

11
24.



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGÓN

**"SISTEMA INTEGRAL DE ADMINISTRACION ESCOLAR
MODULO PROFESORES"**

T E S I S

QUE PARA OBTENER EL TITULO DE :

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

LETICIA CERDA GARRIDO

ASESOR :

ING. ERNESTO PEÑALOZA ROMERO

MÉXICO

1997

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



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.

DEDICATORIAS Y AGRADECIMIENTOS

A mis compañeros del Instituto Mexicano del Petróleo: Socorro, Hugo y Rossy por las veces que me auxiliaron en mis tareas.

A la profesora Silvia Vega por permitirme continuar cuando mi tiempo se había terminado evitando con que mi carrera se truncara.

Al Ing. Ernesto Peñaloza por la oportunidad que me brindó con este proyecto y aunque se que no fue lo que él esperaba espero no haberle fallado demasiado.

A mis compadres José Luis Curiel, Patricia Martínez, Enrique Palmero y Marco Antonio Hernández por su apoyo y motivación, gracias Pepé por los jalones de oreja que tanta falta me hacen, pero sobre todo gracias a los cuatro por la confianza que han depositado en mí, espero no defraudarlos.

A mis hermanos: Susana, Ramón, José, Eduardo y María de la Luz por lo especial que siempre me han hecho sentir y por todo su apoyo y cariño.

A mis Padres les agradezco la vida que me dieron y todo el apoyo que siempre me han brindado, y sobre todo les agradezco que se sientan orgullosos de mí, aunque no lo merezca.

A mi Amado Esposo le doy un agradecimiento muy especial porque sin su apoyo no habría podido terminar mi carrera ni elaborar este proyecto ya que no solo brindo su apoyo en una forma pasiva sino que se involucro totalmente en todo lo referente a mis estudios participando de mis proyectos y realizando muchas actividades que me facilitaron los tramites para llegar hasta esta meta, gracias TE AMO con toda mi alma.

Por último pero de una manera muy especial le dedico este trabajo a mi pequeño hijo **Alexel Juan Tenorio Cerdá** ya que fue él quien se constituyo en mi aliciente principal y a la vez quiero pedirle perdón por los momentos de atención que le robe para poder concluir este proyecto. **Bebé** eres la razón de mi vida.

LE DOY GRACIAS A DIOS

Por la oportunidad que me
dio para terminar
mis estudios y terminar
este proyecto y por
todas las cosas
que me ha dado en la vida.

ÍNDICE

OBJETIVOS GENERALES

INTRODUCCIÓN

CAPITULO I. RESEÑA HISTÓRICA

| | |
|---|----|
| I.1. Automatización de Oficinas..... | 2 |
| I.2. Fundamentos de Bases de Datos..... | 7 |
| I.3. Ciclo de Desarrollo de los Sistemas..... | 16 |

CAPITULO II. ANÁLISIS POR SECCIÓN (CONTROL DE PROFESORES)

| | |
|---|----|
| II.1. Funciones de la Coordinación..... | 34 |
| II.2. Descripción del Control de Profesores..... | 57 |
| II.3. Planteamiento de la Solución..... | 57 |
| II.3.1. Identificación de los Requerimientos del Usuario..... | 58 |
| II.3.2. Propuesta de Solución..... | 59 |
| II.3.3. Información Requerida..... | 60 |

CAPITULO III. DISEÑO

| | |
|--|----|
| III.1. Selección de Software para el Diseño del Sistema..... | 63 |
| III.2. Diseño del Sistema..... | 70 |
| III.2.1. Estructuras de Bases de Datos..... | 70 |
| III.2.1.1. PROF CAT..... | 71 |
| III.2.1.2. PCICO971..... | 72 |
| III.2.1.3. PAICO971..... | 73 |
| III.2.1.4. PMICO971..... | 74 |
| III.2.1.5. PRICO971..... | 75 |
| III.2.1.6. PEICO971..... | 76 |
| III.2.1.7. TIPOMOV..... | 77 |
| III.2.1.8. DMCICO..... | 78 |
| III.2.1.9. CAT..... | 78 |
| III.2.2. Diagrama Modular..... | 80 |
| III.2.3. Explicación de Diagrama Modular..... | 84 |

CAPITULO IV. IMPLANTACIÓN

| | |
|--|-----|
| IV.1. Presentación al Usuario..... | 108 |
| IV.2. Programa de Capacitación..... | 110 |
| IV.3. Pruebas Generales del Sistema..... | 110 |
| IV.4. Descripción del Proceso Computarizado..... | 111 |

CONCLUSIONES

APÉNDICE A. Programas Fuentes
APÉNDICE B. Manual de Operación
APÉNDICE C. Manual de Procedimientos
BIBLIOGRAFÍA

OBJETIVOS GENERALES

- 1.1 Desarrollar un sistema de información, que mejore al existente, para apoyar en las tareas administrativas cotidianas dentro de las coordinaciones de la E.N.E.P. plantel ARAGÓN.**
- 1.2 Integrar y administrar la información de las coordinaciones.**
- 1.3 Crear la documentación necesaria para la actualización del sistema.**
- 1.4 Dar la información necesaria para que la información pueda ser trasladada para su procesamiento o alimentación a un sistema futuro.**

INTRODUCCIÓN

Ultimamente el avance tecnológico ha ocasionado que la ciencia de la computación adquiera gran importancia permitiendo la utilización de esta en cualquier área del conocimiento humano.

La humanidad desde los albores de la civilización, ha requerido información en sus intentos por administrar las organizaciones. De igual manera en la actualidad se tiene la necesidad de contar con información confiable y oportuna. ("Quien tiene la información tiene el poder").

Debido a que las necesidades de información en las organizaciones modernas son extensas y complejas es aconsejable auxiliarse de las computadoras para producir dicha información.

El cambiante campo de la electrónica ha permitido en los últimos años que las computadoras hayan simplificado su operación y disminuido tanto su tamaño como su costo. De tal manera que cada día es más accesible el adquirir equipo de computo para realizar las actividades propias de la oficina y la industria y porque no, incluso las del hogar. Por tal motivo la computadora ha penetrado como herramienta de productividad primordialmente en oficinas, bancos, escuelas, etc., es decir, en todos los centros de trabajo que requieren del manejo de grandes cantidades de información.

La Coordinación de Ingeniería en Computación, de la Escuela Nacional de Estudios Profesionales Campus Aragón, no es la excepción, ya que sus actividades requieren del manejo de grandes cantidades de información, siendo una de estas la realización de las propuestas de profesores en cada período escolar, así como la evaluación de los mismos y su control de asistencia.

Es por esto que la **Escuela Nacional de Estudios Profesionales, plantel "Aragón"** se ve en la necesidad de contar con un sistema integral, que le permita optimizar la administración de banco de horas, grupos-materia, control de profesores, exámenes extraordinarios, tesis y egresados, y que además logre la integración de información de todas las coordinaciones que conforman el plantel logrando con esto contar con una herramienta eficiente y eficaz para la toma de decisiones.

En la actualidad existe un sistema llamado **CRONOS**, sin embargo en el caso de este sistema, los módulos (**Banco De Horas, Exámenes Extraordinarios, Grupos Normales y Profesores**) se encuentran dispersos (**sistemas cerrados**), lo que genera dificultad y repetición de procesos (captura), que ocasiona principalmente pérdidas en tiempo; no deberemos olvidar que la carga de trabajo en las coordinaciones se eleva en los inicios y términos de semestre, esto es debido primordialmente a los tiempos existentes para entregar información de inscripciones y calificaciones respectivamente, sin olvidar el apoyo que proporcionan estas áreas a profesores, alumnos y la carga administrativa habitual.

Adicionalmente las necesidades de información procesada han crecido y dicho sistema ha permanecido estático por lo que se carece de algunas salidas (**Reportes**) que apoyen los procesos administrativos, tal es el caso de las **Propuestas** de horarios para profesores, al igual que la generación de acuerdos.

El principal factor que motivo la reprogramación del sistema y motivo la necesidad de partir de ceros, fue en esencia la nula información existente de **CRONOS** (información interna) y el no poder contar con información sobre el algoritmo de almacenamiento de información, o poder contar con dichos archivos en algún formato exportable o entendible al 100%.

Las características iniciales del programa, fueron el alto índice de integración que debería existir entre sus partes, sin olvidar la responsabilidad que tiene cada módulo en específico.

De esta manera las características deseables del nuevo sistema, consistían en conseguir una similitud con el sistema **CRONOS**; en sus pantallas de captura (para minimizar el tiempo de capacitación de los usuarios finales), como en los reportes que se emiten (Para de esta forma continuar con la documentación interna actual, debido a que a mostrado su eficiencia); siendo estos los requisitos principales, considerando la necesidad de incrementarlos, con reportes para la toma de decisiones ó adicionales; para una administración más sistematizada, con lo que se consigue minimizar aun más el error, así como la agilización de los procesos.

En este sentido, se encontró que el banco de horas, profesores y grupos normales; tenían un alto índice de integración, tan grande incluso que no se alcanzaba a percibir a simple vista los alcances y obligaciones de cada uno.

En el primer capítulo se hace énfasis en la conveniencia del uso de la computadora como una herramienta de la administración y los beneficios que esto implica, haciendo referencia a los conceptos de automatización de oficinas incluyendo una analogía entre el procesamiento manual de información y el procesamiento automático de datos, así mismo se introducen los conceptos

básicos sobre el manejo de información mediante el uso de bases de datos y ciclo de desarrollo de sistemas.

El capítulo segundo describe las funciones que realiza la Coordinación de Ingeniería en Computación, en cuanto a lo que a Control de Profesores se refiere. Mostrando una semejanza de la forma en que se realizaba anteriormente el control de profesores de forma manual. Esta dedicado también a identificar los requerimientos del usuario para solventar las necesidades de información al inicio y término de cada período escolar. De igual manera marca el análisis realizado para definir las necesidades de los usuarios y se presenta la solución propuesta para satisfacer dichas necesidades.

El tercer capítulo esta dedicado a la explicación del diseño del sistema, partiendo de la selección del software utilizado para el desarrollo de los programas, nos muestra la estructura de los archivos que forman las bases de datos utilizadas. Cuenta con el diagrama modular del sistema así como la explicación del mismo.

El Capítulo cuarto trata el proceso de implantación desde la presentación al usuario y la puesta en servicio del sistema desarrollado hasta las actividades realizadas para capacitar al usuario final, describe también las pruebas generales realizadas del sistema. Por último se detalla paso a paso el proceso automatizado.

Finalmente se encuentran los apéndices que contienen el listado completo de todos los programas fuentes que componen el sistema para la automatización del Control de Profesores, además del manual de operación para el sistema y una guía donde se describen los pasos necesarios para actualizarlo, tanto al principio como posteriormente, para que el sistema logre su objetivo.



CAPITULO I RESEÑA HISTORICA



CAPITULO I. RESEÑA HISTÓRICA

- I.1. Automatización de Oficinas**
- I.2. Fundamentos de Bases de Datos**
- I.3. Ciclo de Desarrollo de los Sistemas**

CAPITULO I**RESEÑA HISTÓRICA**

Herbert Simon (Premio Nobel en 1978) considera al ordenador como la cuarta gran apertura en la historia, para ayudar al hombre en su proceso de pensar y su habilidad en la toma de decisiones. La primera fue el invento de la escritura, que ayudó a la memoria del hombre a realizar trabajos de la mente. Los dos restantes eventos anteriores a los ordenadores fueron, el sistema numérico arábigo con su cero y valores posicionales y el invento de la geometría y cálculo analíticos, que permitieron la solución de problemas complejos, en la teoría científica. Ahora el ordenador electrónico digital combina las ventajas y atributos de todas estas posibilidades y lo pone a disposición para la toma de decisiones y para la gestión de organizaciones. (Robert G. Murdick y Joel E. Ross, Information Systems for Modern Management, Prentice Hall, Englewood Cliffs, N. Y., 1971, pág. 238)

I.1. AUTOMATIZACIÓN DE OFICINAS

Desde el inicio de la civilización, las personas han procesado datos, sin embargo, durante el siglo XX ha habido un enorme incremento tanto en número como en variedad de dispositivos mecánicos y electrónicos para procesar datos, con lo cual el manejo y control de la información se han facilitado enormemente.

La automatización de una oficina es la convergencia de herramientas de cómputo que permiten a los empleados y ejecutivos la manipulación efectiva de la información de una forma rápida y eficiente.

- Se obtiene una reducción de costos debido a que se elimina el procesamiento redundante de la misma información.
- Se consigue una mejor calidad de presentación en la elaboración de reportes, oficios, cartas, etc.
- El control y la calidad de servicio se ven incrementados al tener todos los servicios comunicados entre sí.

- Uno de los resultados de la automatización es la sensible reducción de las funciones de mano de obra.
- Se logra un mayor control en cuanto a las actividades desarrolladas por el personal de la oficina.
- En lo que respecta al ahorro de tiempo este es bastante considerable.
- El procesamiento de la información se mejora notablemente.

PROCESADORES HUMANOS DE DATOS

Los procesadores humanos de datos tienen la capacidad de realizar las siguientes funciones: entrada, producir un producto final, almacenar, calcular y tomar decisiones lógicas. Pero estas funciones son independientes unas de otras y por ello se necesita otra función adicionalmente, para coordinar las funciones citadas. Se requiere saber exactamente que hacer en cada una de las funciones y en qué secuencia, dicha función adicional es el Control, la cual se encuentra en el centro de las demás funciones.

Para que el procesador humano de datos aprendiera que hacer y en que secuencia fue necesario que alguien le formulara una lista de instrucciones, indicando todos los pasos a seguir para producir la información requerida. Esta persona debió tener cuidado de escribir las instrucciones en un lenguaje que el encargado del procesamiento de datos pudiera entender. Dichas instrucciones pueden estar sobre la mesa y así poder consultarla para saber que hacer, considerándose esto como un medio de almacenamiento.

PROCESADORES DE DATOS AUTOMÁTICOS

Un ordenador es más procesador de datos que un procesador humano de datos. Si un ordenador tiene que procesar datos, tiene también que aceptar datos de entrada del exterior, tiene que tener acceso a cierto tipo de almacenamiento, tiene que ejecutar funciones aritméticas y lógicas, y tiene que estar en condiciones de producir una salida de información. Y naturalmente, como el procesador humano de datos, el ordenador tiene que disponer de cierta posibilidad de control de forma que sepa qué hacer con estas cinco funciones.

El medio por el cual se obtienen datos del exterior y se convierten en tal forma que el ordenador los pueda usar se llama dispositivo de entrada. Similarmente, el que es capaz de producir un resultado final, se denomina dispositivo de salida. La posibilidad de almacenamiento de un ordenador se llama memoria. El almacenamiento de un ordenador es solamente un lugar donde se puede guardar información de forma que el ordenador pueda utilizarla. Las funciones de aritmética, lógica y de control se ejecutan a través de circuitos electrónicos, los cuales juntamente con ciertos tipos de almacenamiento, forman lo que se denomina La Unidad Central de Proceso del ordenador (CPU).

En el procesamiento automático de datos, los dispositivos de entrada leen los datos a procesar y los transmiten electrónicamente a la CPU. La CPU realiza las operaciones aritméticas necesarias, pudiendo hacer uso también de los datos del almacenamiento así como almacenar otros datos que sean necesarios. Finalmente la CPU transmite la información resultante a los dispositivos de salida. Estos dispositivos convierten las señales electrónicas en resultados finales de salida. Por ello la CPU hace la misma clase de cosas que el procesador humano de datos. Ambos reciben datos con qué trabajar (dependiendo de sus posibilidades de entrada), también pueden acceder al almacenamiento si lo necesitan, ejecutan operaciones aritméticas y producen un resultado; y lo más importante de todo, ambos la CPU y el procesador humano de datos son capaces de controlar todas estas funciones. El método con el que la CPU ejecuta la función de control es particularmente interesante.

Para que un ordenador sepa lo que tiene que hacer, también necesita una lista de instrucciones. Esta lista se encuentra en el almacenamiento del ordenador durante la ejecución de un trabajo, y la unidad de control del ordenador en base a las instrucciones, determina qué hay que hacer. La lista de instrucciones se llama programa, y la persona que escribe las instrucciones en un lenguaje que el ordenador entiende se llama programador.

El ordenador sólo sigue las instrucciones que el programador le ha dado, no entiende lo que esta haciendo.

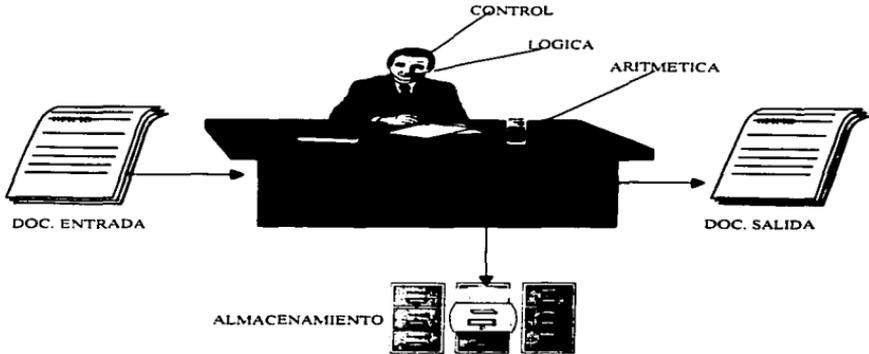
Las instrucciones que dirigen la unidad de control de un ordenador se preparan por personas y se almacenan en el almacenamiento del ordenador. El ordenador hace solamente lo que se le ordena y nada más. No comprende el trabajo que está haciendo. No tiene idea si el trabajo está bien o mal. No tiene opinión sobre el trabajo ni tampoco intuición. No adquiere experiencia haciendo el trabajo una y otra vez, porque el sólo sigue instrucciones y éstas son siempre las mismas cada vez que se ejecuta el mismo trabajo. Por ello, a pesar de existir alguna similitud en la forma que los ordenadores y las personas procesan datos, hay una enorme diferencia entre ellos.

Para poder estar en condiciones de utilizar un ordenador para procesar datos, se tiene que tener primeramente un conocimiento global de lo que se espera que haga el sistema y de como ha de hacerlo.

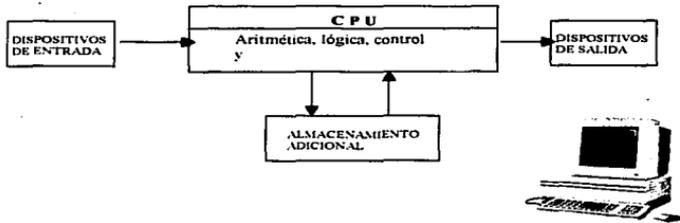
Cuando se desea utilizar ordenadores para realizar sistemas es conveniente contar con un método preciso que describa el sistema. Dicho método puede ser diseñar un diagrama que representa como funciona el sistema.

Para describir un sistema comercial manual se puede dibujar un diagrama del mismo que elimine dificultades que se encuentran en una descripción narrativa. Se utilizan símbolos especiales para: entrada, salida, almacenamiento y ejecución. El diagrama describe el sistema de una forma clara y precisa.

PROCESADOR DE DATOS HUMANO



PROCESADOR DE DATOS AUTOMATICO



I.2. FUNDAMENTOS DE BASES DE DATOS

En el campo de la informática una de las actividades más importantes será sin duda el desarrollo de las bases de datos. Cada vez es mayor el interés que se pone en los datos con miras a conseguir una mejor organización de las empresas.

Los archivos de datos que utilizan las computadoras crecen en volumen e importancia de manera impresionante.

Para lograr la flexibilidad de uso que es esencial en la mayoría de las aplicaciones comerciales es necesario tomar en cuenta dos aspectos del diseño de la base de datos. Primero, los datos deben ser independientes de los programas que los utilizan, de modo que los pueda enriquecer y reestructurar sin que resulte necesario modificar los programas existentes. Segundo, debe ser posible interrogar y explorar la base de datos sin necesidad de recurrir a la tediosa operación de escribir programas utilizando los lenguajes convencionales de programación.

Una de las principales tareas del Administrador de Base de Datos es desarrollar el modelo conceptual también llamado modelo de la empresa, este modelo expresa las entidades y sus relaciones y es la herramienta para representar la organización de los datos.

Este modelo se utiliza para organizar, visualizar, planear y comunicar ideas y debe ser independiente del sistema de manejo de datos, el modelo conceptual debe transcribirse a un modelo lógico usando como una estructura fundamental de un sistema de manejo de Base de Datos este modelo debe transportarse a un modelo físico.

Debido a que el costo de almacenamiento de datos ha disminuido notablemente y a que ha habido constantes mejoras en las facilidades para transmitirlos a distancia, las bases de datos desempeñarán un papel de primer plano en la evolución futura de la industria.

SISTEMA DE BASE DE DATOS

"La base de datos puede definirse como una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es la de servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de

los programas que los usan; se emplean métodos bien determinados para incluir datos nuevos y para modificar o extraer los datos almacenados".

"Una base de datos es cualquier conjunto de información almacenada en cualquier tipo de soporte de forma que sea accesible y manipulable por un sistema informático".

Una base de datos no es lo mismo que el programa informático que se utiliza para explotarla. Aunque muchas veces podemos oír y leer la expresión "base de datos" referida a los programas, parece más apropiado reservar a estos últimos el término genérico de "sistemas de gestión de bases de datos" (DBMS-Data Base Management System) y, en consecuencia, utilizar "base de datos" refiriéndonos al conjunto de la información almacenada.

Para poder manipular la información de una base de datos, existe un programa llamado, DBMS o Sistema Manejador de Base de Datos (Data Base Management System). Este programa debe de cumplir con las siguientes características:

- a) Integridad
- b) Privacidad
- c) Independencia de los datos

INTEGRIDAD

Se refiere a la habilidad del manejador de la base de datos de proteger la base de datos de un error o mal funcionamiento del Software o del Hardware.

Debe ser capaz de conocer un problema, reportarlo, reconstruir la parte dañada de la base y continuar en el proceso que se estaba realizando, el conjunto de estos procedimientos es a veces llamado recuperación de información.

PRIVACIDAD

Identifica la capacidad de proteger la base de datos contra cualquier intento de acceso no permitido o intento de modificación de la información existente. Como cada empresa necesita un nivel determinado de privacidad, siempre se vuelve más crítica la situación si la base de datos va a ser accesada por terminales.

INDEPENDENCIA DE LOS DATOS

Es la capacidad de que la base de datos se modifique estructuralmente y no tengan que hacerse cambios en las relaciones ya definidas ni tampoco en los programas de aplicación ya existentes. Esto quiere decir que si es necesario agregar una nueva relación, no se modifique la estructura anterior de los registros definidos ni de los programas ya existentes, sino que únicamente se altera la estructura de la parte que se agrega.

En resumen un sistema manejador de una base de datos DBMS es una pieza de software que es usada para manejar una base de datos.

Una base de datos es una colección de registros relacionados que están almacenados en dispositivos de acceso directo de una computadora. Si las relaciones no se encuentran presentes o no son necesarias entre los registros, decimos que la información que se tiene almacenada corresponde a un archivo.

Un archivo, consta de dos partes principales: los registros y los campos (estos últimos forman parte de los primeros) que en conjunto con otros archivos forman la base de datos.

CAMPO: Es la unidad elemental de datos.

REGISTRO LÓGICO: Es la parte en donde se definen las características de la información a almacenar como son: nombre del campo, tipo de datos y tamaño del campo, también se conoce como estructura del archivo.

REGISTRO FÍSICO: Son los datos en sí, los cuales se almacenarán en el orden en que se definieron en el registro lógico.

ARCHIVO: Es el conjunto de Registros Físicos.

BASE DE DATOS: Es el conjunto de Archivos relacionados entre si por medio de un campo común.

Está claro que se trata de conceptos muy relacionados, en la medida en que sería poco operativo (por no decir otra cosa) tener una serie de datos sin programa que los maneje, o programa sin datos que manejar.

Desde el punto de vista de la información almacenada, en una primera instancia, suele distinguirse entre bases de datos referenciales y bases de datos fuentes.

Estas últimas contienen todo el texto o los datos originales, mientras que las primeras remiten a la fuente original, y de la misma sólo incorporan una referencia; las primeras almacenan documentos secundarios (referencias) y las segundas primarios (originales, fuentes).

Un sistema de Base de Datos consiste en un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a ellos, este grupo integrado de programas es utilizado para dar apoyo a las Bases de Datos y de esta manera formar un sistema de Base de Datos.

OBJETIVOS DE UN SISTEMA DE BASE DE DATOS

El objetivo principal es crear un ambiente en el que sea posible guardar y recuperar información en forma conveniente y eficiente.

1. El manejo de grandes cantidades de información
2. Cuidar la seguridad de la información almacenada en la Base de Datos
3. Proporcionar a los usuarios una visión abstracta de la información

Existen dos clases de usuarios:

1. El programador de la aplicación que se encarga de diseñar los programas de aplicación que utiliza el sistema de Base de Datos. Estos programas operan sobre los datos en todas las formas acostumbradas como son: Recuperación, inserción, eliminación o modificación de información.
2. Usuario final: Este puede tener acceso a la Base de Datos únicamente por medio de la aplicación antes mencionada.

ENTIDAD

Todo lo que se puede representar en una Base de Datos, este término se utiliza para referirse a cualquier objeto distinguible que ha de representar en nuestra base.

ATRIBUTOS

Denominamos atributo a la columna o columnas en una Base de Datos y al número de atributos dominio.

DICCIONARIO DE DATOS

Es una estructura de datos que almacena la información relativa a la Base de Datos. Se trata de un archivo especial del banco de datos que contiene las definiciones descriptivas de todos los elementos de datos, registros y archivos usados por el sistema.

ARQUITECTURA DE UN SISTEMA DE BASE DE DATOS

Se divide en módulos que se encargan de cada una de las tareas del sistema en general estos módulos son:

1. **Manejador de Archivos:** Se encarga de la asignación de espacio en el disco así como de la estructuras de datos que van a emplear.
2. **Manejador de Base de Datos:** Tiene como función la conexión entre los datos de bajo nivel almacenados en las bases y los programas de aplicación.
3. **Procesador de consultas:** Traduce las proposiciones en lenguaje de consulta a instrucciones de bajo nivel que pueda entender el lenguaje de manejo de datos.
4. **El precompilador del lenguaje de manejo de datos:** Convierte las proposiciones en lenguaje de manejo de datos incrustados en un programa de aplicación en llamadas normales a procedimientos. El precompilador debe interactuar con el procesador de consultas para generar el código apropiado.
5. **El compilador del lenguaje de definición de datos.** Convierte las proposiciones en un conjunto de tablas que contienen datos tales tablas se almacenan después en el diccionario de datos.

MANEJADOR DE BASE DE DATOS

Requerimos de un módulo o parte fundamental dentro del sistema, para la simplificación, acceso y manejo de los datos, a este le denominamos manejador de Base de Datos, el cual constituye la interfaz entre los datos de bajo nivel almacenados en nuestra Base de Datos y los programas de aplicación y consultas hechos al sistema, las tres tareas principales son:

- a) Interacción con el manejador de archivos en donde el Manejador de Base de Datos traduce las diferentes proposiciones del Lenguaje de Manejo de Datos a comandos de sistema operativo de bajo nivel.
- b) Puesta en práctica de la seguridad. En este punto el manejador conserva y aplica la seguridad de los datos en el sistema de Base de Datos.
- c) Control de concurrencia. Cuando varios usuarios actualizan la Base de Datos en forma concurrente es posible que no se conserve la consistencia de los datos.

MODELO RELACIONAL

El modelo de datos relacional representa la Base de Datos como un conjunto de tablas, este modelo se compone de un cierto número de tablas llamadas relaciones cada columna de una tabla corresponde a un atributo y al conjunto de valores que abarca un atributo se le llama dominio. Al conjunto de los atributos de una relación se le llama extensión de la relación, al número de atributos se le llama grado de la relación. A las filas de las relaciones se les llama tuplas o pares-n-tuplas se llama extensión de una relación al conjunto de las tuplas que tiene en un momento dado.

En una Base de Datos de relación se divide en tres tipos de condiciones de integridad.

- 1) La tupla no puede ser nula en ningún componente de la clave.
- 2) No debe haber tuplas con claves iguales.
- 3) Que todo valor de una clave foránea aparezca también como valor de la clave primaria.

El modelo relacional se divide para su estudio en una estructura que esta compuesta por:

Una relación: Corresponde a lo que hasta ahora se ha llamado tabla.

Una tupla: Fila de la tabla.

Un atributo: Una columna.

No. Tuplas: Es la cardinalidad de la base

No. de atributos: Es el grado de la Base

La clave primaria es un identificador único para la tabla y puede ser una columna o combinación de columnas.

Una Base de Datos relacional es percibida por el usuario como una colección de diversos grados que varían con el tiempo.

MODELO JERÁRQUICO

En el modelo jerárquico, los datos y las relaciones se presentan por medio de registros y ligas. Los registros se organizan para formar conjuntos de árboles en vez de gráficas arbitrarias, esta base consiste en un conjunto de registros que se conectan entre sí por medio de ligas. Una liga es una asociación entre dos registros exclusivamente. La estructura de datos jerárquico define relaciones entre registros sobre la base de que ciertos registros deben existir antes que otros. Un registro que tienen que existir antes que otros. Un registro que tiene que existir antes que cualquier otro se llama registro-padre y el registro dependiente se llama registro-hijo o registro de descendencia. Cuando haya más de un registro hijo, cada uno con independencia debiendo su existencia al mismo registro padre se llaman fraternales. Los niveles de las relaciones padre descendencia puede ser bastante profundas, mientras un registro puede ser hijo en relación con un registro, puede ser padre de otros registros. En estructuras jerárquicas el primer registro padre en la jerarquía se llama registro raíz. Los padres siempre tienen que existir antes que sus hijos. Los registros de nivel alto tienen que existir antes que los registros de nivel bajo puedan añadirse a la estructura. Cada registro en una estructura jerárquica tiene que tener su propia clave singular relativa a su pariente.

MODELO DE RED

La estructura de red puede considerarse como una forma extendida de la estructura jerárquica, la distinción principal entre estos dos modelos es que en la estructura jerárquica en un registro hijo tiene uno y sólo un registro padre. En la estructura de la red un registro hijo puede tener cualquier número de padres una Base de Datos de RED se compone de dos conjuntos.

- a) Conjunto de ligas (Conjunto múltiple de ocurrencia liga)
- b) Conjunto de registros (Conjunto múltiple de ocurrencia de registros).

Cada liga implica dos tipos de registro, un tipo de registro padre y un tipo de registro hijo. Cada ocurrencia de un tipo de liga, consiste en una sola ocurrencia del tipo de registro padre junto con un conjunto ordenado de múltiples ocurrencias del tipo padre-hijo.

Cada ocurrencia de una liga representa una interrelación de una o muchas entre la ocurrencia padre y las ocurrencias hijos.

Una forma de establecer tales conexiones es mediante una cadena de apuntadores que se origina en la ocurrencia padre, recorre todas las ocurrencias hijo y regresa por último con su padre.

Un lenguaje de manipulación de datos de red consiste en un conjunto de operadores para procesar datos representados en forma de registros y de ligas.

MODELO ENTIDAD-RELACION

El modelo de datos Entidad-Relación (E-R) esta basado en una percepción de un mundo real que consta de una serie de objetos básicos llamados entidades, y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo especificar un esquema empresarial. Este esquema representa la estructura lógica general de la base de datos.

Una entidad es un objeto que existe y que puede distinguirse de otros objetos. La distinción se logra relacionando cada objeto con una serie de atributos que lo describen. Una relación es una asociación entre varias entidades. Los conjuntos de todas las entidades y de todas las relaciones del mismo tipo se denominan conjunto de entidades y conjunto de relaciones, respectivamente.

Una tarea importante dentro de la modelación de bases de datos consiste en especificar la forma en que se van a distinguir las entidades y las relaciones.

Conceptualmente, las entidades y relaciones individuales son distintas, pero desde el punto de vista de las bases de datos, su diferencia debe expresarse en términos de sus atributos. Para hacer estas distinciones, a cada conjunto de entidades se le asigna una llave primaria. La llave primaria es un conjunto de uno o más atributos que, juntos, permiten identificar en forma única una entidad dentro del conjunto de entidades o una relación dentro del conjunto de relaciones.

La estructura lógica general de una base de datos puede expresarse en forma gráfica mediante un diagrama E-R. Una base de datos que se ajuste a un diagrama E-R también puede representarse como una serie de tablas. Para cada conjunto de entidades y para cada conjunto de relaciones en la base de datos

existe una tabla única a la que se le asigna el nombre de conjunto de entidades o de relaciones correspondientes. Cada tabla tiene varias columnas a las que también se asignan nombre únicos.

Una limitación del modelo E-R es que no se pueden expresar relaciones entre las relaciones. La solución es emplear la agregación, que es una abstracción en la que los conjuntos de relaciones se tratan como conjuntos de entidades de más alto nivel. Así, puede considerarse a un conjunto de relaciones y los conjuntos de entidades a los que asocia como una entidad de alto nivel que se maneja en la misma forma que cualquier otra entidad.

MODELO SEMÁNTICO

Los modelos tradicionales están orientados a registros, e incluso las entidades u objetos del mundo real todavía no pueden ser directamente expresados en los modelos tradicionales. Una de las ventajas del modelado de datos semántico es que está más orientado al usuario y que no lo restringe a usar ninguna estructura de implementación particular.

En el modelo semántico los tipos de objetos y entidades son llamados clases. Una clase es un medio de estructuración y no implica ninguna implementación física. Cada entidad es un miembro de alguna clase. Las entidades en una base semántica están organizadas dentro de una colección significativa de entidades que comparten características comunes.

I.3. CICLO DE DESARROLLO DE LOS SISTEMAS

Para el desarrollo de un sistema en primera instancia se efectúa una investigación sobre las posibles soluciones de proceso de datos al problema, posteriormente se formalizan los planes para el establecimiento de los procedimientos que se utilizarán en dichas soluciones, a tales procedimientos se les conoce como diseño del sistema y por último se realiza la implantación.

El analista debe realizar el diagnóstico de los sistemas y un examen general de la situación para definir el problema, elaborando un modelo que exprese las interrelaciones entre las variables y las restricciones contenidas en el mismo, siendo necesario que trabaje en conjunto con las personas usuarias de la información y con las que deben tomar las decisiones.

El enfoque de sistemas se refiere a la manera en que la administración concibe la interrelación entre los subsistemas de la organización y sus directrices son:

- 1) Integración: Los subsistemas del sistema se deben integrar de tal manera que se aprovechen las interrelaciones y la interdependencia de los elementos.
- 2) Comunicación: Debe haber una comunicación continua entre los subsistemas.
- 3) Método Científico: Se deben aplicar las diferentes técnicas de la ciencia administrativa.
- 4) Orientación hacia las decisiones: A fin de hacer más eficaz la función administrativa de planeación y control, la administración podrá concentrar sus esfuerzos en la toma de decisiones.
- 5) Tecnología: En la medida de lo posible es recomendable que el analista utilice la tecnología moderna como auxiliar en la implantación de las técnicas que se originan a partir de las directrices anteriores; en este aspecto la computadora mejora de manera significativa la integración, la comunicación, el método científico y la toma de decisiones.

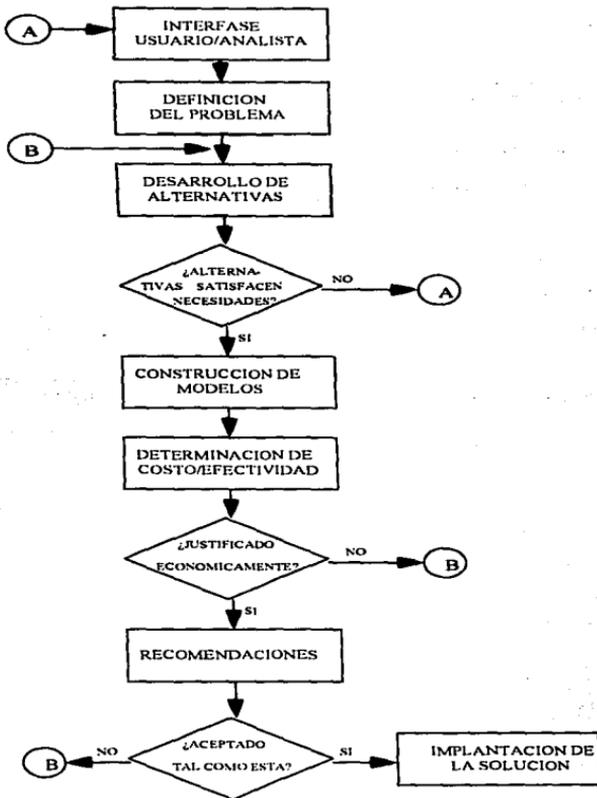
Primeramente, alguien debe pedir un análisis de sistemas, dicha petición puede venir de algún jefe de departamento, del gerente o del propio departamento de proceso de datos. Una vez aprobado por la gerencia, empezaría el análisis del sistema, preparando al finalizarlo, un reporte de lo encontrado y el cual se le

transmitirá a la gerencia, para que se tome la decisión final sobre la acción a tomar.

Dentro del contexto del enfoque a la resolución de problemas, hay seis etapas básicas para la aplicación del desarrollo de sistemas:

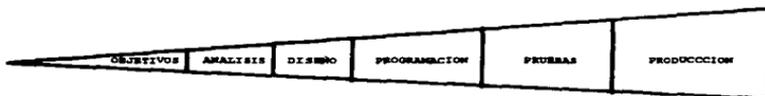
- 1.- Definición y formulación del problema
- 2.- Desarrollo de soluciones alternativas
- 3.- Construcción de modelos que formalicen las alternativas
- 4.- Determinación del costo/eficiencia de las alternativas
- 5.- Presentación de las recomendaciones
- 6.- Implantación de la alternativa seleccionada

La realización de un desarrollo de sistema tiene como uno de sus objetivos principales, maximizar la eficacia de la solución con un costo mínimo.

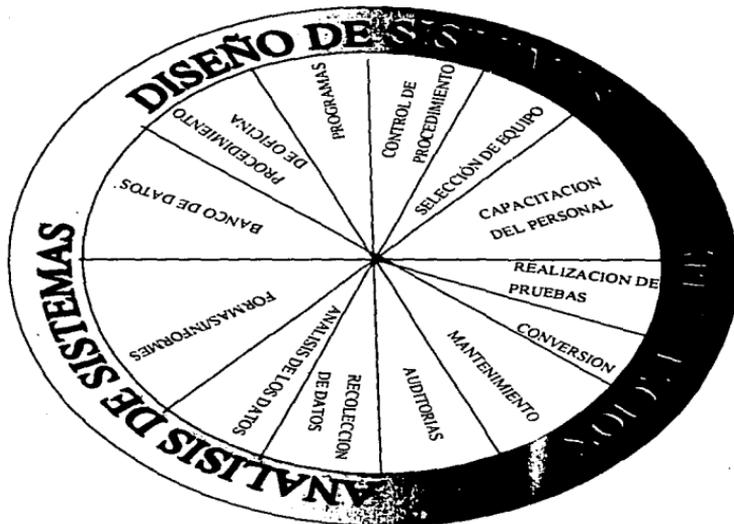


Enfoque de sistemas mediante el uso de la técnica ENALIM

- Debemos entender completamente el problema antes de diseñar una solución.
- Hacer que el usuario distinga entre sus deseos y sus necesidades.
- Más tiempo en análisis, menos tiempo en implantación.

