: FALLO LA B
		7		Y	N		OCOPY :NUM_DEPARTAMENTO INTO :MOV_NUM_DEPARTAMENTO
		8		Y	N		OCOPY GLOBAL.TIPO_MOVIMIENTO INTO :MOV_CLAVE_TIPO_MOVIMIENTO
		9		Y	N		OCOPY :RFC INTO :MOV_RFC
		10		Y	N		OCOPY :HORAS_TECNIA2 INTO HORAS_TECNIA
		11		Y	N		OCOPY :HORAS_PRACTICA2 INTO HORAS_PRACTICA
		12		Y	N		UPDATE BANCO HORAS SET HORAS_EJECUCION = :HORAS_YO_TALES+HORAS_TERCIDAS WHERE BAN_CLAVE_CATEGORIA = :MOV_CLAVE_CATEGORIA AND BAN_NUM_DEPARTAMENTO = :MOV_NUM_DEPARTAMENTO

Fecha:

Aplicación: mvoingre
Brique: PROFESORES

Hora:

Comp	Nombre	N paso	Etiqueta	A' R'	Etiqu. existe	Etiqu. falla	Texto del Trigger
ANTIGUEDAD	KEY-NXTFLD	1		Y	N		DEVENACRO GOFLO DIRECCIONES.DIRECCION; =====
ESTADO_CIVIL	KEY-NXTFLD	1		Y	N		DEVENACRO CASE ESTADO_CIVIL IS WHEN 'S' THEN ENDSTEP; WHEN 'C' THEN ENDSTEP; WHEN OTHERS THEN ENOTRIG MESSAGE 'Los posibles val ores para el estado civil sort: S o C' END CASE; =====
		2		Y	N		DEVENACRO GOBLN CATEGORIAS; =====
NOMBRE	KEY-NXTFLD	1		Y	N		DEVENACRO GOFLO DIRECCIONES.DIRECCION; =====
RFC	KEY-NXTFLD	1		Y	N		DEVENACRO CASE GLOBAL.TIPO MOVIMIENTO IS WHEN '2' THEN GOSTEP BUSCA HISTORICO; WHEN '7' THEN GOSTEP BUSCA HISTORICO; WHEN OTHERS THEN GOSTEP NORMAL; END CASE; =====
		2	BUSCA_HISTOR	N	N	FIN	SELECT HIS_PRO_RFC,HIS_PRO_NOMBRE,HIS_PRO_CLAVE_NA CIONALIDAD,HIS_PRO_SEXO, HIS_PRO_ESTADO_CIVIL,HIS_PRO_ANTIGUEDAD,DIREC CION_CASA,HIS_PRO_CP, HIS_PRO_TELEFONO INTO :RFC,:NOMBRE,:PRO_CLAVE_NACIONALIDAD,:SEXO,:E STADO_CIVIL,:ANTIGUEDAD, :DIRECCIONES.DIRECCION,:DIRECCIONES.CP,:DIREC CIONES.TELEFONO FROM HISTORICO PROFESORES WHERE :RFC = HIS_PRO_RFC FOR UPDATE OF HIS_PRO_RFC,HIS_PRO_NOMBRE,HIS_PRO_C LAVE_NACIONALIDAD, HIS_PRO_SEXO,HIS_PRO_ESTADO_CIVIL,HIS_PRO_A NTIGUEDAD,DIRECCION_CASA, HIS_PRO_CP,HIS_PRO_TELEFONO =====
		3	NORMAL	N	N	FIN	TERMINAR
4	TRAER_DIRECCI	N	N			TERMINAR	Mensaje opp: El profesor no esta registrado. =====
5		N	N				Mensaje opp: El profesor no tiene direccion domicilio =====
6		N	N				Mensaje opp: El profesor no tiene direccion de oficina. =====
7	TERMINAR	Y	N	nada			DEVENACRO GOFLO PROFESORES.NOMBRE; =====
8	FIN	N	Y				SELECT '1' FROM SYSTEM.DUAL 157

Mensaje opp: El profesor ya tiene un acrobamiento.
- 17sotochooff

C a m p o	Nombre	U paso	Etiqueta	R? R?	Etq. éxito	Etq. falla	Texto del Trigger
RFC	KEY-NXTFLD	8	FIN	N	Y	Y	DEBERA SER CLAREC;
							9
	KEY-PRVFLD	1			Y	N	DEBERA SER GOFLU DEPARTAMENTOS.NUM DEPARTAMENTO;
SEXO	KEY-NXTFLD	1			Y	N	DEBERA SER CASE 'SEXO IS
	KEY-PRVFLD	1			Y	N	WHEN 'N' THEN NEXTFLD; WHEN 'F' THEN NEXTFLD; WHEN 'O' THEN NEXTFLD; WHEN 'I' THEN NEXTFLD; WHEN OTHERS THEN ENDTRIG MESSAGE 'Sexo no valido, solo N (O) o F (I)'; END CASE;
	KEY-PRVFLD	1			Y	N	DEBERA SER GOBLK NACIONALIDADES;