TRADICIONAL**ENALIM****CICLO VITAL DE UN SISTEMA DE INFORMACIÓN Y LAS PRINCIPALES ACTIVIDADES ASOCIADAS CON CADA ETAPA DEL CICLO**

- 1.- Análisis de sistemas
- 2.- Diseño de sistemas
- 3.- Implantación de sistemas



Las actividades que se realizan en estas tres etapas tienen relación con las seis etapas que se describieron anteriormente:

El análisis de sistemas es análogo a la etapa de definición y formulación del problema.

El diseño de sistemas incluye las actividades de las etapas 2 a 5, es decir, desarrollo de soluciones alternativas, construcción de modelos que formalizan las alternativas, determinación del costo/eficiencia de las alternativas y por último presentación de las recomendaciones.

La implantación del sistema es en sí la implantación de las alternativas seleccionadas.

El empleo del concepto de ciclo vital es una excelente manera de mostrar la naturaleza viable y dinámica del sistema de información en sí, además constituye una exposición general de la metodología del desarrollo de sistemas.

Para contar con un sistema funcional de información que satisfaga diversas necesidades, todos los datos medibles deben ser organizados de manera que sea fácil registrarlos, almacenarlos, procesarlos, recuperarlos y comunicarlos según los requieran los usuarios, proporcionar información a los administradores, en apoyo de las actividades de planeación, control y toma de decisiones, haciendo uso del banco de datos, el cual es la base del sistema de información. En realidad, el banco de datos es una colección de archivos de datos. A su vez, el archivo es un conjunto de registros formados por campos de datos.

ANÁLISIS DE SISTEMAS

El análisis de sistemas consiste en separar estos, en partes que los constituyen, con el fin de estudiarlas y evaluarlas para ver si hay un mejor método que satisfaga las necesidades de la administración.

El análisis de sistemas hace uso tanto de métodos cuantitativos, como de factores cualitativos, como el juicio, la creatividad, el sentido común y la experiencia.

Cuándo y dónde se debe empezar un análisis de sistemas es hasta cierto punto arbitrario. Se pueden requerir un gran número de sesiones de discusión y varios estudios y experiencias empíricos antes de efectuar el análisis de sistemas.

Las razones básicas para iniciar un análisis de sistemas son:

- Solución de problemas
- Nuevas necesidades
- Implantación de una nueva idea o tecnología
- Mejoramiento general de los sistemas

Comúnmente la aplicación del análisis de sistemas al desarrollo de un sistema de información abarca un período más largo que el que se requiere para la resolución de problemas en general.

El analista de sistemas es la persona que sirve de interfaz entre los usuarios del sistema de información y los técnicos que trabajan en el sistema, tales como programadores, administradores del banco de datos y operadores de las máquinas, identifica que tipo de información se requiere, en que forma se requiere y las fuentes de información, convierte en un plan detallado los requerimientos de información que se especificaron en la etapa de investigación. Debe poner atención en: formas de entrada y salida, la estructura física y lógica del banco de datos, rutinas de escritorio, programas para computadora y controles del procesamiento. Asimismo debe saber las fechas de arranque, capacitar y coordinar al personal que empleará el sistema, instruir al personal de computación; instalar nuevos procedimientos y formas, realizar pruebas con el sistema y mantenerse alerta para percibir descuidos y omisiones en las actividades de las etapas descritas anteriormente.

A fin de efectuar un análisis de sistemas, el analista debe definir primero el alcance, es decir, los límites, dentro de los cuales va a trabajar. Enseguida debe formular una proposición para efectuar un análisis de sistemas. Dicha proposición es una comunicación que permite, tanto al solicitante como al usuario conocer lo que se ha puesto en claro en la fase de análisis.

Fuentes de datos para estudio del análisis de sistemas

- El sistema actual
- Otras fuentes Internas (personas, documentos y relaciones)
- Fuentes externas

Ventajas del estudio del Sistema Actual

- 1.- Efectividad del sistema actual
- 2.- Ideas para el diseño
- 3.- Reconocimiento de recursos
- 4.- Conocimiento para hacer la conversión
- 5.- Punto de partida común

Desventajas

- El costo
- Obstáculos innecesarios

Las Fuentes Internas.- Las necesidades de información las pueden indicar mejor quienes utilizan esa información. una fuente secundaria es la papelería que se utiliza actualmente.

Las Fuentes Externas.- El estudio de sistemas de información similares que existen en otras organizaciones, libros y publicaciones especializadas, instructivos y folletos que proporcionan los proveedores de equipo y sistemas de computación.

MÉTODOS PARA LA RECOLECCIÓN DE DATOS

- **Análisis de los niveles de decisión:** El analista entrevista a los funcionarios clave con el fin de clasificar los recursos principales de la organización.
- **Análisis del flujo de información:** Al analista le interesa saber que información necesita una persona respecto a otra y a su vez que información debe proporcionar.
- **Análisis de Entradas y Salidas:** Se describe cada entrada y salida sin aclarar la forma en que la entrada se convirtió en salida.

Hay 3 métodos principales para obtener datos:

- 1) El análisis de los niveles de decisión divide el sistema en base a los recursos.

Se definen los puntos de decisión que controlan esos recursos y la administración formula las reglas de decisión adecuadas. El analista determina la manera de producir la información necesaria ara satisfacer los requisitos de esas reglas de decisión.

- 2) El análisis del flujo de información indica cual es necesaria, quien la requiere y donde puede obtenerse.
- 3) El análisis de entradas y Salidas muestra simplemente la entrada de datos y la salida de información que se realizan en el sistema, sin ocuparse de las decisiones que originó dicha información.

Los riesgos durante la recopilación de datos son:

- 1) Utilizar datos incorrectos
- 2) Hacer suposiciones incorrectas, conscientes o inconscientes
- 3) Verificar sin necesidad todas y cada una de las fuentes potenciales de información.

Entre las técnicas para analizar los datos de estudio figuran las relaciones, factores, diagramas de flujo, tablas de decisión, gráficas, etc.

El analista de sistemas reúne toda la información sobre el equipo que se está usando actualmente: el personal que hace el trabajo, los documentos y reportes requeridos y el volumen de datos procesados en los diversos intervalos de tiempo, junto con cualquier otra información pertinente respecto al trabajo. La recopilación y organización de esta información se pueden clasificar, usualmente, en cinco categorías:

1. Métodos directos: Estos incluyen entrevistas y observación del sistema como existe actualmente. El analista entrevistará a las personas que están trabajando para determinar que es lo que hacen.
2. Documentación: El analista reúne muestras de todos los documentos escritos usados en el sistema actual, tanto aquellos creados manualmente como los que son preimpresos.
3. Proceso: El proceso incluye la operación completa del sistema y define las interrelaciones de equipo, personal y documentos. También se incluyen aquí los datos, tipos de archivos, proceso y salida deseada.
4. Organización del análisis: Después que se ha reunido la información, se organiza en alguna forma bosquejada o gráfica que presente el panorama completo.
5. Necesidades: Como parte de su trabajo, el analista entrevistará a todo el personal relacionado con el trabajo para encontrar los reportes, documentos, formas, etc., que serían los más útiles para ellos al hacer su trabajo. Todos los empleados y la gerencia deben ser incluidos en esta fase.

Cuando se ha reunido la información sobre el sistema actual, se consideran todas las alternativas al sistema.

Una vez terminado el análisis se debe preparar el informe de terminación del análisis del sistema, el cual servirá para comprobar si se tomó en cuenta todos los requisitos y necesidades de la organización.

Estas técnicas sirven también para comunicar los resultados a diversas personas y para redactar el informe de terminación del análisis de sistemas. Las dos áreas principales del informe de terminación son:

- 1) Las recomendaciones con respecto al diseño preliminar
- 2) La viabilidad del trabajo de sistemas

La viabilidad comprende cuatro áreas:

- a) Viabilidad técnica
- b) Viabilidad económica
- c) Viabilidad de operación
- d) Viabilidad de programación

El resultado final del análisis de sistemas conducirá a una de estas decisiones:

- 1) Suspender todo trabajo de sistemas con relación con el proyecto.
- 2) Esperar cierto tiempo, hasta que ocurran otros eventos, momento en el cual será conveniente seguir adelante.
- 3) Modificar el trabajo actual o combinarlo con otro proyecto de sistemas.
- 4) Continuar con el trabajo sujetándolo a consideraciones posteriores, en especial las de costo/efectividad.
- 5) Continuar con el trabajo de sistemas, sin restricción alguna.

DISEÑO GENERAL DE SISTEMAS

Una vez analizado el problema se especifican los formatos para los archivos de entrada y salida. Se construye el diagrama de la corrida para mostrar el flujo de datos y las corridas de proceso. El diagrama estructural nos da una panorámica general de lo que sucede en cada corrida de la computadora. Y, finalmente, el diagrama técnico y las tablas de decisiones detallan los procedimientos específicos que se usarán en cada corrida. Por consiguiente, el diseño representa un plano exacto y detallado de la operación completa.

El diseño de sistemas se refiere a la formulación de especificaciones para el nuevo sistema o subsistema propuesto de manera que satisfaga los requisitos determinados durante la fase de análisis.

El diseño de sistemas puede definirse como el acto de delinear, planear, bosquejar o disponer muchos elementos reuniéndolos en un conjunto viable y unificado.

La fase de diseño se ocupa de como debe desarrollarse el sistema para que pueda satisfacer las necesidades de los usuarios.

Elementos que deben conocerse en el Proceso de Diseño

- 1.- Los recursos de la organización: El analista procura obtener un provecho óptimo de los recursos con que cuenta la organización en cuanto a personal, máquinas, materiales, dinero y métodos, ya que uno de los objetivos del diseño de sistemas consiste en utilizar esos recursos tan efectivamente como sea posible.
- 2.- Las necesidades de información de los usuarios: El objetivo primordial del sistema consiste en proporcionar información que satisfaga las necesidades de información de los usuarios. Aunque rara vez puede diseñarse un sistema que pueda satisfacer por completo las necesidades de todos los usuarios.
- 3.- Las necesidades de otros sistemas: Constituyen objetivos hacen los cuales se dirige el diseño y restricciones a lo que puede abarcar el diseño final.
- 4.- Los métodos de procesamiento de datos: La capacidad de cada método para realizar operaciones con los datos influye en el diseño específico y en la operación del sistema.

- 5.- Las operaciones con los datos: Dependiendo del diseño final se tomará la decisión de que operaciones serán incluidas y en que secuencia se realizaran.
- 6.- Las herramientas del diseño: Durante el proceso de diseño, el analista se ayuda mucho con el uso de diagramas de flujo, tablas de decisión y técnicas de modelado.

ETAPAS BÁSICAS DEL PROCESO DE DISEÑO

- 1.- Definir el objetivo del sistema: Generalmente puede definirse destacando determinadas características de todas las necesidades de información. Cuando un sistema se ha diseñado para alcanzar un objetivo, generalmente posee cierta flexibilidad en cuanto a la forma de lograr dicho objetivo.
- 2.- Desarrollar un modelo conceptual: Si el analista considera las necesidades específicas de información, la estructura de la organización y las diversas restricciones, podrá producir un modelo de diseño específico.
- 3.- Aplicar las restricciones: La tarea de lograr una combinación buena u óptima de recursos y objetivos es un problema bastante complejo que el analista debe resolver durante la fase de diseño del sistema.

MÉTODO PARA DEFINIR LAS ENTRADAS, LAS SALIDAS Y LAS ACTIVIDADES DE PROCESAMIENTO

- 1.- Identificar el resultado más importante del sistema.
- 2.- Señalar los datos específicos necesarios para obtener ese resultado.
- 3.- Identificar los datos específicos de entrada necesarios para obtener los campos de información.
- 4.- Describir las operaciones de procesamiento de datos, particularmente los algoritmos lógicos y de cálculo, que deben aplicarse a los datos de entrada para producir la información deseada.

- 5.- Identificar los elementos de entrada que se pueden introducir una sola vez y quedar almacenados para usarlos en operaciones subsecuentes de procesamiento.
- 6.- Seguir efectuando los pasos 1-5, para cada resultado requerido y por orden de prioridad, hasta haberlas considerado en su totalidad.
- 7.- Establecer el banco de datos que pueda sustentar al sistema en la forma más efectiva, considerando las necesidades, los métodos de procesamiento y los puntos comunes de los datos.
- 8.- En base a las restricciones impuestas al desarrollo, en las prioridades y en las estimaciones de costo, eliminar los casos extremos de entrada, salida y procesamiento.
- 9.- Señalar los diferentes puntos de control para regular las actividades de procesamiento y garantizar la calidad.
- 10.- Diseñar los formatos de entrada y salida que mejor se adapten al diseño del sistema.

IMPLANTACIÓN

La implantación involucra dos procedimientos:

- 1) La preparación del dato de entrada para que esté listo para el proceso
- 2) La escritura del programa para obtener el proceso necesario.

Los programas son procesados y checados hasta que procesen exitosamente los datos de prueba. Finalmente, estarán listos para ser usados en el nuevo sistema y serán usados cada vez que se requiera procesar.

En el proceso de implantación intervienen la capacitación y la instrucción del personal, la prueba del sistema, así como la conversión de sistemas.

Las personas son el elemento fundamental de todo sistema. Es necesario proporcionarles instrucción y capacitación adecuadas, tanto al principio como posteriormente, para que el sistema logre su objetivo.

Si se quiere implantar con éxito un nuevo sistema de información, a toda persona que participe en él se le debe informar, en primer lugar, de las responsabilidades que asume y, en segundo, de lo que va a obtener del sistema.

Generalmente, el término "instrucción" se usa cuando se trata de informar a los usuarios con respecto a lo que el sistema requiere y lo que puede proporcionar.

Por diversas razones, a menudo esta instrucción se reduce al mínimo o se omite durante el periodo de implantación de un sistema.

La capacitación del personal de operación presenta dos aspectos que el analista debe considerar. Primero, este personal debe ser entrenado en el manejo del nuevo sistema. Segundo el entrenamiento debe continuar a medida que se va modificando el sistema o se va necesitando nuevo personal.

Métodos de Instrucción y Capacitación del Personal

- 1.- **Seminarios e instrucción en grupo:** Este método permite llegar simultáneamente a un gran número de personas, y es muy útil cuando se trata de presentar un panorama general del sistema.
- 2.- **Capacitación a base de procedimientos:** Mediante este método se proporcionan al empleado los procedimientos escritos que describen sus actividades.
- 3.- **Preceptoría:** Capacita e instruye no sólo al personal de operación, sino también a los usuarios de los productos del sistema.
- 4.- **Simulación:** Consiste en simular el ambiente de trabajo. La práctica real demuestra que se cometen menos errores y se evitan las repeticiones cuando el empleado se enfrenta a la verdadera situación.
- 5.- **Entrenamiento directo:** Capacitar al personal de operación consiste en ponerlo a realizar el trabajo.

Realización de pruebas del sistema

La prueba es la actividad final que precede a la implantación. En realidad, las pruebas parciales se efectúan a muchos niveles, el mayor de los cuales lo constituye la prueba general del sistema.

- La prueba del sistema es una actividad que, igual que la capacitación del personal, exige una planeación cuidadosa por parte del analista.

La prueba del sistema de computadora implica probar todos los programas que se implantarán para apoyar el sistema. Todos los elementos de entrada se preparan y se procesan con el fin de producir los resultados previstos, los cuales se verifican en cuanto a exactitud y confiabilidad.

CONVERSIÓN DE SISTEMAS

Cuando se aplica al sistema de información, el término conversión, describe los cambios introducidos en las actividades relacionadas con la recopilación, almacenamiento, recuperación y procesamiento de datos, y con la presentación de la información.

La conversión del equipo: Actualmente, la conversión rara vez exige la reprogramación. En muchos casos, la nueva computadora compatible con el equipo antiguo. Dicho de otro modo, la estructura física y la operación del nuevo equipo no tienen ningún efecto en la lógica de los programas.

La conversión del método de procesamiento de datos: Esta conversión describe el cambio de un método de procesamiento de datos a otro diferente.

Generalmente con la conversión del método de procesamiento de datos va aparejada una serie de extensos cambios de procedimiento.

La conversión de procedimientos: La conversión de procedimientos puede abarcar tanto el cambio de actividades como la alteración de la secuencia en que se realizan esas actividades.

Si es capaz de relacionar lógicamente las actividades de conversión de acuerdo con el tipo que representan, el analista tendrá una buena idea del enfoque que debe aplicar a la conversión general de los sistemas de la organización.

ENFOQUES DE LAS CONVERSIONES

- 1.- **Conversión directa:** La conversión directa es la implantación del nuevo sistema con la supresión inmediata del anterior.
- 2.- **Conversión en paralelo:** La conversión en paralelo describe una situación en la que el viejo y el nuevo sistema operan simultáneamente durante cierto tiempo.
- 3.- **Conversión modular:** Se realiza la implantación por partes de un sistema, es decir, se puede instalar en una determinada área, donde se pone a prueba, luego en una segunda área, y así sucesivamente.
- 4.- **Conversión por fases:** Se lleva a cabo la implantación fragmentando el sistema, se va instalando cada segmento del nuevo sistema, hasta implantarlo todo.

El éxito de la conversión de un sistema depende en gran parte de la forma en que el analista prepare la creación y la conversión de los archivos de datos que requiere el nuevo sistema.

En la práctica, generalmente el plan de conversión se divide en dos etapas, que podrían denominarse: plan general y plan específico. Como ya se dijo, el plan general de conversión señala la programación del trabajo de sistemas que se efectúa durante la fase de desarrollo. El plan específico, elaborado en las etapas finales de prueba, indica los procedimientos especiales de entrenamiento del personal, el plan y el programa para la creación y conversión de archivos, los criterios aplicables para la aceptación y los procedimientos especiales para iniciar el control.

Para lograr una exitosa implantación del sistema, ayudará la preparación de planes de conversión bien concebidos, que luego se comunicarán a todo el personal interesado de la organización.

SECUELA DE LA IMPLANTACIÓN

Una vez implantado el nuevo sistema, la participación del analista no termina necesariamente.

Uno de los problemas que afrontará el analista durante el periodo posterior a la implantación consiste en distinguir entre las mejoras y otros refinamientos que se le sugieren, y los verdaderos problemas que presenta el sistema.

Las sugerencias de mejoras al sistema son bien recibidas y alentadas; pero dichas sugerencias se reunirán y evaluarán una vez terminada la implantación.

Una vez implantado el sistema, el analista desempeña la función de consultor. Inmediatamente después de la implantación, ayuda al personal de operación y a los usuarios a entender el nuevo sistema, asistiéndolos en la solución de los problemas que se presenten.



CAPITULO II
ANALISIS POR SECCION
CONTROL DE PROFESORES



CAPITULO II. ANÁLISIS POR SECCION (CONTROL DE PROFESORES)

II.1. Funciones de la Coordinación

II.2. Descripción del Control de Profesores

II.3. Planteamiento de la Solución

II.3.1. Identificación de los Requerimientos del Usuario

II.3.2. Propuesta de Solución

II.3.3. Información Requerida

CAPITULO II**ANÁLISIS POR SECCIONES
(SECCIÓN DE CONTROL DE PROFESORES)**

Las necesidades constantes de desarrollo y mejoramiento de las actividades administrativas que realiza la Coordinación de Ingeniería en Computación del plantel obligan al uso de métodos automatizados de manejo de información para un mejor desempeño de sus funciones.

II.1. FUNCIONES DE LA COORDINACIÓN

La Coordinación de la carrera de Ingeniería en Computación es el área encargada de realizar la apertura de grupos, formulación de horarios, así como de la asignación de profesores de acuerdo a las necesidades de cada semestre.

Según el manual de procedimientos Académico Administrativo, la Jefatura de Carrera es una instancia central en el desarrollo académico del plantel, ya que su compromiso es atender la planeación, coordinación y seguimiento de las acciones de docencia, investigación, extensión y apoyo administrativo del área correspondiente y el Secretario Técnico es uno de los elementos básicos para que el trabajo de cada carrera se desempeñe de la mejor manera posible, ya que el responsable funge como un apoyo permanente en las labores cotidianas de la carrera respectiva, de acuerdo a sus necesidades de índole académica y administrativa.

De las funciones de la Coordinación vamos a centrar nuestra atención solamente en las actividades que tienen que ver con la contratación de profesores, la evaluación de los mismos y su control de asistencia.

En cuanto a la contratación del personal académico el objetivo es designar al personal académico que cubra cuantitativa y cualitativamente los requerimientos del área en sus labores de docencia, investigación y extensión. Dicha contratación se supedita a las necesidades de la planta académica, lo que implica ajustarse al banco de horas correspondiente.

Cuando el profesor desea renunciar a su cargo entrega carta de renuncia al jefe de carrera el cual elabora propuesta de renuncia y selecciona un nuevo profesor para sustitución.

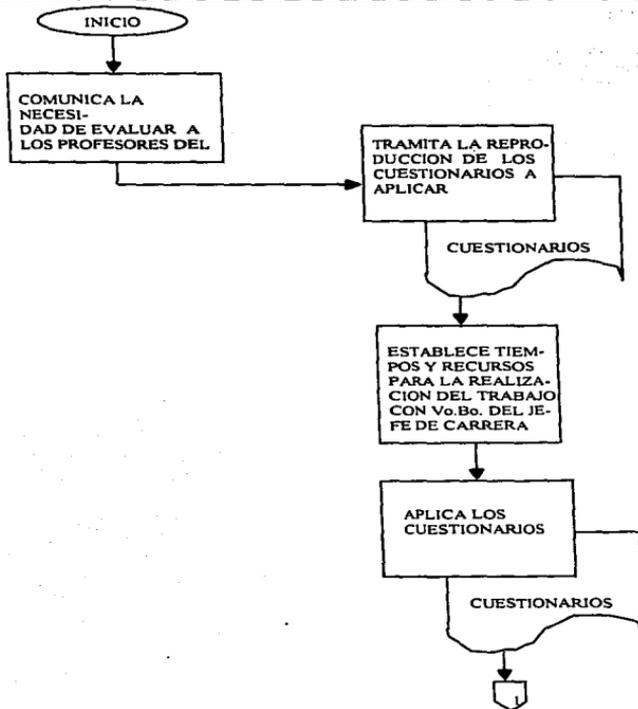
El objetivo de la aplicación de la evaluación de profesores es valorar semestralmente el desempeño de los profesores para obtener información que permita fortalecer logros académicos del área.

En cuanto al control y verificación de asistencia del personal académico, el objetivo es supervisar la asistencia de los profesores del área para verificar la impartición de sus clases en los días y horarios asignados.

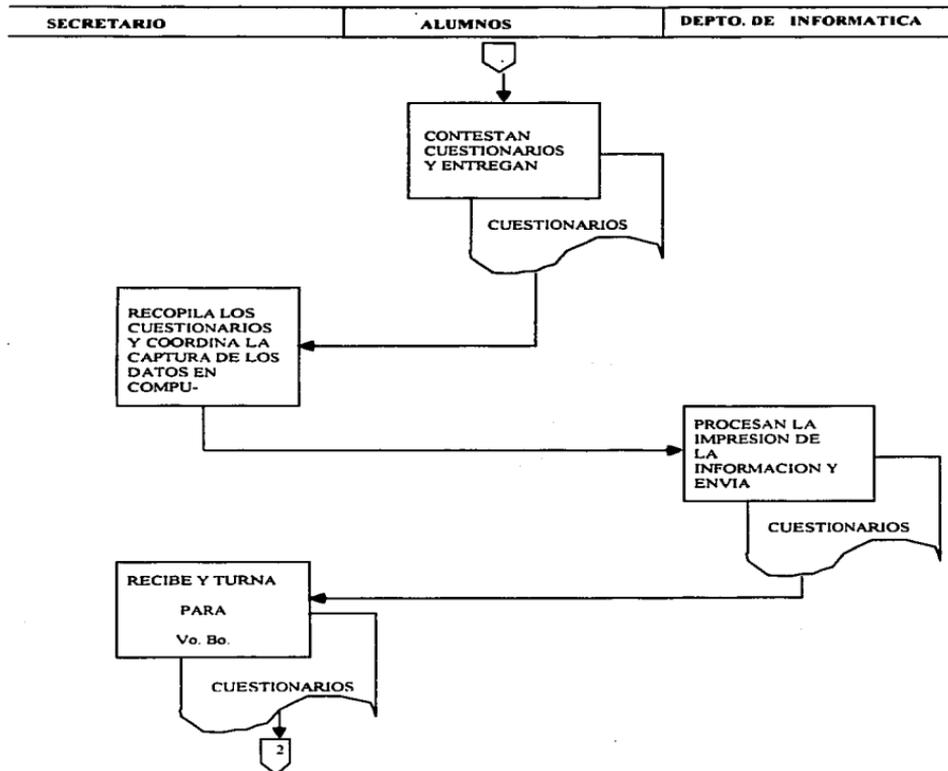
SEGUIMIENTO DE LA EVALUACION DE PROFESORES

JEFE DE

SECRETARIO



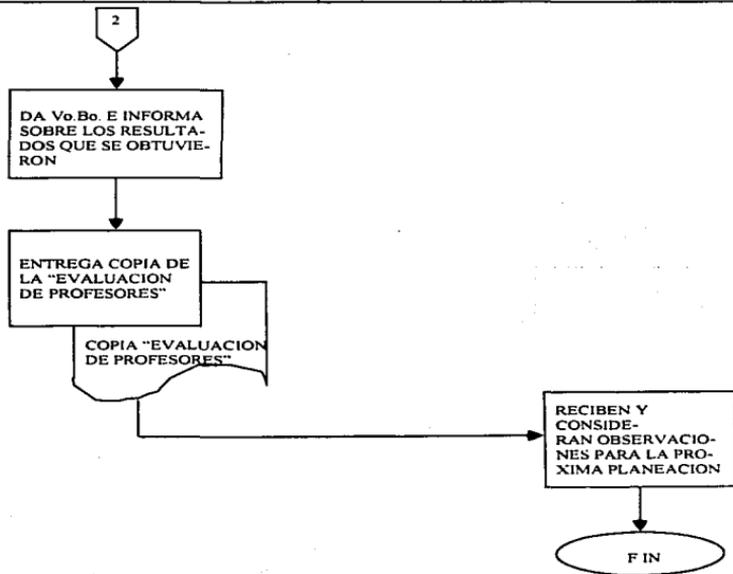
SEGUIMIENTO DE LA EVALUACION DE PROFESORES



SEGUIMIENTO DE LA EVALUACION DE PROFESORES

JEFE DE

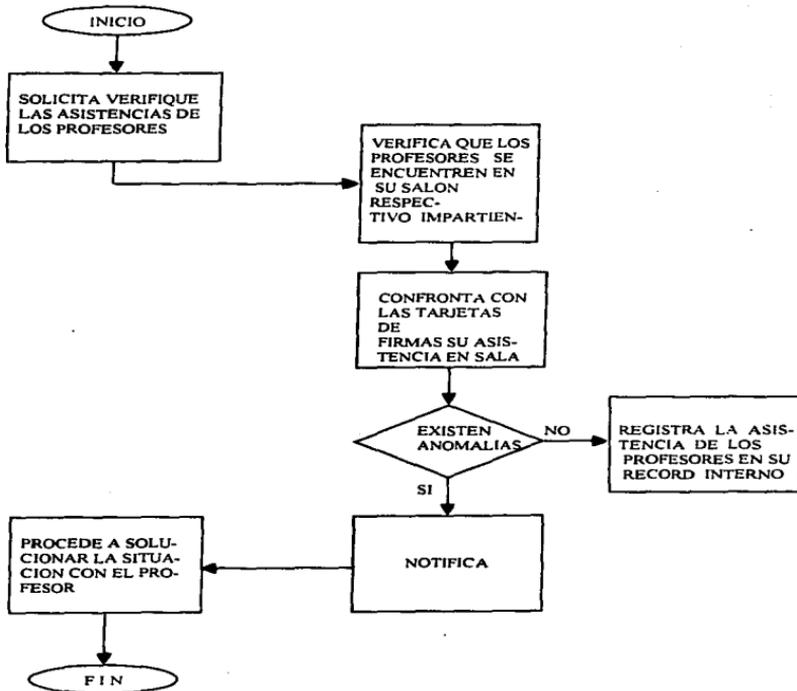
PROFESORES



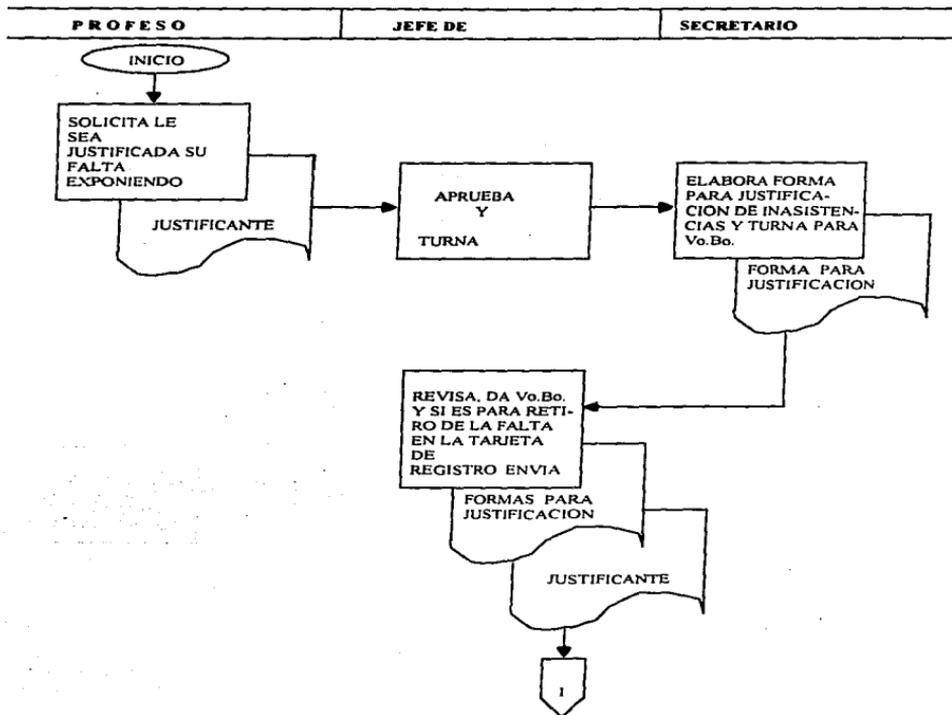
CONTROL Y VERIFICACION DE ASISTENCIAS DEL PERSONAL

JEFE DE

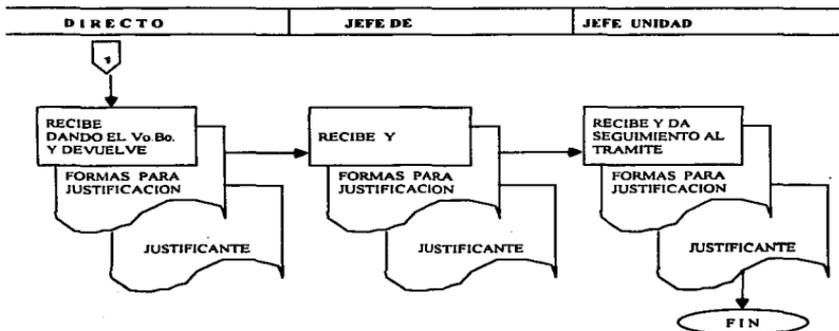
SECRETARIO



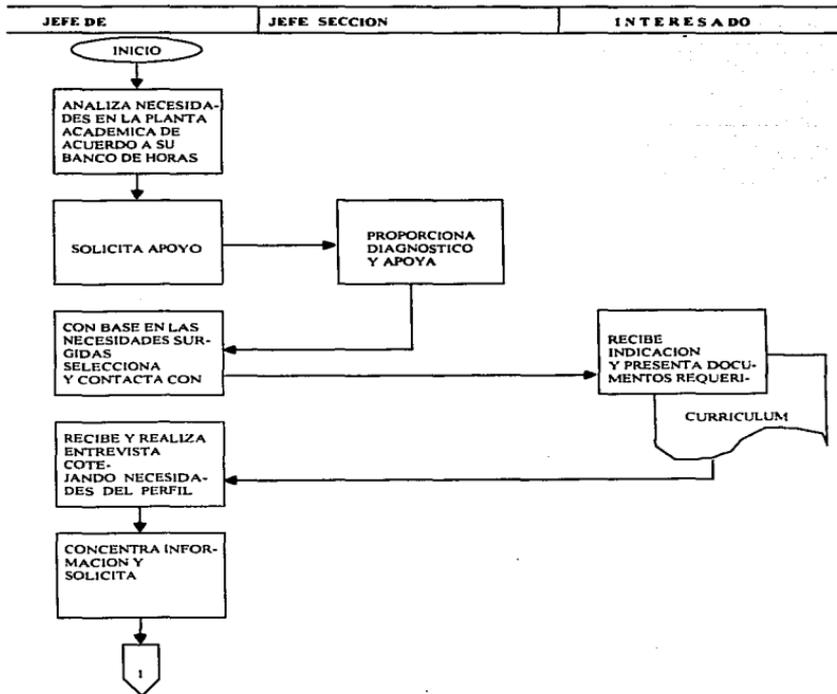
JUSTIFICACION DE INASISTENCIA DEL PERSONAL



JUSTIFICACION DE INASISTENCIA DEL PERSONAL

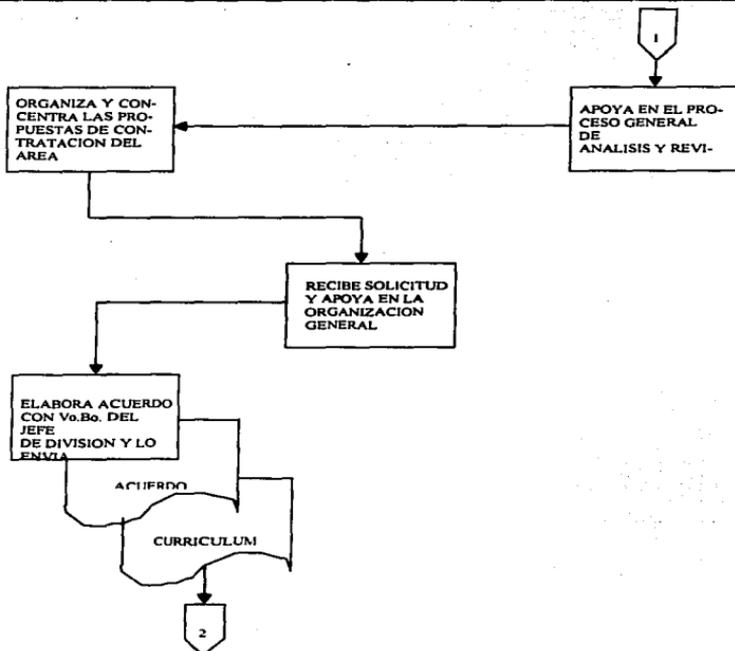


CONTRATACION DE PERSONAL

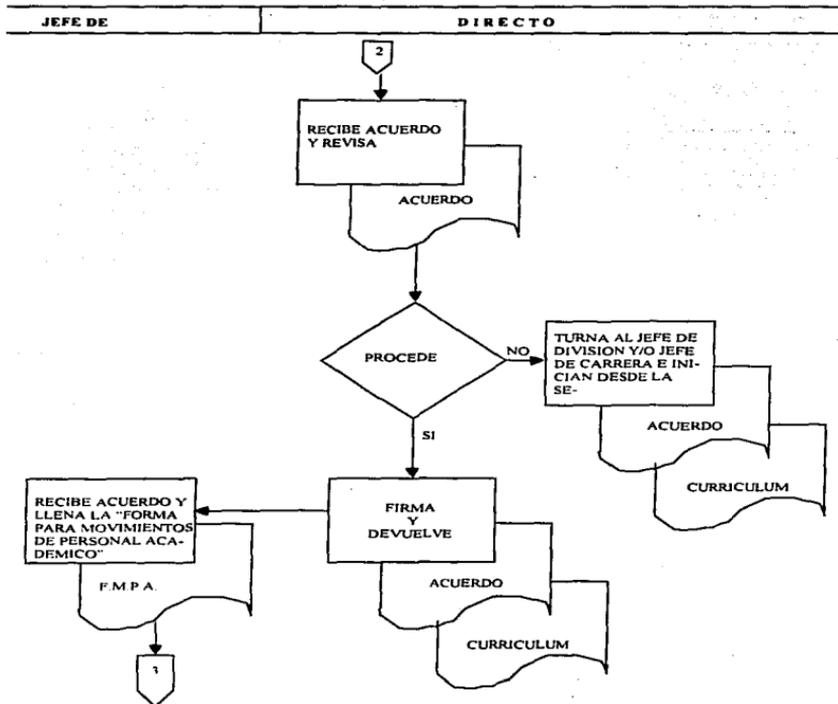


CONTRATACION DE PERSONAL

| | | |
|---------|--------------|------------|
| JEFE DE | JEFE SECCION | SECRETARIO |
|---------|--------------|------------|

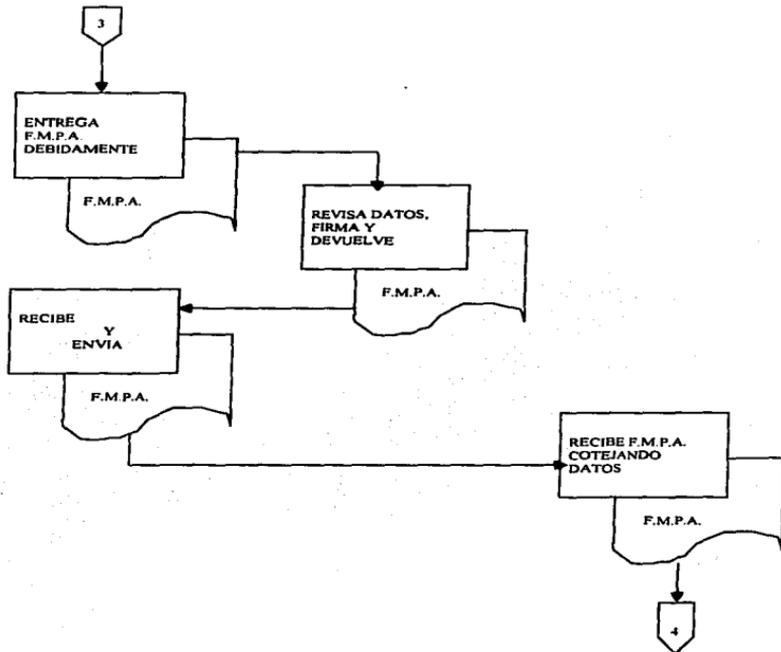


CONTRATACION DE PERSONAL



CONTRATACION DE PERSONAL

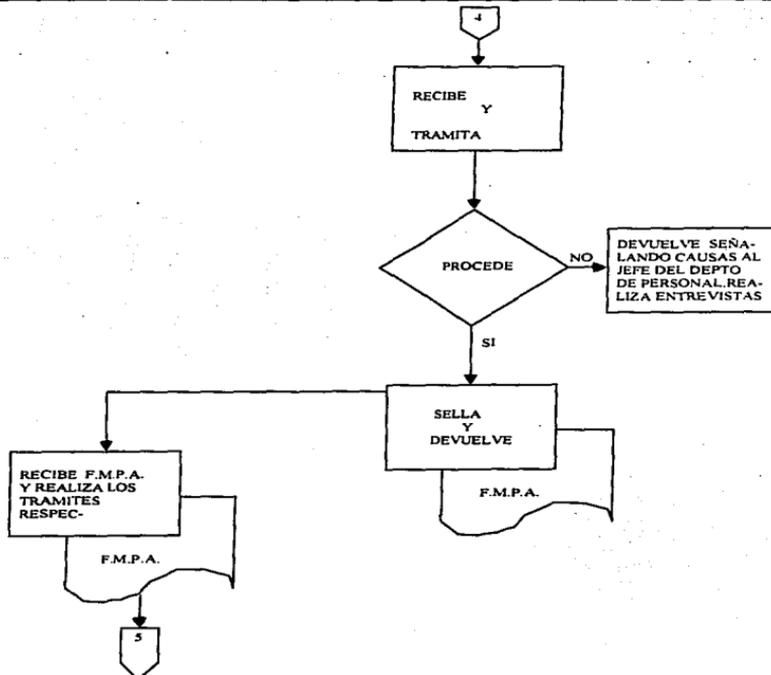
| | | |
|-----------------|------------|-------------------------|
| JEFE DE CARRERA | INTERESADO | JEFE DEPTO. DE PERSONAL |
|-----------------|------------|-------------------------|



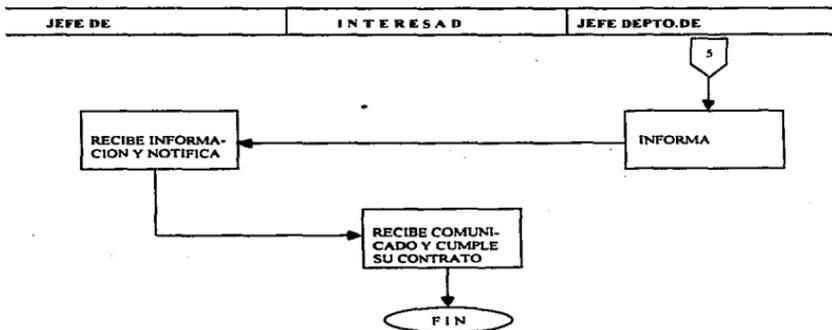
CONTRATACION DE PERSONAL

JEFE DEPTO. DE PERSONAL

DIRECTOR GENERAL DE PERSONAL



CONTRATACION DEL PERSONAL



PROCESO ADMINISTRATIVO.

Para una mayor comprensión del sistema, es pertinente hacer una reseña del proceso administrativo a automatizar.

El proceso sigue esencialmente los siguientes puntos:

- 1.- Inicio de semestre: Se generan los grupos - materia para el semestre buscando posteriormente concordar la información con la contenida dentro del Banco de Horas.
- 2.- En caso de existir modificaciones, estas se realizan (altas, bajas, términos de interinatos, etc.).
- 3.- Se elaboran las **PROPUESTAS** de horarios para profesores.
- 4.- A partir del punto anterior se obtendrán, los horarios reales de impartición de los grupos materia y la reafirmación del total de grupos - materia a cubrir por los profesores.
- 5.- En el caso de existir modificaciones, se realizan.
- 6.- En el caso de que por los movimientos las horas por ejercer de un profesor aumenten o disminuyan, es indispensable que se realicen acuerdos, para informar al departamento de personal sobre la procedencia y destino de las horas.

Con lo expresado anteriormente, se puede considerar que el proceso es simple, pero ahora bien, se deberá cuidar que no se exceda el total de horas por categoría, al igual que en el momento en que algún profesor exceda o no cumpla con el total de horas por ejercer se tendrá una variación no deseada dentro del sistema.

Deberemos recordar que los puntos 5 y 6, se pueden dar en cualquier momento durante el semestre, al no poder prever sobre todo, la incapacidad física o defunción de algún profesor.

Al no existir comunicación entre los diferentes módulos del sistema **CRONOS**, encontramos el principal punto para la sustitución del mismo, debido a que llega el momento en el que se puede perder el control de la información, y se depende en gran medida de los reportes emitidos, al igual que de los reportes realizados fuera de línea, de la misma forma, no existe una comunicación del sistema entre las diferentes coordinaciones, para compartir información. Lo cual va en contra de la automatización de procesos.

PROCESO DISTRIBUIDO

La filosofía principal, sobre la cual fue diseñado el sistema **ZEUS**, no es otra que la de proceso distribuido (Cliente / Servidor); esta nos plasma la necesidad de contar con 2 elementos fundamentales:

- **Cliente.**
- **Servidor.**

De esta manera, el cliente realiza una serie de pedidos al servidor; y este a su vez proporcionara la información solicitada, como podemos ver es un trabajo conjunto, con lo que se consigue una división en las tareas de procesamiento. Las características deseables del cliente son:

- Gran capacidad de memoria.
- Gran espacio de almacenamiento.
- Gran poder de procesamiento.
- Poder de impresión.

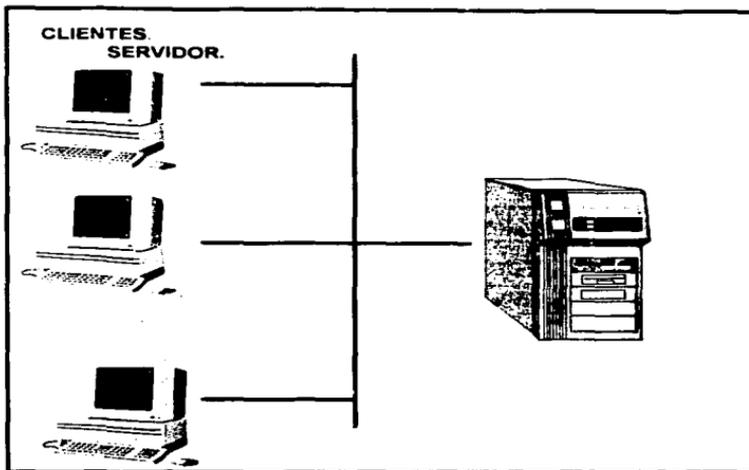
Estas características son deseables, ya que en dicha computadora se realizara el procesamiento y actualización del banco de datos del sistema, así como la impresión de las **PROPUESTAS**, las Tarjetas de Control de Asistencia y los Reportes. Por otra parte el cliente solamente necesitara.

- Memoria para ejecutar una aplicación.

Recordemos que el programa para actualizar nuestro banco de datos (**ZEUS**), deberá encontrarse en el servidor de archivos, con lo que evitaremos en caso de modificaciones omitir la actualización a alguna estación de trabajo.

Como se puede observar en la figura, este tipo de configuración nos permite, que una gran cantidad de usuarios consulten o modifiquen la información contenida dentro de nuestro banco de datos, al mismo tiempo que comparten otros recursos como impresoras. La topología de la figura; es la empleada en la actualidad en el edificio de gobierno de la **E.N.E.P. Aragón** y es conocida como Topología de Bus basada en una red Novell, conectada por cable coaxial.

Es conveniente contar con un sistema de respaldo, para prevenir las contingencias que pudiera tener el servidor de archivos, y de esta manera salvaguardar nuestros datos.



Este tipo de configuración nos permite contar con un medio seguro y único para el almacenamiento de nuestros datos, algunas características primordiales del mismo son:

| HECHO. | IMPLICA. |
|--|---|
| Concentración de los datos en un solo lugar. | Información actualizada en cualquier momento. Carencia de redundancias innecesarias en los medios de almacenamiento. |
| Bajo costo en el equipo de computo (clientes). | No resulta indispensable HDD, grandes cantidades de memoria ó impresoras en las estaciones de trabajo. |
| Compartición de recursos. | Compartir periféricos conectados al servidor: Impresoras, módem, etc. |

DIAGRAMA GENERAL DEL SISTEMA "ZEUS".

Tomando en consideración, que uno de los puntos fundamentales de este proyecto es la integración de múltiples sistemas; el paso inicial dentro de nuestro análisis será identificar la estructura general del mismo; para de esta forma, comprender la importancia del modulo **PROFESORES** dentro del sistema.

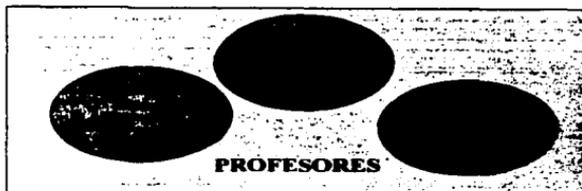


Como podemos observar en la figura, se cuenta con un programa principal llamado **ZEUS**, el cual servirá como integrador del sistema.

De la misma manera se puede observar que dicho sistema se subdivide en cinco:

- **BANCO DE HORAS.** Auditor de horas en asignación.
- **HORARIOS NORMALES.** Generador de formatos de alta de horarios normales y administración del banco de grupos para inscripción.
- **PROFESORES.** Generador de banco de profesores y propuestas.
- **EXTRAORDINARIOS.** Generador de formatos para alta de grupos extraordinarios y administración de actas.
- **UTILERIAS.** Utilerías del sistema integral.

De la figura anterior se deduce que los diversos sistemas contemplados en el segundo nivel se encuentran completamente dispersos (Sistemas cerrados); mas sin embargo los sistemas cuentan con una serie de procesos de comunicación para realizar la integración de la información, de esta manera; ninguno de los sistemas se encuentra aislado.



De la figura que se encuentra a la izquierda; tomando en consideración a **PROFESORES** como un conjunto universo podemos notar que los demás módulos son parte integral del

mismo; esto es debido a la gran interconexión que existen entre sistemas. Esto es apoyado con una serie de programas de intercomunicación entre ellos; los cuales se emplean para realimentar a cada uno de los subsistemas; la intención de esto es generar subsistemas cerrados. Como se indico anteriormente, cada uno de los subsistemas es responsable de su información.

Esta unión que existe, es uno de los principales problemas encontrados dentro del análisis inicial, debido a que a simple vista no se encontraban en forma real los límites y responsabilidades dentro de tres módulos principales, los cuales son:

- > **BANCO DE HORAS.**
- > **PROFESORES.**
- > **GRUPOS NORMALES.**

De esta manera, lo primero fue crear una tabla de responsabilidades de información, para de esta forma no interferir o crear redundancias innecesarias y sobre todo crear la verdadera integración de los mismos, la presente tabla esta creada desde la perspectiva del modulo **PROFESORES**:

| MODULO. | RESPONSABILIDADES Y CARACTERÍSTICAS ESPECIALES. |
|---------|--|
| | <p>Proporcionara a profesores el total de horas por ejercer en la coordinación activa, tanto como en otras (integración global del sistema), esto es para generar las propuestas de horarios para el departamento de personal.</p> <p>De manera informativa dictaminara si el banco se excede, o las horas por ejercer de un profesor se han excedido, para tomar las consideraciones pertinentes.</p> |

| MODULO. | RESPONSABILIDADES Y CARACTERÍSTICAS ESPECIALES. |
|------------------|--|
| PROFESORES. | ➤ Apoyara en el mantenimiento del banco interno del BANCO DE HORAS . |
| GRUPOS NORMALES. | ➤ Proporcionara la información de horas asignadas para su integración dentro del BANCO DE HORAS . |

➤ Se deberá tener la capacidad de dar de alta un profesor desde la pantalla de captura (Comunicación directa con el modulo de PROFESORES) y capturar únicamente los datos principales, los cuales son:

- R.F.C. del profesor.
- Nombre del profesor.
- Si el profesor es funcionario.
- La categoría del profesor.

Esto es debido a que son los datos fundamentales para poder ejercer el banco particular de categoría.

BANKBAJA()

Formato

BANKBAJA(<pro_rfc>,<limite>)

Parámetros

<pro_rfc>

Cadena de caracteres que contiene el R.F.C. del profesor.

<limite>

Indica los límites de búsqueda, para un valor de .t. la búsqueda será en todas las coordinaciones, para un valor de .f. la búsqueda solamente será en la coordinación actual.

Devuelve

Un arreglo, que contendrá los nombres de las coordinaciones en las que el profesor tiene horas asignadas, dependiendo del limite.

Nulo en caso de que el profesor no tenga grupos asignados dentro del limite.

Descripción

Esta función se emplea para comunicarle al sistema de PROFESORES si el profesor tiene horas asignadas dentro del límite, esto se emplea en las bajas de profesores, para prevenir que dicho profesor tenga horas asignadas y sea dado de baja.

VARIABLES GLOBALES :

| NOMBRE. | CONTENIDO. |
|------------------|--|
| VM_DIR_CARRERA | Directorio que contiene los archivos de datos de la carrera. |
| VM_AÑO LINEA | Año activo completo, ejemplo : 1996 |
| VM_PERIODO LINEA | Periodo activo ("1" o "2") |
| VM_DIR SISTEMA | Directorio principal del sistema ZEUS. |

BAPRHOR()**Formato**

BAPRHOR(<pro_rfc>)

Parámetros

<pro_rfc> Cadena de caracteres que contiene el R.F.C. del profesor.

Devuelve

Un arreglo, que contendrá los nombres de las coordinaciones en las que el profesor tiene horas asignadas, dependiendo del límite.

Nulo en caso de que el profesor no tenga grupos asignados dentro del límite.

Descripción

Esta función se emplea para comunicarle al sistema de PROFESORES si el profesor tiene horas asignadas dentro del límite, esto se emplea en las bajas de profesores, para prevenir que dicho profesor tenga horas asignadas y sea dado de baja.

VARIABLES GLOBALES :

| NOMBRE. | CONTENIDO. |
|----------------|---|
| VM_DIR_CARRERA | Directorio que contiene los archivos de datos de la |

| | |
|------------------|--|
| | carrera. |
| VM_ANIO_LINEA | Año activo completo, ejemplo : 1996 |
| VM_PERIODO_LINEA | Periodo activo ("1" o "2") |
| VM_DIR_SISTEMA | Directorio principal del sistema ZEUS. |

BASIGHO()**Formato**

BASIGHO(<carrera>,<periodo>,<categoria>)

Parámetros

| | |
|-------------|--|
| <carrera> | Cadena de caracteres que contiene la ruta de acceso a los archivos de datos de la carrera. |
| <periodo> | Cadena de caracteres que contiene el periodo activo. |
| <categoria> | Variable numérica que contendrá la clave de la categoría a procesar. categoría = 1 profesores de asignatura. categoría = 2 ayudantes de profesor. categoría = 3 profesores de carrera. categoría = 4 funcionarios. categoría = 5 personal por honorarios. categoría = 6 técnicos académicos. |

Devuelve

Nulo en cualquier caso.

Descripción

Esta función se emplea para comunicarle al sistema de PROFESORES si el profesor tiene horas asignadas dentro del límite, esto se emplea en las bajas de profesores, para prevenir que dicho profesor tenga horas asignadas y sea dado de baja.

VARIABLES GLOBALES :

| NOMBRE. | CONTENIDO. |
|----------------|--|
| VM_DIR_CARRERA | Directorio que contiene los archivos de datos de la carrera. |

| | |
|------------------|--|
| VM ANIO LINEA | Año activo completo, ejemplo : 1996 |
| VM PERIODO LINEA | Periodo activo ("1" o "2") |
| VM DIR SISTEMA | Directorio principal del sistema ZEUS. |

COPRTCA()**Formato**

COPRTCA(<dbf>, <ntx>, <color>, <tipo>)

Parámetros

| | |
|---------|---|
| <dbf> | Cadena de caracteres que contiene el nombre del archivos de datos a emplear. |
| <ntx> | Cadena de caracteres que contiene el nombre del archivo indice a emplear. |
| <color> | Cadena de caracteres que contiene el color a emplear dentro de los despliegues de la función. |
| <tipo> | Parámetro booleano que indica si el catalogo de profesores será el general (modulo profesores), o el de la coordinación (Banco de horas). .t. Emplea las bases de datos de profesores. .f. Emplea las bases de datos de Banco de H. |

Devuelve

Nulo en caso de error o cancelación del proceso.

R.F.C. del profesor seleccionado.

Descripción

Esta función que da soporte tanto a Banco De Horas como a Profesores, se emplea para mostrar el catalogo de profesores, empleando un objeto BROWSE, con la variante de que tiene incorporado este a su vez un objeto GET, esto es para realizar selecciones directas.

VARIABLES GLOBALES :

| NOMBRE. | CONTENIDO. |
|----------------|--|
| VM_DIR_CARRERA | Directorio que contiene los archivos de datos de la carrera. |
| VM ANIO LINEA | Año activo completo, ejemplo : 1996 |

II.2. DESCRIPCION DEL CONTROL DE PROFESORES

El documento principal en la Contratación de Profesores es la propuesta de contratación llamada "Forma para movimientos de personal académico" por lo tanto mencionaremos a continuación la forma en que se procesan dichas formas actualmente:

Primeramente se debe contar con los horarios asignados para el semestre por cada profesor. A continuación se realiza la comparación de cada una de las propuestas del semestre anterior con el horario actual para generar las propuestas del semestre actual, una vez hechas las modificaciones pertinentes se entregan a la secretaria los formatos para que los mecanografié, cuando está terminada la propuesta se le entrega al profesor para que la firme, se envía al departamento de personal para revisión y en caso de corrección hay que volver a mecanografiar la propuesta hasta que este correcta para ser sellada. Cabe mencionar que este proceso se lleva de uno a dos meses.

Cuando ya están las propuestas se le entregan a la secretaria con las tarjetas de control de asistencia por cada grupo-materia para que las mecanografié. Cuando ha concluido el vaciado de la información en las tarjetas se elabora la relación de las tarjetas que serán enviadas a la Unidad Académica donde serán foliadas, selladas y firmadas, cuando ya están debidamente requisitadas se entregan en la sala de firmas el primer día de clases. En caso de error se realizan altas, bajas o cambios.

Uno o dos meses antes de terminar el semestre se evalúa el desempeño de los profesores mediante la aplicación de cuestionarios, los cuales son contestados por los alumnos, cuando ya se realizó la evaluación al profesor se entregan los cuestionarios en la Coordinación donde se captura y se generan los listados, actualmente se hace uso del programa CAPTURA y se envía la información al departamento de informática donde realizan el diagnóstico devolviendo los resultados hasta que termina el semestre, retrasando con esto la asignación de estímulos.

II.3. PLANTEAMIENTO DE LA SOLUCIÓN

Una vez que se ha enfocado y entendido el problema lo importante es poder emitir una propuesta de solución que satisfaga las necesidades requeridas. Siendo esta una de las etapas fundamentales del desarrollo de sistemas. Para

poder conseguir esto es necesario que la comunicación entre el diseñador del sistema y el usuario final sea continua y efectiva.

Lo idóneo para lograr el análisis de los requerimientos que han de ser solventados es tomar en cuenta todas las posibles situaciones que pudieran presentarse durante la operación del sistema. Para poder determinar dichas situaciones se hace necesario que el diseñador identifique claramente a las personas que estarán involucradas en el control de profesores y su relación organizacional dentro de la Coordinación de Ingeniería en Computación.

Así mismo identificar las áreas ajenas a la Coordinación de Ingeniería en Computación que tienen que ver con el uso y distribución de la información que se genera del Control de Profesores y por último cuales serán los beneficios que puede aportar la implantación del Sistema.

II.3.1. IDENTIFICACIÓN DE LOS REQUERIMIENTOS DEL USUARIO

Se sostuvieron en primera instancia entrevistas con el Secretario Técnico de la Coordinación de Ingeniería en Computación, quien proporcionó una lista de requerimientos que debería contemplar el sistema para de esta manera cubrir las necesidades que se tienen, ya que el conoce a fondo todo el seguimiento que se realiza con la información generada en cuanto a control de profesores, incluyendo los procedimientos manuales involucrados en ellos.

En dichas entrevistas fue posible establecer las bases para el desarrollo del sistema, cuyas características deberían ser las que se describen a continuación:

- * Contener la información que se requiere para el control de profesores, tanto datos personales como administrativos, tales como RFC, nombre, domicilio, teléfono, categoría, horas a impartir, etc., permitiendo de una manera rápida y efectiva efectuar consultas a esta información, mediante el uso del RFC como campo clave para un fácil acceso.
- * Las consultas o modificaciones de la información podrán realizarse en forma simultánea por dos o más personas.
- * Generar las formas para movimientos de personal académico con las cuales se proponen a los profesores que se han de contratar.
- * Impresión de tarjetas de control de asistencia y reportes administrativos.

- * Realizar la captura de la información contenida en la hoja de respuesta de los cuestionarios para evaluación de profesores y emitir reportes.

Por último, se puede observar que la expectativa del personal de la Coordinación de Ingeniería en Computación es contar con una herramienta confiable que realice las verificaciones que anteriormente realizaban apoyándose en información de semestres anteriores tanto de horarios como de propuestas.

II.3.2. PROPUESTA DE SOLUCIÓN

Ya que se tienen bien claros los requerimientos del usuario y tomando en consideración la problemática del control de profesores, se pudieron establecer las características con que debería contar el sistema. A continuación se presenta una propuesta de solución, la cual pretende realizar de manera adecuada el manejo de información.

Se desarrollara un sistema capaz de manejar la información correspondiente a cada profesor mediante el uso del RFC como campo llave, el cual deberá permitir la actualización y consulta de los datos generales del profesor, así como la verificación del horario asignado contra el horario autorizado a cada profesor. El cual aunque fue diseñado para la Coordinación de Ingeniería en Computación podrá ser utilizado en todas y cada una de las Coordinaciones del Plantel.

El objetivo fundamental del Sistema será el Control de Profesores desde su Contratación y Control de Asistencia hasta la Evaluación de su desempeño.

El Sistema pretende generar las Formas para Movimientos de Personal Académico con las cuales se proponen a los profesores que se han de contratar.

El Sistema será capaz de realizar la impresión de Tarjetas de Control de Asistencia, así como de los reportes administrativos.

Uno de los objetivos del Sistema es permitir la captura de la información contenida en la hoja de respuestas de los cuestionarios para la Evaluación de Profesores y emitir reportes de diagnóstico.

II.3.3. INFORMACIÓN REQUERIDA

Determinar la información requerida para la correcta operación del Sistema es imprescindible desde el inicio del proyecto ya que para realizar la normalización de los archivos es necesario tener bien claro que información resulta realmente útil y de esta manera evitar redundancias, este análisis se realizó partiendo de las pláticas sostenidas con el Secretario Técnico de la Coordinación de Ingeniería en Computación, llegando a la conclusión que las bases de datos que se ocuparan serán las siguientes:

- Un archivo que contenga información general del profesor como son RFC, nombre, domicilio, teléfono, beneficiarios del seguro de vida, etc. Mediante el cual se pueda proporcionar al usuario los datos personales de cada profesor.

Cabe aclarar que este archivo contendrá los registros de todos los profesores del plantel mientras que los demás solo contendrán los registros de los profesores que imparten clases en la Coordinación que este haciendo uso del Sistema.

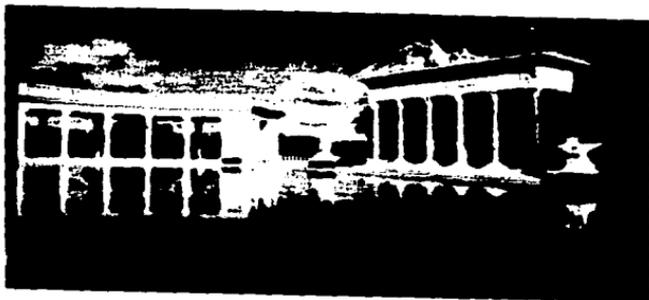
- Un archivo que contenga información administrativa del profesor como es su categoría, antigüedad, horas del semestre anterior, horas a impartir el semestre actual, etc.
- Un archivo que contenga la información correspondiente a las materias que se imparten en la carrera como clave de materia, descripción y tipo.
- Un archivo que contenga la información referente a grupos, materias y horarios en los que impartirá clases cada profesor el semestre actual y otro similar del semestre anterior. Estos archivos serán proporcionados por el Sistema de Horarios.
- Un archivo con las horas autorizados para cada profesor, el cual será proporcionado por el Sistema de Banco de Horas.

Los siguientes archivos son generados por el sistema en caso de no existir:

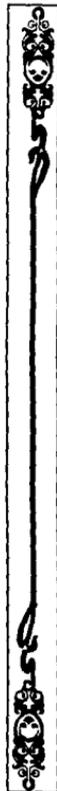
- Un archivo para la generación de propuestas automáticas y otro similar para la generación de propuestas manuales que contenga el tipo de movimiento, materia, grupo, horario, etc.
- Un archivo que contenga las propuestas revisadas y listas para imprimir, del cual también podrán imprimirse las tarjetas de control de asistencia.

- * Un archivo que contenga la información necesaria para realizar la Evaluación de Profesores como son grupo, materia y campos para almacenar la respuesta a cada pregunta.
- * Un archivo que contenga los promedios de la Evaluación de cada profesor.
- * Un archivo que contenga el catálogo de categorías.
- * Un archivo que contenga el catálogo de tipos de movimiento.

- * Un archivo que contenga los registros de los profesores que no tienen horario asignado.
- * Un archivo que contenga los registros de los profesores cuyas horas asignadas excedan a sus horas autorizadas.



CAPITULO III DISEÑO



CAPITULO III. DISEÑO

III.1. Selección de Software para el Diseño del Sistema

III.2. Diseño del Sistema

III.2.1. Estructuras de Bases de Datos

III.2.1.1. PROFCAT

III.2.1.2. PCICO971

III.2.1.3. PAICO971

III.2.1.4. PMICO971

III.2.1.5. PRICO971

III.2.1.6. PEICO971

III.2.1.7. TIFOMOV

III.2.1.8. DMCICO

III.2.1.9. CAT

III.2.2. Diagrama Modular

III.2.3. Explicación de Diagrama Modular

CAPITULO III**DISEÑO**

Teniendo debidamente delimitados los problemas a que se enfrenta la coordinación de computación al inicio de cada semestre para poder generar las propuestas de contratación y las tarjetas de control de asistencia, así como la evaluación de cada profesor al finalizar el semestre, se realiza un análisis para plantear una posible solución y posteriormente realizar el diseño final del sistema que dará solución a los problemas que fueron detectados durante la etapa de dicho análisis.

En el presente capítulo se realiza el diseño formal del sistema que será presentado al usuario final, es aquí donde se desarrollan las opciones propuestas para solucionar el problema.

III.1. SELECCIÓN DE SOFTWARE PARA EL DISEÑO DEL SISTEMA

La implementación del sistema se realizó mediante un manejador de base de datos capaz de manipular la información de una forma rápida y flexible.

Para tomar dicha decisión se tomaron en cuenta las siguientes alternativas:

- DBASE III PLUS
- DBASE IV
- FOXPRO 2.5
- CLIPPER 5.2

y las siguientes consideraciones ¹:

INTERPRETE:

Un programa interpretador es aquél que recibe una instrucción e inmediatamente la traduce al lenguaje máquina, le añade otras instrucciones que requiere y la ejecuta. El intérprete realiza todo el proceso descrito anteriormente para cada instrucción que ejecuta. Si se escribe un conjunto de instrucciones en un programa, cada una de las instrucciones debe ser interpretada antes de ejecutarse. Si el programa se ejecuta múltiples veces, todas las instrucciones del programa deben ser interpretadas cada vez que el programa se ejecute. Esto hace que la ejecución de un programa intérprete sea muy lenta.

COMPILADOR:

Un programa compilador recibe todas las instrucciones de los programas realizados y los traduce al lenguaje máquina de una sola vez. Enseguida se realiza un proceso de encadenamiento donde se le añaden algunas instrucciones que requiere para su funcionamiento y se guarda en un archivo ejecutable del sistema operativo. De esta manera, se genera un programa escrito en lenguaje máquina con todas las instrucciones necesarias para su ejecución. Si se quiere ejecutar el programa múltiples veces, la computadora simplemente lee el archivo generado y lo ejecuta. Esto hace que la ejecución del programa sea bastante rápida; sin embargo, si se realiza alguna modificación al código original, es necesario compilar otra vez todo el programa.

Considerando que los manejadores de bases de datos como DBASE III PLUS, DBASE IV Y FOXPRO requieren de un intérprete que realice la traducción de las instrucciones de los programas que en ellos se desarrollan, línea por línea, lo cual hace su ejecución lenta y que además requieren que el software que compone dichos manejadores este instalado en la computadora donde ha de ejecutarse la aplicación, se opto por la última alternativa, ya que cuando un programa es interpretado cada vez que se corre, se pierde una gran cantidad de tiempo del procesador.

Un programa compilado corre 10 o más veces más rápido que un intérprete. No se puede alterar el código en un programa compilado, para cambiar las líneas, se debe utilizar el código fuente, esto proporciona un grado de seguridad a la aplicación ya que el usuario no tendrá acceso a los programas fuentes y no podrá alterarlos.

En Clipper se escribe la aplicación, se compila y enlaza y se puede distribuir este software desarrollado sin necesidad de que el usuario compre el paquete de software, para poder correr la aplicación creada, lo cual la hace más atractiva comercialmente hablando.

CLIPPER 5.2 es un manejador de bases de datos más potente, ya que en realidad es un compilador y no un intérprete como las otras alternativas, es decir, convierte el programa fuente en código máquina lo cual hace su ejecución más rápida y además el producto es un programa ejecutable autónomo, es decir, no se requiere el software de CLIPPER instalado en la máquina donde va a ejecutarse.

VENTAJAS OFRECIDAS POR CLIPPER ²

- Programas más rápidos
- Programas más fáciles de entender (y de mantener)
- Modernización de la programación
- Economía de memoria
- Creación de funciones definidas por el usuario

La primera versión de Clipper se elaboró para ser sólo un compilador de programas Dbase, pero Clipper ha seguido su evolución creando un software donde el Programador/Analista puede contar con un Lenguaje de Programación estructurado, controlando todo el ambiente de desenvolvimiento, volviéndose así un Lenguaje de Programación Real, independiente e incomparable con cualquier otro Software administrador de base de datos.³

El compilador de Clipper (CLIPPER.EXE) funciona básicamente de la siguiente forma: primero el compilador carga en la memoria el programa fuente (.PRG), después analiza todo el programa revisando cada línea de código para encontrar errores solo una vez, verificando las directivas y sintaxis en general. Si todo es satisfactorio, el compilador traduce el código en lenguaje máquina. El lenguaje máquina es un código binario que el microprocesador de la computadora puede ejecutar directamente, cuando el compilador traduce del archivo ASCII al lenguaje máquina, el resultado es un archivo llamado código objeto(.OBJ). El código objeto es ligado con las otras piezas del lenguaje máquina, las cuales contienen instrucciones y rutinas necesarias para completar el archivo del programa objeto, una vez que el programa es "linkeado", mediante el RTLINK, podemos utilizarlo directamente sin la necesidad de otros programas, es decir, se genera un archivo ejecutable (.EXE).

El uso de funciones y procedimientos es la base de la estructura de los programas desenvueltos en Clipper. Las funciones y procedimientos son un grupo de declaraciones que ejecutan una tarea específica. La diferencia entre las procedures y las funciones es que la función puede regresar valores para la rutina que la llamó a través del comando RETURN, y las procedures no regresan valores.

Las variables en Clipper pueden ser creadas a partir de una simple atribución (:=) y pueden ser iniciadas con las siguientes declaraciones:

Variable Local

Estas variables son creadas para manipular valores sólo en una rutina. Normalmente, este tipo de variable es utilizada para manejar valores dentro de una función definida por el usuario, existe sólo para la rutina o programa que lo creó.

Variable Static

Esta variable existe sólo para la rutina o programa que la creó, pero cuando la rutina o programa termina, la variable continúa en la memoria.

Variable Privada

Las variables del tipo PRIVATE pueden ser manejadas también por las funciones o programas que son llamados por el programa que las creó.

Variable Pública

Las variables PUBLIC son opcionales. Cuando es declarada, la variable permanecerá en la memoria hasta que el programa cierre la ejecución o la misma se liberada a través del comando RELEASE.

Una vez creada la variable PUBLIC, se puede manejar por todos los otros módulos del programa.

TIPOS DE DATOS**Carácter**

Los datos manejados en Clipper como tipo "C", o sea, carácter, son conocidos en otros lenguajes como tipos alfanuméricos. Este tipo de dato está formado por los caracteres de la tabla ASCII y deben tener un tamaño definido.

Los datos caracteres deben ser manejados entre delimitadores. En Clipper, podemos utilizar como delimitadores dos comillas (" "), dos apóstrofes (' ') y corchetes ([]). Este tipo de dato es usual para tratamiento de datos que fueron formados por letras y/o número.

Numérico

Este tipo de dato es básico para cálculos aritméticos. Está formado por los números de 0 hasta 9, operadores matemáticos, el signo de punto decimal y la coma.

Los datos numéricos son utilizados para el manejo de valores pasivos de cálculo.

Memo

El tipo de dato **Memo** es utilizado para manejar informaciones de tamaño variable, o sea, el volumen de informaciones puede ser pequeño o grande.

Un campo definido como el tipo **Memo** es inicializado con el tamaño de 10 byte;., sin embargo, si fuese necesario almacenar más de 10 bytes, no habrá ningún obstáculo.

El campo **Memo** es compatible con el tipo de dato Carácter, pero este último posee el tamaño fijo. En una variable o campo **Memo**, podemos almacenar hasta 65.535 bytes.

Fecha

Este tipo de dato es utilizado en Clipper para el manejo de fechas de calendario. Podemos, por ejemplo, calcular la diferencia de días, meses o años entre dos o más fechas.

Las fechas son procesadas en el tamaño básico de 8 bytes y pueden ser presentadas en diferentes formatos.

Bloque de Códigos ("Code Block")

Los bloques de códigos son un tipo de datos especiales que poseen instrucciones ejecutables. Es posible almacenar dentro de una variable de bloque de código instrucciones procesables por el lenguaje, como esquemas de cálculos o llamadas a funciones.

Arreglos

Los arreglos poseen la característica de ser un tipo especial de variable que puede almacenar varios valores (informaciones) al mismo tiempo. En esta super variable llamada "arreglo" se puede guardar todos los valores de una lista de datos.

CONFIGURAR EL AMBIENTE PARA CLIPPER

Para un mejor trabajo y desempeño durante la ejecución o desarrollo de las aplicaciones Clipper, aconsejamos que se coloque en el archivo AUTOEXEC.BAT, algunas líneas de comandos de configuración de ambiente que deben ser pasados al D.O.S, a fin de esclarecer algunas informaciones para la aplicación Clipper. Tales declaraciones son las siguientes:

```
SET CLIPPERCMD=<opciones de compilación>
```

Esta línea de instrucción creará una variable de sistema operativo que será leída por el CLIPPER.EXE en el momento de compilar el programa. <opciones de compilación> son parámetros entendidos por el compilador como /M/W, entre otros, que quedarán siempre señalados como opciones estándar de compilación.

```
SET TMP=\CLIPPER5\TMP
```

Esta línea creará la variable TMP que contendrá un señalamiento de directorio que creará archivos temporarios durante la compilación (CLIPPER.EXE) y "linkeo" (RTLINK) de programas, en el caso que la memoria convencional de la computadora no sea la suficiente para el proceso. Tales archivos serán creados y luego eliminados por los respectivos programas.

```
SET PLL=\CLIPPER5\PLL
```

Señala en cuál directorio del disco están almacenadas las bibliotecas "pre-linkeadas", o sea las PLL's. Este directorio será buscado por las aplicaciones que serán generadas a partir de PLL's.

```
SET CLIPPER=F055
```

Esta línea de comando creará la variable de sistema de nombre CLIPPER que contenga el parámetro F(n), donde n es un número entero que representa la cantidad de archivos a ser abiertos al mismo tiempo por la aplicación Clipper. En la línea de comando anterior se asume que los programas desarrollados en

Clipper abrirán un máximo de 55 archivos al mismo tiempo. Hasta 250 archivos pueden ser abiertos en Clipper al mismo tiempo.

PATH\CLIPPER5\BIN;\DOS;(otro directorio);,...

Señala a través del comando **PATH** de D.O.S. directorios ubicados como estándar, o sea, en el caso que estén trabajando en otros directorios del disco podrá acceder en cualquier momento a programas ejecutables contenidos en la lista de direcciones opcionales señaladas por **PATH\CLIPPER5\BIN**, y a la localización de programas **CLIPPER.EXE** y **RTLINK.EXE**, vital para la compilación y "linkeo" de programas en Clipper5.

SET LIB=\CLIPPER5\LIB

Esta línea crea la variable de memoria **LIB** que apuntará para el programa **RTLINK**, cuando éste es ejecutado para el "linkeo" de programas, el directorio de los archivos **LIB's** a ser buscados durante el proceso de "linkeo". Normalmente las bibliotecas estándar de Clipper son colocadas en el subdirectorio **CLIPPER5\LIB**.

¹ Manual de referencia rápida del Clipper 5.01 / Enrique Pérez Villaseñor- Pedro Pablo Torija Rojas / Ventura

² Clipper 5.01 Avanzado / José A. Ramalho / Mc Graw Hill

³ Clipper 5.2 / Gorki Starlin C. Oliveira / Ventura

III.2. DISEÑO DEL SISTEMA

Con base en la información proporcionada por el personal administrativo de la Coordinación de Ingeniería en Computación en cuanto a las consideraciones que debían incluirse en las propuestas de contratación, las tarjetas de control de asistencia y la evaluación de profesores, se desarrollaron los módulos del sistemas para cubrir dichos requerimientos.

En este apartado se muestra el Sistema de Control de Profesores en forma modular, así como la explicación del funcionamiento de cada uno de los módulos y funciones del Sistema y por último se detalla la estructura de las bases de datos que se emplean en el sistema y el significado de algunos campos y registros de control contenidos en ellas.

III.2.1. Estructuras de Bases de Datos.