Fecha:

Aplicación: nvoingpp
 Bloque: nacionalidad

Hora:

Campos	Nombre	# paso	Etiqueta	A? R?	Etio. éxito	Etio. falla	Texto del Trigger	
NOMBRE_NACIO	KEY-NXTREC	1		Y	N		<pre> @XENACRO NXTREC; @XENACRO CASE :CLAVE_NACIONALIDAD IS WHEN '' THEN PRVREC;ENDTRIG MESSAGE 'Ya no hay na s nacionalidades'; WHEN OTHERS THEN NULL; END CASE; </pre>	
		2		Y	N			
	KEY-PRVREC	1		Y	N		<pre> @XENACRO PRVREC; </pre>	
Nivel Bloque	KEY-EYEORF	1		Y	N		<pre> @COPY '1' GLOBAL.TIPO_CALL @XENACRO CALLORY NACIONALIDADES; @COPY GLOBAL.CLAVE_NACIONALIDAD :CLAVE_NACIONALIDA D @COPY GLOBAL.NOMBRE_NACIONALIDAD :NOMBRE_NACIONALI DAD </pre>	
		2		Y	N			
		3		Y	N			
		4		Y	N			
	KEY-NXTFLD	1		Y	N	CONTINUA	FIN	<pre> SELECT CLAVE_NACIONALIDAD,CLAVE_NACIONALIDAD INTO :CLAVE_NACIONALIDAD,:PRO_CLAVE_NACIONALIDAD FROM NACIONALIDADES WHERE :NOMBRE_NACIONALIDAD=NOMBRE_NACIONALIDAD Mensaje err: Esta nacionalidad no se tiene registrada. !!! FIN @XENACRO GOFLD PROFESORES.SEXO; </pre>
		2		CONTINUA	Y	N		
	KEY-PRVFLD	1		Y	N			<pre> @XENACRO GOFLD DELEGACIONES.CLAVE_DELEGACIONI; </pre>

CAPITULO V.
CONCLUSIONES Y RECOMENDACIONES.

V. CONCLUSIONES Y RECOMENDACIONES.

La promesa de CASE es incrementar la productividad en la generación de software y mejorar su calidad, en las empresas que lo utilizan.

Para un gran número de compañías, CASE ha cumplido su promesa y las organizaciones han reportado impresionantes aumentos en la productividad, típicamente en un rango del 30 al 40%. Estas mismas compañías han reportado importantes mejoras en calidad, tales como: mejor documentación, menos "BUGS" y errores en los programas que se encuentran trabajando en producción, disminución de cambios en los requerimientos, un mejor control y predicibilidad de los proyectos.

Sin embargo, también hay compañías que han fallado al tratar de alcanzar los beneficios prometidos por CASE. Se estima que del 4 al 7% de las organizaciones que han tratado de utilizar CASE, han detenido su uso, dado que no cumplió sus compromisos.

Entonces, se podría plantear el siguiente cuestionamiento:

¿Por qué algunas empresas fallan en la utilización de CASE, mientras otras logran tremendos éxitos?

A continuación se dan algunas razones, por las cuales puede fracasar la implantación de CASE.

- La funcionalidad y/o el nivel de integración de las herramientas CASE es el adecuado para ser usado fácilmente.
- No se implantan una metodología, estándares y procedimientos.
- Las prácticas de administración del proyecto son inadecuadas.
- El soporte de la gerencia es insuficiente.
- Los programadores tienen una actitud negativa hacia CASE y se resisten al cambio.
- La necesidad de capacitación y entrenamiento en herramientas CASE son subestimados.

Obviamente, las compañías que escogen CASE deben considerar muchos aspectos técnicos, administrativos y de planeación, para poder asegurar que podrán implantar esta tecnología exitosamente.

Los siguientes pasos son una guía de lo que deben considerar las organizaciones antes de la implantación de CASE.

Un punto importante para las empresas nacionales, es saber que en los Estados Unidos, hay cientos de compañías que emplean la tecnología CASE.

De 862 organizaciones encuestadas por "Sentry Survey" reportaron que el 52% de ellas tenían una experiencia promedio de 7 sistemas hechos con CASE.

Como las diferentes compañías tienen distintas metas en software, así como prioridades, no existe la opción "correcta" para escoger herramientas CASE y metodologías estructuradas, apropiadas para cada empresa, y como cada organización conoce sus propias metas, la elección depende de cada una de ellas.

Pero, aunque se encontraran las herramientas y metodología adecuadas, ello no puede garantizar el éxito, si la organización no está preparada para cambiar a CASE.

Las empresas que cambien a CASE, deberán firmar un acuerdo con su alta gerencia para tener la educación adecuada en las herramientas y metodologías para todos sus usuarios, y hacer un plan detallado para implantar CASE.

Los gerentes deberán brindar soporte a todas las fases de la implantación de CASE, dado que la transición es costosa y lleva tiempo.

Para muchas organizaciones, una transición total a CASE puede llevarse de 2 a 5 años. Durante este período, las empresas no sólo deberán introducir nuevas herramientas, sino también nuevos estándares y procedimientos, y tal vez reorganizar sus departamentos de sistemas de información.