A continuación se detalla la estructura de las bases de datos que se emplean en el sistema y el significado de algunos campos y registros de control contenidos en ellas.

En este archivo se almacenará el catálogo de profesores del plantel, dicho catálogo contendrá datos personales del profesor, es decir, registro federal de contribuyentes, nombre del profesor, si es o no funcionario, domicilio particular, clave de personal, teléfono particular, teléfono de oficina, extensión telefónica, si esta o no titulado, el nombre, sexo y parentesco de dos personas beneficiarias del seguro de vida.

En el campo flag se registrara un "" cuando el registro se haya dado de alta con la función BAALTPR.

Archivo de datos: PROFCA1.DBF
 Archivo índice: PRORFC.NTX
 Campo llave: RFC
 Alias: PRO

| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|-------------------|---------------|-----------------|------------------------|
| RFC | CARÁCTER | 13 | rfc del profesor |
| NOM_PRO | CARÁCTER | 32 | nombre del profesor |
| FUNSI0 | CARÁCTER | 1 | profesor funcionario |
| DOMICILIO | CARÁCTER | 50 | domicilio del profesor |
| CLAVE_PERCARÁCTER | | 2 | clave de personal |
| TEL_CASA | CARÁCTER | 9 | teléfono particular |
| TEL_TRAB | CARÁCTER | 9 | teléfono de trabajo |
| EXT | CARÁCTER | 5 | extensión telefónica |
| TITULADO | CARÁCTER | 2 | profesor titulado |
| BENEFICIA1 | CARÁCTER | 30 | nombre beneficiario 1 |
| SEXB1 | CARÁCTER | 1 | sexo beneficiario 1 |
| PARB1 | CARÁCTER | 10 | parentesco benef. 1 |
| BENEFICIA2 | CARÁCTER | 30 | nombre beneficiario 2 |
| SEXB2 | CARÁCTER | 1 | sexo beneficiario 2 |
| PARB2 | CARÁCTER | 10 | parentesco benef. 2 |
| FLAG | CARÁCTER | 1 | bandera actualización |

Archivo de datos: PCIC0871.DBF
 IPC-INICIALES DE LA COORDINACION PERIODO
 Archivo índice: COORRFC.NTX
 Campo llave: RFC
 Alias: COOR

| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------------|---------------------|-----------------------|------------------------|
| RFC | CARÁCTER | 13 | rfc del profesor |
| NOM_PRO | CARÁCTER | 32 | nombre del profesor |
| CATEGORIA | CARÁCTER | 3 | categoría del profesor |
| ANTIGUEDAD | FECHA | 8 | antigüedad en la coord |
| ACTIVO | CARÁCTER | 2 | activo en la coord |
| FE_INICIO | FECHA | 8 | fecha de inicio |
| FE_LIMITE | FECHA | 8 | fecha limite |
| HOR_TEO_AN | NUMÉRICO | 4.1 | horas teóricas ant |
| HOR_PRA_AN | NUMÉRICO | 4.1 | horas prácticas ant |
| AREA_O_JEF | CARÁCTER | 33 | otra area o jefatura |
| CAT_O_JEF | CARÁCTER | 3 | categoría en otra area |
| HOR_O_JEF | NUMÉRICO | 4.1 | horas en otra jefatura |
| DEP_O_DEP | CARÁCTER | 10 | otra dependencia |
| CAT_O_DEP | CARÁCTER | 3 | categoría en otra dep |
| HOR_O_DEP | NUMÉRICO | 4.1 | horas en otra depend |
| HOR_TEO | NUMÉRICO | 4.1 | horas teóricas actual |
| HOR_PRA | NUMÉRICO | 4.1 | horas prácticas actual |
| BANCO_HORA | NUMÉRICO | 4.1 | horas autorizadas BH |
| FLAG | CARÁCTER | 1 | bandera actualización |
| PROPUESTA | CARÁCTER | 1 | cuenta con propuesta |
| EVALUACION | CARÁCTER | 1 | cuenta con evaluación |



| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|-----------------------|
| RFC | CARÁCTER | 13 | rfc del profesor |
| CVE_ASIG | CARÁCTER | 4 | clave de asignatura |
| ASIGNATURA | CARÁCTER | 32 | nombre de asignatura |
| GRUPO | CARÁCTER | 4 | grupo |
| TIPOMOV | CARÁCTER | 2 | tipo de movimiento |
| TEORICA | NUMÉRICO | 4.1 | horas teóricas |
| PRACTICA | NUMÉRICO | 4.1 | horas prácticas |
| HORARIO | CARÁCTER | 40 | horario de la materia |
| LUNES | CARÁCTER | 13 | horario del lunes |
| MARTES | CARÁCTER | 13 | horario del martes |
| MIERCOLES | CARÁCTER | 13 | horario del miércoles |
| JUEVES | CARÁCTER | 13 | horario del jueves |
| VIERNES | CARÁCTER | 13 | horario del viernes |
| SABADO | CARÁCTER | 13 | horario del sábado |
| FLAG | CARÁCTER | 2 | bandera índice |



| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|-----------------------|
| RFC | CARÁCTER | 13 | rfc del profesor |
| CVE_ASIG | CARÁCTER | 4 | clave de asignatura |
| ASIGNATURA | CARÁCTER | 32 | nombre de asignatura |
| GRUPO | CARÁCTER | 4 | grupo |
| TIPOMOV | CARÁCTER | 2 | tipo de movimiento |
| TEORICA | NUMÉRICO | 4.1 | horas teóricas |
| PRACTICA | NUMÉRICO | 4.1 | horas prácticas |
| HORARIO | CARÁCTER | 40 | horario de la materia |
| LUNES | CARÁCTER | 13 | horario del lunes |
| MARTES | CARÁCTER | 13 | horario del martes |
| MIERCOLES | CARÁCTER | 13 | horario del miércoles |
| JUEVES | CARÁCTER | 13 | horario del jueves |
| VIERNES | CARÁCTER | 13 | horario del viernes |
| SABADO | CARÁCTER | 13 | horario del sábado |
| FLAG | CARÁCTER | 2 | bandera índice |

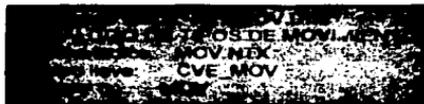


| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|-----------------------|
| RFC | CARÁCTER | 13 | rfc del profesor |
| CVE_ASIG | CARÁCTER | 4 | clave de asignatura |
| ASIGNATURA | CARÁCTER | 32 | nombre de asignatura |
| GRUPO | CARÁCTER | 4 | grupo |
| TIPOMOV | CARÁCTER | 2 | tipo de movimiento |
| TEORICA | NUMÉRICO | 4.1 | horas teóricas |
| PRACTICA | NUMÉRICO | 4.1 | horas prácticas |
| HORARIO | CARÁCTER | 40 | horario de la materia |
| LUNES | CARÁCTER | 13 | horario del lunes |
| MARTES | CARÁCTER | 13 | horario del martes |
| MIERCOLES | CARÁCTER | 13 | horario del miércoles |
| JUEVES | CARÁCTER | 13 | horario del jueves |
| VIERNES | CARÁCTER | 13 | horario del viernes |
| SABADO | CARÁCTER | 13 | horario del sábado |
| FLAG | CARÁCTER | 2 | bandera índice |



| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|-------------------------|
| PROFESOR | CARÁCTER | 30 | nombre del profesor |
| RFC | CARÁCTER | 13 | rfc del profesor |
| GRUPO | CARÁCTER | 4 | grupo |
| MATERIA | CARÁCTER | 20 | materia |
| CLAVE | CARÁCTER | 4 | clave de la materia |
| SEMESTRE | CARÁCTER | 4 | semestre de evaluación |
| A01 | NUMÉRICO | 2 | autoevaluación preg. 1 |
| A02 | NUMÉRICO | 2 | autoevaluación preg.2 |
| A03 | NUMÉRICO | 2 | autoevaluación preg.3 |
| A04 | NUMÉRICO | 2 | autoevaluación preg.4 |
| A05 | NUMÉRICO | 2 | autoevaluación preg.5 |
| A06 | NUMÉRICO | 2 | autoevaluación preg.6 |
| A07 | NUMÉRICO | 2 | autoevaluación preg.7 |
| A08 | NUMÉRICO | 2 | autoevaluación preg.8 |
| A09 | NUMÉRICO | 2 | autoevaluación preg.9 |
| A10 | NUMÉRICO | 2 | autoevaluación preg. 10 |
| A11 | NUMÉRICO | 2 | autoevaluación preg.11 |
| A12 | NUMÉRICO | 2 | autoevaluación preg.12 |
| A13 | NUMÉRICO | 2 | autoevaluación preg.13 |
| P01 | NUMÉRICO | 2 | evaluación prof.preg 1 |
| P02 | NUMÉRICO | 2 | evaluación prof.preg 2 |
| P03 | NUMÉRICO | 2 | evaluación prof.preg 3 |
| P04 | NUMÉRICO | 2 | evaluación prof.preg 4 |
| P05 | NUMÉRICO | 2 | evaluación prof.preg 5 |
| P06 | NUMÉRICO | 2 | evaluación prof.preg 6 |
| P07 | NUMÉRICO | 2 | evaluación prof.preg 7 |

| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|------------------------|
| P08 | NUMÉRICO | 2 | evaluación prof.preg 8 |
| P09 | NUMÉRICO | 2 | evaluación prof.preg 9 |
| P10 | NUMÉRICO | 2 | evaluación |
| prof.preg10 | | | |
| P11 | NUMÉRICO | 2 | evaluación |
| prof.preg11 | | | |
| P12 | NUMÉRICO | 2 | evaluación |
| prof.preg12 | | | |
| P13 | NUMÉRICO | 2 | evaluación |
| prof.preg13 | | | |
| P14 | NUMÉRICO | 2 | evaluación |
| prof.preg14 | | | |
| P15 | NUMÉRICO | 2 | evaluación |
| prof.preg15 | | | |
| P16 | NUMÉRICO | 2 | evaluación |
| prof.preg16 | | | |
| P17 | NUMÉRICO | 2 | evaluación |
| prof.preg17 | | | |
| COMENTARIO | CARÁCTER | 60 | comentario |



| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|----------------------|
| CVE_MOV | CARÁCTER | 2 | clave del movimiento |
| MOVIMIENTO | CARÁCTER | 20 | tipo de movimiento |



| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|------------------------|
| CVE_MAT | CARÁCTER | 4 | clave de materia |
| DES_MAT | CARÁCTER | 32 | descripción de materia |
| TIPO | CARÁCTER | 1 | tipo de materia |



| NOMBRE DEL CAMPO | TIPO DE CAMPO | TAMAÑO DE CAMPO | DESCRIPCIÓN |
|------------------|---------------|-----------------|--------------------|
| CVE_CAT | CARÁCTER | 3 | clave de categoría |
| CATEGORIA | CARÁCTER | 24 | categoría |

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
 INSTITUTO DE INVESTIGACIONES Y ENSEÑANZA EN QUÍMICA

VARIABLES GLOBALES EN ZEUS UTILIZADAS POR PROFESORES

VM_INICIAL

Se utiliza para las iniciales de las carreras; por ejemplo, Ingeniería en Computación tiene la inicial ICQ, de Ingeniería Mecánica Eléctrica es IME, etc.

VM_CVECAR

Esta variable se emplea para las claves de las carreras, como 035 para Ingeniería en Computación, es de tipo carácter.

VM_ANIO_LINEA

Contiene los cuatro dígitos del año en que trabajará Zeus, la variable es tipo carácter, ejemplo: 1997.

VM_DIR_SISTEMA

Es utilizada para la ruta de acceso del sistema ZEUS, bajo la cual se encuentra el archivo ejecutable, las base de datos y demás archivos generales.

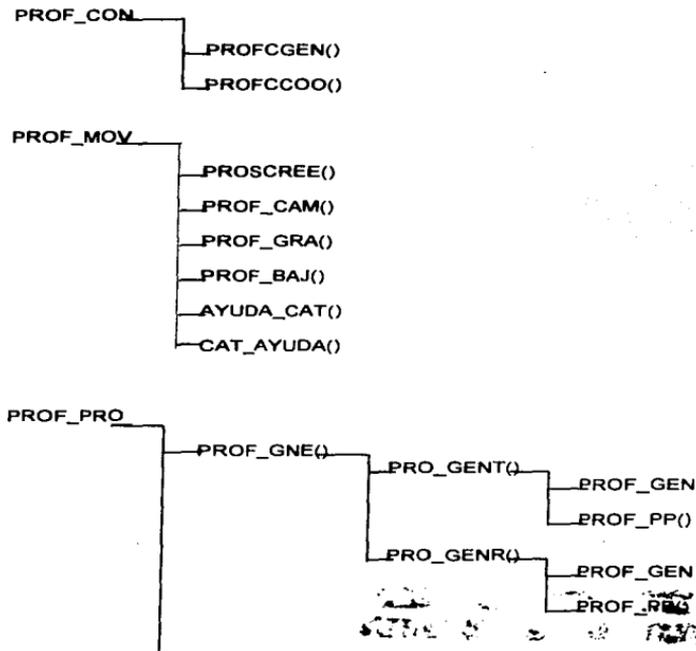
VM_DIR_CARRERA

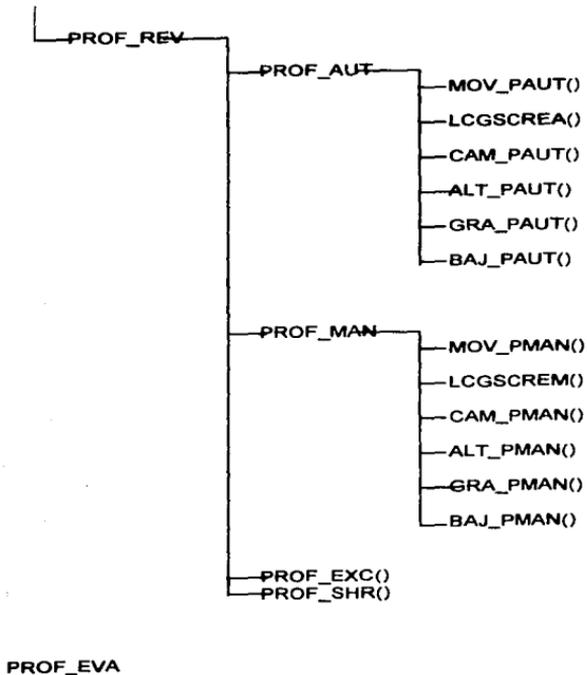
Retiene toda la ruta de acceso para cada una de las carreras, en donde están sus archivos, por ejemplo: C:ZEUSICO.

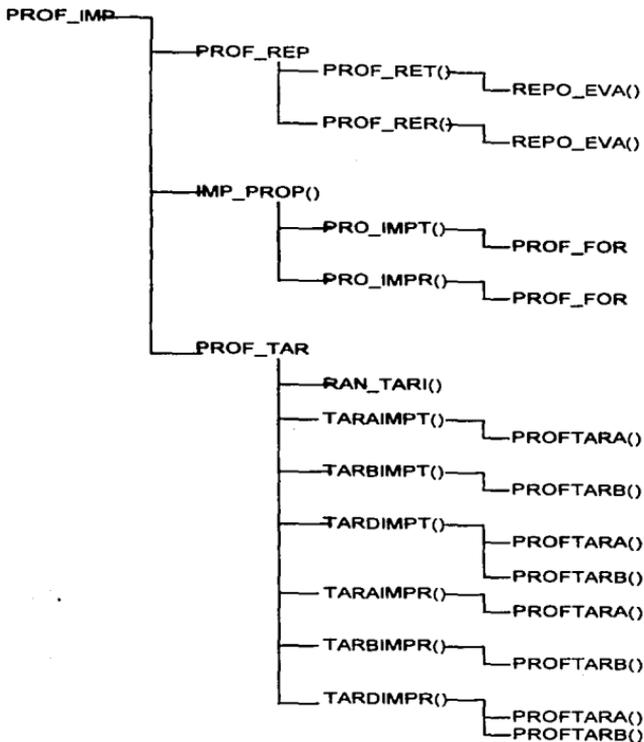
**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

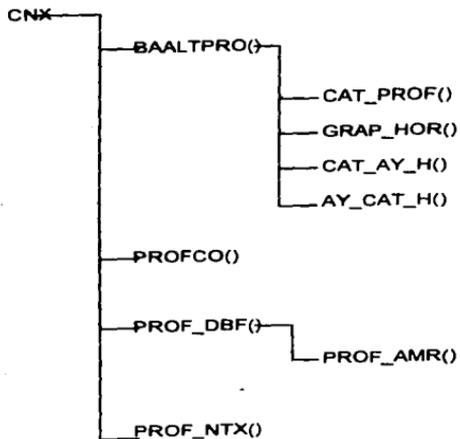
III.2.2. Diagrama Modular

A continuación se muestra el Sistema de Control de Profesores en forma modular.

PROF_LCG







III.2.3. EXPLICACIÓN DE DIAGRAMA MODULAR

A continuación se explica el funcionamiento de cada uno de los módulos y funciones del Sistema de Control de Profesores.

Función BaAltPro()

PROF_CNX.PRG

Función que verifica que exista un profesor y envía el RFC y el NOMBRE, si no existe presenta una pantalla de captura para darlo de alta.

El formato de la función es:

BAALTPRO(rfc)

donde:

RFC Es una cadena con el R.F.C. del profesor

Cuando se ejecuta esta función se verifica que exista el RFC tanto en el catálogo general de profesores como en el catálogo de profesores de la coordinación, si el RFC existe únicamente debe devolver RFC, NOMBRE.

Si existe en PROF-CAT pero en el catálogo de profesores de la coordinación no, se presenta una pantalla mostrando el RFC, el nombre del profesor, si es o no funcionario y se debe pedir la categoría para darlo de alta en el catálogo de profesores de la coordinación.

Si existe en el catálogo de profesores de la coordinación pero en el catálogo de profesores del plantel no se presenta una pantalla mostrando el RFC, el nombre del profesor y su categoría y nos pregunta si es funcionario o no para darlo de alta en el catálogo de profesores del plantel.

Si no existe en ninguno de los dos archivos presenta una pantalla mostrando el RFC y pidiendo nombre del profesor, si es funcionario o no y su categoría para darlo de alta en ambos archivos.

En cualquiera de los casos se le asigna una bandera a los registros que se dan de alta para después completarles la información. Si se genera una alta se avisa al Sistema de Banco de Horas transmitiendo el RFC, el Nombre del profesor, la Categoría y si es Funcionario, mediante la función BAALTPR2().

Se asignan variables globales y las variables que definen el entorno en cuanto a colores y ruta de los archivos.

En esta función se abren los siguientes archivos, los cuales se cierran al terminar la función:

| | |
|---|---|
| PROFCAT (pro) | Catálogo de profesores del plantel |
| PC+Iniciales de la carrera+Período (coor) | Catálogo de profesores de la coordinación |
| CAT (cat) | Catálogo de categorías |

Esta función se auxilia de las siguientes funciones:

| | |
|------------|---|
| CAT_PROF() | Función que muestra un browse con las categorías de profesores. |
| GRAP_HOR() | Función que graba los cambios al catálogo de profesores. |
| CAT_AY_H() | Función que activa el ENTER. |
| AY_CAT_H() | Función que muestra un browse de las categorías. |

Función Choose()

PROF_IMP.PRG

Función que permite seleccionar el tipo de impresora.

El menú nos presenta las siguientes opciones:

REMOTA LÁSER
REMOTA MATRIZ
LOCAL LÁSER
LOCAL MATRIZ

Dependiendo de la opción que se elija se asignaran los caracteres de escape para la impresión.

Función Imp_Prop()**PROF_IMP.PRG**

Función que presenta el menú de impresión de propuestas.

En primera instancia nos solicita el número de copias que deseamos, así como renglón y columna iniciales.

Las opciones del menú son:

TODOS PRO_IMPT()(Imprimir todas las propuestas)

RANGO PRO_IMPR()(Imprimir rango de propuestas)

Función ProfCo()**PROF_CNX.PRG**

Función que transmite el nombre de un profesor, haciendo uso del catálogo de profesores del plantel.

El formato de la función es:

PROFCO(rfc)

donde:

RFC Es una cadena con el R.F.C. del profesor

Al llamar ha esta función se debe transmitir un RFC este se busca en el catálogo de profesores del plantel y si se encuentra se devuelve el nombre del profesor, de lo contrario se transmite un valor nulo.

Función ProfTaRA()

PROF_IMP.PRG

Función que realiza el formato del registro de asistencia del personal docente de la cara "A".

Desde esta función se configura la impresora y por último se imprime el formato de la tarjeta.

Función ProfTaRB()

PROF_IMP.PRG

Función que realiza el formato del registro de asistencia del personal docente de la cara "B".

Desde esta función se configura la impresora y por último se imprime el formato de la tarjeta.

Programa PROF_AUT.PRG

Programa que permite revisar la información antes de imprimir las propuestas automáticas.

En este programa se abren los siguientes archivos:

PR+iniciales de la coordinación+período(rev)

El cual contiene la información de las propuestas revisadas y listas para imprimir.

TIPOMOV (mov) (Catálogo de Tipos de Movimiento)

En este programa se también se hace uso de los siguientes archivos de datos:

PCICO971 (Contiene información del profesor dependiendo de la coordinación (coor))

PAICO971 (Contiene información para las propuestas automáticas (paut))

PMICO971 (Contiene información para las propuestas manuales (pman))

En este programa se hace un llamado al actualizador MOV_PAUT() el cual realiza los movimientos al archivo de propuestas automáticas.

Este programa muestra la mascarilla de captura mediante la función LCGSCREA() y solicita el RFC del profesor más el índice de registro de la propuesta que se va a revisar, tiene la alternativa de consultar mediante un browse de ayuda del archivo de propuestas automáticas, esta ayuda se despliega al dar ENTER o al oprimir la tecla de función F2.

Este programa cuenta con una función para dar de alta algún registro ALT_PAUT() la cual se activa al oprimir la tecla de función F1 y otra para dar de baja BAJ_PAUT() esta se activa al oprimir la tecla de función F3.

Si lo que se desea es hacer cambios a la información se presenta para su edición con la función CAM_PAUT().

Si se realiza una alta primeramente se verifica si el RFC existe en el catálogo de profesores, posteriormente se busca en los archivos de propuestas automáticas y manuales para verificar cual es el último índice para ese RFC y poder asignarle un número consecutivo.

Ya sea que se realice un cambio o una alta se hará un llamado a la función GRA_PAUT() para grabar los movimientos al archivo de propuestas automáticas.

En el caso de la clave de la materia solo se permitirá su edición si está es de tipo carácter.

Se cuenta con ayuda para el tipo de movimiento, dicha ayuda se activa con la tecla de función F2.

En caso de que el horario cambie se desplegara una pantalla anexa para modificar también la información que se imprimirá en las tarjetas de control de asistencia.

Antes de aceptar el movimiento se verifica que el total de horas asignada no rebase las horas autorizadas, si esto sucediera se manda un mensaje de error y se permite modificar el número de horas autorizadas.

Se actualiza la información de manera simultánea en el archivo de propuestas automáticas y en el archivo de propuestas revisadas.

Por último se va al catálogo de profesores para asignarle un asterisco en el campo propuesta para tener identificados los profesores que tienen lista su propuesta.

Programa PROF_CNX.PRG

Programa que contiene funciones de comunicación con los otros sistemas que componen el Sistema Integral de Administración.

Programa PROF_CON.PRG

Programa que realiza la consulta al catálogo de profesores es llamado pro el programa PROF_LCG.

En este programa se abren los siguientes archivos de datos, los cuales se cierran al concluir el programa:

PROFCAT (Contiene los datos personales del profesor (pro))

PCICO971 (Contiene información del profesor dependiendo de la coordinación (coor))

CAT (Catálogo de categorías (cat))

Presenta un menú de opciones para especificar que tipo de información se desea consultar, ya sea personal o de datos generales, haciendo un llamado a las siguientes funciones:

PROFCGEN() (Consulta de datos personales)

PROFCCOO() (Consulta de datos generales)

Función Prof_DBF0**PROF_CNX.PRG**

Función que verifica que existan los archivos necesarios para el proceso.

En esta función se asignan variables globales para crear los archivos necesarios para el funcionamiento del sistema.

El archivo PROFCAT (Catálogo de profesores del plantel) y el catálogo de materias deben existir ya que el sistema no puede generarlos.

El archivo PC+Iniciales de la coordinación+período (Catálogo de profesores de la coordinación) es indispensable para el funcionamiento del sistema pero si no se encuentra se hará una copia del catálogo del semestre anterior.

Los archivos PA+Iniciales de la coordinación+período (Archivo de propuestas automática), PM+Iniciales de la coordinación+período (Archivo de propuestas manuales), PR+Iniciales de la coordinación+período (Archivo de propuestas revisadas), si no existen serán creados mediante la función PROF_AMR().

El archivo PE+Iniciales de la coordinación+período (Archivo de evaluaciones) si no existe se generará.

El archivo CAT (Catálogo de categorías) si no existe se generará.

El archivo TIPOMOV (Catálogo de tipo de movimiento) si no existe se generará.

Esta función también verifica que existan los índices necesarios, de no ser así los genera.

Programa PROF_EVA.PRG

Programa que realiza la captura de evaluación de profesores.

En este programa se abren los siguientes archivos, los cuales se cierran al terminar el programa:

PC+Iniciales de la coordinación+Período

(Contiene el catálogo de profesores de la coordinación coor))

PE+Iniciales de la coordinación+Período

(Archivo de evaluaciones a profesores (eva))

Nos pide el nombre del profesor que se va a evaluar, se cuenta con una pantalla de ayuda de la que podemos seleccionar el profesor, esta ayuda se activa al oprimir la tecla de función F2.

Si el nombre del profesor se tecléa se verificará que exista en el catálogo de profesores, si existe nos despliega una pantalla donde se capturaran los valores que vienen en el formato de evaluación al profesor, esta pantalla estará presente hasta que se indique que la información es correcta.

Cuando la información este correcta se grabara en el archivo de evaluaciones a profesores.

Este programa se ejecutara hasta que le indiquemos que no deseamos capturar otro registro.

Función Prof_EXC()

PROF_REV.PRG

Función que realiza la búsqueda de los profesores que tienen más horas asignadas en comparación con sus horas autorizadas.

Verifica si existe el archivo EXC_HRS.DBF (exc), de ser así limpia el archivo; si no existe lo crea. Posteriormente utiliza el catálogo de profesores para realizar la búsqueda de profesores con aumento de horas.

Para realizar dicha búsqueda primeramente se establece una comunicación con el Sistema de Banco de Horas mediante la función BAPRHOR(), transmitiendo el RFC del profesor que se va a verificar, de esta función obtenemos un arreglo del cual solo utilizamos la siguiente información:

- (1) Horas Autorizadas en la Coordinación
- (7) Horas Teóricas del Semestre Actual
- (8) Horas Prácticas del Semestre Actual

(Para mayor referencia ver Tesis del Sistema de Banco de Horas)

Se suman las horas teóricas y las horas practicas para comparar el resultado con las horas autorizadas, si estas últimas son menores que las primeras se da de alta el registro en el archivo EXC_HRS, cuando se han revisado todos los registros del catálogo de profesores se despliega en pantalla una ventana con los profesores que tienen mas horas asignadas que autorizadas.

Programa PROF_FOR.PRG

Programa que realiza el formato de la propuesta para la contratación de profesores.

En este programa se abren los siguientes archivos, los cuales se cierran al terminar el programa:

- PROFCAT (Contiene el catálogo de profesores)
- CAT (Catálogo de categorías)

Además se utilizan los siguientes archivos:

PC+Iniciales de la coordinación+Período

(Contiene el catálogo de profesores de la coordinación(coor))

PR+Iniciales de la coordinación+Período

(Archivo de propuestas revisadas y listas para imprimir (prop))

Desde este programa se configura la impresora, también se convierte la fecha al formato necesario para la propuesta.

Para imprimir la propuesta se toman datos de los catálogos de profesores y del archivo de propuestas revisadas.

Los datos para la forma de seguro de vida únicamente se imprimirán cuando se trate de un nuevo ingreso.

Se imprimirá una propuesta para cada tipo de movimiento que tenga un profesor.

Programa PROF_FUN.PRG

Este programa contiene un conjunto de funciones que son utilizadas por los programas que forman el Sistema de Control de Profesores.

CUADRO() (Dibuja un cuadro con o sin sombra)

SOMLCG() (Realiza la sombra tenue)

MENUV() (Forma el menú de opciones vertical)

MENUH() (Forma el menú de opciones horizontal)

MENSAJEE() (Envía cuadro con mensaje de un renglón con o sin sonido y retardo)

FILTRO() (Avisa si una condición no es correcta y permite reintentar)

REPORTE() (Rutina de envío de reporte y chequeo de impresor)

SUSPENDER() (Permite suspender la impresión)

CLRSCR() (Limpia la pantalla con efectos)

BROWPRO() (Si un campo requiere de un código, pulsando ENTER se puede llamar esta función para que el usuario, elija en una ventana la opción que desee)

Esta función se auxilia de las siguientes funciones: AYU_DATO(), COD_AYUDA(), SOL127()

Este programa no utiliza ningún archivo de datos.

Programa PROF_GEN.PRG

Programa que prepara la información para generar las propuestas de profesores.

Para generar la información de las propuestas de profesores y las tarjetas de control de asistencia es necesario realizar una comunicación con el sistema de horarios mediante la función HORA_RFC_HOEX en la cual debemos transmitir el RFC, el AÑO y PERIODO en que se este trabajando, de dicha función se recibe un arreglo matricial del cual se toma la siguiente información:

- () Rfc Titular
- () Rfc Suplente
- () Horario
- () Clave de Materia
- () Grupo
- () Horas

(Para mayor referencia ver Tesis de Sistema de Horarios)

Una vez que se cuenta con esta información se hace un llamado a la función PROF_PP().

Función Prof_Gne()**PROF_PRO.PRG**

Función que muestra el menú para generar la información de las propuestas de profesores y las tarjetas de control de asistencia.

Esta función primeramente envía un mensaje de aviso para advertir que la información se va a modificar dando las opciones de continuar o cancelar.

Las opciones que muestra el menú son:

| | | |
|-------|------------|---|
| TODOS | PRO_GENT() | Genera la información para las propuestas para todos los profesores |
| RANGO | PRO_GENR() | Genera la información para un rango |

Programa Prof_Imp.PRG

Programa que muestra el menú para imprimir las evaluaciones, las propuestas de profesores y/o tarjetas de control de asistencia.

En este programa se abren los siguientes archivos, los cuales se cierran al terminar el programa:

PC+Iniciales de la coordinación+Período

(Contiene el catálogo de profesores de la coordinación(coor))

PR+Iniciales de la coordinación+Período

(Archivo de propuestas revisadas y listas para imprimir (prop))

Antes que nada este programa manda un mensaje para advertir que la información puede no estar actualizada, dando la opción de cancelar o continuar.

El menú de impresión nos muestra las siguientes opciones:

| | | |
|------------|------------|---|
| Evaluación | PROF_REP | (Imprime la evaluación de profesores) |
| Propuestas | IMP_PROP() | (Imprimir las propuestas de profesores) |

Tarjetas

PROF_TAR

(Imprime las tarjetas de
control de asistencia)**Programa PROF_LCG.PRG**

El sistema de Control de Profesores comienza haciendo la declaración global de sus variables, definiendo el medio ambiente del sistema, se asignan las variables para controlar los colores, así como variables para el manejo de la ruta de acceso a los archivos.

De igual manera se asignan las variables para formar los nombres de archivo que van a ser utilizados:

```
lccprocoo 'PC'+inicial de la coordinación+período  
lccproaut 'PA'+inicial de la coordinación+período  
lccproman 'PM'+inicial de la coordinación+período  
lccprorev 'PR'+inicial de la coordinación+período  
lccgeva 'PE'+inicial de la coordinación+período  
lccgmat 'DMC'+inicial de la coordinación
```

Se tiene la pantalla de presentación del Sistema de Control de Profesores que contiene el menú principal con las siguientes opciones:

```
CONSULTA PROF_CON(Consulta al archivo de profesores)  
MOVIMIENTOS PROF_MOV(Actualización al archivo de profesores)  
PROPUESTAS PROF_PRO(Generación de propuestas y tarjetas)  
EVALUACIÓN PROF_EVA(Evaluación a profesores)  
IMPRESIÓN PROF_IMP(Impresión de propuestas, tarjetas y/o  
evaluación)
```

Este programa no utiliza ningún archivo de datos.

Programa PROF_MAN.PRG

Programa que permite revisar la información antes de imprimir las propuestas manuales.

En este programa se abren los siguientes archivos:

PR+iniciales de la coordinación+período(rev)

El cual contiene la información de las propuestas revisadas y listas para imprimir.

TIPOMOV (mov) (Catálogo de Tipos de Movimiento)

En este programa se hace un llamado al actualizador MOV_PMAN() el cual realiza los movimientos al archivo de propuestas manuales.

Este programa muestra la mascarilla de captura mediante la función LCGSCREM() y solicita el RFC del profesor más el índice de registro de la propuesta que se va a revisar, tiene la alternativa de consultar mediante un browse de ayuda del archivo de propuestas manuales, esta ayuda se despliega al dar ENTER o al oprimir la tecla de función F2.

Este programa cuenta con una función para dar de alta algún registro ALT_PMAN() la cual se activa al oprimir la tecla de función F1 y otra para dar de baja BAJ_PMAN() esta se activa al oprimir la tecla de función F3.

Si lo que se desea es hacer cambios a la información se presenta para su edición con la función CAM_PMAN().

Si se realiza una alta primeramente se verifica si el RFC existe en el catálogo de profesores, posteriormente se busca en los archivos de propuestas automáticas y manuales para verificar cual es el último índice para ese RFC y poder asignarle un número consecutivo.

Ya sea que se realice un cambio o una alta se hará un llamado a la función GRA_PMAN() para grabar los movimientos al archivo de propuestas manuales.

En el caso de la clave de la materia solo se permitirá su edición si está es de tipo carácter.

Se cuenta con ayuda para el tipo de movimiento, dicha ayuda se activa con la tecla de función F2.

En caso de que el horario cambie se desplegara una pantalla anexa para modificar también la información que se imprimirá en las tarjetas de control de asistencia.

Antes de aceptar el movimiento se verifica que el total de horas asignada no rebase las horas autorizadas, si esto sucediera se manda un mensaje de error y se permite modificar el número de horas autorizadas.

Se actualiza la información de manera simultánea en el archivo de propuestas manuales y en el archivo de propuestas revisadas.

Por último se va al catálogo de profesores para asignarle un asterisco en el campo propuesta para tener identificados los profesores que tienen lista su propuesta.

Programa PROF_MOV.PRG

Programa que realiza los movimientos al catálogo de profesores, es llamado por el programa PROF_LCG.

En este programa se abren los siguientes archivos de datos, los cuales se cierran al concluir el programa:

PROFCAT (Contiene los datos personales del profesor (pro))
PCICO971 (Contiene información del profesor dependiendo de la
coordinación (coor))
CAT (Catálogo de categorías (cat))

Este programa muestra la mascarilla de captura mediante la función PROSCREE() y solicita el RFC del profesor que se va a actualizar, tiene la alternativa de consultar mediante un browse de ayuda del catálogo de profesores, esta ayuda se despliega al dar ENTER o al oprimir la tecla de función F2.

Al dar un RFC se busca en el catálogo de profesores si no se encuentra se da de alta pidiendo los datos del profesor, si se encuentra se presenta la información del profesor permitiendo la edición con la función PROF_CAM(). En ambos casos se hace un llamado a la función PROF_GRA() para grabar los cambios al catálogo de profesores.

Cuando hay algún movimiento en el catálogo de profesores se realiza una comunicación con el sistema de banco de horas mediante la función BAALTPR2 transmitiendo el RFC, Nombre del profesor, si es funcionario, su categoría.

Este programa cuenta con una función para dar de baja a algún profesor y se activa al oprimir la tecla de función F3.

En el catálogo de profesores de la coordinación se podrá dar de baja un profesor solo si la fecha de baja es mayor a la fecha de inicio de propuesta pero menor que la fecha de término, además debe coincidir la fecha de antigüedad con la fecha de inicio, si todo esto se cumple se debe verificar que no existan movimientos en horarios, ni en banco de horas, ni en extraordinarios, esto se logra mediante las siguientes funciones:

HORABAJA() (Ver tesis de horarios)

BANKBAJA()(Ver tesis de banco de horas)
EXTRBAJA()(Ver tesis de extraordinarios)

Solo si no se tiene ningún movimiento se permitirá que el registro sea dado de baja.

Función Prof_NTX()**PROF_CNX.PRG**

Función que genera los índices de todos los archivos necesarios para el funcionamiento del sistema.

Función Prof_PP()**PROF_GEN.PRG**

Función que genera la información para las propuestas de profesores y las tarjetas de control de asistencia.

En esta función se convierte el horario en que se imparte la materia al texto adecuado como debe ser impreso en la propuesta de profesor, así como en las tarjetas de control de asistencia.

Si las horas anteriores y las horas actuales son iguales la propuesta se generara de manera automática y se e le considerara como prorroga y se incluirá en el archivo PA+Iniciales de la Coordinación+Período; de no ser así se generara la propuesta de forma manual, incluyendo la información del horario actual, estos registros se incluirán en el archivo PM+Iniciales de la Coordinación+Período.

Programa PROF_PRO.PRG

Programa que muestra el menú para generar y revisar las propuestas de profesores y tarjetas de control de asistencia es llamado por el programa PROF_LCG.

En este programa se abren los siguientes archivos de datos, los cuales se cierran al concluir el programa:

PCICO971 (Contiene información del profesor dependiendo de la coordinación (coor))

PAICO971 (Contiene información para las propuestas
automáticas (paut))
PMICO971 (Contiene información para las propuestas
manuales (pman))
DMCICO (Catálogo de materias de la coordinación (mat))

Presenta un menú de opciones para seleccionar si se quiere generar o revisar la información para las propuestas de profesores y tarjetas de control de asistencia.

Para generar la información de las propuestas se hace un llamado a la función PROF_GNE(), si se va a revisar dicha información se ejecuta el programa PROF_REV.

Programa PROF_REP.PRG

Programa que muestra el menú de impresión del reporte de evaluación.

En este programa se abre el archivo

PE+Iniciales de la coordinación+Período

(Archivo de Evaluaciones (eva))

el cual se cierra al terminar el programa.

Las opciones del menú son las siguientes:

TODOS PROF_RET()

RANGO PROF_RER()

Función Prof_ReR()

PROF_REP.PRG

Función que imprime el reporte de la evaluación para un rango de profesores.