CASE significa una manera más disciplinada y estandarizada de desarrollo de software. Esta nueva forma de trabajar puede implicar cambios mayores en sus sistemas de trabajo. La misión de los altos gerentes es evitar que su personal de sistemas se resista a CASE antes de tener una implantación exitosa.

Una forma fácil en que las organizaciones pueden hacer su transición, es permitiendo a sus desarrolladores y personal de mantenimiento, así como de soporte técnico, participar en la evaluación y selección de las herramientas CASE. Cada individuo deberá sentir que su participación representa una oportunidad para el éxito, y su experiencia puede contribuir en el establecimiento de los estándares y procedimientos corporativos.

La gerencia deberá entonces reconocer y recompensar a aquellos que adopten CASE y hagan un uso eficiente de esta tecnología.

Las compañías que deseen implantar CASE, deberán planear cuidadosamente su estrategia, ya que varias empresas han tratado de adoptar CASE, antes de considerar costos iniciales y el impacto final, han visto que sus herramientas CASE se dejan de emplear muy pronto. Esto se debe a que simplemente compran y traen nuevas herramientas, sin planear como introducir CASE a la cultura existente en la organización.

Algunas recomendaciones para efectuar una planeación paso a paso son:

- Definir el proceso del ciclo de vida del software, especificando los pasos a seguir, los estándares y los procedimientos que controlan dicho proceso, como se asegura la calidad para sus productos de software.

- Definir el orden de aplicación de la tecnología CASE para el proceso, basándose en las necesidades y prioridades de la organización.

- Seleccionar las herramientas CASE para automatizar y reforzar los estándares y procedimientos.

- Probar las metodologías y herramientas CASE en proyectos piloto.

- Introducir las metodologías y herramientas CASE aprobadas, a gran escala en la organización a través de programas de entrenamiento y reuniones gerenciales.

- Evaluar el impacto de CASE, aplicando medidas preestablecidas para medir el éxito; mantener reuniones para continuar mejorando el proceso del ciclo de vida del software, así como la calidad de los productos.

- Estos pasos pueden variar dependiendo de si la empresa ya cuenta con estándares y procedimientos automatizado, para el desarrollo de software. Si es así, deberán seleccionar herramientas CASE que soporten la automatización de su proceso existente en el desarrollo de software; Si no, se deberán definir estos procesos, y después elegir las herramientas CASE. Obviamente, las empresas deberán tener procesos bien definidos antes de intentar automatizarlos usando CASE.

- Es importante recordar que la tecnología CASE consiste de dos partes indivisibles (Las herramientas y la metodología estructurada). Mientras las herramientas CASE automatizan el software, la metodología define los pasos del proceso, las entradas y salidas de cada fase, la calidad y las verificaciones de control. Por esta razón el implantar CASE debe seguir una estrategia planeada por la metodología. La elección de las herramientas CASE será dirigida por la elección de esta metodología.

Las organizaciones que deseen cambiar a CASE, deberán ponerlo a prueba con un proyecto piloto, el cual debe ser real, más no crítico. Además deberán definir antes de iniciar este proyecto piloto, como se medirá el éxito y que metas se deberán alcanzar con CASE.

Las empresas tal vez quieran usar herramientas y/o metodologías de manera paralela en los proyectos piloto, para así poder compararlos.

Los participantes en el proyecto piloto, deben ser desarrolladores de software que estén de acuerdo en usar la tecnología CASE, y algunos de los participantes deberán tener experiencia desarrollando sistemas. Cuando el proyecto termine, la empresa podrá emplear los resultados para sugerir que herramientas y metodología CASE son las adecuadas.

Hay que convencer a la gerencia y a los desarrolladores de los beneficios que obtendrán en sus resultados si emplean CASE; definir los estándares y procedimientos que emplearán para forzar así el uso de la tecnología CASE.

Después de lograr el éxito en el proyecto piloto, los participantes pueden pasar a ser miembros de otros proyectos con CASE, así extenderán su experiencia y entusiasmo por toda la organización.

Las empresas que implantan CASE, usualmente asignan esta tarea a un grupo en especial de profesionales del software dentro de sus organizaciones, (por ejemplo centros de desarrollo, el grupo de estándares y procedimientos, el grupo de tecnología avanzada o el grupo de certificación de calidad), los cuales evalúan nuevos métodos para desarrollar software. Si tal grupo no existe ahora, es buen momento para considerar su creación.

Las organizaciones que deciden implantar CASE, no deben esperar un éxito de la noche a la mañana, ya que el proceso requiere de una cuidadosa planeación, un gasto substancial de dinero y de recursos humanos, así como un fuerte compromiso por parte de la alta gerencia con el personal de desarrollo de sistemas.

La mayoría de las empresas que implanten CASE, encontrarán que esta tecnología les ayudará a realizar e incluso exceder sus metas en el desarrollo de sistemas. Simplemente DEBERAN SER PACIENTES.