Esta función pide el RFC INICIAL y el RFC FINAL del rango que se desea imprimir, se cuenta con ayuda mediante un browse del catálogo de profesores con los registros que tengan evaluación para imprimir, del cual pueden seleccionar el

RFC que se desee, si el RFC se teclea de manera manual se verifica que exista en el catálogo de profesores y tenga lista su evaluación, cuando se trata del RFC FINAL también verifica que no sea menor que el RFC INICIAL.

Desde esta función se configura la impresora y se imprime el reporte de la evaluación al ejecutar la función REPO_EVA() el cual se explicará más adelante.

Función Prof_ReT()**PROF_REP.PRG**

Función de impresión del reporte de la evaluación de todos los profesores desde esta función se configura la impresora y se imprime el reporte de la evaluación al ejecutar la función REPO_EVA() el cual se explicará más adelante.

Programa PROF_REV.PRG

Programa que muestra el menú para revisar la información de profesores las opciones que nos presenta son las siguientes:

| | | |
|-------------------|------------|--|
| PROP. AUTOMÁTICAS | PROF_AUT | (Revisa las propuestas Automáticas) |
| PROP. MANUALES | PROF_MAN | (Revisa las propuestas Manuales) |
| AUMENTO DE HORAS | PROF_EXC() | (Revisa los profesores que tienen aumento de horas) |
| PROF. SIN HORARIO | PROF_SHR() | (Revisa los profesores que no tienen horario asignado) |

Función Prof_SHr()**PROF_REV.PRG**

Función que realiza la búsqueda de los profesores que no tienen horario asignado.

Verifica si existe el archivo SIN_HOR.DBF (sinh), de ser así limpia el archivo, si no existe lo crea, posteriormente utiliza el catálogo de profesores para realizar la búsqueda de profesores sin horario asignado.

Para realizar dicha búsqueda primeramente se hace una comunicación con el sistema de horarios mediante la función HORA_RFC_HOEX(), transmitiendo el RFC del profesor que se va a verificar, así como el año y período en que se este trabajando.

(Para mayor referencia ver Tesis del Sistema de Horarios)

Si dicha función nos devuelve un valor NULO, damos de alta el registro en el archivo SIN_HOR, cuando se han revisado todos los registros del catálogo de profesores se despliega en pantalla una ventana con los profesores que no tienen horario asignado.

Programa PROF_TAR.PRG

Programa que muestra el menú para imprimir las tarjetas de control de asistencia.

En este programa se abren los siguientes archivos, los cuales se cierran al terminar el programa:

PC+Iniciales de la coordinación+Período

(Contiene el catálogo de profesores de la coordinación(coor))

PR+Iniciales de la coordinación+Período

(Archivo de propuestas revisadas y listas para imprimir (prop))

En principio pide a partir de que mes se van a imprimir las tarjetas, si se da ENTER se muestra una lista con los meses del año para que se seleccione el mes de inicio, los otros meses se calculan de manera automática tomando como punto de partida el mes inicial; además solicita renglón y columna inicial.

El menú nos presenta las siguientes opciones:

TODOS-CARA A TARAIMPT() (Imprime la cara "A" de
todas las tarjetas)

| | |
|--------------|--|
| TODOS-CARA B | TARBIMPT() (Imprime la cara "B" de todas las tarjetas) |
| TODOS-AMBAS | TARDIMPT() (Imprime ambas caras de todas las tarjetas) |
| RANGO-CARA A | TARAIMPR() (Imprime la cara "A" de un rango de tarjetas) |
| RANGO-CARA B | TARBIMPR() (Imprime la cara "B" de un rango de tarjetas) |
| RANGO-AMBAS | TARDIMPR() (Imprime ambas caras de un rango de tarjetas) |

Función Pro_GenR()**PROF_PRO.PRG**

Función que genera la información para la propuesta de un rango de profesores.

Esta función pide el RFC INICIAL y el RFC FINAL del rango que se desea procesar, se cuenta con ayuda mediante un browse del catálogo de profesores del cual pueden seleccionar el RFC que se desee, si el RFC se teclea de manera manual se verifica que exista en el catálogo de profesores cuando se trata del RFC FINAL también verifica que no sea menor que el RFC INICIAL.

Se genera la información para el rango seleccionado del catálogo de profesores.

Primeramente se borran los registros del rango de los archivos de propuestas automáticas y propuestas manuales.

Posteriormente se realiza una comunicación con el Sistema de Banco de Horas mediante la función BAPRHOR() en la cual se transmite el R.F.C. y de ella recibimos un arreglo de ocho parámetros en el siguiente orden:

- (1) Horas Autorizadas en la Coordinación
- (2) Horas Asignadas en otras Coordinaciones
- (3) Nombre de la Coordinación en la que se tenga mayor número de horas asignadas y un asterisco por cada una de las otras coordinaciones en las que también se tengan horas asignadas como parte del nombre de

- otra jefatura
- (4) Categoría que tenga el profesor en otra coordinación
 - (5) Horas Teóricas del Semestre Anterior
 - (6) Horas Prácticas del Semestre Anterior
 - (7) Horas Teóricas del Semestre Actual
 - (8) Horas Prácticas del Semestre Actual

(Para mayor referencia ver Tesis del Sistema de Banco de Horas)

Cuando ya se cuenta con esta información se ejecuta el programa PROF_GEN el cual se explicara posteriormente.

Función Pro_GenT()**PROF_PRO.PRG**

Función que genera la información para la propuesta de todos los profesores.

En esta función se zapean los siguientes archivos:

- PA+iniciales de la coordinación+período(Paut)
- PM+iniciales de la coordinación+período(Pman)

Se generara la información para todo el archivo del catálogo de profesores.

Priviamente se realiza una comunicación con el Sistema de Banco de Horas mediante la función BAPRHOR() en la cual se transmite el RFC y de ella recibimos un arreglo de ocho parámetros en el siguiente orden:

- (1) Horas Autorizadas en la Coordinación
- (2) Horas Asignadas en otras Coordinaciones
- (3) Nombre de la Coordinación en la que se tenga mayor número de horas asignadas y un asterisco por cada una de las otras coordinaciones en las que también se tengan horas asignadas como parte del nombre de otra jefatura
- (4) Categoría que tenga el profesor en otra coordinación
- (5) Horas Teóricas del Semestre Anterior
- (6) Horas Prácticas del Semestre Anterior
- (7) Horas Teóricas del Semestre Actual
- (8) Horas Prácticas del Semestre Actual

(Para mayor referencia ver Tesis del Sistema de Banco de Horas)

Cuando ya se cuenta con esta información se ejecuta el programa PROF_GEN el cual se explicara posteriormente.

Función Pro_ImpR()**PROF_IMP.PRG**

Función que imprime las propuestas de un rango de profesores.

Esta función pide el RFC INICIAL y el RFC FINAL del rango que se desea imprimir, se cuenta con ayuda mediante un browse del catálogo de profesores con los registros que tengan lista su propuesta para imprimir, del cual pueden seleccionar el RFC que se desee, si el RFC se teclea de manera manual se verifica que exista en el catálogo de profesores y tenga lista su propuesta, cuando se trata del RFC FINAL también verifica que no sea menor que el RFC INICIAL.

Se imprimen las propuestas del rango seleccionado mediante la ejecución del programa PROF_FOR el cual se explicará más adelante.

Función Pro_ImpT()**PROF_IMP.PRG**

Función que imprime las propuestas de todos los profesores.

Verifica en el catálogo de profesores de la coordinación que profesores tienen lista para imprimir su propuesta, esto lo verifica en el campo propuesta, ya que solo se imprimirán las propuestas de aquellos profesores que en su registro tengan un "*" en dicho campo.

Cuando se cumpla esta condición se ejecuta el programa PROF_FOR que se explicará más adelante.

Función Ran_Tar()**PROF_TAR.PRG**

Función que selecciona un rango de tarjetas.

Esta función pide el RFC INICIAL y el RFC FINAL del rango que se desea procesar, se cuenta con ayuda mediante un browse del catálogo de profesores, del cual pueden seleccionar el RFC que se desee, si el RFC se teclea de manera

manual se verifica que exista en el catálogo de profesores, cuando se trata del RFC FINAL también verifica que no sea menor que el RFC INICIAL.

Función Repo_Eva()**PROF_REP.PRG**

Función que realiza el reporte de la evaluación.

En esta función se calcula los promedios para cada una de las preguntas, esto se logra sumando todos los valores de cada una de las preguntas de cada grupo-materia de un profesor y dividiendo este total entre el número de cuestionarios.

Por último se imprime el reporte de la evaluación.

Función TarAImpR()**PROF_TAR.PRG**

Función que imprime un rango de tarjetas de asistencia cara "A" mediante la función PROFTARA().

Función TarAImpT()**PROF_TAR.PRG**

Función que imprime todas las tarjetas de asistencia por la cara "A", mediante la función PROFTARA().

Función TarBImpR()**PROF_TAR.PRG**

Función que imprime un rango de tarjetas de asistencia cara "B" mediante la función PROFTARB().

Función TarBImpT()**PROF_TAR.PRG**

Función que imprime todas las tarjetas de asistencia por la cara "B", mediante la función PROFTARB().

Función TarDImpR()**PROF_TAR.PRG**

Función que imprime un rango de tarjetas de asistencia ambas caras. Primero se imprime la cara "A" del rango de tarjetas mediante la función PROFTARA(), una vez terminado este proceso envía un mensaje avisando que la impresión de la cara "A" de las tarjetas ha concluido y pide que se prepare la cara "B" de las tarjetas dando la opción de cancelar o continuar, si se continua se imprimirá la cara "B" de las tarjetas mediante la función PROFTARB().

Función TarDImpT()**PROF_TAR.PRG**

Función que imprime todas las tarjetas de asistencia por ambas caras; primero se imprime la cara "A" de todas las tarjetas mediante la función PROFTARA(), una vez terminado este proceso envía un mensaje avisando que la impresión de la cara "A" de las tarjetas ha concluido y pide que se prepare la cara "B" de las tarjetas dando opción de cancelar o continuar.

Si se continua se imprimirá la cara "B" de las tarjetas mediante la función PROFTARB().

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON
UNIDAD ADMINISTRATIVA
SECRETARIA DE CARRERA DE INGENIERIA EN COMPUTACION

| | | | | | | |
|---|---------------------------------|---|--|-------------------|--------------------|----------------|
| NOMBRE: NIETO TORRES ANTONIO | | R.F.C.: NITA460623 | | | | |
| DIRECCION: FAUSTO VEGAS 740 COL. ESCUADRON 201 | | TELEFONO: 670-44-43 | | | | |
| TIPO DE GOBIERNO: PROCEGA | | CATEGORIA: PROFESOR ASIGNATURA 'A' | | | | |
| FECHA INICIO: 16/ENE/97 | FECHA LECTIV: 15/JUL/97 | NUMERO DE HORAS: 18.0 | | | | |
| NUMERO DE HORAS DEL SEMESTRE ANTERIOR: TEORICAS (12.0) | PRACTICAS (0.0) | TOTAL (12.0) | | | | |
| Imparte clases en otra(s) Jefaturas(S) de Carrera de E.N.E.P. Aragón (Indique cual) A R E A : _____ CATEGORIA: _____ NUMERO DE HORAS: _____ | | | | | | |
| Imparte clases en otra(s) Dependecia(s) de la U.N.A.M. (Indique cual) DEPENDENCIA: _____ CATEGORIA: _____ NUMERO DE HORAS: _____ | | | | | | |
| DATOS PARA LA FORMA DE SEGURO DE VIDA | | | | | | |
| NOMBRE DE LOS BENEFICIARIOS | SENO | PARENTESCO | PORCENTAJE % | | | |
| | | | | | | |
| CLAVE | ASIGNATURA | GRUPO | HORARIO | H-TEORICAS | H-PRACTICAS | H-TOTAL |
| 1400 | ANALISIS DE SISTEMAS Y SEÑALES | 1407 | MAÑTES, JUEVES, (11:15-13:15) | 4.0 | 0.0 | 4.0 |
| 0712 | PROBABILIDAD Y ESTADISTICA | 1408 | LABES, MIÉRCOLES, VIERNES, (11:30-13:00) | 4.5 | 0.0 | 4.5 |
| 1720 | DISEÑO LÓGICO | 1407 | MAÑTES, JUEVES, (09:00-11:00) | 4.0 | 0.0 | 4.0 |
| 1400 | ANALISIS DE SISTEMAS Y SEÑALES | 1406 | MAÑTES, JUEVES, (07:00-09:00) | 4.0 | 0.0 | 4.0 |
| 0712 | PROBABILIDAD Y ESTADIST. (LJWB) | | LABES (10:00-11:30) | 0.0 | 1.5 | 1.5 |
| TOTAL DE HORAS: | | | | 16.5 | 1.5 | 18.0 |

San Juan de Aragón, Edo. de Méx., a 09 de Junio de 1997

JEFE DE CARRERA

INTERESADO

ING. JUAN CASTALDI PEREZ

NIETO TORRES ANTONIO

DEPARTAMENTO DE PERSONAL



Universidad Nacional Autónoma de México

Campus Aragón

Carrera: _____

Semestre: _____



REGISTRO DE ASISTENCIA DEL PERSONAL DOCENTE EN EL MES DE _____

| DIA | FIRMA | DIA | FIRMA | DIA | FIRMA |
|-----|-------|-----|-------|-----|-------|
| 1 | | 11 | | 21 | |
| 2 | | 12 | | 22 | |
| 3 | | 13 | | 23 | |
| 4 | | 14 | | 24 | |
| 5 | | 15 | | 25 | |
| 6 | | 16 | | 26 | |
| 7 | | 17 | | 27 | |
| 8 | | 18 | | 28 | |
| 9 | | 19 | | 29 | |
| 10 | | 20 | | 30 | |
| | | | | 31 | |

| OBSERVACIONES | DIA | HORARIO | CONCEPTO | TOTAL | % |
|---------------|-----|---------|--------------------------|-------|---|
| | L | | | | |
| | M | | CURSOS PROGRAMADOS | | |
| | MA | | CURSOS INMEDIADOS | | |
| | J | | FALTAS | | |
| | V | | PORCENTAJE DE ASISTENCIA | | |
| | S | | | | |

POSICIÓN

MATENCIÓN

GRUPO

A.F.C.

HORARIO

REGISTRO DE ASISTENCIA DEL PERSONAL DOCENTE EN EL MES DE _____

| DIA | FIRMA | DIA | FIRMA | DIA | FIRMA |
|-----|-------|-----|-------|-----|-------|
| 1 | | 11 | | 21 | |
| 2 | | 12 | | 22 | |
| 3 | | 13 | | 23 | |
| 4 | | 14 | | 24 | |
| 5 | | 15 | | 25 | |
| 6 | | 16 | | 26 | |
| 7 | | 17 | | 27 | |
| 8 | | 18 | | 28 | |
| 9 | | 19 | | 29 | |
| 10 | | 20 | | 30 | |
| | | | | 31 | |

REGISTRO DE ASISTENCIA DEL PERSONAL DOCENTE EN EL MES DE _____

| DIA | FIRMA | DIA | FIRMA | DIA | FIRMA |
|-----|-------|-----|-------|-----|-------|
| 1 | | 11 | | 21 | |
| 2 | | 12 | | 22 | |
| 3 | | 13 | | 23 | |
| 4 | | 14 | | 24 | |
| 5 | | 15 | | 25 | |
| 6 | | 16 | | 26 | |
| 7 | | 17 | | 27 | |
| 8 | | 18 | | 28 | |
| 9 | | 19 | | 29 | |
| 10 | | 20 | | 30 | |
| | | | | 31 | |

POSICIÓN

MATENCIÓN

GRUPO

**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
CAMPUS ARAGON UNIDAD DE PLANEACION
DEP. DE INFORMATICA EN COLABORACION CON ING. EN COMPUTACION
REPORTE DE EVALUACION POR PROFESOR**

| GRUPO : MATERIA : PROFESOR : | 1787 CONTROL ANALOGICO LOPEZ TRUJANO ISAAC | CLAVE DE LA MATERIA : No. de Cursel. : | 0112 28 | SEMESTRE : No. Pag. 5 | 87-1 |
|--|--|---|---------------------------|--------------------------|----------------------|
| PARTE I : PREGUNTAS | AUTOEVALUACION DEL ALUMNO | EVALUACION CUANTITATIVA | EVALUACION CUALITATIVA | EVALUACION MINIMA | EVALUACION MAXIMA |
| A) PLANEACION DEL CURSO | | 9.59 | EXCELENTE | | |
| 1 - EN QUE MOMENTO DEL SEMESTRE TE INTERESASTE POR CONOCER EL PROGRAMA DE LA MATERIA ? | | | | | |
| B) CONOCIMIENTO DE LA MATERIA | | 7.93 | BUENO | | |
| 2 - QUE NIVEL DE COMPRENSION CONSIDERAS QUE ALCANZASTE SOBRE LOS CONTENIDOS REVISADOS EN EL CURSO ? | | | | | |
| C) METODOLOGIAS Y ESTRATEGIAS DIDACTICAS | | 7.31 | BUENO | | |
| 3 - PARTICIPASTE EN LAS DIVERSAS ACTIVIDADES DE ENSEÑANZA APRENDIZAJE DESARROLLADAS EN CLASE ? | | | | | |
| D) SEGUIMIENTO Y EVALUACION | | 9.03 | BUENO | | |
| 4 - CUMPLISTE CON LOS REQUISITOS ESTABLECIDOS PARA TU EVALUACION ? | | | | | |
| E) DISCIPLINA Y PUNTUALIDAD | | 9.17 | EXCELENTE | | |
| 5 - ASISTE CON PUNTUALIDAD Y CONSTANCIA AL CURSO ? | | | | | |
| PROMEDIO DE LA EVALUACION DEL ALUMNO | | 8.61 | EXCELENTE | | |
| PARTE II : EVALUACION DEL ALUMNO HACIA EL PROFESOR | | | | | |
| A) PLANEACION DEL CURSO | | 9.66 | EXCELENTE | 9.66 | 9.66 |
| 1 - EN QUE MOMENTO DEL SEMESTRE PRESENTO EL PROGRAMA DEL CURSO ? | | | | | |
| 2 - SE OBSERVO UNA PREPARACION CONSTANTE DE LAS CLASES EN EL TRANCURSO DEL SEMESTRE ? | | | | | |
| B) CONOCIMIENTO DE LA MATERIA | | 9.02 | EXCELENTE | 9.02 | 9.02 |
| 3 - SE PERCIBIO QUE TIENE LOS CONOCIMIENTOS NECESARIOS SOBRE LA MATERIA ? | | | | | |
| 4 - SE ALCANZARON LOS OBJETIVOS PLANTEADOS AL INICIO DEL CURSO ? | | | | | |
| C) METODOLOGIA Y ESTRATEGIAS DIDACTICAS | | 9.16 | EXCELENTE | 9.16 | 9.16 |
| 5 - HUBO CONTINUIDAD Y COHERENCIA EN EL DESARROLLO DE LOS CONTENIDOS DEL CURSO ? | | | | | |
| 6 - SE PROFICIO EN LOS ALUMNOS UNA ACTITUD REFLEXIVA, PARTICIPATIVA Y DE COMUNICACION GRUPAL ? | | | | | |
| D) SEGUIMIENTO Y EVALUACION | | 8.38 | BUENO | 8.38 | 8.38 |
| 7 - REALIZO UNA EVALUACION DEL CURSO EN FORMA CONTINUA Y GRADUAL QUE PERMITIO CONOCER TU APROVECHAMIENTO ESCOLAR ? | | | | | |
| 8 - PROMOVIO LA INVESTIGACION Y TAREAS EXTRA CLASE PARA FORTALECER LOS CONTENIDOS REVISADOS EN EL PROGRAMA DE LA MATERIA ? | | | | | |
| E) DISCIPLINA Y PUNTUALIDAD | | 8.24 | BUENO | 8.24 | 8.24 |
| 9 - ASISTIO CON PUNTUALIDAD Y CONSTANCIA A CLASES ? | | | | | |
| 10 - LA RELACION MAESTRO-ALUMNO SE ESTABLECIO EN UN MARCO DE RESPETO ? | | | | | |
| PROMEDIO DE LA EVALUACION DEL PROFESOR : | | 8.89 | EXCELENTE | | |
| E= EXCELENTE | (8.4 - 7.0) | B= BUENO | (6.9 - 6.0) | R= REGULAR | (5.9 - 5.0) |
| | | | | D= DEFICIENTE | (10 - 8.5) |

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
 CARPUS ARAGON
 1mo. SEMESTRE 97-2
 EVALUACION GLOBAL DE PROFESORES EN ORDEN POR PROMEDIO

| C. R. | CPO. | MONEDRE DE LA MATERIA | NOMBRE DEL PROFESOR | MC | PROM | EVALUACION |
|-------|------|----------------------------------|---------------------------------|----|------|------------|
| 0480 | 2308 | METODOS NUMERICOS | RAMIREZ FLORES LUIS | 48 | 9.71 | EXCELENTE |
| 0129 | 2557 | DINAMICA DE SISTEMAS FISICOS | HERNANDEZ GUTIERREZ JORGE E | 11 | 9.22 | EXCELENTE |
| 0071 | 2307 | ELECTRICIDAD Y MAGNETISMO | BACON VERA | 70 | 9.66 | EXCELENTE |
| 0129 | 2508 | DINAMICA DE SISTEMAS FISICOS | VERDE CRUZ ABEL | 18 | 9.63 | EXCELENTE |
| 0129 | 2558 | DINAMICA DE SISTEMAS FISICOS | LUNA ESCORZA PABLO | 6 | 9.63 | EXCELENTE |
| 0401 | 2458 | INGENIERIA DE SISTEMAS | PORTILLA VAZQUEZ HUGO | 15 | 9.57 | EXCELENTE |
| 0138 | 2258 | DISPOSITIVOS ELECTRONICOS | HERNANDEZ GUTIERREZ JORGE E | 5 | 9.56 | EXCELENTE |
| 1720 | 2659 | DISEÑO LOGICO | HERRARDO GONZALEZ ANTONIA | 25 | 9.53 | EXCELENTE |
| 1200 | 2208 | ALGEBRA LINEAL | LEON Y BERBER RAUL W. | 11 | 9.51 | EXCELENTE |
| 0762 | 2058 | RECURSOS Y NECESIDADES DE MEX | COLUMBA VAZQUEZ NATILDE | 13 | 9.51 | EXCELENTE |
| 1400 | 2457 | DINAMICA DE SISTEMAS Y SEÑALES | VERDE CRUZ ABEL | 62 | 9.50 | EXCELENTE |
| 0112 | 2758 | CONTROL ANALOGICO | LOPEZ TRUJANO ISSAC I. | 13 | 9.49 | EXCELENTE |
| 0063 | 2258 | CALCULO VECTORIAL | RENDIZ ACUVEDO GERARDO | 25 | 9.48 | EXCELENTE |
| 0232 | 2307 | INTRODUCCION A LA ECONOMIA | GARCIA VILLANUEVA RA. DEL PILAR | 11 | 9.48 | EXCELENTE |
| 0840 | 2458 | SISTEMAS OPERATIVOS | GUTIERREZ OROZCO RICARDO | 15 | 9.45 | EXCELENTE |
| 0075 | 2208 | PROGRAM ESTRUCT. Y CARAC LENGUAJ | PENALOZA ROMERO ERNESTO | 59 | 9.43 | EXCELENTE |
| 0742 | 2058 | RECURSOS Y NECESIDADES DE MEX | GUTIERREZ CASTILLO ALMA ROSA | 15 | 9.41 | EXCELENTE |
| 0744 | 2057 | RECONOCIMIENTO DE PATRONES | MONTERROSA ESCOBAR ALICIA A. | 12 | 9.40 | EXCELENTE |
| 1110 | 2158 | ALGEBRA | LEON Y BERBER RAUL W. | 9 | 9.38 | EXCELENTE |
| 0875 | 2257 | PROGRAM ESTRUCT. Y CARACT LENG. | GUTIERREZ OROZCO RICARDO | 38 | 9.34 | EXCELENTE |
| 8789 | 2888 | SISTEMAS DE INFORMACION | GUTIERREZ OROZCO RICARDO | 30 | 9.35 | EXCELENTE |
| 8019 | 2259 | ADMINISTRACION CONTAB. Y COSTOS | ZARAGOZA BUCMAIN RODOLFO | 3 | 9.33 | EXCELENTE |
| 0742 | 2067 | RECURSOS Y NECESIDADES DE MEX | MARTINEZ ITURBE FERNANDO | 11 | 9.32 | EXCELENTE |
| 0840 | 2408 | SISTEMAS OPERATIVOS | GUTIERREZ OROZCO RICARDO | 35 | 9.32 | EXCELENTE |
| 0112 | 2707 | CONTROL ANALOGICO | ZUNIGA VILLEGAS BENITO | 9 | 9.27 | EXCELENTE |
| 1110 | 2157 | ALGEBRA | FLORES LOPEZ ENILLO LUIS | 8 | 9.27 | EXCELENTE |
| 1400 | 2407 | ANALISIS DE SISTEMAS Y SEÑALES | MIZTO TORRES ANTONIO | 68 | 9.19 | EXCELENTE |
| 0063 | 2208 | CALCULO VECTORIAL | SANCHEZ MARTINEZ RICARDO | 45 | 9.18 | EXCELENTE |
| 0075 | 2259 | PROGRAMACION ESTRUCTURADA | GONZALEZ BEDOLLA JOSE | 10 | 9.17 | EXCELENTE |
| 1109 | 2157 | CALCULO DIFERENCIAL E INTEGRAL | GONZALEZ ROMERO JUAN DE DIOS | 13 | 9.13 | EXCELENTE |
| 0480 | 2357 | METODOS NUMERICOS | SUZMAN VAZQUEZ GUILLERMO | 5 | 9.12 | EXCELENTE |
| 0613 | 2907 | ORGANIZ. Y ADMIN. DE CEN DE COMP | GONZALEZ BEDOLLA JOSE | 35 | 9.12 | EXCELENTE |
| 0019 | 2207 | ADMINISTRACION CONTAB. Y COSTOS | IBARRA ROSAS ALBERTO | 33 | 9.11 | EXCELENTE |
| 0076 | 2557 | BASES DE DATOS | GONZALEZ HERMANDEZ GABRIELA | 18 | 9.11 | EXCELENTE |
| 0434 | 2858 | COMPILADORES | BLANCO BAUTISTA ROBERTO | 5 | 9.08 | EXCELENTE |
| 0407 | 2438 | INGENIERIA DE PROGRAMACION | CRUZ LUEVANO BLANCA ESTELA | 12 | 9.07 | EXCELENTE |
| 1200 | 2258 | ALGEBRA LINEAL | LEBUS CRUZ JOSE BOLDORES | 22 | 9.04 | EXCELENTE |
| 1715 | 2457 | MEJICION E INSTRUMENTACION | VARGAS ALCALIA JOSÉ HUMBERTO | 36 | 9.02 | EXCELENTE |
| 0789 | 2857 | SISTEMAS DE INFORMACION | JUAREZ ORTEGA ISRAEL | 38 | 8.97 | EXCELENTE |
| 1200 | 2257 | ALGEBRA LINEAL | SARRIEMTO HERMANDEZ JOSE A. | 22 | 8.95 | EXCELENTE |
| 2135 | 2008 | ROBOTICA | ISLAS ARGUELLO ALJAMDRD | 19 | 8.94 | EXCELENTE |
| 0362 | 2657 | INVESTIGACION DE OPERACIONES | PORTILLA VAZQUEZ HUGO | 24 | 8.93 | EXCELENTE |
| 0994 | 2208 | OPATIVIA DE HUMADIDADES | RODRIGUEZ PEREZ RA. DE LOURDES | 34 | 8.90 | EXCELENTE |
| 1109 | 2107 | ALGEBRA | AREZCUA CAMPOVERDE ALFONSO | 42 | 8.89 | EXCELENTE |
| 1720 | 2658 | DISEÑO LOGICO | LOPEZ CARRETO JUAN MANUEL | 9 | 8.87 | EXCELENTE |
| 0407 | 2657 | INGENIERIA DE PROGRAMACION | ORTIZ PENA ANTONIO | 16 | 8.85 | EXCELENTE |
| 0613 | 2957 | ORGANIZ. Y ADMIN. DE CEN COMPUT | JUAREZ ORTEGA ISRAEL | 41 | 8.85 | EXCELENTE |
| 0071 | 2358 | ELECTRICIDAD Y MAGNETISMO | CARDJAL PINAL RAUL | 15 | 8.73 | EXCELENTE |
| 8405 | 2657 | SISTEMAS OPERATIVOS | LORIS ZAULICHA FERNANDO | 15 | 8.68 | EXCELENTE |
| 0114 | 2808 | CONTROL DIGITAL | VEGA RUMYOV SILVIA | 48 | 8.64 | EXCELENTE |

UNIDAD DE PLANEACION

DEPARTAMENTO DE INFORMATICA

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
 CAMPUS ARAGON
 ING. EN COMPUTACION SEMESTRE: 97-2
 EVALUACION GLOBAL DE PROFESORES EN ORDEN POR PROMEDIO

| C. N. | CPD | NOMBRE DE LA MATERIA | NOMBRE DEL PROFESOR | NC | PROM. | EVALUACION |
|-------|------|---------------------------------|------------------------------|----|-------|------------|
| 0019 | 2209 | ADMN. CONTABILIDAD Y COSTOS | CUTIERREZ CASTILLO ALNA ROSA | 43 | 8.64 | EXCELENTE |
| 1200 | 2207 | ALGEBRA LINEAL | MORGAN URQUIEZ RAFAEL | 30 | 8.63 | EXCELENTE |
| 0362 | 2658 | INVESTIGACION DE OPERACIONES | RESENDIZ GRANADOS ANTONIO | 6 | 8.57 | EXCELENTE |
| 0905 | 2907 | TEMAS ESPECIALES DE COMPUTACION | JUAREZ ORTEGA ISRAEL ANTONIO | 29 | 8.56 | EXCELENTE |
| 0712 | 2457 | PROBABILIDAD Y ESTADISTICA | FLORES ZAVALETA FERNANDO | 29 | 8.19 | BUENO |
| 0480 | 2358 | METODOS NUMERICOS | MENCHACA ROCHA ROSA MARIA | 17 | 8.04 | BUENO |
| 0633 | 2557 | PROGRAMACION DE SISTEMAS | MARTINEZ PISCADOR GERARDO | 6 | 7.75 | BUENO |
| 0071 | 2308 | ELECTRICIDAD Y MAGNETISMO | ARISTA PATINO FRANCISCO | 10 | 6.40 | REGULAR |



CAPITULO IV IMPLANTACION



CAPITULO IV. IMPLANTACIÓN

IV.1. Presentación al Usuario

IV.2. Programa de Capacitación

IV.3. Pruebas Generales del Sistema

IV.4. Descripción del Proceso Computarizado

CAPITULO IV**IMPLANTACIÓN**

Quando el Sistema ya ha sido desarrollado, la siguiente etapa es la puesta en operación, es decir hacer entrega del Sistema al usuario final para que se realicen las pruebas necesarias que permitirán evaluar si el Sistema cumple de una manera confiable con los objetivos para los cuales fue creado.

Si se quiere implantar con éxito un nuevo sistema de información, a toda persona que participe en él se le debe informar, en primer lugar, de las responsabilidades que asume y en segundo, de lo que va a obtener del sistema.

En el proceso de implantación el diseñador debe especificar el plan de capacitación del sistema y las pruebas que se han de efectuar del mismo.

Una vez implantado el sistema el analista desempeña la función de consultor. Inmediatamente después de la implantación, ayuda a los usuarios a entender el nuevo sistema, asistiéndolo en la solución de los problemas que se presenten, así como de las mejoras necesarias.

IV.1. PRESENTACIÓN AL USUARIO

En este apartado se describe la presentación del Sistema de Control de Profesores al personal de la Coordinación de Ingeniería en Computación y al de algunas otras Coordinaciones del Plantel.

La presentación al usuario se llevó a cabo cuando se presumía que el Sistema cumplía con los requerimientos necesarios para solucionar la problemática que implica el Control de Profesores sobre todo en cuanto a la generación de las Propuestas.

En la demostración del Sistema a manera de introducción se dio una plática en la cual se menciono la conveniencia de utilizar el Sistema Integral ZEUS y los beneficios que obtendrán al realizar sus procesos administrativos haciendo uso del mismo.

El punto de partida fue indicar al usuario la forma mediante la cual se debe acceder al Sistema y se describió a detalle su funcionamiento.

En cuanto al módulo de Control de Profesores su mayor beneficio es la generación automática de la propuesta "Formato para Movimientos de Personal Académico" una vez que se tienen debidamente especificados los horarios de cada profesor.

Contará con un catálogo de profesores en el cual encontrará tanto sus datos personales como administrativos el cual podrá ser consultado de una manera fácil y rápida.

Se presentaron los formatos impresos tanto de la Propuesta que será entregada al Departamento de Personal como de las Tarjetas de Control de Asistencia y los reportes de Evaluación de Profesores.

Finalmente se pidió a los usuarios de las otras Coordinaciones dieran su opinión en cuanto a lo que les parecía que le hiciera falta al Sistema, motivo por el cual sugirieron se agregaran algunos reportes administrativos. En cuanto a la operación del sistema opinaron que era sencilla y "amigable".

Se hizo la presentación de la propuesta impresa que será entregado al departamento de personal y de la impresión de tarjetas de control de asistencia.

IV.2. PROGRAMA DE CAPACITACIÓN

Como se mencionó anteriormente las personas son el elemento fundamental de todo sistema. Es necesario proporcionarles instrucción y capacitación adecuadas, tanto al principio como posteriormente, para que el sistema logre su objetivo.

Generalmente, el término "Instrucción" se usa cuando se trata de informar a los usuarios con respecto a lo que el sistema requiere y lo que puede proporcionar.

Por diversas razones, a menudo esta instrucción se reduce al mínimo o se omite durante el período de implantación de un sistema, como es el caso del Sistema que nos ocupa, debido a que el usuario final sería el Secretario Técnico de la Coordinación de Ingeniería en Computación, por lo tanto el programa de capacitación consistió únicamente en una explicación general del Sistema.

En cuanto a los usuarios de otras Coordinaciones, se elaboró un Manual de Operación que indica explícitamente cada módulo del Sistema para lograr el objetivo de capacitar al usuario final.

IV.3. PRUEBAS GENERALES DEL SISTEMA

La prueba del Sistema implica probar todos los programas que se implantarán para apoyar el Sistema. Todos los elementos de entrada se preparan y se procesan con el fin de producir los resultados previstos, los cuales se verifican en cuanto a exactitud y confiabilidad.

Antes de realizar la presentación del sistema, se realizaron pruebas de todo tipo que garantizara la eficiencia del Sistema, verificando que las modificaciones en la información se reflejara correctamente en las bases de datos.

Se realizaron pruebas desde la captura de información hasta la impresión de propuestas y demás reportes.

IV.4. DESCRIPCION DEL PROCESO COMPUTARIZADO

En este punto se describe el procedimiento para realizar el Control de Profesores con el proceso automatizado.

A continuación se da una explicación detallada de las actividades realizadas para la generación e impresión de propuestas, la impresión de las tarjetas de control de asistencia y sus reportes administrativos, así como la captura de la evaluación de profesores y la emisión de los reportes correspondientes.

Como podemos observar el documento principal en el Control de Profesores es la Propuesta de contratación llamada "Forma para Movimientos de Personal Académico", por lo tanto mencionaremos a continuación la forma en que se procesan dichas formas mediante el proceso computarizado.

En primera instancia debemos tener debidamente actualizado el catálogo de profesores mediante la opción de Movimientos del módulo PROFESORES del Sistema ZEUS y bien definido nuestro Banco de Horas, haciendo uso del módulo BANCO DE HORAS del Sistema ZEUS, para poder realizar la asignación de Horarios por medio del módulo HORARIOS del mismo Sistema ZEUS, cuando dicha asignación ha sido concluida podemos decir que contamos con la información necesaria para la generación de la información que debe contener la Propuesta "Forma para Movimientos de Personal Académico", esto se logra con la opción Propuestas del módulo de Control de Profesores del Sistema ZEUS donde primeramente se selecciona generar propuestas donde el sistema verifica si el número de horas del semestre actual es igual al del semestre anterior de ser así se genera la propuesta como una prorroga en forma automática, de lo contrario se deberá generar la propuesta manualmente para lo cual se presentara el RFC con toda la información contenida en el archivo de horarios actual además del dato de horas autorizadas por el banco de horas enviando el mensaje "La propuesta no puede ser generada automáticamente debe realizarse manualmente". Cabe aclarar que al decir manualmente se refiere a hacerlo utilizando el sistema pero el usuario decide que tipo de movimiento será y con cuantas horas.

Una vez que han sido generadas las propuestas se debe revisar para lo cual al dar el RFC se presentará la información permitiendo que sea modificada. Se debe aclarar que en el caso tanto de la clave como del nombre de asignatura únicamente se permitirá su modificación cuando esta pertenezca a una materia de descarga.

Se debe tomar en cuenta que solo las propuestas revisadas podrán ser impresas.

Cuando ya han sido revisadas las Propuestas se imprimen en original y copia para que los profesores las firmen y puedan ser tramitadas.

De igual manera pueden ser impresas en este momento las Tarjetas de Control de Asistencia, ya que este proceso utiliza la misma información que en las propuestas, se va a imprimir una tarjeta por cada grupo-materia que imparte el profesor las cuales una vez firmadas y selladas se entregan en la sala de firmas el primer día de clases.

Por lo que respecta a la Evaluación de Profesores en cuanto ya se tienen concentrados los cuestionarios en la Coordinación se procede a capturar los resultados en la opción Evaluación del módulo de Control de Profesores del Sistema ZEUS. En el momento en que la captura sea concluida es posible efectuar la impresión de los reportes de diagnóstico.

Como puede apreciarse el proceso automatizado reduce el trabajo que realizan las secretarías ya que no tienen que mecanografiar una y otra vez las propuestas y los formatos de control de asistencia.

Es conveniente hacer mención de que tanto en el período 97-1 como en el período 97-2 las propuestas que presentó la Coordinación de Ingeniería en Computación al Departamento de Personal fueron generadas haciendo uso del módulo de Control de Profesores del Sistema ZEUS logrando una disminución de tiempo considerable en el manejo de información.

CONCLUSIONES

Es un hecho comprobado que al hacer uso de equipo de computo para llevar a cabo el manejo de información se obtienen grandes beneficios. Debido a que se puede tener acceso a la información de forma inmediata, pudiendo realizar consultas o modificaciones de la misma, de manera rápida y eficiente.

En base a los resultados obtenidos durante el semestre 97-1 podemos concluir que la implantación del sistema trajo consigo grandes beneficios en lo que respecta al manejo de información.

La etapa de análisis de un sistema de información es el punto climático, en el que se deslumbra sobre todo la permanencia del mismo a través del tiempo, al igual que el buen funcionamiento del mismo a corto plazo, logrando dar claridad a las necesidades de los usuarios finales; necesidades que rara ocasión son concretas.

El poder en un principio, determinar algunas de las modificaciones que pudiese tener el proceso administrativo, y de esta manera prevenir en la medida de lo posible modificaciones extensas, dentro del sistema **ZEUS**, una modificación de ultimo momento, genero la reprogramación en un módulo, gracias a un buen análisis y a la división de procesos, dichas modificaciones fueron mínimas para el módulo **PROFESORES**, apoyado este éxito en la generación de prototipos y simuladores para los procesos.

Uno de los más grandes problemas en un análisis es comprender el proceso que deseamos modelar, es por lo que dentro de cualquier equipo de analistas de sistemas, se integra al menos uno de los usuarios finales, en el caso de este sistema en particular el **Ing. Ernesto Peñaloza Romero**, sin el cual este trabajo no podría salir a la luz.

El desconocimiento, por parte de los usuarios de que información me puede dar su información y el miedo al cambio, generan frustración y hermetismo, no es raro, que un usuario acostumbrado a una herramienta, considere la existencia de un proceso sin ella.

La unión repentina de un grupo de trabajo completamente heterogéneo, brinda a cada uno de los módulos un toque especial, el trabajo en equipo, algo que casi nunca se puede lograr, en esta ocasión brindo grandes frutos, una gran experiencia que difícilmente se repetirá, lamentablemente, por condiciones especiales, algunos integrantes fueron sustituidos, lo que trajo demoras a nivel general.

El análisis de un sistema integral, inicia siempre por determinar las obligaciones de cada uno de los módulos, su interacción, con esto generamos las fronteras del mismo.

De la misma forma, para instituciones como la **U.N.A.M.** el empleo de nuevas herramientas (programas), empleando el equipo existente, ayudara para emplear esos recursos en otros rubros. Cuando me fue planteado el proyecto lo único que pude decir "Gracias por brindarme la oportunidad de aportarle un beneficio, a la institución que me formo como profesionista".

En la actualidad dicho sistema se encuentra operando en la coordinación de Ingeniería en Computación, **cuna** de la inquietud del mismo.

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGÓN

COORDINACIÓN DE INGENIERÍA EN COMPUTACIÓN

PROGRAMAS FUENTES

SISTEMA DE CONTROL DE PROFESORES

VER. 1.0

MÉXICO, 1997

```

*****
* PROGRAM-ID: Prof Lcg.prg ver 1.0
* AUTHOR: Leticia Cerd Garrido
* REMARKS: Mene Principal del Sistema de Control de Profesores
* MODIFIED: 17 Enero, 1997
* ULTIMA MODIFICACION: 23 DE ENERO DE 1997
* ERNESTO PEVALCOZA ROMERO
*****
MEMVAR getlist,vm_dir_sistema,vm_dir_carrera,vm_tit_carr,vm_anio_linea,;
vm_periodo_linea,lcgperiodo,lcgpaleta,lcgtinte,lcgtinte0,;
lcgtinte1,lcgtinte2,lcgtinte3,lcgtinte4,lcgtinte5,lcgtinte6,;
lcgenpana,lcgerror,lcginvisible,lcgvideo,lcgunam,lcgpanold,;
lcgruta,lcgeva,lcgmat,csuayuda,lcgtit1,lcgtit2,lcgtit_sem,lcgpan,;
lcglin,lcgunamp,lcgopm,titulo,texto,columna,lcgpunam,lcgtinte7,;
lcgzeus,vm_inicial,delact,lcgprococ,lcgproaut,lcgproman,;
lcgprorev
PUBLIC vm_dir_sistema,vm_dir_carrera,vm_tit_carr,vm_anio_linea,;
vm_periodo_linea,lcgperiodo,vm_inicial,lcgruta,lcgzeus,lcgeva,;
lcgmat,lcgprococ,lcgproaut,lcgproman,lcgprorev
/*****
* MEDIO AMBIENTE DEL SISTEMA
*****/
lcgpanold:=SAVESCREEEN(0,0,24,79)
*CLEAR
SET CURSOR OFF
SET ESCAPE ON
SET MESSAGE TO 23 CENTER
SET PROCEDURE TO PROF_FUN
SET PROCEDURE TO PROF_CNK
#include "grump.ch"
#include "fileio.ch"
#include "setcurs.ch"
#include "inkey.ch"

#define T BORRAR 127
#define T_INSERT 22

/*****
* ASIGNACION DE COLORES
*****/

* DEBEMOS ASIGNAR EN BASE A LOS COLORES DEL SISTEMA!!!
lcgpaleta:=lcgtinte:"W*/B"//AZUL
lcgtinte0:="W*/B,N/W,BG/GR+"//AZUL
lcgtinte1:="W*/BG" //CIAN
lcgtinte2:="W*/BG+" //GRIS
lcgtinte3:="W*/G" //VERDE
lcgtinte4:="N*BG" //NEGRO CIAN
lcgtinte5:="B*/RB+" //MAGENTA
lcgtinte6:="W*/N,GRGR+" //MARRON CLARO
lcgtinte7:="B*/W+" //GRIS CON NEGRO
lcgenpana:="B*/W+" //AZUL SUAVE
lcgerror:="W*/W,W/W" //ROJO
lcginvisible:="W*/W,W/W" //DE ACUERDO CON EL TINTE BASICO
lcgvideo:="GR*/B,BG/N,BG/GR+,B,W/B"
lcgunam:="GR*/B"

*****VARIABLES DE SISTEMA*****
vm_dir_sistema := 'C:\ZEUS'
vm_dir_carrera := 'C:\ZEUS\ICO'
vm_tit_carr := 'INGENIERIA EN COMPUTACION'
vm_anio_linea := '1997'
vm_periodo_linea:='1'

```

```

vm_inicial      := 'ICO'
*****

/*****
* VARIABLES PUBLICAS
*****/
lcgperiodo := alltrim(vm_ano_linea)
lcgperiodo := substr(lcgperiodo, len(lcgperiodo) -1, 2)
lcgperiodo := lcgperiodo + alltrim(vm_periodo_linea)
lcgruta := vm_dir_carrera
lcgzeus := vm_dir_sistema
lcgprocoo := 'PC'+vm_inicial+lcgperiodo
lcgproaut := 'PA'+vm_inicial+lcgperiodo
lcgproman := 'PM'+vm_inicial+lcgperiodo
lcgprorev := 'PR'+vm_inicial+lcgperiodo
lcgeva := 'PE'+vm_inicial+lcgperiodo
lcgmat := 'DMC'+vm_inicial

/*****
* PANTALLA DE PRESENTACION
*****/
suayuda:=0
SET COLOR TO lcgtinte
SET DEFAULT TO &lcgruta
lcgtit1 = "Universidad Nacional Autonoma de Mexico < ARAGON >"
lcgtit2 = "Sistema de Control de Profesores"
vm_tit_carr = " " + vm_tit_carr + " "
lcgtit_sem = " " + vm_ano_linea + "-" + vm_periodo_linea + " "
CUADRO(lcgtinte, 0, 0, 24, 79)
@ 0, 0 TO 24, 79 DOUBLE
@ 4, 1 TO 4, 78
lcgpan:= REPLICATE(' ', 76)
lcglin=7
DO WHILE lcglin<23
  @ lcglin, 2 SAY lcgpan
  ++lcglin
ENDDO
SETCOLOR(lcgunam)
@ 1, 40-LEN(lcgtit1)/2 SAY lcgtit1
@ 2, 40-LEN(lcgtit2)/2 SAY lcgtit2
SETCOLOR(lcgtinte)
@ 1, 1 SAY DTC( DATE() )
@ 1, 70 SAY TIME()
@ 3, 2 SAY vm_tit_carr
@ 3, 70 SAY lcgtit_sem
lcgunamp=SAVESCREEN(0, 0, 24, 79)
/*****
* MENU PRINCIPAL
*****/

/* eliminar la sig. linea*/
PROF_DBF()

RESTSCREEN(0, 0, 24, 79, lcgunamp)
lcgopm=1
DO WHILE lcgopm#0
  CUADRO(lcgtinte, 5, 2, 7, 77)
  TITULO:={ "CONSULTA " , " MOVIMIENTOS " , " PROPUESTAS " , " EVALUACION " , ;
           " IMPRESION " , " SALIR " }
  TEXTO :={"CONSULTA AL ARCHIVO DE PROFESORES", ;
          "ACTUALIZACION AL ARCHIVO DE PROFESORES", ;
          "GENERACION DE PROPUESTAS Y TARJETAS", ;
          "EVALUACION A PROFESORES", ;

```

```

                "IMPRESION DE PROPUESTAS, TARJETAS Y/O EVALUACION",;
                "SALIDA DEL SISTEMA"}
COLUMNNA:={5,16,30,43,56,68}
SET KEY 19 TO
SET KEY 4 TO
MENUH(6,TITULO,TEXTO,COLUMNNA)
MENU TO lcgopm
lcpunam=SAVESCREEN(0,0,24,79)
DO CASE
CASE lcgopm=1
DO PROF CON //CONSULTAS
CASE lcgopm=2
DO PROF MOV //MOVIMIENTOS
CASE lcgopm=3
DO PROF_PRO //PROPUESTAS
CASE lcgopm=4
DO PROF EVA //EVALUACIONES
CASE lcgopm=5
DO PROF_IMP // IMPRESIONES
CASE lcgopm=6.OR.LASTKEY()=27 //SALIDA
EXIT
CASE LASTKEY() = 27
EXIT
ENDCASE
ENDDO
CLOSE DATAB
RELEASE
lcpaleta,lcgtinte,lcgtinte0,lcgtinte1,lcgtinte2,lcgtinte3,lcgtinte4,;
lcgtinte5,lcgtinte6,lcgenpana,lcgerror,lcginvisible,lcgvideo,lcg
unam,; lcgpanold,lcgruta,lcgeva,lcgmat,suayuda,lcgtit1,lcgtit2,lcgtit_s
em,; lcgpan,lcglin,lcgunamp,lcgopm,titulo,texto,columnna,lcgpunam,lcgz
eus,; lcgtinte7,lcgprocoa,lcgproaut,lcgproman,lcgprorev
RETURN

```

```

*****
* PROGRAM-ID:      Prof_Con.prg ver 1.0 llamado por Prof_Lcg.prg
* AUTHOR:         Leticia Cerd Garrido
* REMARKS:        Programa que Realiza la Consulta al catalogo de
*                 Profesores
* MODIFIED:       08 Junio, 1996.
*****
#include "lnkey.ch"
MEMVAR getlist,lcgtinte,lcgopco,titulo,texto,lcgpccon,lcgpunan,lcgruta,;
lcgzeus
USE &lcgzeus\profcat ALIAS pro NEW SHARED
SET INDEX TO &lcgzeus\profrfc.&lcgzeus\pronom
USE &lcgruta&lcgprocoo ALIAS coor NEW SHARED
SET INDEX TO &lcgruta\coorrfc.&lcgruta\coornom
USE &lcgruta\cat INDEX &lcgruta\cat ALIAS cat NEW SHARED

CUADRO (lcgtinte,8,2,12,15,"S") ;lcgopco=1
DO WHILE lcgopco#0
  TITULO:={" GENERICO ","COORDINACION"," SALIR "}
  TEXTO :={"CONSULTA DE DATOS PERSONALES","CONSULTA DE DATOS
  GENERALES"," "SALIR AL MENU ANTERIOR"}
  MENUV(8,1,TITULO,TEXTO)
  SET KEY 19 TO ; SET KEY 4 TO
  MENU TO lcgopco
  lcgpccon=SAVESCREEN(0,0,24,79)
  DO CASE
    CASE lcgopco=1
      PROFCGEN()
    CASE lcgopco=2
      PROFCCOO()
    CASE lcgopco=3.OR.LASTKEY()=27
      EXIT
  ENDCASE
  RESTSCREEN(0,0,24,79,lcgpccon)
ENDDO
RESTSCREEN(0,0,24,79,lcgpunan)
CLOSE pro
CLOSE coor
CLOSE cat
RELEASE
lcgopco,titulo,texto,lcgopcl,lcgpccon1,lcgcon,lcgpccon,lcgrfc,lcgrev,;
lcgopc2,lcgpccon2,lcgvtota,seleold,lcgvpb2,lcgvb2,lcgvben2,lcgv
b1,;
lcgvb1,lcgvben1,lcgvpcper,lcgvfunc,lcgvttitu,lcgvext,lcgvtrab,;
lcgvtcasa,lcgvdomi,lcgvnomb,lcgvhdep,lcgvvdep,lcgvdepe,lcgvhjef,;
;
lcgvcejef,lcgvjefd,lcgvprac,lcgvteor,lcgvflim,lcgvfina,lcgvcate,;
lcgvacti,lcgvanti,lcgvnomb,lcgvjefa,lcgvbanh,lcgvvota

RETURN

/*****
* PROFCGEN()
* Función que realiza la consulta de datos personales
*****/
FUNCTION PROFCGEN()
MEMVAR getlist,lcgtinte,lcgopcl,titulo,texto,lcgpccon1,lcgcon,lcgpccon,;
lcgrfc,lcgcampo,lcgrev,lcgpunan
CUADRO (lcgtinte,10,11,14,24,"S") ; lcgopcl=1
DO WHILE lcgopcl#0
  TITULO:={"PROFESORES"," R.F.C. "," SALIR "}
  TEXTO :={"CONSULTA ORDENADA POR PROFESOR","
  "CONSULTA ORDENADA POR R.F.C. ","SALIR AL MENU ANTERIOR"}
  MENUV(10,11,TITULO,TEXTO)

```

```

SET KEY 19 TO ; SET KEY 4 TO
MENU TO lcgopc1
lcgpcn1=SAVESCREEN(0,0,24,79)
SELE pro
DO CASE
  CASE lcgopc1=1
    SET ORDER TO 2
  CASE lcgopc1=2
    SET ORDER TO 1
  CASE lcgopc1=3.OR.LASTKEY()=27
    EXIT
ENDCASE
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpcn1)
CUADRO(lcgtinte,8,20,10,59,"S")
SET CURSOR ON
DECLARE LCGCAMPO[2]
LCGCAMPO[1] := "RFC"
LCGCAMPO[2] := "NOM PRO"
SET KEY K F2 TO COD_AYUDA
lcgrfc := SPACE(13)
@9,22 SAY "RFC DEL PROFESOR <F2> :" GET lcgrfc PICT "@!:"
READ
IF (LASTKEY() = K_F2 .OR. lcgrfc = SPACE(13)).AND. LASTKEY() <> 27
  lcgrfc := .F.
  lcgrfc := BROWPRO("PRO",LCGCAMPO)
  @9,46 SAY lcgrfc PICT "@!:"
ENDIF
RELEASE LCGCAMPO
SET KEY K F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgrfc) <> 0)
  SELE pro
  SET ORDER TO 1
  SEEK lcgrfc
  IF FOUND()
    SET CURSOR OFF
    CONPROF()
    SET CURSOR ON
  ELSE
    ALERT("EL PROFESOR NO EXISTE EN EL CATALOGO")
  ENDIF
ENDIF
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpcn1)
ENDDO
RESTSCREEN(0,0,24,79,lcgpcn1)
RETURN nil

/*****
* PROFCCOO()
*
* Funcion que realiza la consulta de datos generales
*****/
FUNCTION PROFCCOO()
MEMVAR getlist,lcgpcn,lcgtinte,lcgopc2,titulo,texto,lcgpcn2,lcgcampo,;
lcgrev,lcgrfc,lcgpcn1
CUADRO(lcgtinte,11,11,15,24,"S") ; lcgopc2=1
DO WHILE lcgopc2#0
  TITULO:={"PROFESORES"," R.F.C. "," SALIR "}
  TEXTO:={"CONSULTA ORDENADA POR PROFESOR";
         "CONSULTA ORDENADA POR R.F.C.", "SALIR AL MENU ANTERIOR"}
  MENUV(11,11,TITULO,TEXTO)
  SET KEY 19 TO ; SET KEY 4 TO
  MENU TO lcgopc2

```

```

lcgpcon2=SAVESCREEN(0,0,24,79)
SELE coor
DO CASE
  CASE lcgpoc2=1
    SET ORDER TO 2
  CASE lcgpoc2=2
    SET ORDER TO 1
  CASE lcgpoc2=3.OR.LASTKEY()=27
    EXIT
ENDCASE
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpunam)
CUADRO(lcgtinte,8,20,10,59,"S")
SET CURSOR ON
DECLARE LCGCAMPO[2]
LCGCAMPO[1] := "RFC"
LCGCAMPO[2] := "NOM_PRO"
SET KEY K_F2 TO COD_AYUDA
lcgrfc := SPACE(13)
@9,22 SAY "RFC DEL PROFESOR <F2> :" GET lcgrfc PICT "@!"
READ
IF (LASTKEY() = K_F2 .OR. lcgrfc = SPACE(13)).AND. LASTKEY() <> 27
  lcgreve := .F.
  lcgrfc := BROWPRO("COOR",LCGCAMPO)
  @9,46 SAY lcgrfc PICT "@!"
ENDIF
RELEASE LCGCAMPO
SET KEY K_F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgrfc) <> 0)
  SELE coor
  SET ORDER TO 1
  SEEK lcgrfc
  IF FOUND()
    SET CURSOR OFF
    CONCOOR()
    SET CURSOR ON
  ELSE
    ALERT("EL PROFESOR NO EXISTE EN EL CATALOGO")
  ENDIF
ENDIF
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpcon2)
ENDDO
RESTSCREEN(0,0,24,79,lcgpcon)
RETURN nil

/*****
* CONPROP()
*
* Función que realiza la consulta de datos personales del profesor
*****/
FUNCTION CONPROP()
  MENVAR getlist,lcgrfc,lcgpunam,lcgtinte
  LOCAL
  lcgvpb2,lcgvb2,lcgvben2,lcgvpb1,lcgvb1,lcgvben1,lcgvcp,lcgvfunc,;
  lcgvtitu,lcgvext,lcgvtrab,lcgvtcasa,lcgvdomi,lcgvnomb
  CUADRO(lcgtinte,8,9,20,75,"S")
  @ 9,22 SAY "RFC DEL PROFESOR <F2> :" GET lcgrfc PICT "@!"
  @11, 7 SAY "Nombre: "
  @12, 7 SAY "Domicilio: "
  @13, 7 SAY "Tel.fono de Casa: "
  @14, 7 SAY "Tel.fono de Oficina: "
  @14,44 SAY "Ext: "
  @15, 7 SAY "Funcionario: "
  @15,29 SAY "Titulado: "

```

```

@15,49 SAY "Clave Personal: "
@16, 7 SAY "Primer Beneficiario: "
@17, 7 SAY "Sexo: "
@17,34 SAY "Parentesco: "
@18, 7 SAY "Segundo Beneficiario: "
@19, 7 SAY "Sexo: "
@19,34 SAY "Parentesco: "

lcvgnomb := pro->nomb_pro
lcvgdomi := pro->domicillio
lcvgtcasa:= pro->tel_casa
lcvgttrab:= pro->tel_trab
lcvgext := pro->ext
lcvgtitu := pro->titulado
lcvgfunc := pro->funsio
IF lcvgfunc = '*'
  lcvgfunc := 'SI'
ELSE
  lcvgfunc := 'NO'
ENDIF
lcvgcper := pro->clave_per
lcvgben1 := pro->beneficial
lcvgsb1 := pro->sexb1
IF lcvgsb1 = 'f'.OR.lcvgsb1 = 'F'
  lcvgsb1 := 'FEMENINO'
ELSEIF lcvgsb1 = 'm'.OR.lcvgsb1 = 'M'
  lcvgsb1 := 'MASCULINO'
ENDIF
lcvvpb1 := pro->parb1
lcvben2 := pro->beneficia2
lcvgsb2 := pro->sexb2
IF lcvgsb2 = 'f'.OR.lcvgsb2 = 'F'
  lcvgsb2 := 'FEMENINO'
ELSEIF lcvgsb2 = 'm'.OR.lcvgsb2 = 'M'
  lcvgsb2 := 'MASCULINO'
ENDIF
lcvvpb2 := pro->parb2

MENSAJEE (" < ESC > PARA CONTINUAR " ,22,20)
@11,16 SAY lcvgnomb PICT "@!"
@12,18 SAY lcvgdomi PICT "@!"
@13,26 SAY lcvgtcasa PICT "@!"
@14,29 SAY lcvgttrab PICT "@!"
@14,50 SAY lcvgext PICT "@!"
@15,21 SAY lcvgfunc PICT "@!"
@15,40 SAY lcvgtitu PICT "@!"
@15,66 SAY lcvgcper PICT "@!"
@16,29 SAY lcvgben1 PICT "@!"
@17,14 SAY lcvgsb1 PICT "@!"
@17,48 SAY lcvvpb1 PICT "@!"
@18,29 SAY lcvben2 PICT "@!"
@19,14 SAY lcvgsb2 PICT "@!"
@19,48 SAY lcvvpb2 PICT "@!"

DO WHILE !INKEY(0) = K_ESC
ENDDO
RETURN nil

/*****
* CONCOOR()
* Funcion que realiza la consulta de datos generales del profesor
*****/
FUNCTION CONCOOR()

```

MEMVAR getlist,vm_tit_carr,lcgrfc,seleold,lcgtinte,banentry
LOCAL lcgvhdep,lcgvcddep,lcgvdepe,lcgvhjeff,lcgvjeff,lcgvjefd,lcgvprac,
lcgvteor,lcgvflim,lcgvfini,lcgvcate,lcgvacti,lcgvanti,lcgvnomb,
lcgvjefa,lcgvbanh,lcgvkota

CUADRO(lcgtinte,8,2,22,77)

@ 9,22 SAY "RFC DEL PROFESOR <F2> : " GET lcgrfc PICT "@!"
@10, 3 SAY "Nombre: "
@11, 3 SAY "Antigüedad: "
@11,35 SAY "Activo: "
@11,55 SAY "Horas Aut.: "
@13,22 SAY "JEFATURA DE "
@13,34 SAY vm_tit_carr
@14, 3 SAY "Categoría: "
@15, 3 SAY "Fecha Inicio: "
@15,40 SAY "Fecha Límite: "
@16, 3 SAY "Técnicas: "
@16,25 SAY "Pr cticas: "
@16,50 SAY "Total: "
@18,11 SAY "OTRA JEFATURA"
@19, 3 SAY "Area: "
@20, 3 SAY "Categoría: "
@21, 3 SAY "Número de Horas: "
@18,50 SAY "OTRA DEPENDENCIA"
@19,42 SAY "Dependencia: "
@20,42 SAY "Categoría: "
@21,42 SAY "Número de Horas: "

lcgvnomb := coor->nom_pro
lcgvcate := coor->cat_goria
lcgvanti := coor->antigüedad
lcgvacti := coor->activo
lcgvfini := coor->fe_inicio
lcgvflim := coor->fe_limite
* OTRAS JEFATURAS
BANENTRY := BAPRROR(lcgrfc)
SELE coor
IF banentry=NIL .AND. LEN(banentry)>0
 lcgvbanh := BANENTRY[1]
 lcgvjefa := BANENTRY[3]
 lcgvjef := BANENTRY[4]
 lcgvhjeff := BANENTRY[2]
 lcgvteor := BANENTRY[7]
 lcgvprac := BANENTRY[8]
ELSE
 lcgvbanh := 0
 lcgvjefa := SPACE(10)
 lcgvjef := SPACE(3)
 lcgvhjeff := 0
 lcgvteor := 0
 lcgvprac := 0
ENDIF
lcgvkota := lcgvteor + lcgvprac
RELEASE BANENTRY
* OTRAS DEPENDENCIAS
lcgvdepe := coor->dep_o_dep
lcgvcddep := coor->cat_o_dep
lcgvhdep := coor->hor_o_dep

MENSAJEE (" < ESC > PARA CONTINUAR " ,22,20)
@10,12 SAY lcgvnomb PICT "@!"
@11,17 SAY lcgvanti PICT "@d"
@11,46 SAY lcgvacti PICT "@!"

```

@11,67 SAY lcgbanh PICT "###.#"
* CATEGORIA EN LA COORDINACION
IF lcgvcate#space(3)
  seleold := SELE()
  SELE cat
  SEEK lcgvcate
  IF FOUND()
    @14,16 SAY cat->categoria
  ENDIF
  SELE(seleold)
ENDIF
@15,18 SAY lcgvfini PICT "@D"
@15,55 SAY lcgvflim PICT "@D"
@16,17 SAY lcgvteor PICT "###.#"
@16,40 SAY lcgvprac PICT "###.#"
lcvvtota = lcvvteor + lcvvprac
@16,61 SAY lcvvtota PICT "###.#"
* OTRAS JEFATURAS
@19,9 SAY lcvvjefa PICT "@!:"
IF (lcvvjefa <> SPACE(10))
  IF lcvvjef#SPACE(3)
    seleold := SELE()
    SELE cat
    SEEK lcvvjef
    IF FOUND()
      @20,13 SAY cat->categoria
    ENDIF
  SELE(seleold)
ENDIF
@21,21 SAY lcvvhjef PICT "###.#"
ENDIF
* OTRAS DEPENDENCIAS
@19,55 SAY lcvvdpe PICT "@!:"
IF (lcvvdpe <> SPACE(10))
  IF lcvvdpe#SPACE(3)
    seleold := SELE()
    SELE cat
    SEEK lcvvdpe
    IF FOUND()
      @20,53 SAY cat->categoria
    ENDIF
  SELE(seleold)
ENDIF
@21,59 SAY lcvvhdep PICT "###.#"
ENDIF
DO WHILE !INKEY(0) = K_ESC
ENDDO
RETURN nil
// EOF()

```

```

*****
* PROGRAM-ID:      Prof_Mov.prg ver 1.0      llamado por Prof_Lcgr.prg
* AUTHOR:         Leticia Cerd Garrido
* REMARKS:        Programa que Realiza los Movimientos al Catalogo de
*                 Profesores
*                 08 Junio, 1996.
*****
#include "inkey.ch"
MEMVAR
getclst,lcgenpana,lcgtinte,lcgcampo,lcgalt,lcgrfc,lcgvhdep,lcgvcddep,;
lcgvdepe,lcgvhjef,lcgvcjef,lcgvjefd,lcgvprac,lcgvteor,lcgvflim,;
lcgvfini,lcgvcate,lcgvpb2,lcgvsb2,lcgvben2,lcgvpb1,lcgvsb1,lcgvbe
n1,;
lcgvcpet,lcgvfunc,lcgvacti,lcgvtitu,lcgvanti,lcgvext,lcgvtrab,;
lcgvcasa,lcgvdomi,lcgvnomb,lcgvnam,lcgvjeft,lcgvrev,lcgvbanh,lcgr
uta,;
lcgzeus,lcgcont

lcgcont := .T.
DO WHILE (lcgcont).OR.LASTKEY() <> 27

  USE &lcgzeus\profcat INDEX &lcgzeus\profic ALIAS pro NEW SHARED
  USE &lcgruta\lcgprocoo INDEX &lcgruta\coorrfc ALIAS coor NEW SHARED
  USE &lcgruta\cat INDEX &lcgruta\cat ALIAS cat NEW SHARED

  SELE coor
  GO TOP

  SET CURSOR ON
  SET KEY K_F3 TO
  SET KEY K_CTRL_DEL TO
  SETCOLOR(lcgenpana)
  CLRSCR (5, 130)
  CUADRO(lcgtinte,0,0,24,79,"S")
  PROSCREE()
  DECLARE LCGCAMPO[3]
  LCGCAMPO[1] := "RFC"
  LCGCAMPO[2] := "NOM_PRO"
  LCGCAMPO[3] := "FLAG"
  SET KEY K_F2 TO COD_AYUDA

  lcgalt := 0
  lcgrfc := SPACE(13)
  @1,10 SAY "RFC DEL PROFESOR <F2> : " GET lcgrfc PICT "@!"
  READ
  IF (LASTKEY() = K_F2 .OR. lcgrfc = SPACE(13)).AND. LASTKEY() <> 27
    lcgvrev := .F.
    lcgrfc := BROWPRO("COOR",LCGCAMPO)
    @1,35 SAY lcgrfc PICT "@!"
  ENDIF
  RELEASE LCGCAMPO
  SET KEY K_F2 TO
  IF (LASTKEY() <> 27 .AND. LEN(lcgrfc) <> 0)
    SELE coor
    SEEK lcgrfc
    IF !FOUND()
      PUBLIC lcgvnomb := SPACE(32),lcgvdomi := SPACE(50),;
        lcgvtcasa:= SPACE(9),lcgvtrab := SPACE(9),;
        lcgvext := SPACE(5),lcgvanti := DATE(),;
        lcgvtitu := SPACE(2),lcgvacti := SPACE(2),;
        lcgvfunc := SPACE(1),lcgvcpet := SPACE(1),;
        lcgvben1 := SPACE(30),lcgvsb1 := SPACE(1),;
        lcgvpb1 := SPACE(10),lcgvben2 := SPACE(30),;
        lcgvsb2 := SPACE(1),lcgvpb2 := SPACE(10),;
        lcgvcate := SPACE(3),lcgvfini := DATE(),lcgvbanh:=0,;

```

```

0.;          lcgvflim := DATE() ,lcgvteor := 0,   lcgvprac :=
0.;          lcgvjefa := SPACE(10),lcgvcejef := SPACE(3),lcgvhjef:=
          lcgvdepe := SPACE(10),lcgvcddep := SPACE(3),lcgvhdep:= 0
          lcgalt=1
          PROF_GRA()
          ELSE
          PROF_CAM()
          ENDF
          ELSE
          lcgcont = .F.
          ENDF
          COMMIT
          CLOSE pro
          CLOSE coor
          CLOSE cat
ENDDO
SET KEY K_F3 TO
SET KEY K_CTRL_DEL TO
RESTSCREEN(0,0,24,79,lcgpunam)
RELEASE
lcgalt,lcgrfc,lcgvhdep,lcgvcddep,lcgvdepe,lcgvhjef,lcgvcejef,lcgvjefd,;
vben2,;   lcgvpbac,lcgvteor,lcgvflim,lcgvfini,lcgvceate,lcgvppb2,lcgvsb2,lcg
vant1,;   lcgvpbi,lcgvb1,lcgvben1,lcgvcpbr,lcgvfunc,lcgvacti,lcgvtitu,lcg
vgvext,lcgvtrab,lcgvtcasa,lcgvdomi,lcgvnomb,lcgvjefa,lcgrev,lc
gbanh,;   lcgcont,lcgvttota,seleold,lcgcatvar,lcgptr,lcgrest,lcgresp,lcgcat
P.;
lcgdate,seguirb,lcgpba,seguir1,seguir,respaja
RETURN

/*****
* PROSCREE()
*
* Funcion que despliega la mascara de captura
*****/
FUNCTION PROSCREE()
MEMVAR getlist,vm_tit_carr
@ 3, 3 SAY "Nombre: "
@ 4, 3 SAY "Domicilio: "
@ 5, 3 SAY "Tel.fono de Casa: "
@ 6, 3 SAY "Tel.fono de Oficina: "
@ 6,40 SAY "Ext: "
@ 7, 3 SAY "Antig edad: "
@ 7,35 SAY "Titulado: "
@ 7,60 SAY "Activo: "
@ 8, 3 SAY "Funcionario: "
@ 8,35 SAY "Clave Personal: "
@ 9, 3 SAY "Primer Beneficiario: "
@10, 3 SAY "Sexo: "
@10,45 SAY "Parentesco: "
@11, 3 SAY "Segundo Beneficiario: "
@12, 3 SAY "Sexo: "
@12,45 SAY "Parentesco: "
@14,20 SAY "JEFATURA DE "
@14,32 SAY vm_tit_carr
@15, 3 SAY "Categoria: "
@16, 3 SAY "Fecha Inicio: "
@16,35 SAY "Fecha Limite: "
@17, 3 SAY "Tecnicas: "
@17,35 SAY "Pr cticas: "
@17,60 SAY "Total: "

```

```

@19,15 SAY "OTRA JEFATURA"
@20, 3 SAY "Area: "
@21, 3 SAY "Categoria: "
@22, 3 SAY "Número de Horas: "
@19,46 SAY "OTRA DEPENDENCIA"
@20,44 SAY "Dependencia: "
@21,44 SAY "Categoria: "
@22,44 SAY "Número de Horas: "
RETURN nil

```

```

/*****
* PROF_CAM()

```

```

* Función que realiza los cambios al catalogo de profesores
*****/

```

```

FUNCTION PROF_CAM()

```

```

MEMVAR getllist,lcgrfc,lcgvnomb,lcgvdomi,lcgvtcasa,lcgvtrab,lcgvext,;
lcgvitu,lcgvfunc,lcgvpcper,lcgvben1,lcgvsb1,lcgvpb1,lcgvben2,;
lcgvsb2,lcgvpb2,lcgvcate,lcgvanti,lcgvacti,lcgvfini,lcgvflim,;
lcgvteor,lcgvprac,lcgvtotal,lcgvjefa,lcgvcjef,lcgvhjef,lcgvdepe,;
lcgvdepe,lcgvhdep,seleold,lcgvbanh,banentry

```

```

LOCAL seguir1

```

```

/*****
* SE CARGA LA INFORMACION DE LA BASE DE DATOS A LAS VARIABLES
*****/

```

```

SELE pro
SEEK lcgrfc
IF !FOUND()
  APPEND BLANK
  REPLA pro->rfc WITH lcgrfc
ENDIF

```

```

lcgvnomb := pro->nom pro
lcgvdomi := pro->domicilio
lcgvtcasa:= pro->tel_casa
lcgvtrab := pro->tel_trab
lcgvext := pro->ext
lcgvitu := pro->titulado
lcgvfunc := pro->funσιο
IF lcgvfunc = ''
  lcgvfunc := 'S'
ELSE
  lcgvfunc := 'N'
ENDIF

```

```

lcgvpcper := pro->clave_per
lcgvben1 := pro->beneficial
lcgvsb1 := pro->sexb1
lcgvpb1 := pro->parb1
lcgvben2 := pro->beneficia2
lcgvsb2 := pro->sexb2
lcgvpb2 := pro->parb2
SELE coor

```

```

lcgvcate := coor->categoria
lcgvanti := coor->antiguedad
lcgvacti := coor->activo
lcgvfini := coor->fe_inicio
lcgvflim := coor->fe_limite
* OTRAS JEFATURAS

```

```

BANENTRY := BAPRHO(lcgrfc)
SELE coor

```

```

IF banentry!=NIL .AND. LEN(banentry)>0
  lcgvbanh := BANENTRY[1]
  lcgvjefa := BANENTRY[3]
  lcgvcjef := BANENTRY[4]

```

```

lcvghjef := BANENTRY[2]
lcvgteor := BANENTRY[7]
lcvvprac := BANENTRY[8]
ELSE
  lcvbanh := 0
  lcvvjefa := SPACE(10)
  lcvvcjef := SPACE(3)
  lcvghjef := 0
  lcvgteor := 0
  lcvvprac := 0
ENDIF
RELEASE BANENTRY
lcvvtota := lcvgteor + lcvvprac
* OTRAS DEPENDENCIAS
lcvvdepe := coor->dep_o_dep
lcvvcdep := coor->cat_o_dep
lcvvhdep := coor->hor_o_dep
seguiri = '.T.'

* MIENTRAS NO ESTEN CORRECTOS LOS DATOS
DO WHILE (seguir1)
  @ 3,12 SAY lcvgnomb PICT "@!"
  @ 4,15 SAY lcvgdomi PICT "@!"
  @ 5,22 SAY lcvgtcasa PICT "@!"
  @ 6,25 SAY lcvgttrab PICT "@!"
  @ 6,46 SAY lcvgtxt PICT "@!"
  @ 7,16 SAY lcvgtanti PICT "@D"
  @ 7,46 SAY lcvgtitu PICT "@!"
  @ 7,69 SAY lcvgtacti PICT "@!"
  @ 8,17 SAY lcvgtfunc PICT "@!"
  @ 8,52 SAY lcvgtper PICT "@!"
  @ 9,25 SAY lcvgtben1 PICT "@!"
  @10,13 SAY lcvgtvb1 PICT "@!"
  @10,59 SAY lcvgtvp1 PICT "@!"
  @11,25 SAY lcvgtben2 PICT "@!"
  @12,13 SAY lcvgtvb2 PICT "@!"
  @12,59 SAY lcvgtvp2 PICT "@!"
  * CATEGORIA EN LA COORDINACION
  IF lcvgtcate#space(3)
    seleold := SELE()
    SELE cat
    SEEK lcvgtcate
    IF FOUND()
      @15,15 SAY cat->categoria
    ENDIF
    SELE(seleold)
  ENDIF
  @16,18 SAY lcvgtfini PICT "@D"
  @16,50 SAY lcvgtfilm PICT "@D"
  @17,14 SAY lcvgteor PICT "###.#"
  @17,47 SAY lcvvprac PICT "###.#"
  lcvvtota = lcvgteor + lcvvprac
  @17,68 SAY lcvvtota PICT "###.#"
  * OTRAS JEFATURAS
  @20,10 SAY lcvvjefa PICT "@!"
  IF (lcvvjefa <> SPACE(10))
    IF lcvvcjef#SPACE(3)
      seleold := SELE()
      SELE cat
      SEEK lcvvcjef
      IF FOUND()
        @21,15 SAY cat->categoria
      ENDIF
      SELE(seleold)
    ENDIF
  ENDIF