Como resultado de haber realizado el sistema banco de horas siguiendo la metodología CASE, se vio que se tiene un buen control de desarrollo del sistema, ya que en todo momento se sabe que se deberá ir obteniendo.

Durante el desarrollo del sistema, se presentaron algunos problemas de tipo técnico, al principal problema fue la limitante de memoria que tenía la Computadora Personal que se utilizó, ya que los productos Oracle consumen gran cantidad de la misma, y como el programa fue creciendo para completar su programación, ya no cabía en la memoria, por lo que se optimizaron algunas rutinas.

Por otro lado se pudo comprobar que la aplicación, por estar desarrollada en los productos de Cuarta Generación de Oracle, es portable a cualquiera de las plataformas en las que dichos productos corren, no importando así el sistema operativo, ni el Hardware. La manera como comprobamos dicha portabilidad fue, porque se inicio el desarrollo en la VAX del CECAFI, la cual corre bajo un sistema operativo VMS, posteriormente, se llevaron los programas a la computadora personal con sistema operativo DOS.

Cabe aclarar que la computadora personal que se utilizó, tiene 4 MB de RAM. Mínima cantidad recomendada de RAM para que pueda correr el sistema.

En lo que respecta a la seguridad de la información, el sistema tiene passwords que se deben proporcionar para poder ejecutar diversos movimientos o reportes.

También se pueden tener usuarios con diferentes tipos de privilegios, con lo cual se restringen los accesos a ciertas funciones que solo puede efectuar la persona responsable del Banco de Horas.

El sistema cuenta con una diversidad de reportes, que son los más utilizados en la División, entre los Jefes de departamento, la Secretaria Auxiliar y la Jefatura de la División, sin embargo si se presenta la necesidad de crear otro reporte, o explotar la información de otra manera, es posible, ya que la estructura de la base de datos se tiene, y los reportes se hacen en SQL, que es un lenguaje de consulta de datos, con una sintaxis muy similar a la manera en la que se quiere realizar las cosas, como es de cuarta generación, basta "decir" que se quiere, y no cómo se hace, el manejador de la base de datos se encarga de ejecutar la petición.

En lo que respecta al sistema en si, se ha trabajado con una de las personas que mayor conocimiento tiene del manejo y administración del banco de horas, y toda esta experiencia se ha visto reflejada en el sistema, ya que se han validado cada uno de los movimientos, y revisando que las funciones se hagan como deben. Se han creado una gran cantidad de validaciones en los programas y reportes, para que se apeguen a la normatividad académica.

Algunas de las validaciones más importantes son:

Siempre que se introduce un movimiento, se verifica si el RFC ya está registrado, para evitar duplicados en los datos generales de los profesores, esto se ve reforzado por la creación de un índice único sobre el RFC.

Posteriormente, toda la información no debe ser modificada por el usuario, se tiene en varios catálogos, como son los departamentos de la división, las delegaciones, las nacionalidades, las categorías y las asignaturas de la División. Dichos catálogos pueden ser consultados desde cualquiera de las formas que se hicieron para efectuar los movimientos, se puede extraer información, más no modificarla. Para poder modificar dichos catálogos, se deberá entrar al sistema como administrador del mismo, e ir a la opción de modificación a los catálogos.

Existe otra funcionalidad, que dependiendo de la fecha de inicio del movimiento, automáticamente se calcula el sueldo del movimiento, basándose en un catálogo de tabuladores, en el cual se van introduciendo los nuevos tabuladores, y se conservan los anteriores.

En lo que respecta a las asignaturas, ya se tienen cargadas en el catálogo correspondiente todas las asignaturas de la DIE, y se tienen asociadas al departamento correspondiente, por lo que no se podrá asignar ninguna asignatura que no corresponda al departamento asociado al movimiento, evitando así inconsistencias de la información.

Asociado con los horarios y asignaturas, se tiene un reporte, el cual podrá ser comparado con el listado de horarios emitido por el CECAFI, para encontrar inconsistencias entre los horarios reportados por el CECAFI y los reportados por cada departamento.

Se ha puesto especial énfasis en el manejo de las horas máximas permitidas por categoría, para evitar cualquier error en el cálculo de las mismas.

Por otro lado el sistema cuenta con un reporte para listar los movimientos que deberán tener una prórroga dentro de un rango de fechas, con este reporte se puede notificar a los Jefes de departamento cuales de sus profesores deberán recibir una prórroga de contrato o bien una baja en el siguiente consejo técnico, evitando con ello que algún profesor pueda quedar descubierto en sus contratos.

Algo de lo más importante del sistema, es que integra todas las funciones que realizan entre varios sistemas existentes, y que permite tener un control del banco de horas, control de cada uno de los movimientos de los profesores, mantener siempre un estatus de los movimientos, teniendo así toda la información ya cargada dentro del sistema y solo irá variando el estatus del movimiento, dependiendo de en donde se encuentre. Posteriormente, se imprimen las formas únicas de los movimientos, y finalmente se hace la generación del reporte hacia el consejo técnico, en el cual toda la información proviene de las mismas tablas de donde se tienen los movimientos, con lo cual la veracidad de la información se asegura.

Es así que se puede afirmar que se realizó un sistema de calidad, y sobre todo funcional, ya que se satisfacen los requerimientos de la Secretaría Auxiliar de la DIE.

También, como recomendaciones se pueden hacer las siguientes:

El sistema puede ser instalado en cualquier división de la Facultad de Ingeniería, o incluso de la UNAM, pero se deberán cargar los catálogos correspondientes para cada una.

El sistema se puede mejorar, por ejemplo hacerlo para correr en red, con lo cual cada departamento de la División correspondiente, podrá hacer su propio manejo del banco de horas, pero todos accediendo a una sola base de datos, que podría estar en la Secretaría Auxiliar. Esto se puede llevar a cabo por que a cada departamento se le dan los privilegios necesarios, para que solo vea la información correspondiente a su departamento.

Así entonces se pueden hacer las siguientes conclusiones:

1. CASE incrementa la productividad y calidad en el software.
2. Considerar una serie de factores antes de emplear CASE.
3. El Ingeniero en Computación, dada su formación académica, es capaz de poder entender, evaluar e implantar CASE dentro de las empresas y dependencias que requieran de sus servicios profesionales.
4. El sistema banco de horas es un excelente ejemplo de como emplear CASE en la vida real.
5. El sistema cuenta con una serie de validaciones, las cuales permiten asegurar que los movimientos se efectuen como lo marca la normatividad universitaria.
6. Se cuenta con un sistema capaz de efectuar todas las funciones relacionadas con el banco de horas, altas, bajas, licencias, promociones, reportes diversos, manejo del banco de horas, impresión de formas únicas, reportes para el consejo técnico.
7. El sistema puede utilizarse en varias divisiones de la facultad.
8. El sistema ha sido liberado, pero puede ser mejorado, y esto está contemplado dentro del ciclo de vida del software.
9. Se puede dar mantenimiento facilmente al sistema, por que se tiene toda la documentación necesaria.
10. Con CASE se obtuvo un sistema de calidad.
11. En México actualmente el Ingeniero en Computación tiene un amplio campo de desarrollo en el área de la Ingeniería de Programación, que puede atacar con CASE.
12. CASE es la opción actual en la Ingeniería de Programación.

Anexo A. Manual de Usuario

Anexo A. Manual de Usuario.

A continuación se verá como se efectúa un movimiento, por ejemplo, una alta por nuevo ingreso.

1. Encender el equipo.
2. Cambiarse al directorio C:\TESIS\PROGRAMA:
C> cd tesis\programa
3. Iniciar la base de datos con el siguiente comando:
C> oracle
4. Iniciar el sistema del Banco de Horas:
C> banco
5. Después de esto, se desplegará el menú "APPLICATION MENU" que se muestra en la siguientes hojas.
6. Seleccionar la opción número 1 que es el "Banco de Horas".
7. Esto nos llevará al menú "Banco de Horas" que se muestra en las siguientes hojas.
8. En este menú se deberá seleccionar el tipo de movimiento, en este caso será la opción 1 "Altas".
9. Al seleccionarla, se despliega el menú "ALTAS", en el cual se deberá tomar la opción 1 "Nuevo ingreso".
10. Al tomar el tipo de movimiento correspondiente, se despliega la pantalla de datos generales del profesor. Esta pantalla es similar para todos los tipos de movimientos.
11. Lo primero que hay que proporcionar es el número de departamento al que corresponde el movimiento; es importante señalar que en cada campo, al posicionarse envía un mensaje de ayuda, donde describe brevemente que significa ese campo, existen algunos campos como son número de departamento, la clave de la nacionalidad, la clave de la categoría, la clave de la asignatura. Los cuales cuentan con una ayuda más específica, y a la cual se entra con la tecla de función <F8> y se llega a otra pantalla, donde están los posibles valores para ese campo.
12. Aquí se mueve con las teclas de <Pantalla Siguiente> y <Pantalla Previa>, hasta posicionarse en el valor requerido, ahí se da <F7> para seleccionarlo o <ESC> para cancelar y regresar a la pantalla 3. En el campo correspondiente se desplegará el valor previamente seleccionado.
13. Para moverse entre los campos, se hace con la tecla <ENTER> o con la tecla <TAB>.
14. Un ejemplo de este tipo de pantallas de ayuda se muestra para las claves de las asignaturas.