```

```

    @22,21 SAY lcgvhjef PICT "##.#"
ENDIF
* OTRAS DEPENDENCIAS
@20,58 SAY lcgvdepe PICT "@!"
IF (lcgvdepe <> SPACE(10))
  IF lcgvcdep#SPACE(3)
    seleold := SELE()
    SELE cat
    SEEK lcgvcdep
    IF POUND()
      @21,55 SAY cat->categoria
    ENDIF
    SELE(seleold)
  ENDIF
  @22,62 SAY lcgvhdep PICT "##.#"
ENDIF
PROF_GRA()
seguir1 := .F.
IF LASTKEY() = K_ESC
  seguir1 = .F.
ENDIF
ENDDO
RETURN nil

/*****
* PROF_GRA()
*
* Función que graba los cambios al catalogo de profesores
*****/
FUNCTION PROF_GRA()
#define T_BORRAR 127
MEMVAR
getlist, lcgcatvar, lcgptr, lcgRFC, lcgvtitu, lcgvacti, lcgvfunc, lcgvsb1, ;
lcgsb2, lcgcatp, lcgvcate, seleold, lcgvteor, lcgvprac, lcgvtota, lcgba
nh, ;
lcgvjefa, lcgalt, lcgvcjef, lcgvhjef, lcgvdepe, lcgvcdep, lcgrest, ;
lcgsanta, lcgtime, lcgvnomb, lcgvdomi, lcgvtcasa, lcgvttrab, lcgvext, ;
lcgvCper, lcgvben1, lcgvpb1, lcgben2, lcgvpb2, lcgfunc, lcgvanti, lcgvfi
ni, ;
lcgvflim, lcgvhdep, lcgtime6, lcgresp, lcgtime2, lcgvben2, lcgcat1, lc
gcat2
LOCAL seguir
SET KEY K_F3 TO PROF_BAJ()
SET KEY K_CTRL DEL TO PROF_BAJ()
lcgcatvar=SPACE(3)
seguir = .T.
lcgptr := 1
DO WHILE (seguir)
  MENSAAJE ("<F2> AYUDA EN CATEGORIAS <F3> ◊ <CTR-DEL>
BORRAR", 23,10)
  @ 1,35 SAY lcgRFC PICT "@!"
  @ 2,53 SAY "HORAS AUTORIZADAS: "
  @ 2,73 SAY lcgbanh PICT "##.#"
  @ 3,12 GET lcgvnomb PICT "@!"
  @ 4,15 GET lcgvdomi PICT "@!"
  @ 5,22 GET lcgvtcasa PICT "@!"
  @ 6,25 GET lcgvttrab PICT "@!"
  @ 6,46 GET lcgvext PICT "@!"
  @ 7,16 GET lcgvanti PICT "@D"
  @ 7,46 GET lcgvtitu PICT "@!" VALID lcgvtitu $ "SI,NO, "
  @ 7,69 GET lcgvacti PICT "@!" VALID lcgvacti $ "SI,NO, "
  @ 8,17 GET lcgvfunc PICT "@!" VALID lcgvfunc $ "S,N, "
  @ 8,52 GET lcgvcper PICT "@!" VALID lcgvcper $ "A,B, "
  @ 9,25 GET lcgvben1 PICT "@!"
  @10,13 GET lcgvsb1 PICT "@!" VALID lcgvsb1 $ "M,F, "

```

```

@10,59 GET lcgvpb1      PICT "@1"
@11,25 GET lcgvben2    PICT "@1"
@12,13 GET lcgvsb2    PICT "@1" VALID lcgvsb2 $ "M,F, "
@13,59 GET lcgvpb2    PICT "@1"
READ
IF (LASTKEY() <= T BORRAR .AND. LASTKEY() <= K_ESC)
  DECLARE LCGCATP[2]
  LCGCATP[1] := "CVE CAT"
  LCGCATP[2] := "CATEGORIA"
  SET KEY K_F2 TO CAT_AYUDA
  lcgcatvar = lcgvcate
  @15,15 CLEAR TO 15,38
  @15,15 GET lcgcatvar PICT "@1" ;
  VALID lcgcatvar $ "ASA,ASB,AYA,AYB,CAA,CAB,CAC,TA , PH , "
  READ
  AYUDA_CAT()
  lcgvcate = lcgcatvar
  seleold := SELE()
  SELE cat
  SEEK lcgvcate
  IF FOUND()
    @15,15 SAY cat->categoria
  ENDIF
  SELE(seleold)
  @16,18 GET lcgvfini PICT "@D"
  @16,50 GET lcgvfilim PICT "@D"
  READ
  @17,14 SAY lcgvteor PICT "###.#"
  @17,47 SAY lcgvprac PICT "###.#"
  lcgvtota = lcgvteor + lcgvprac
  @17,68 SAY lcgvtota PICT "###.#"
  * OTRAS JEFATURAS
  @20,10 SAY lcgvjefa PICT "@1"
  IF lcgalt=0
    seleold := SELE()
    SELE cat
    SEEK lcgvjef
    IF FOUND()
      @21,15 SAY cat->categoria
    ENDIF
    SELE(seleold)
  ENDIF
  @22,21 SAY lcgvhjef PICT "###.#"
  * OTRAS DEPENDENCIAS
  @20,58 GET lcgvdepe PICT "@1"
  READ
  IF (lcvdepe <> SPACE(10))
    lcgcatvar = lcgvcdep
    @21,56 CLEAR TO 21,78
    @21,56 GET lcgcatvar PICT "@1" ;
    VALID lcgcatvar $ "ASA,ASB,AYA,AYB,CAA,CAB,CAC,TA , PH , "
    READ
    AYUDA_CAT()
    RELEASE LCGCATP
    SET KEY K_F2 TO
    lcgvcdep = lcgcatvar
    seleold := SELE()
    SELE cat
    SEEK lcgvcdep
    IF FOUND()
      @21,55 SAY cat->categoria
    ENDIF
    SELE(seleold)
    @22,62 GET lcvhdep PICT "###.#"
    READ

```

```

ENDIF
lcgresp = 1
lcgpanta = SAVESCREEN(0,0,24,79)
CUADRO(lcgtinte2,10,30,13,47,"S")
@11,33 SAY " CORRECTO ?"
@12,32 PROMPT "SI"
@12,35 PROMPT "NO"
@12,38 PROMPT "CANCELAR"
MENU TO lcgresp
RESTSCREEN(0,0,24,79,lcgpanta)
SETCOLOR(lcgtinte)
DO CASE
CASE lcgresp = 1
  SELE pro
  SEEK lcgrefc
  IF !FOUND()
    APPEND BLANK
  ENDIF
  IF RLOCK()
    REPLACE pro->rffc WITH lcgrefc
    REPLACE pro->nom_pro WITH lcgvnomb
    REPLACE pro->domicilio WITH lcgvdomi
    REPLACE pro->stel_casa WITH lcgvtcasa
    REPLACE pro->stel_trab WITH lcgvttrab
    REPLACE pro->ext WITH lcgvext
    REPLACE pro->titulado WITH lcgvtitu
    REPLACE pro->clave_per WITH lcgvppep
    REPLACE pro->beneficial WITH lcgvben1
    REPLACE pro->sexb1 WITH lcgvsb1
    REPLACE pro->parb1 WITH lcgvpb1
    REPLACE pro->beneficia2 WITH lcgvben2
    REPLACE pro->sexb2 WITH lcgvsb2
    REPLACE pro->parb2 WITH lcgvpb2
    REPLACE pro->flag WITH lcgvflag
    IF lcgvfunc='S'
      WITH SPACE(1)
    ELSE
      lcvfunc='*'
    ELSE
      lcvfunc='SPACE(1)'
    ENDIF
  ENDIF
  REPLACE pro->funσιο WITH lcgvfunc
ENDIF
SELE coor
SEEK lcgrefc
IF !FOUND()
  APPEND BLANK
ENDIF
IF RLOCK()
  REPLACE coor->rffc WITH lcgrefc
  REPLACE coor->categoria WITH lcgvcate
  REPLACE coor->nom_pro WITH lcgvnomb
  REPLACE coor->antiguedad WITH lcgvanti
  REPLACE coor->activo WITH lcgvacti
  REPLACE coor->fe_inicio WITH lcgvfini
  REPLACE coor->fe_limite WITH lcgvflim
  REPLACE coor->hor_teo WITH lcgvteor
  REPLACE coor->hor_pra WITH lcgvprac
  REPLACE coor->area_o_jef WITH lcgvjefa
  REPLACE coor->flag WITH SPACE(1)
  IF coor->area_o_jef<>SPACE(10)
    REPLACE coor->cat_o_jef WITH lcgvcjef
    REPLACE coor->hor_o_jef WITH lcgvhjef
  ELSE
    REPLACE coor->cat_o_jef WITH SPACE(3) , coor->hor_o_jef
  ENDIF
ENDIF
WITH 0

```

```

REPLACE coor->dep_o_dep WITH lcgvdpep
IF coor->dep_o_dep<>SPACE(10)
  REPLACE coor->cat_o_dep WITH lcgvdpep
  REPLACE coor->hor_o_dep WITH lcgvhdep
ELSE
  REPLACE coor->cat_o_dep WITH SPACE(3) , coor->hor_o_dep
WITH 0
  ENDIF
ENDIF
lcgcat1 := SUBSTR(lcgvcate,1,2)
lcgcat2 := SUBSTR(lcgvcate,3,1)
BAALTPR2(lcgrfc,lcgvnomb,lcgvfunc,lcgcat1,lcgcat2)
seguir = .F.
CASE lcgresp = 2
  seguir = .T.
CASE lcgresp = 3 .OR. LASTKEY() = 27
  seguir = .F.
ALERT(" LOS DATOS HAN SIDO CANCELADOS ")
KEYBOARD (27)
ENDCASE
ELSE
  seguir = .F.
ENDIF
ENDDO
RETURN nil

/*****
* CAT_AYUDA()
*
* Función que activa el ENTER
*****/
FUNCTION CAT_AYUDA
MEMVAR getlist.lcgcatvar
lcgcatvar=SPACE(3)
KEYBOARD CHR(K_ENTER)
RETURN nil

/*****
* AYUDA_CAT()
*
* Función que muestra un browse de las categorías
*****/
FUNCTION AYUDA_CAT
MEMVAR getlist.seleold,lcgcatvar,lcgcatp,lcgrev
seleold:=SELE()
SELE cat
IF (LASTKEY()=K_F2 .OR. lcgcatvar=SPACE(3)) .AND. LASTKEY()<>27
  lcgrev := .F.
  lcgcatvar := BROWPRO("CAT",LCGCATP)
ENDIF
SELE(seleold)
RETURN nil

/*****
* PROF_BAJ()
*
* Función que realiza la baja de un registro
*****/
FUNCTION PROF_BAJ()
MEMVAR getlist.lcgtinte6,lcgrfc,lcgdate,seguirb,lcgpbaj,lcgtinte
FIELD RFC,NOM,PRO
LOCAL respbaja
respbaja = 2
lcgpbaj = SAVESCREEN(0,0,24,79)
CUADRO(lcgtinte6,10,25,13,60,"5")

```

```

@11.27 SAY "" DESEA BORRAR ESTE REGISTRO ?"
@12.35 PROMPT "SI"
@12.45 PROMPT "NO"
MENU TO respbaja
SETCOLOR(lcgintec)
RESTSCREEN(0,0,24,79,lcgpbaj)
DO CASE
  CASE respbaja = 1
    SELE coor
    SEEK lcgffc
    IF FOUND()
      lcgdate := DATE()
      IF lcgdate > coor->fe_inicio .OR. lcgdate = coor->fe_inicio
        IF lcgdate < coor->fe_limite .OR. lcgdate = coor->fe_limite
          IF coor->fe_inicio = coor->antiguedad
            IF RLOCK()
              DELE
              CLOSE coor
              USE &lcgruta\&lcgprocoo ALIAS coor NEW EXCL
              PACK
              INDEX ON rfc TO &lcgruta\coorffc
              INDEX ON nom_pro TO &lcgruta\coornom
              CLOSE coor
              USE &lcgruta\&lcgprocoo ALIAS coor NEW SHARED
              SET INDEX TO &lcgruta\coorffc,&lcgruta\coornom
              ALERT("REGISTRO DADO DE BAJA")
            ELSE
              ALERT("EL REGISTRO NO ESTA BLOQUEADO")
            ENDIF
          ELSE
            ALERT("EL REGISTRO NO SE PUEDE DAR DE BAJA")
          ENDIF
        ELSE
          ALERT("EL REGISTRO NO SE PUEDE DAR DE BAJA")
        ENDIF
      ELSE
        ALERT("LOS DATOS HAN SIDO CANCELADOS ")
      ENDIF
    CASE respbaja = 2
      ALERT (" OPERACION CANCELADA ")
    ENDCASE
  KEYBOARD CHR(27)+CHR(T_BORRAR)
  RETURN nil
// EOF()

```

```

*****
* PROGRAM-ID: Prof.Pro.prg ver 1.0 llamado por Prof_Lcg.prg *
* AUTHOR: LetiGia Cerd Garrido *
* REMARKS: Programa que Muestra el Menú para Generar, Revisar *
* las Propuestas y tarjetas de Profesores *
* MODIFIED: 17 Enero, 1997. *
*****
MEMVAR getlist,lcgtinte,lcgopp,titulo,texto,lcgpcnpp,lcgpunam,lcgruta

USE &lcgruta\&lcgprocoo ALIAS coor NEW SHARED
SET INDEX TO &lcgruta\coorrfc,&lcgruta\coornom
USE &lcgruta\&lcgproaut INDEX &lcgruta\autrfc ALIAS paut NEW EXCL
USE &lcgruta\&lcgproman INDEX &lcgruta\manrfc ALIAS pman NEW EXCL
USE &lcgruta\&lcgmat INDEX &lcgruta\mat ALIAS mac NEW SHARED

CUADRO (lcgtinte,8,30,12,41,"S") ;lcgopp=1
DO WHILE lcgopp#0
  TITULO := {"GENERAR ","REVISAR "," SALIR "}
  TEXTO := {"GENERAR LA INFORMACION PARA LA PROPUESTA",;
           "REVISAR LA INFORMACION DE LA PROPUESTA",;
           "SALIR AL MENU ANTERIOR"}
  MENUV(8,30,TITULO,TEXTO)
  SET KEY 19 TO ; SET KEY 4 TO
  MENU TO lcgopp
  lcgpcnpp=SAVESCREEN(0,0,24,79)
  DO CASE
    CASE lcgopp=1 ; PROF_GNE()
    CASE lcgopp=2 ; DO PROF_REV
    CASE lcgopp=3.OR.LASTKEY()=27
      EXIT
  ENDCASE
  RESTSCREEN(0,0,24,79,lcgpcnpp)
ENDDO
RESTSCREEN(0,0,24,79,lcgpunam)
CLOSE coor
CLOSE paut
CLOSE pman
CLOSE mat
RELEASE lcgopp,titulo,texto,lcgpcnpp,lcgconfg,colorold,lcgpgen,lcgops,;
lcgpcnng,lcgthorant,lcgthoract,lcgtbancon,lcgfiag,lcgrfci,lcgrfrc
f. ;
lcgban,lcgrfcp,lcgrev

RETURN

/*****
* PROF_GNE()
*
* Funcion que muestra el menú para generar la información de las
propuestas
*****/
FUNCTION PROF_GNE()
MEMVAR
getlist,lcgconfg,colorold,lcgpgen,lcgerror,lcgtinte,lcgops,titulo,;
texto,lcgpcnng,lcgpcnpp
lcgconfg=2
colorold:=SETCOLOR()
lcgpgen=SAVESCREEN(0,0,24,79)
CUADRO(lcgerror,15,20,18,65,"S")
@16,22 SAY "LA INFORMACION DE PROPUESTAS SE MODIFICARA"
@17,30 PROMPT "CONTINUAR"
@17,45 PROMPT "CANCELAR"
SET KEY 19 TO ; SET KEY 4 TO
MENU TO lcgconfg
SETCOLOR(colorold)
RESTSCREEN(0,0,24,79,lcgpgen)

```

```

DO CASE
  CASE lcgconfig=1
    CUADRO(lcgtinte,10,39,14,47,"S") ;lcgpg=1
  DO WHILE lcgpg#0
    TITULO:={"TODOS","RANGO","SALIR"}
    TEXTO :={"GENERAR PARA TODOS","GENERAR DE UN RANGO","SALIR AL MENU
  ANTERIOR"}
    MENUV(10,39,TITULO,TEXTO)
    SET KEY 19 TO ; SET KEY 4 TO
    MENU TO lcgpg
    lcgpcng=SAVESCREEN(0,0,24,79)
  DO CASE
    CASE lcgpg=1 ; PRO_GENT()
    CASE lcgpg=2 ; PRO_GENR()
    CASE lcgpg=3.OR.LASTKEY()=27
      EXIT
    ENDCASE
    RESTSCREEN(0,0,24,79,lcgpcng)
  ENDDO
  CASE lcgconfig=2.OR.LASTKEY()=27
    RESTSCREEN(0,0,24,79,lcgpcng)
    RETURN nil
  ENDCASE
  RESTSCREEN(0,0,24,79,lcgpcng)
  RETURN nil

/*****
* PRO_GENT()
*
* Función que genera la información para la propuesta de todos los
* profesores
*****/
FUNCTION PRO_GENT()
MEMVAR
  getlist,lcgpnam,lcgunam,lcgrfcp,lcgthorant,lcgthoract,lcgtbancoh,;
  lcgtinte,lcgpcng,lcgflag,banentry,htteorica,htpractica,hteorica,
;
  hpractica
  CLRSCR (5,130)
  RESTSCREEN(0,0,24,79,lcgpnam)
  CUADRO(lcgunam,11,9,14,39,"S")
  SET CURSOR OFF
  @12,12 SAY "GENERACION DE INFORMACION" ; @13,13 SAY "PROPUESTAS POR
  PROFESOR"
  CUADRO(lcgunam,16,40,19,70,"S")
  @17,41 SAY "ESPERE...PROCESO EN EJECUCION" ; @18,46 SAY "GRACIAS POR
  ESPERAR"
  SELE paut ; ZAP
  SELE pman ; ZAP
  SELE coor
  SET ORDER TO 1
  GO TOP
  DO WHILE !EOF()
    lcgrfcp := coor->rfc
    BANENTRY := BAPRHR(lcgrfcp)
    SELE coor
    IF BANENTRY!=NIL .AND. LEN(banentry)>0
      IF RLOCK()
        REPLACE coor->banco_hora WITH BANENTRY[1]
        REPLACE coor->hor_o_jef WITH BANENTRY[2]
        REPLACE coor->area_o_jef WITH BANENTRY[3]
        REPLACE coor->cat_o_jef WITH BANENTRY[4]
        REPLACE coor->hor_tec_an WITH BANENTRY[5]
        REPLACE coor->hor_pra_an WITH BANENTRY[6]
        REPLACE coor->hor_tec_ WITH BANENTRY[7]

```

```

        REPLACE coor->hor_pra      WITH BANENTRY(8)
    ENDIF
ENDIF
RELEASE BANENTRY
lcgthorant := coor->hor_teo_an + coor->hor_pra_an
lcgthoract := coor->hor_teo + coor->hor_pra
lcgtbancoh := coor->banco_hora
lcgflag    := 0
hteorica  := 0
htpractica := 0
hteorica  := 0
hpractica := 0
* IF lcgthoract <= lcgtbancoh
* DO PROF_GEN
* ENDF
  SELE coor
  IF RLOCK()
    REPLACE coor->hor_teo WITH hteorica
    REPLACE coor->hor_pra WITH htpractica
  ENDF
  SKIP
ENDDO
SET CURSOR ON
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpcong)
RETURN nil

/*****
* PRO_GENR()
*
* Función que genera la información de propuestas de un rango de
* profesores
*****/
FUNCTION PRO_GENR()
MEMVAR
getlist,lcgpunam,lcgrfci,lcgrfcf,lcgban,lcgunam,lcgtinte,lcgrfcp,;
lcgthorant,lcgthoract,lcgtbancoh,lcgpcong,lcgflag,lcgrev,lcgcampo
* ;
    hteorica,htpractica,hteorica,hpractica

LOCAL banentry
#include "inkey.ch"
CLRSCR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
lcgrfci:=SPACE(13); lcgrfcf:=SPACE(13); lcgban=0
CUADRO(lcgunam,9,10,12,40,"S")
@10,13 SAY "GENERACION DE INFORMACION" ; @11,14 SAY "PROPUESTAS POR
PROFESOR"
CUADRO(lcgunam,13,25,17,55,"S")
DO WHILE lcgban=0.OR.LASTKEY()=27
  DECLARE LCGCAMPO[2]
  LCGCAMPO[1] := "RFC"
  LCGCAMPO[2] := "NOM_PRO"
  SET KEY K F2 TO COD_AYUDA
  SET CURSOR ON
  @14,26 SAY "R.F.C. INICIAL:" GET lcgrfci PICT "@!"; READ
  IF (LASTKEY() = K_F2 .OR. lcgrfci = SPACE(13)).AND. LASTKEY() <> 27
    lcgrev := .F.
    lcgrfci := BROWPRO("COOR",LCGCAMPO)
    @14,42 SAY lcgrfci PICT "@!";
  ENDF
  RELEASE LCGCAMPO
  SET KEY K F2 TO
  IF (LASTKEY() <> 27 .AND. LEN(lcgrfci) <> 0)
    SELE coor

```

```

SET ORDER TO 1
SEEK lcgrfci
IF !FOUND() ; ALERT("NO EXISTE EN EL CATALOGO DE PROFESORES")
    lcgrfci=SPACE(13)
ELSE ; lcgban=1 ; ENDIF
ENDIF
ENDDO
lcgban=0
DO WHILE lcgban=0.OR.LASTKEY()=27
    DECLARE LCGCAMPO[2]
    LCGCAMPO[1] := "RFC"
    LCGCAMPO[2] := "NOM_PRO"
    SET KEY K F2 TO COD AYUDA
    @16,26 SAY "R.F.C. FINAL: " GET lcgrfcf PICT "@!"; READ
    IF (LASTKEY() = K.F2 .OR. lcgrfcf = SPACE(13)).AND. LASTKEY() <> 27
        lcgrcv := .F.
        lcgrfcf := BROWPRO("COOR",LCGCAMPO)
        @16,42 SAY lcgrfcf PICT "@!"
    ENDIF
    RELEASE LCGCAMPO
    SET KEY K F2 TO
    IF (LASTKEY() <> 27 .AND. LEN(lcgrfcf) <> 0)
        SELE coor
        SEEK lcgrfcf
        IF !FOUND()
            ALERT("NO EXISTE EN EL CATALOGO DE PROFESORES")
        ;lcgrfcf=SPACE(13)
        ELSEIF lcgrfcf < lcgrfci
            lcgrfcf = SPACE(13)
        ALERT("EL R.F.C. FINAL DEBE SER MAYOR QUE EL INICIAL")
        ELSE
            lcgban=1
        ENDIF
    ENDIF
ENDIF
ENDDO
CUADRO(lcgunam,18,40,21,70,"6")
SET CURSOR OFF
@19,41 SAY "ESPERE...PROCESO EN EJECUCION" ; @20,46 SAY "GRACIAS POR
ESPERAR"
SELE coor
SEEK lcgrfci
IF FOUND()
    DO WHILE coor->rfc=>lcgrfci.AND.coor->rfc<=>lcgrfcf
        lcgrfcf:=coor->rfc
        SELE paut
        SEEK lcgrfcf
        IF FOUND()
            DO WHILE paut->rfc = lcgrfcf
                IF RLOCK() ; DELE; ENDIF
            SKIP
        ENDDO
    ENDIF
    SELE pman
    SEEK lcgrfcf
    IF FOUND()
        DO WHILE pman->rfc = lcgrfcf
            IF RLOCK() ; DELE; ENDIF
        SKIP
    ENDDO
ENDIF
SELE coor
SKIP
ENDDO
SELE paut ; PACK
SELE pman ; PACK

```

```

ENDIF
SELE coor
SEEK lcgrfci
IF FOUND()
DO WHILE coor->rfc=>lcgrfci.AND.coor->rfc<lcgrfci
lcgrfcp := coor->rfc
BANENTRY := BAPRHOR(lcgrfcp)
SELE coor
IF banentry=NIL .AND. LEN(banentry)>0
IF RLOCK()
REPLACE coor->banco_hora WITH BANENTRY [1]
REPLACE coor->hor_o_jef WITH BANENTRY [2]
REPLACE coor->area_o_jef WITH BANENTRY [3]
REPLACE coor->cat_o_jef WITH BANENTRY [4]
REPLACE coor->hor_teo_an WITH BANENTRY [5]
REPLACE coor->hor_pra_an WITH BANENTRY [6]
REPLACE coor->hor_teo WITH BANENTRY [7]
REPLACE coor->hor_pra WITH BANENTRY [8]
ENDIF
ENDIF
RELEASE BANENTRY
lcgthorant := coor->hor_teo_an + coor->hor_pra_an
lcgthoract := coor->hor_teo + coor->hor_pra
lcgtbancoh := coor->banco_hora
lcglag := 0
htteorica := 0
htpractica := 0
htteorica := 0
htpractica := 0
* IF lcgthoract <= lcgtbancoh
* DO PROF_GEN
* ENDF
SELE coor
IF RLOCK()
REPLACE coor->hor_teo WITH htteorica
REPLACE coor->hor_pra WITH htpractica
ENDIF
SKIP
ENDDO
ENDIF
SET CURSOR ON
SETCOLOR(14,14)
RESTSCREEN(0,0,24,79,lcgpcong)
RETURN nil
// EOF()

```

```

*****
* PROGRAM-ID: Prof_Eva.prg ver 1.0 llamado por Prof_LCG.prg *
* AUTHOR: Leticia Cerd Garrido *
* REMARKS: Programa que Realiza la captura de Evaluacion de *
* Profesores *
* MODIFIED: 07 Junio, 1997. *
*****
#include "inkey.ch"
MEMVAR getlist,lcgtinte2,lcgtinte6,lcgtinte,lcgruta,lcgcampo,lcgeva, ;
vm_inicial,vm_tit_carr

LOCAL
eal,ea2,ea3,ea4,ea5,ep1,ep2,ep3,ep4,ep5,ep6,ep7,ep8,ep9,ep10,com,cor, ;
mas,lcgnom,lcgppo,lcgmat,lcgvcve,oldcolor,lcgrfc,lcgpcce,lcgrev

USE &lcgruta&\&lcgprocoo INDEX &lcgruta\&coornom ALIAS coor NEW SHARED
USE &lcgruta\&lcgeva INDEX &lcgruta\eva ALIAS eva NEW SHARED

lcgpcce=SAVESCREEN(0,0,24,79)
SET KEY 19 TO
SET KEY 4 TO
mas="S"
lcgnom=SPACE(30)
lcgppo=SPACE(4)
lcgmat=SPACE(20)
lcgvcve=SPACE(4)
DO WHILE LASTKEY() <> 27.AND. (mas="S".OR.mas="s")
CUADRO (lcgtinte2,1,2,24,77,"S")
eal=0;ea2=0;ea3=0;ea4=0;ea5=0;com=SPACE(60);cor="N"
ep1=0;ep2=0;ep3=0;ep4=0;ep5=0;ep6=0;ep7=0;ep8=0;ep9=0;ep10=0
CLRSCR (5, 130)
DO WHILE LASTKEY() <> 27.AND. (cor="N".OR.cor="n")
@1,15 SAY "ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGON"
@2,10 SAY "CUESTIONARIO DE EVALUACION DEL PROCESO ENSEANZA"
APRENDIZAJE
DECLARE LCGCAMPO[2]
LCGCAMPO[1] := "NOM_PRO"
LCGCAMPO[2] := "RFC"
SET KEY K_F2 TO COD_AYUDA
lcgnom := SPACE(30)
@4,5 SAY "NOMBRE DEL PROFESOR: " GET lcgnom PICT "@!"
READ
IF (LASTKEY() = K_F2 .OR. lcgnom = SPACE(30)).AND. LASTKEY() <> 27
lcgrev := .F.
SELE coor
lcgnom := BROWPRO("COOR",LCGCAMPO)
@4,27 SAY lcgnom PICT "@!"
ENDIF
RELEASE LCGCAMPO
SET KEY K_F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgnom) <> 0)
SELE coor
SEEK lcgnom
IF !FOUND()
ALERT("EL NOMBRE DEL PROFESOR NO EXISTE")
ELSE
@05,05 SAY "NOMBRE DE LA MATERIA: " GET lcgmat PICT "@!"
@05,40 SAY "CLAVE DE LA MATERIA: " GET lcgvcve PICT "@!"
@06,05 SAY "GRUPO: " GET lcgppo PICT "@!"
@06,30 SAY "CARRERA: "
@06,40 SAY vm_tit_carr PICT "@!"
@07,19 SAY "PARTE I AUTOEVALUACION DE ALUMNO"
@07,52 SAY "PARTE II EVALUACION DEL ALUMNO HACIA EL
PROFESOR"
@08,05 SAY "A) PLANEACION DEL CURSO"

```

```

"abcdABCD"  @09,05 SAY "1) " GET ea1 PICT "X" VALID ea1 $
"abcdABCD"  @11,05 SAY "A) PLANEACION DEL CURSO"
"abcdABCD"  @12,09 SAY "2) " GET ea2 PICT "X" VALID ea2 $
"abcdABCD"  @14,05 SAY "A) PLANEACION DEL CURSO"
"abcdABCD"  @15,09 SAY "3) " GET ea3 PICT "X" VALID ea3 $
"abcdABCD"  @17,05 SAY "A) PLANEACION DEL CURSO"
"abcdABCD"  @18,09 SAY "4) " GET ea4 PICT "X" VALID ea4 $
"abcdABCD"  @20,05 SAY "A) PLANEACION DEL CURSO"
"abcdABCD"  @21,09 SAY "5) " GET ea5 PICT "X" VALID ea5 $
"abcdABCD"  @09,44 SAY "1) " GET ep1 PICT "X" VALID ep1 $
"abcdABCD"  @10,44 SAY "2) " GET ep2 PICT "X" VALID ep2 $
"abcdABCD"  @12,44 SAY "3) " GET ep3 PICT "X" VALID ep3 $
"abcdABCD"  @13,44 SAY "4) " GET ep4 PICT "X" VALID ep4 $
"abcdABCD"  @15,44 SAY "5) " GET ep5 PICT "X" VALID ep5 $
"abcdABCD"  @16,44 SAY "6) " GET ep6 PICT "X" VALID ep6 $
"abcdABCD"  @18,44 SAY "7) " GET ep7 PICT "X" VALID ep7 $
"abcdABCD"  @19,44 SAY "8) " GET ep8 PICT "X" VALID ep8 $
"abcdABCD"  @21,44 SAY "9) " GET ep9 PICT "X" VALID ep9 $
"abcdABCD"  @22,57 SAY "10) " GET ep10 PICT "X" VALID ep10 $
"abcdABCD"  @23,3 SAY "COMENTARIOS:" GET com PICT "@"
SET CURSOR ON
READ
oldcolor:= SETCOLOR()
CUADRO(lcgtint=6,12,18,14,56,"S")
@13,20 SAY "ES CORRECTA LA INFORMACION S/N?" GET cor PICT
"@";
VALID cor $ "Snen"
READ
SETCOLOR(oldcolor)
ENDIF
ENDIF
ENDDO
IF cor="S".OR.cor="s"
  SELE coor
  SEEK lcgnom
  IF FOUND()
    lcgRFC=coor->rfc
    SELE eva
    APPEND BLANK
    REPLA eva->profesor WITH lcgnom, eva->grupo WITH lcgppo
    REPLA eva->materia WITH lcgmat, eva->clave WITH lcgcv
    REPLA eva->a01 WITH ea1, eva->a02 WITH ea2, eva->a03 WITH ea3
    REPLA eva->a04 WITH ea4, eva->a05 WITH ea5, eva->p01 WITH ep1
    REPLA eva->p02 WITH ep2, eva->p03 WITH ep3, eva->p04 WITH ep4
    REPLA eva->p05 WITH ep5, eva->p06 WITH ep6, eva->p07 WITH ep7
    REPLA eva->p08 WITH ep8, eva->p09 WITH ep9, eva->p10 WITH ep10
    REPLA eva->comentario WITH com, eva->rfc WITH lcgRFC
  ELSE
    ALERT("EL NOMBRE DEL PROFESOR NO EXISTE")
  ENDIF
ENDIF

```

```
ENDIF
CUADRO(lcgtinte6,12,18,14,56,"S")
@13,20 SAY "DESEA CAPTURAR OTRO REGISTRO S/N?" GET mas PICT "@!";
      VALID mas $ "SNan"
READ
ENDDO
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpcoe)
CLOSE coor
CLOSE eva
RELEASE
ea1,ea2,ea3,ea4,ea5,ep1,ep2,ep3,ep4,ep5,ep6,ep7,ep8,ep9,ep10,com,;
cor,mas,lcgnom,lcggpo,lcgmat,lcgcve,oldcolor,lcgrfc,lcgpcoe,lcgr
ev
RETURN
*/ EOF()
```

```

*****
* PROGRAM-ID:      Prof Imp.prg ver 1.0      llamado por Prof_Lcg.prg
* AUTHOR:         Leticia Cerdá Garrido
* REMARKS:        Programa que Muestra el Menú para Imprimir las eva-
*                 luaciones, las Propuestas y tarjetas de Profesores
* MODIFIED:       26 Abril, 1997.
*****
#include "inkey.ch"
MEMVAR getlist,lgconf,colorold,lgerror,lgpconp,lgtime,lgopi,;
        titulo,texto,lgpconi,opc,lgcrsup,lgcizq,lggrinf,lgcder,lggruta
.;
PUBLIC inicia,little,entrega
        inicia,little,entrega

lgconf=1
colorold:=SETCOLOR()
lgpconp=SAVESCREEN(0,0,24,79)
*CUADRO(lgerror,10,20,13,64,"S")
*@11,22 SAY "LA INFORMACION PUEDE NO ESTAR ACTUALIZADA"
*@12,30 PROMPT "CONTINUAR"
*@12,45 PROMPT "CANCELAR"
SET KEY 19 TO ; SET KEY 4 TO
*MENU TO lgconf
SETCOLOR(colorold)
DO CASE
CASE lgconf=1
        USE &lggruta&lgprocoo INDEX &lggruta\coorffc ALIAS coor NEW
SHARED
        USE &lggruta&lgprorev ALIAS prop NEW SHARED
        SET INDEX TO &lggruta\proprffc,&lggruta\proptip
        RESTSCREEN(0,0,24,79,lgpconp)
        lgcrsup:=8 ; lgcizq:=56 ; lggrinf:=13 ; lgcder:=70
/* entrega := BA_CUAL_()
        IF entrega=NIL .AND. LEN(entrega)>0
                IF entrega = 1 .OR. entrega = 3
                        INICIA = CHR(27)+'E'
                                &&& --- RESET
IMPRESORA
        LITTLE = CHR(27)+'(s18h(s8S' &&& '(s25H' &&& --- LETRA
PEQUEVA SUPER CONDENCED
        ENDIF
        IF entrega = 2
                INICIA = CHR(27)+'@'
                                &&& --- RESET IMPRESORA
                LITTLE = CHR(27)+'P'+CHR(015)
                                &&& --- LETRA PEQUEVA SUPER
CONDENCED
        ENDIF
        ENDIF
        REPORTE()*/
        CHOOSE()
/* IF OPC="N"
        RETURN
ENDIF */
RESTSCREEN(0,0,24,79,lgpconp)
CUADRO(lgtime,8,56,13,67,"S")
lgopi=1
DO WHILE lgcopi<=0
        TITULO:={"EVALUACION","PROPUUESTAS"," TARJETAS"," SALIR "}
        TEXTO :={"IMPRIME LA EVALUACION DE PROFESORES";;
                "IMPRIME LAS PROPUUESTAS";;
                "IMPRIME LAS TARJETAS DE ASISTENCIA";;
                "SALIR AL MENU ANTERIOR"}
        MENUV(8,55,TITULO,TEXTO)
        SET KEY 19 TO ; SET KEY 4 TO
        MENU TO lgcopi
        lgcpconi=SAVESCREEN(0,0,24,79)
DO CASE

```

```

CASE lcgopi=1
  DO PROF REP
CASE lcgopi=2
  IMP_PROP()
CASE lcgopi=3
  DO PROF TAR
CASE lcgopi=4.OR.LASTKEY()=27
  EXIT
ENDCASE
RESTSCREEN(0,0,24,79,lcgpconi)
ENDDO
CASE lcgconf=2.OR.LASTKEY()=27
  RESTSCREEN(0,0,24,79,lcgpcomp)
  RETURN
ENDCASE
RESTSCREEN(0,0,24,79,lcgpcomp)
CLOSE coor
CLOSE prop
RELEASE
lcgconf, colorold, lcgpcomp, lcgopi, titulo, texto, lcgpconi, opc, lcgrsup, ;
lcgciqz, lcgrinf, lcgeder, lcgopi, lcgpconi, lcgren, lcgcot, legtotal
;
;
lcgpcopy, lcgrfcp, lcgrfci, lcgrfcf, lcgban, lcgrfcp, lcgre, seleold, ;
lcgimp, lcghoj, movver, inicia, movhor, medl, hofi, cahofi, alma, totline
;
;
black, white, rmedl, lcgpimp, big, nor, little, blackn, normal, lcgfecha,
;
lcgfecha, lcgmov, lcgcot, lcgcotj, lcgcotd, lcgmesc, lcgmesc, lcgrfcp
;
;
lcgkey, lcgkey1, lcgmec, lcgkcat, lcgkcatj, lcgkcatd, lcgfini, lcgmeci,
;
lcgflim, lcgmesc, lcgmesc, lcgtot_an, lcgttee, lcgtpra, lcgttot, cont

RETURN

/*****
* IMP_PROP()
*
* Funcion que presenta el menf de impresion de propuestas
*****/
FUNCTION IMP_PROP()
MEMVAR getlist, lcgopi, lcgpconi, lcgpconi, lcgtinte, titulo, texto, ;
lcgpcopia, lcgrn, lcgcot, lcgpcopy
PUBLIC lcgpcopia

lcgpcopy=SAVESCREEN(11,56,17,79)
SET CURSOR ON
CUADRO(lcgtinte, 11,56,15,76,"S") ;lcgpcopia=1;lcgren=0 ;lcgcot=0
@12,57 SAY "NUMERO DE COPIAS:"
@12,75 GET lcgpcopia PICT "9" RANGE 1,3
@13,57 SAY "RENGLON INICIAL:" GET lcgren PICT "99" RANGE -9,10
@14,57 SAY "COLUMNA INICIAL:" GET lcgcot PICT "99" RANGE -9,10
READ
lcgren:= lcgren-1
lcgcot:= lcgcot-1
SETCOLOR(lcgtinte)
RESTSCREEN(11,56,17,79,lcgpcopy)
SET CURSOR OFF
CUADRO(lcgtinte, 11,65,15,73,"S")
lcgopi=1
DO WHILE lcgopi<=>0
  TITULO:={"TODOS", "RANGO", "SALIR"}
  TEXTO :={"IMPRIMIR TODOS", "IMPRIMIR UN RANGO", "SALIR AL MENU
ANTERIOR"}
  MENUV(11,65,TITULO,TEXTO)