A P P L I C A T I O N M E N U

SELECT AN APPLICATION FROM THE LIST BELOW

- > 1 Banco de Horas
- 2 SQL*Menu Development of Dynamic Menus
- 3 Exit

Make your choice: 1

v Fri Dec 08 13:20:16 1995 Replace (6ap16)

Banco de Horas
DivisiAn de IngenieríA Eléctrica

- > 1 Altas
- 2 Bajas
- 3 Reportes
- 4 Modificaciones
- 5 Menú anterior
- 6 Salir

Make your choice: 1

Escoja su opción:

v Fri Dec 05 13:21:05 1995 OSC DBG Replace BANCO (BANCO)

	BANCO DE HORAS	FECHA: 08-DEC-95 HORA : 13:21
ALTAS 1. NUEVO INGRESO 2. REINGRESO 3. PRORROGA 4. OTRO NOMBRAMIENTO 5. AUMENTO DE HORAS 6. PROMOCION 7. REANUDACION DE LABORES 8. LICENCIAS 0. Menú anterior OPCION: 1_		

Seleccione el tipo de alta. _

Char Mode: Insert Page 1

Count: *0

NUEVO INGRESO DEPARTAMENTO —	BANCO DE HORAS _____	FECHA: 08-DEC-95 HORA : 13:23		
RFC	Nombre	Antigüedad _____ meses		
Domicilio Casa	Colonia	CP	Teléfono	
Clave _____	Delegación _____			
Domicilio Oficina	Colonia	CP	Teléfono	
Clave _____	Delegación _____			
Nacionalidad _____	Sexo _____	Estado Civil _____		

DEPARTAMENTO —	BANCO DE HORAS	FECHA: 08-DEC-95 HORA : 13:32
	Clave	Asignatura
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____
	_____	_____

Proporcione el numero de departamento en el que trabaja el profesor. —F8—>Ayud:
Char Mode: Replace Page 1 Count: *0

15. Cada campo valida automáticamente que se haya introducido un tipo de dato preestablecido (Por ejemplo numérico, caracteres, fecha).

16. A continuación se posicionará el cursor al campo "RFC", ahí habrá de teclearse el RFC del profesor y dar <ENTER>, de manera automática se checará si ya se tiene registrado el RFC, si ya se tiene se detendrá y no dejará pasar de este campo, si no existe, se posicionará en el campo de "Nombre".

17. Es importante mencionar que existen campos que por seguridad no son modificables, ya que contienen información que no es factible modificar por el usuario, Por ejemplo, los nombres de los departamentos, asignaturas, etc.

18. Así se continuará navegando por los campos de la pantalla de datos generales y se irán desplegando los mensajes correspondientes, al llegar al campo "Estado Civil" y al dar <ENTER>, aparecerá la pantalla de los datos correspondientes al movimiento.

19. En esta pantalla, se copiarán los datos del número de departamento, el nombre del departamento y el "RFC". Estos datos se toman de la pantalla de datos generales.

20. El cursor se detendrá en el campo "Clave", donde se dará la clave de la categoría del movimiento; este campo cuenta con ayuda adicional con la tecla <F8>, al dar <ENTER> se desplegará la descripción de la categoría en el campo "categoría", se pondrá el "Estatus" con el valor "B" (Bitácora).

21. En el campo de la categoría se verifica que el departamento tenga esa categoría dada de alta.

22. Así se posicionará sobre los campos "Horas Adicionales", en los cuales hay que dar el número de horas teóricas y prácticas del movimiento.

23. Cabe aclarar que en este caso, por ser una alta por nuevo ingreso, los campos de "Horas Actuales" tendrán un valor de cero; en caso de otros tipos de movimientos, en estos campos se desplegarán los totales de horas tanto de teoría como de práctica que tenga el profesor dentro de la división sin importar en que departamento las tenga.

24. En el campo de "Horas Adicionales Prácticas", al dar <ENTER>, se validará que se cumplan las restricciones establecidas en los reglamentos universitarios, en cuanto al número máximo de horas de teoría, horas de práctica y el total de horas; permitidas para cada categoría. Por ejemplo, para un ayudante, no se permiten horas de teoría.

NUEVO INGRESO		BANCO DE HORAS		FECHA: 08-DEC-95	
DEPARTAMENTO —				HORA : 13:23	
<u> </u>		<u> </u>		<u> </u>	
RFC		Clave		Categoría	
<u> </u>		<u> </u>		<u> </u>	
Horas Actuales		Horas Adicionales		Horas	
Teóricas Prácticas		Teóricas Prácticas		Totales	
<u> </u>		<u> </u>		<u> </u>	
A partir de:		Con límite al:		Artículo 51	
<u> </u>		<u> </u>		<u> </u>	
				sueldo	
				N\$: <u> </u>	
				Horario:	
				Lun. <u> </u>	
				Mar. <u> </u>	
Clave		Asignatura		Grupo	
<u> </u>		<u> </u>		<u> </u>	
				Mie. <u> </u>	
				Jue. <u> </u>	
				Vie. <u> </u>	
				Sab. <u> </u>	
		Observaciones:		Grabar	
		<u> </u>		<u> </u>	

25. Posteriormente, se detendrá en el campo "A partir de", donde se dará la fecha inicial, a partir de la cual tendrá validez el movimiento, al dar <ENTER>, se pasará al campo "Con límite al", este campo se calcula por un año a partir de la fecha inicial, si no fuera el caso, es decir, el movimiento es por menos tiempo, sólo se dará la fecha final que se quiere. Estos dos campos validan que automáticamente que se de una fecha valida y que la fecha final sea mayor a la inicial.

26. El siguiente campo, es "Artículo 51", este campo admite únicamente dos posibles valores "S" o "N", y sólo se admite el valor "S", en aquellas categorías preestablecidas en el estatuto universitario. Si se diera en alguna categoría en que no es permitido el "Artículo 51", automáticamente se regresará el valor "N".

27. De ahí, se pasa a campo "Clave" de asignatura, en donde se dará la clave de la asignatura que tendrá el movimiento. Este campo cuenta con ayuda más amplia, dando <F8>. Al dar <ENTER> se desplegará el nombre de la asignatura en el campo "Asignatura", en este caso, el cursor no se posicionará en el campo "Asignatura", ya que no esta permitido que cualquier usuario modifique los nombres de las asignaturas.

28. Posteriormente, hay que dar el horario que se comprometio por parte del profesor, para desempeñar sus labores con la DIE, y el grupo en el cual dará su clase, y así poder checar con la base de datos de horarios del CECAFI.

29. También se tiene un campo "Observaciones", el cual es un campo donde se puede escribir cualquier comentario en formato libre.

30. Finalmente, se posicionará en el campo "Grabar Cambios", en donde se dará "S" si se quiere guardar los datos; de lo contrario dar una "N" para ignorar todos los datos que se han dado para este movimiento, después de dar <ENTER>, se regresará al campo "Departamento" de la pantalla de datos generales, donde se podrá iniciar otro movimiento del mismo tipo. Si se quiere cambiar de tipo de movimiento, en la pantalla de datos generales dar <ESC> y regresará al menú de "ALTAS".

31. Así se podrá navegar por todos los tipos de movimientos, y cuando se termine de capturar todos los movimientos correspondientes, se deberá acceder al menú de "REPORTES", en este menú seleccionar la opción de "Procesos".

32. Cuando se despliegue el "Menú de Procesos a la Base de Datos", primero obtener el reporte de "Bitácora", este reporte muestra todos los movimientos que han sido capturados para este Consejo Técnico, y de aquí se podrá verificar que sean los adecuados.

33. Una vez que se está satisfecho con el contenido del reporte de bitácora, se pasará a "Generar el Consejo Técnico".

34. La generación del Consejo Técnico, incluye los reportes de movimientos que se van a solicitar, así como el cuadro de resumen.

35. Cuando ya se ha generado el Consejo Técnico, se deberán imprimir las formas únicas, para ello seleccionar la opción 3 de el menú de procesos.

36. Al terminar con la impresión de formas únicas, se deberá hacer la integración de movimientos a la base de datos definitiva.

37. Al hacer la integración de los movimientos, se cambiará el estatus de éstos a "P" Pendiente.

38. Cuando el movimiento sea aprobado por el Consejo Técnico, y el profesor firme su contrato, entonces se deberá cambiar el estatus a "E", que significa ejercido.

REPORTES

División de Ingeniería Eléctrica

- > 1 Generales
- 2 Resúmenes
- 3 Catálogos
- 4 Procesos
- 5 Menú Anterior
- 6 Salir

Make your choice: 1

Escoja su opción:

v Fri Dec 06 13:26:31 1995 OSC DBG Replace BANCO (REPORT)

Menú de Procesos a la Base de Datos

División de Ingeniería Eléctrica

- > 1 Bitacora (80)
- 2 Generar Consejo Técnico (132)
- 3 Impresión de formas Únicas (132)
- 4 Integración de movimientos
- 5 Movimientos Pendientes (80)
- 6 Imprimir (80 columnas)
- 7 Imprimir (132 columnas)
- 8 Menú Anterior
- 9 Salir

Make your choice: 1

Escoja su opción:

v Fri Dec 06 13:30:30 1995 OSC DBG Replace BANCO (PROC)

GLOSARIO.

Análisis de datos: Revisar un análisis de entidades, a la luz de datos conocidos.

Análisis de entidades: Definir y entender las cosas de importancia sobre las cuales el negocio necesita saber o tener información, y las relaciones existentes entre ellas.

Análisis funcional: Identificar lo que actualmente se hace en el negocio, y lo que necesita hacerse en el futuro.

Atributo: Cualquier detalle que sirve para calificar, identificar o expresar el estado de ocurrencia de una entidad.

Columna: Atributo, Campo.

Dominio: Tipo de dato, Conjunto de valores con características comunes y que por lo tanto son sujetos a las mismas validaciones.

Entidad: Cualquier persona, lugar o cosa de importancia, sobre la cual se requiere mantener, desplegar y correlacionar información; Algo que puede existir por si mismo sea real o imaginariamente.

Evento: Una circunstancia que provoca la ejecución de una función de negocio.

Evento de cambio: Cualquier punto en la vida de la empresa, en el que, bajo ciertas condiciones especificadas, una ocurrencia de una entidad, es creada o destruida, el valor de un atributo es cambiado o una relación es conectada o desconectada.

Evento de sistema: La terminación de una o más funciones de negocio, que disparan la ejecución de otras funciones.

Evento de tiempo real: Un punto en la vida de la empresa, en las que, bajo ciertas condiciones, el tiempo alcanza una fecha y hora.

Función elemental: Una función de negocio que, una vez comenzada, debe ser terminada o desechada totalmente; si se puede tener en etapas intermedias, entonces no es función elemental.

Función de negocio: Aquello que la empresa hace o necesita hacer, no la manera en como la hace.

Integridad: Validar que lo que un usuario autorizado pretenda hacer, este correcto.

Integridad de entidades: La llave primaria completa debe ser no nula, de tal forma que los renglones sean plenamente identificables correspondiendo así con la ocurrencias de las entidades en el mundo real. Los valores de las columnas deben pertenecer a los dominios definidos por éstas.

Integridad referencial: El valor de una llave foránea de la tabla T1, que identifica a la llave primaria de la tabla T2, debe corresponder al valor de la llave primaria de algún renglón en T2 o ser totalmente nula. Una llave foránea no-nula que no corresponde a ninguna llave primaria identifica a un objeto inexistente.

Llave alterna: Cualquier llave candidata que no sea primaria, es llave alterna. Las llaves alternas pueden ser nulas.

Llave artificial: Cuando una tabla no tiene llave candidata simple, puede ser recomendable introducir una llave artificial. También se utilizan para una tabla identificada por varias llaves foráneas; para la tabla del lado 1 de la relación 1:M y las tablas contienen mucho renglones. Además al utilizar las llaves artificiales, se elimina redundancia (y consecuentemente ahorran espacio). Permiten que la codificación de proposiciones SQL (especialmente la cláusula WHERE) sean más cortas. Son una forma de normalización. Son un campo sin significado y difícil de conocer para el usuario. Es necesario su mantenimiento (creación y borrado).

Llave candidata: Combinación de columnas, cuyo valor identifica en forma única los renglones de una tabla, es la llave candidata. En una tabla puede haber varias llaves candidatas y se pueden trasladar (si son compuestas).

Llave compuesta: formada por dos o más columnas.

Llave foránea: Una llave foránea es una columna o combinación de ellas, cuyos valores son los mismos que los de la llave primaria de otra tabla. Las llaves foráneas y las llaves primarias se definen en el mismo dominio. Las llaves foráneas representan relaciones 1:1 y 1:M. Una llave foránea puede ser parte de una llave primaria compuesta.

Llave primaria: Todos los renglones de una tabla deben ser distintos y por lo tanto debe haber al menos una llave candidata. Una llave candidata, debe ser elegida, como llave primaria, tal que sea ÚNICA, MINIMA y NO NULA.

Llave simple: Consiste de una sola columna.

Mecanismo: Una técnica particular para automatizar una función de negocio.

Normalización: La normalización es una técnica para diseñar tablas en forma normal. Una tabla en forma normal no tiene redundancia.

El Método consiste en llevar a cabo una descomposición sin pérdida, lo que implica separar la tabla en otras más pequeñas, sin perder información. Existen seis formas normales ONF - 5NF.

Aunque la 3NF es suficiente para lograr los efectos positivos.

Cada forma normal está dentro del contexto de la forma normal precedente, lo que significa que para estar en la forma normal N, la tabla debe de estar en la forma normal N-1 y además satisfacer alguna restricción adicional.

RDBMS: Relational Database Management System (Sistema Manejador de Base de Datos Relacional).

Redundancia: La redundancia es indeseable, pues puede provocar inconsistencias debido a los problemas de actualización. La redundancia controlada puede mejorar el performance.

Relación: Cualquier forma significativa en la que dos cosas de cualquier tipo pueden estar asociadas (toda relación debe tener un nombre).

Región: Ocurrencia de una entidad, Registro.

Seguridad: Asegurar que los usuarios tengan autorización para hacer lo que pretenden.

BIBLIOGRAFIA.

- Barker Richard, "CASE*METHOD Entity Relationship Modelling"
Addisson-Wesley, Gran Bretaña 1990.
- Barker Richard, "CASE*METHOD Task And Deliverables"
Addisson-Wesley, Gran Bretaña 1991.
- Chistopher Ken, "DOS 5 The Basics"
WILEY, U.S.A. 1991.
- Perry James y Leter Joseph, "Understanding ORACLE"
SYBEX, U.S.A. 1989.
- Rangel Gutierrez Reymundo, "Apuntes de Progremación Estructurada"
UNAM, Facultad de Ingenieria 1985.
- Oracle, "Detailed Systems Analysis"
Oracle, Reino Unido 1988.
- Oracle, "SQL*FORMS Class Notes"
Oracle Corporation, U.S.A. 1987.
- Oracle, "SQL*PLUS Class Notes"
Oracle Corporation, U.S.A. 1988.
- Oracle, "SQL*Report Class Notes"
Oracle Corporation, U.S.A. 1988.
- Oracle, "Introduction to CASE*Dictionary Class Notes"
Oracle Corporetion, U.S.A. 1989.
- Oracle, "Oracle Magazine, The SQL Detabese Journal" Otoño 1989,
Verano 1990. U.S.A.