```

```

SET KEY 19 TO ; SET KEY 4 TO
MENU TO lcgop11
lcgpccon11=SAVESCREEN(0,0,24,79)
DO CASE
  CASE lcgop11=1
    PRO_IMPT()
  CASE lcgop11=2
    PRO_IMPR()
  CASE lcgop11=3.OR.LASTKEY()=27
    EXIT
ENDCASE
  RESTSCREEN(0,0,24,79,lcgpccon11)
ENDDO
  RESTSCREEN(0,0,24,79,lcgpccon1)
  RETURN nil

/*****
* PRO_IMPT()
*
* Función que imprime las propuestas de todos los profesores
*****/
FUNCTION PRO_IMPT()
MEMVAR getlist,lcgpunam,lcgunam,lcgrfcf,lcgtinte,lcgpccon1,lcgpcopia
PUBLIC lcgpcopia
CLRSCR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
CUADRO(lcgunam,11,8,15,20,"S")
@12,10 SAY "IMPRESION" ; @13,13 SAY "DE" ; @14,10 SAY "PROPUESTA"
CUADRO(lcgunam,16,30,19,60,"S")
@17,31 SAY "ESPERE... PROCESO EN EJECUCION"
@18,36 SAY "GRACIAS POR ESPERAR"
SELE COOR
GO TOP
DO WHILE !EOF()
  IF COOR->propuesta='*'
    lcgrfcf=COOR->rfc
    DO PROF_FOR
  ENDDIF
  SELE COOR
  SKIP
ENDDO
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpccon1)
RETURN nil

/*****
* PRO_IMPR()
*
* Función que imprime las propuestas de un rango de profesores
*****/
FUNCTION PRO_IMPR()
MEMVAR
getlist,lcgpunam,lcgrfcf,lcgrfcf,lcgban,lcgunam,lcgtinte,lcgrfcf, ;
lcgpccon1,lcgrev,lcgcampo,seleold,lcgpcopia
PUBLIC lcgpcopia
CLRSCR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
lcgrfcf:=SPACE(13); lcgrfcf:=SPACE(13); lcgban=0
CUADRO(lcgunam, 9,10,12,38,"S")
@10,14 SAY "IMPRESION DE PROPUESTAS" ; @11,19 SAY "R A N G O"
CUADRO(lcgunam,13,23,17,53,"S")
DO WHILE lcgban=0.AND.LASTKEY()<>27
  DECLARE LCGCAMPO[2]
  LCGCAMPO[1] := "RFC"

```

```

LCGCAMPO[2] := "NOM_PRO"
SET KEY K F2 TO COD_AYUDA
SET CURSOR ON
@14,24 SAY "R.F.C. INICIAL:" GET lcgrfcf PICT "@!" ;READ
IF (LASTKEY() = K_F2 .OR. lcgrfcf = SPACE(13)).AND. LASTKEY() <> 27
  lcgrfv := F
  seleold:= SELE()
  SELE coor
  SET FILTER TO coor->propuesta="*"
  GO TOP
  lcgrfcf := BROWPRO("COOR",LCGCAMPO)
  SET FILTER TO
  SELE(seleold)
  @14,40 SAY lcgrfcf PICT "@!"
ENDIF
RELEASE LCGCAMPO
SET KEY K F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgrfcf) <> 0)
  SELE prop
  SET ORDER TO 1
  SEEK lcgrfcf
  IF !FOUND() ; ALERT("NO EXISTE EN EL ARCHIVO DE PROPUESTAS")
    lcgrfcf=SPACE(13)
  ELSE ; lcgban=1 ;ENDIF
ENDIF
ENDDO
lcgban=0
DO WHILE lcgban=0.AND.LASTKEY()<>27
  DECLARE LCGCAMPO[2]
  LCGCAMPO[1] := "RFC"
  LCGCAMPO[2] := "NOM_PRO"
  SET KEY K F2 TO COD_AYUDA
  @16,24 SAY "R.F.C. FINAL: " GET lcgrfcf PICT "@!" ;READ
  IF (LASTKEY() = K_F2 .OR. lcgrfcf = SPACE(13)).AND. LASTKEY() <> 27
    lcgrfv := F
    seleold:= SELE()
    SELE coor
    SET FILTER TO coor->propuesta="*"
    GO TOP
    lcgrfcf := BROWPRO("COOR",LCGCAMPO)
    SET FILTER TO
    SELE(seleold)
    @16,40 SAY lcgrfcf PICT "@!"
  ENDIF
  RELEASE LCGCAMPO
  SET KEY K F2 TO
  IF (LASTKEY() <> 27 .AND. LEN(lcgrfcf) <> 0)
    SELE prop
    SEEK lcgrfcf
    IF !FOUND()
      ;lcgrfcf=SPACE(13)
    ELSEIF lcgrfcf < lcgrfcf
      lcgrfcf = SPACE(13)
      ALERT("EL R.F.C. FINAL DEBE SER MAYOR QUE EL INICIAL")
    ELSE
      lcgban=1
    ENDIF
  ENDIF
ENDIF
ENDDO
CUADRO(lcgunam,18,38,21,68,"S")
@19,39 SAY "ESPERE...PROCESO EN EJECUCION" ; @20,44 SAY "GRACIAS POR
ESPERAR"
SELE coor
SEEK lcgrfcf

```

```

IF FOUND()
DO WHILE coor->rffc->lcgrfcf.AND.coor->rffc<=lcgrfcf
IF coor->propuesta = 'A'
lcgrfcf=coor->rffc
DO PROF_FOR
ENDIF
ELSE coor
SKIP
ENDDO
ENDIF
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpconi)
RETURN nil

/*****
* CHOOSE()
*
* Función que permite seleccionar el tipo de impresora
*****/
FUNCTION CHOOSE()
MEMVAR
getlist,lcgimp,lcghoj,movver,incia,movhor,medl,hofi,cahofi,alma,;
totline,black,white,titulo,opc,lcgtinte,rmedl,lcgimp,big,;
nor,little,blackn,normal,lcgrsup,lcgcizq,lcgrinf,lcgcedr,lcgpcopn
PUBLIC
lcgimp,lcghoj,movver,incia,movhor,medl,hofi,cahofi,alma,totline,;
black,white,opc,rmedl,big,nor,little,blackn,normal
CUADRO(lcgtinte,lcgrsup,lcgcizq,lcgrinf,lcgcedr,"S") ;lcgimp=1 ; OPC='N'
DO WHILE lcgimp<=0.AND.LASTKEY()<=27
TITULO:={"REMOTA LASER ", "REMOTA MATRIZ" ;
"LOCAL LASER " "LOCAL MATRIZ "}
TEXTO :={"IMPRESORA REMOTA LASER", "IMPRESORA REMOTA MATRIZ",;
"IMPRESORA LOCAL LASER", "IMPRESORA LOCAL MATRIZ"}
MENUV(lcgrsup,lcgcizq-1,TITULO,TEXTO)
SET KEY 19 TO ; SET KEY 4 TO
MENU TO lcgimp
lcgimp=SAVESCREEN(0,0,24,79)
DO CASE
CASE lcgimp = 1 .OR. lcgimp = 3 &&& LASER
SET MARGIN TO 10
IF PRON() > 0
EJECT
ENDIF
IF lcgimp = 1 .OR. lcgimp = 2
SET DEVICE TO PRINT
* SET PRINTER TO \\spooler\nb
SET DEVICE TO SCREEN
ELSE
!ENDCAP > MESSAGES.TXT
SET PRINTER TO
ENDIF
REPORTE()
IF OPC='N'
RESTSCREEN(0,0,24,79,lcgpcopn)
* CLOSE coor
* CLOSE prop
RETURN(OPC)
ENDIF
BIG = CHR(27)+'(slp(s24s(s4b(s3T' &&& --- CODIGOS DE
ESCAPE letra grande
NOR = CHR(27)+'(s0p4k2s(s8s(s-4b(s3T' &&& --- CODIGOS DE
ESCAPE letra normal
LITTLE = CHR(27)+'(s18h(s8S' &&& '(s25H' &&& --- LETRA PEQUEÑA
SUPER CONDENCED
BLACKN = CHR(27)+'&k0s(s0s7B' &&& --- LETRA NORMAL

```

```

NEGRITA
LINEA MEDL = CHR(27)+'='  &&& --- AVANCE MEDIA
      HOPI = CHR(27)+'&13A'  &&& --- HOJA OFICIO
      ALMA = CHR(27)+'&10H'  &&& --- ALIMENTACION
AUTOMATICA
      CAHOPI = CHR(27)+'&12A'  &&& --- HOJA CARTA
      INICIA = CHR(27)+'E'  &&& --- RESET
IMPRESORA
      MOVHOR = CHR(27)+'&11O'  &&& --- IMPRESION
HORIZONTAL
      MOVVER = CHR(27)+'&10O'  &&& --- IMPRESION
VERTICAL
      BLACK = CHR(27)+'(s7B'  &&& --- LETRA NEGRITA
      WHITE = CHR(27)+'(s0B'  &&& --- LETRA NORMAL
      @0,0 SAY CAHOPI
      @0,0 SAY MOVVER
CASE lcgimp = 2 .OR. lcgimp = 4  &&&--  MATRIZ
* CUADRO(lcgtinte,14,43,16,76,"S")
* @15,44 PROMPT " HOJAS SUELTAS "
* @15,59 PROMPT " HOJAS CONTINUAS "
SET KEY 19 TO ; SET KEY 4 TO
* MENU TO leghoj
leghoj=1
IF leghoj = 1
  CAHOPI= CHR(27)+'C'+CHR(47)  &&& --- HOJA SUELTA
  TOTLINE = 50
  REPORTE()
  IF OPC="N"
    RESTSCREEN(0,0,24,79,lcgpcnp)
    * CLOSE coor
    *CLOSE prop
    RETURN(OPC)
  ENDIF
ELSE
  CAHOPI= CHR(27)+'C'+CHR(66)  &&& --- HOJA CONTINUA
  TOTLINE = 61
  REPORTE()
  IF OPC="N"
    RESTSCREEN(0,0,24,79,lcgpcnp)
    * CLOSE coor
    *CLOSE prop
    RETURN(OPC)
  ENDIF
ENDIF
SET MARGIN TO 3
IF PROW() > 0
  EJECT
ENDIF
IF lcgimp = 1 .OR. lcgimp = 2
  SET DEVICE TO PRINT
  *SET PRINTER TO \\spooler\nb
  SET DEVICE TO SCREEN
ELSE
  ENDCAP > MESSAGES.TXT
  SET PRINTER TO
ENDIF
REPORTE()
IF OPC="N"
  RESTSCREEN(0,0,24,79,lcgpcnp)
  * CLOSE coor
  * CLOSE prop
  RETURN nil
ENDIF
@0,0 SAY CHR(27)+'E'  &&&+ '1'  &&& --- PARA GRAFICOS

```

ACEPTE LA 'V'

@0,0 SAY CHR(27)+'X'+1'
@0,0 SAY CHR(27)+'K'+0'
BIG = CHR(27)+'143'
NOR = CHR(27)+'10'

GRANDE

LITTLE = CHR(27)+'P'+CHR(015)

CONDENSED

NORMAL = CHR(27)+'P'
MEDL = CHR(27)+CHR(48)
RMEDL = CHR(27)+CHR(50)
HOFI = CHR(27)+'C'+CHR(66)
INICIA = CHR(27)+'@'
BLACK = CHR(27)+'E'
WHITE = CHR(27)+'F'
MOVVER = CHR(27)+'@'
@0,0 SAY CAHOFI

\$\$\$ --- LETRA GRANDE

\$\$\$ --- LETRA NORMAL ELIMINA LA

\$\$\$ --- LETRA PEQUEWA SUPER

\$\$\$ --- LETRA NORMAL

\$\$\$ --- AVANCE MEDIA LINEA

\$\$\$ --- RESTAURA LINEA NORMAL

\$\$\$ --- HOJA CARTA

\$\$\$ --- RESET IMPRESORA

\$\$\$ --- LETRA NEGRITA

\$\$\$ --- LETRA NORMAL

\$\$\$ --- OPCIONAL

ENDCASE

lcpimp:=0

ENDDO

RESTSCREEN(0,0,24,79,lcpimp)

RETURN(OPC)

//EOF()

```

*****
* PROGRAM-ID:      Prof_Aut.prg ver 1.0      llamado por Prof_Rev.prg *
* AUTHOR:         Leticia Cerd Garrido      *
* REMARKS:        Programa que Permite Revizar la Informacion antes *
*                de Imprimir las Propuestas. *
* MODIFIED:       08 Julio, 1996.          *
*****
#include "inkey.ch"
#define T_INSERT 22
MEMVAR getlist,lcgruta
USE &lcgruta\&lcgprorrev ALIAS prop NEW SHARED
SET INDX TO &lcgruta\&lcgprfc,&lcgruta\&lcgproptip
USE &lcgruta\&lcgipomov ALIAS mov NEW SHARED
SET INDX TO &lcgruta\&mov

SELE PAUT
GO TOP
DO WHILE LASTKEY() <> 27
  MOV_PAUT()
ENDDO
SET KEY K_F3 TO
SET KEY K_CTRL_DEL TO
SET KEY K_CTRL_INS TO
CLOSE prop
CLOSE mov
RELEASE lcgcont1,lcgrfc,lcgkey,lcgrev,lcgalt,lcgbanh,lcghrs,lcgvcea,;
lcgvasig,lcgvgrupo,lcgvtmov,lcgvteo,lcgvpra,lcgvtot,lcgvhor,lcgh
old,;
lcgvln,lcgvmar,lcgvmie,lcgvjue,lcgvvie,lcgvsab,lcgvcea,lcgflag
,;
lcgconta,lcgpalt,ptr,x,lcgres,lcgpan,lcgtot,lcgbajp,seguir,r_baj
a,;
lcgcmov,lcgvmov,seleold,lcgrev
RETURN

/*****
* MOV_PAUT()
*
* Funcion que realiza los movimientos al archivo de propuestas
automaticas
*****/
FUNCTION MOV_PAUT()
MEMVAR getlist,lcgcont1,lcgenpana,lcgtinte,lcgpunam,lcgcampo_p,lcgrfc,;
lcgkey,lcgrev,lcgalt,lcgbanh,lcghrs
lcgcont1 := .T.

SET KEY 19 TO ; SET KEY 4 TO

DO WHILE (lcgcont1)
  lcgalt := 0
  SET KEY K_CTRL_INS TO ALT_PAUT()
  SET CURSOR ON
  SET KEY K_F3 TO
  SET KEY K_CTRL_DEL TO
  SETCOLOR(lcgenpana)
  CLRSCR (5, 130)
  RESTSCREEN(0,0,24,79,lcgpunam)
  CUADRO(lcgtinte,8,11,21,76,"S")
  LCGSCREA()
  DECLARE LCGCAMPO P[3]
  LCGCAMPO_P[1] := "RFC"
  LCGCAMPO_P[2] := "FLAG"
  LCGCAMPO_P[3] := "HORARIO"
  SET KEY K_F2 TO COD_AYUDA
  lcgrfc := SPACE(13)

```

```

lcgkey := SPACE(15)
MENSAJEE (" <CTR-INS> DAR DE ALTA UN REGISTRO ",22,20)
@9,19 SAY "RFC DEL PROFESOR:" GET lcgRFC PICT "@"!
READ
IF (LASTKEY() = K_F2 .OR. lcgRFC = SPACE(13)).AND. LASTKEY() <> 27
  lcgrev := .T.
  lcgkey := BROWPRO("PAUT",LCGCAMPO_P)
  lcgRFC := SUBSTR(lcgkey,1,13)
ENDIF
RELEASE LCGCAMPO_P
SET KEY K_F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgRFC) <> 0)
  SELE COOR
  SEEK lcgRFC
  IF FOUND()
    lcgbanh := coor->banco_hora
    lcghrs := coor->hor_teo + coor->hor_pra
  SEEK lcgkey
  IF FOUND()
    lcgRFC := paut->rfc
    @9,37 SAY lcgRFC PICT "@"!
    CAM_PAUT()
  ENDIF
ENDIF
ELSE
  lcgconti = .F.
ENDIF
ENDDO
COMMIT
RETURN nil

/*****
* LCGSCREA()
*
* Función que despliega la pantalla de revisión
*****/
FUNCTION LCGSCREA()
@ 11, 14 SAY "Clave de Asignatura: "
@ 12, 14 SAY "Asignatura: "
@ 13, 14 SAY "Tipo de Movimiento: "
@ 14, 14 SAY "Grupo: "
@ 15, 14 SAY "Teoría: "
@ 16, 14 SAY "Práctica: "
@ 17, 14 SAY "Horario: "
RETURN nil

/*****
* CAM_PAUT()
*
* Función que realiza los cambios al archivo de propuestas automáticas
*****/
FUNCTION CAM_PAUT()
MEMVAR
getlist, lcgvcea, lcgvasig, lcgvgrupo, lcgvtmov, lcgvteo, lcgvptra, lcgvtot, ;
        lcgvhor, lcgshoid, lcgviun, lcgvmar, lcgvmie, lcgvvue, lcgvvie, lcgvsab, ;
        lcgvcea, lcgflag
LOCAL seguir
lcgvcea := paut->cve_asig
lcgvasig := paut->asignatura
lcgvgrupo := paut->grupo
lcgvtmov := paut->tipomov
lcgvteo := paut->teorica
lcgvpra := paut->practica
lcgvtot := paut->teorica+paut->practica

```

```

legvhor := paut->horario
legghold := legvhor
legvluun := paut->lunes
legvmar := paut->martes
legvmie := paut->miercoles
legvjue := paut->jueves
legvvie := paut->viernes
legvsab := paut->sabado
legflag := paut->flag
seguir = .T.
DO WHILE (seguir)
  @ 11,36 SAY legvcvea PICT "@!"
  @ 12,36 SAY legvasig PICT "@!"
  @ 13,36 SAY legvtmov PICT "@!"
  @ 14,36 SAY legvgrupo PICT "@!"
  @ 15,36 SAY legvteo PICT "@!"
  @ 16,36 SAY legvptra PICT "@!"
  @ 17,36 SAY legvhor PICT "@!"
  GRA PAUT()
  seguir := .F.
  IF LASTKEY() = K_ESC
    RETURN nil
  ENDDIF
ENDDO
RETURN nil

/*****
* ALT_PAUT()
*
* Función que realiza las altas al archivo de propuestas automaticas
*****/
FUNCTION ALT_PAUT()
MEMVAR getlist, legvtot, legvptra, legvteo, legvsab, legvvie, legvjue, legvmie, ;
legvmar, legvluun, legvhor, legvtmov, legvgrupo, legvasig, legvcvea, ;
leggrfc, legflag, legkey, legghold, legconta, legpalt, legalt, legbanh, leg
hrs
PUBLIC legvcvea := SPACE(4), legvasig := SPACE(32), ;
SPACE(40), ;
legvluun := SPACE(13), legvmar := SPACE(13), legvmie :=
SPACE(13), ;
legvjue := SPACE(13), legvvie := SPACE(13), legvsab :=
SPACE(13), ;
legvteo := 0, legvptra := 0, legvtot := 0, legflag :=
0, legghold := ' '
legconta := .T.
DO WHILE (legconta)
  legpalt = SAVESCREEN(0,0,24,79)
  @9,37 GET leggrfc PICT "@!"
  IF legalt = 0
    COD_AYUDA()
  ENDDIF
  READ
  @ 9,37 SAY leggrfc PICT "@!"
  SELE coor
  SEEK leggrfc
  IF !FOUND()
    ALERT("NO EXISTE EN EL CATALOGO DE PROFESORES")
    leggrfc:=SPACE(13)
    legalt = 1
    RETURN nil
  ELSE
    SET KEY K_CTRL_INS TO
    legbanh := coor->banco_hora

```

```

lcghrs := coor->hor_teo + coor->hor_pra
SELE paut
SEEK lcgRFC
IF FOUND()
DO WHILE paut->RFC=lcgRFC
SKIP
ENDDO
SKIP-1
lcgflag:=VAL(paut->flag)+1
ELSE
SELE pman
SEEK lcgRFC
IF FOUND()
DO WHILE pman->RFC=lcgRFC
SKIP
ENDDO
SKIP-1
lcgflag:=VAL(pman->flag)+1
ELSE
lcgflag:=1
ENDIF
ENDIF
lcgflag:=STR(lcgflag,2)
lcgkey :=lcgRFC+lcgflag
GRA_PAUT()
lcgconta:=-F.
ENDIF
ENDDO
RESTSCREEN(0,0,24,79,lcgpalt)
SET KEY K_CTRL_INS TO ALT_PAUT()
lcgRFC:=SPACE(13)
RETURN nil

/*****
* GRA_PAUT()
*
* Función que graba los movimientos al archivo de propuestas automáticas
*****/
FUNCTION GRA_PAUT()
#define T BORRAR 127
MEMVAR geE1:at,ptr,lcgRFC,lcgvcvea,x,lcgvasig,lcgvhor,lcghold,lcgtinte,;
lcgtes,lcgpan,lcgtinte2,lcgvgrupo,lcgvcmov,lcgvteo,lcgvpra,lcgvto
t,;
lcgvlun,lcgvmar,lcgvmie,lcgvjue,lcgvvie,lcgvsab,lcgkey,lcgflag,;
lcgbanh,lcghrs,lcgtot,lcgcmov,lcgvcmov,seleold,lcgrev
LOCAL seguir
SET KEY K_CTRL_INS TO
SET KEY K_F3 TO BAJ_PAUT()
SET KEY K_CTRL_DEL TO BAJ_PAUT()
seguir = T.
ptr := 1
o MIENTRAS NO ESTEN CORRECTOS LOS DATOS o NO SE BORRE EL REGISTRO
DO WHILE (seguir)
MEMWAJE (" <F3> c <CTR-DEL> BORRAR ",22,20)
o 9,37 SAY lcgRFC PICT "@!"
o 9,62 SAY "HRS_AUT : "
o 9,71 SAY lcgbanh PICT "###.#"
o10,62 SAY "HRS_ASIG: "
o10,71 SAY lcghrs PICT "###.#"
x=subs(lcgvcvea,1,1)
IF TYPE(x)="#N"
o 11,36 GET lcgvcvea PICT "@!"
o 12,36 GET lcgvasig PICT "@!"
ELSE
o 11,36 SAY lcgvcvea PICT "@!"

```

```

@ 12,36 SAY lcgvasig PICT "@!"
ENDIF
IF (LASTKEY() <> T_BORRAR .AND. LASTKEY() <> K_ESC)
DECLARE LCGCMOV(2)
LCGCMOV[1] := "CVE MOV"
LCGCMOV[2] := "MOVIMIENTO"
SET KEY K_F2 TO MOV_AYUDA
lcgvmov:=lcgvtmov
@ 13,36 CLEAR TO 13,50
@ 13,36 GET lcgvmov PICT "@!" VALID lcgvmov $
*01,02,03,04,05,06,07,08,09,10,11, *
READ
Seleold:=SELE()
SELE mov
IF (LASTKEY()=K_F2 .OR. lcgvmov=SPACE(2)) .AND. LASTKEY() <> 27
lcgrcv := .F
lcgvmov := BROWPRO("MOV",LCGCMOV)
ENDIF
SELE(seleold)
lcgvtmov:=lcgvmov
seleold := SELE()
SELE mov
SEEK lcgvtmov
IF FOUND()
@13,36 SAY mov->movimiento
ENDIF
SELE(seleold)
ENDIF
ENDIF
@ 14,36 GET lcgvgrupo PICT "@!"
@ 15,36 GET lcgvteo PICT "##.#"
@ 16,36 GET lcgvptra PICT "##.#"
@ 17,36 GET lcgvhor PICT "@!"
READ
IF lcgvhor <> lcghold
@ 18,14 SAY "Lunes:" " GET lcgvlun PICT "@!"
@ 18,45 SAY "Martes:" GET lcgvmar PICT "@!"
@ 19,14 SAY "Miercoles:" GET lcgvmie PICT "@!"
@ 19,45 SAY "Jueves:" GET lcgvjue PICT "@!"
@ 20,14 SAY "Viernes:" GET lcgvvie PICT "@!"
@ 20,45 SAY "Sabado:" GET lcgvaab PICT "@!"
READ
ENDIF
IF (LASTKEY() <> T_BORRAR .AND. LASTKEY() <> K_ESC)
lcgres = 1
lcgpan = SAVESCREEN(0,0,24,79)
CUADRO(lcgtinte2,15,30,18,50,"S")
@16,34 SAY " CORRECTO ?"
@17,32 PROMPT "SI"
@17,36 PROMPT "NO"
@17,39 PROMPT "CANCELAR"
MENU TO lcgres
RESTSCREEN(0,0,24,79,lcgpan)
SETCOLOR(lcgtinte)
DO CASE
CASE lcgres = 1
SELE prop
SET ORDER TO 1
SEEK lcgfic
lcgtot = lcgvteo + lcgvptra
IF FOUND()
DO WHILE prop->rfc=lcgrfc
IF prop->rfc + prop->flag = lcgkey
lcgtot = lcgtot
ELSE
lcgtot = lcgtot + prop->teorica + prop->practica

```

```

        ENDIF
        ENDDO
        SKIP
    ENDDO
ENDIF
*
IF lcgtot > lcgbanh
DO WHILE lcgtot > lcgbanh .AND. LASTKEY() <> 27
    seguir = .T.
    lcghrs = lcgtot
    ALERT("LAS HORAS ASIGNADAS SON MAYORES QUE LAS
AUTORIZADAS")
    *
    ELIMINAR LA SIGUIENTE LINEA
    @ 9,72 GET lcgbanh    PICT "###.#"
    READ
ENDDO
/* BORRRAR LAS OCHO LINEAS SIGUIENTES*/
SELE coor
SEEK lcgkey
IF FOUND()
    IF RLOCK()
        REPLACE coor->banco_hora WITH lcgbanh
    ENDIF
ENDIF
ELSE
    SELE prop
    SET ORDER TO 1
    SEEK lcgkey
    IF !FOUND()
        APPEND BLANK
    ENDIF
    IF RLOCK()
        REPLACE prop->rfc WITH lcgRFC
        REPLACE prop->cve_asig WITH lcgvcvcea
        REPLACE prop->asignatura WITH lcgvasig
        REPLACE prop->grupo WITH lcgvggrupo
        REPLACE prop->tipomov WITH lcgvtmov
        REPLACE prop->teorica WITH lcgvteo
        REPLACE prop->practica WITH lcgvpca
        REPLACE prop->horario WITH lcgvhor
        REPLACE prop->lunes WITH lcgvlun
        REPLACE prop->martes WITH lcgvmar
        REPLACE prop->miercoles WITH lcgvmie
        REPLACE prop->jueves WITH lcgvjue
        REPLACE prop->viernes WITH lcgvvie
        REPLACE prop->sabado WITH lcgvsab
        REPLACE prop->flag WITH lcgflag
    ENDIF
    SELE paut
    SEEK lcgkey
    IF !FOUND()
        APPEND BLANK
    ENDIF
    IF RLOCK()
        REPLACE paut->rfc WITH lcgRFC
        REPLACE paut->cve_asig WITH lcgvcvcea
        REPLACE paut->asignatura WITH lcgvasig
        REPLACE paut->grupo WITH lcgvggrupo
        REPLACE paut->tipomov WITH lcgvtmov
        REPLACE paut->teorica WITH lcgvteo
        REPLACE paut->practica WITH lcgvpca
        REPLACE paut->horario WITH lcgvhor
        REPLACE paut->lunes WITH lcgvlun
        REPLACE paut->martes WITH lcgvmar
        REPLACE paut->miercoles WITH lcgvmie
        REPLACE paut->jueves WITH lcgvjue
        REPLACE paut->viernes WITH lcgvvie
    ENDIF

```

```

                REPLACE paut->sabado      WITH lcgvsab
                REPLACE paut->flag        WITH lcgflag
ENDIF
SELE coor
SEEK lcgrfc
IF FOUND()
    IF RLOCK()
        REPLACE coor->propuesta WITH '*'
    ENDIF
ENDIF
seguir = .F.
ENDIF
CASE lcgres = 2
    seguir = .T.
CASE lcgres = 3 .OR. LASTKEY() = 27
    seguir = .F.
    ALERT(" LOS DATOS HAN SIDO CANCELADOS ")
    KEYBOARD (27)
ENDCASE
ELSE
    seguir = .F.
ENDIF
ENDDO
RETURN nil

/*****
* BAJ_PAUT()
*
* Función que da de bajas los registros del archivo de propuestas
* automáticas
*****/
FUNCTION BAJ_PAUT()
MEMVAR getllst,lcgbajp,lcgerror,lcgkey,lcgtinte
LOCAL r_baja
r_baja = 2
lcgbajp = SAVESCREEN(0,0,24,79)
CUADRO(lcgerror,15,25,18,60,"S")
@16,27 SAY " DESEA BORRAR ESTE REGISTRO ?"
@17,35 PROMPT "SI"
@17,45 PROMPT "NO"
MENU TO r_baja
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgbajp)
DO CASE
CASE r_baja = 1
    SELE paut
    IF RLOCK()
        DELE
        PACK
    ELSE
        ALERT("EL REGISTRO NO ESTA BLOQUEADO")
    ENDIF
    SELE prop
    SET ORDER TO 1
    SEEK lcgkey
    IF FOUND()
        IF RLOCK()
            DELE
            PACK
        ELSE
            ALERT("EL REGISTRO NO ESTA BLOQUEADO")
        ENDIF
    ENDIF
CASE r_baja = 2
    ALERT (" OPERACION CANCELADA ")

```

```
ENDCASE  
KEYBOARD CHR(27)+CHR(T_BORRAR)  
RETURN nil
```

```
*/ EOF()
```

```
/******
```

```
* MOV_AYUDA()
```

```
*
```

```
* Función que activa el ENTER
```

```
*****/
```

```
FUNCTION MOV_AYUDA
```

```
MEMVAR getlist,logmovvar
```

```
logmovvar=SPACE(2)
```

```
KEYBOARD CHR(K_ENTER)
```

```
RETURN nil
```

```

*****
* PROGRAM-ID:      Prof_Cnx.prg ver 1.0
* AUTHOR:         Leticia Cerd Garrido
* REMARKS:        Programa que contiene las funciones de comunicaci3n
*                 con los otros sistemas
* MODIFIED:       07 Julio, 1996.
*****
/*****
* BAALTPRO()
* Funci3n que verifica que exista un profesor y envia el rfc y el nombre
* si no existe presenta una pantalla de captura para darlo de alta
*****
#include "inkey.ch"
#include "grump.ch"
#include "fileio.ch"
#include "setcur.ch"
FUNCTION BAALTPRO(RFC)
MEMVAR getlist,lcginte,lcgrfc,lcgvcate,lcgvfunc,lcgvnomb,lcgphor,;
       seleold,oldsele,vm_dir_sistema, vm_dir_carrera,lcgruta,entrega,;
       vm_tit_carr,vm_anio_linea,vm_periодо_linea,lcgperiодо,lcgtinte2,;
       lcgcacVar,colorold,seggra,lcgzeus,lcgfunc,lcgtinte6,lcgcontinua,;
       lcgprocco,vm_inicial

PUBLIC vm_dir_sistema.
vm_dir_carrera,vm_tit_carr,vm_anio_linea,vm_inicial,;
       vm_periодо_linea,lcgperiодо,lcgruta,entrega,lcgcontinua,lcgprocco

entrega := {}
lcgrfc := rfc
lcgphor := SAVESCREEN(0,0,24,79)
colorold := SETCOLOR()
seleold := SELE()
lcginte := "W/B"//AZUL
lcgtinte2 := "W/BG+" //GRIS
lcgtinte6 := "W/N+GRGR+" //MARRON CLARO
lcgruta := vm_dir_carrera
lcgzeus := vm_dir_sistema
lcgprocco := "PC"+vm_inicial+lcgperiодо
*****
USE &lcgzeus\profcac ALIAS pro NEW SHARED
SET INDEX TO &lcgzeus\profcac.&lcgzeus\pronom
USE &lcgruta\lcgprocco ALIAS coor NEW SHARED
SET INDEX TO &lcgruta\coorrfc.&lcgruta\coornom
USE &lcgruta\cat INDEX &lcgruta\cat ALIAS cat NEW SHARED
*****
seggra := .T.
lcgcontinua := .T.
DO WHILE lcgcontinua
SET CURSOR ON
IF LASTKEY() <> 27
SELE coor
SET ORDER TO 1
SEEK lcgrfc
IF !FOUND()
SELE pro
SET ORDER TO 1
SEEK lcgrfc
IF !FOUND()
PUBLIC lcgvnomb := SPACE(32),lcgvfunc := SPACE(1)
MENSAJE ("<F2> AYUDA EN CATEGORIAS",22,25)
CUADRO(lcgtinte,10,15,16,62,"S")
@11,29 SAY "RFC DEL PROFESOR:"
@11,48 SAY lcgrfc PICT "@!"
IF LASTKEY()<>K_ESC
DO WHILE (seggra).AND.LASTKEY()<>27

```

```

@13,17 SAY "Nombre: "
@13,30 GET lcgvnomb PICT "@1"
@14,17 SAY "Funcionario: "
@14,30 GET lcgvfunc PICT "@1" VALID lcgvfunc $ "S,N,

CAT_PROF()
GRAP_HOR()
ENDDO
ENDIF
ELSE
lcvgnomb := pro->nom_pro
lcvgfunc := pro->funsio
MENSAJEE {"<F2> AYUDA EN CATEGORIAS ",23,25)
CUADRO(lcgtinte,10,15,16,62,"S")
@11,29 SAY "RFC DEL PROFESOR: "
@11,48 SAY lcvrffc PICT "@1"
IF LASTKEY()<>K_ESC
DO WHILE (asegra).AND.LASTKEY()<>27
@13,17 SAY "Nombre: "
@13,30 SAY lcvgnomb PICT "@1"
@14,17 SAY "Funcionario: "
IF lcvgfunc='*'
lcvgfunc := 'SI'
ELSE
lcvgfunc := 'NO'
ENDIF
@14,30 SAY lcvgfunc PICT "@1"
CAT_PROF()
GRAP_HOR()
ENDDO
ENDIF
ELSE
lcvgnomb:=coor->nom_pro
lcvcate:=coor->categoria
SELE pro
SEEK lcvrffc
IF !FOUND()
PUBLIC lcvgfunc := SPACE(1)
SELE coor
IF RLOCK()
REPLACE coor->flag WITH '*'
ENDIF
CUADRO(lcgtinte,10,15,16,62,"S")
@11,29 SAY "RFC DEL PROFESOR: "
@11,48 SAY lcvrffc PICT "@1"
IF LASTKEY()<>K_ESC
DO WHILE (asegra).AND.LASTKEY()<>27
@13,17 SAY "Nombre: "
@13,30 SAY lcvgnomb PICT "@1"
@14,17 SAY "Categoria: "
IF lcvcate#space(3)
oldsele := SELE()
SELE cat
SEEK lcvcate
IF FOUND()
@14,30 SAY cat->categoria
ENDIF
SELE(oldsele)
ENDIF
@15,17 SAY "Funcionario: "
@15,30 GET lcvgfunc PICT "@1" VALID lcvgfunc $ "S,N,

READ
GRAP_HOR()

```

```

                ENDDO
            ENDF
        ELSE
            AADD (ENTREGA, {lgrffc, lcgvnomb})
            CLOSE pro
            CLOSE coor
            CLOSE cat
            RELEASE
lgrffc, lcgvcate, lcgvfunc, lcgvnomb, lcgrev, lcgphor, seleold, ;
oldsele, lcgcatvar, colorold, seggra, lcgfunc, , lcgcontin
ua, ;
                lcgcatp, lcgpanta, lgrhor, lcgcat1, lcgcat2, lcgcatp, sel
eant
            RETURN (entrega)
        ENDF
    ENDF
    ENDF
        lcgcontinua := .F.
ENDDO
COMMIT
SELE (seleold)
SETCOLOR (colorold)
RESTSCREEN (0, 0, 24, 79, lcgphor)
IF LASTKEY() = 27
    entrega := nil
ENDIF
CLOSE pro
CLOSE coor
CLOSE cat
RELEASE
lgrffc, lcgvcate, lcgvfunc, lcgvnomb, lcgphor, lcgrev, seleold, oldsele, ;
lgrhor, lcgcat1, lcgcat2, lcgcatp, seleant
RETURN (entrega)

/*****
* CAT_PROF()
*
* Función que muestra un browse con las categorías de profesores
*****/
FUNCTION CAT_PROF()
MEMVAR getlist, lcgcatp, lcgcatvar, lcgvcate, oldsele
PUBLIC lcgvcate := SPACE(3)
@15, 17 SAY "Categoría: "
DECLARE LCGCATP[2]
LCGCATP[1] := "CVE CAT"
LCGCATP[2] := "CATEGORIA"
SET KEY K_F2 TO CAT_AY_H
lcgcatvar := lcgvcate
@15, 30 CLEAR TO 15, 31
@15, 30 GET lcgcatvar PICT "@!" ;
VALID lcgcatvar $ "ASA, ASB, AYA, AYB, CAA, CAB, CAC, TA , PH , "
READ
AY_CAT H()
RELEASE LCGCATP
SET KEY K_F2 TO
lcgvcate := lcgcatvar
oldsele := SELE()
SELE cat
SEEK lcgvcate
IF FOUND()
    @15, 30 SAY cat->categoria
ENDIF
SELE (oldsele)
RETURN nil

```

```

/*****
* GRAP_HOR()
*
* Función que graba los cambios al catalogo de profesores
*****/
FUNCTION GRAP_HOR()
MEMVAR
getlist,lcgcatvar,lcgrfc,lcgvfunc,lcgvcate,seleold,lcgpanta,entrega,;
lcgtinte,lcgvnomb,lcgrhor,lcgtinte2,lcgcat1,lcgcat2,lcgcatp,seggra
a

IF LASTKEY() <> 27
  lcgrhor := 1
  DO WHILE lcgrhor=1.OR.LASTKEY() = 27
    SETCOLOR(lcgtinte)
    lcgpanta = SAVESCREEN(0,0,24,79)
    CUADRO(lcgtinte2,16,50,19,65,"S")
    @17,52 SAY " CORRECTO ?"
    @18,51 PROMPT "SI"
    @18,54 PROMPT "NO"
    @18,57 PROMPT "CANCELAR"
    MENU TO lcgrhor
    RESTSCREEN(0,0,24,79,lcgpanta)
    SETCOLOR(lcgtinte)
    DO CASE
      CASE lcgrhor = 1
        SELE pro
        SELE pro
        SEEK lcgrfc
        IF !FOUND()
          APPEND BLANK
          REPLACE pro->rfc WITH lcgrfc
          REPLACE pro->nom_pro WITH lcgvnomb
          IF lcgvfunc='S'
            lcgvfunc:='.'
          ELSE
            lcgvfunc:=SPACE(1)
          ENDF
          REPLACE pro->funzio WITH lcgvfunc
          REPLACE pro->flag WITH '.'
        ENDIF
        SELE coor
        SEEK lcgrfc
        IF !FOUND()
          APPEND BLANK
          REPLACE coor->rfc WITH lcgrfc
          REPLACE coor->categoria WITH lcgvcate
          REPLACE coor->nom_pro WITH lcgvnomb
          REPLACE coor->flag WITH '.'
        ENDIF
        seggra := .F.
        lcgrhor := 3
        lcgcat1 := SUBSTR(lcgvcate,1,2)
        lcgcat2 := SUBSTR(lcgvcate,3,1)
        BAALTPR2(lcgrfc,lcgvnomb,lcgvfunc,lcgcat1,lcgcat2)
        AADD(ENTREGA,{lcgrfc,lcgvnomb})
      CASE lcgrhor = 2
        seggra := .T.
      CASE lcgrhor = 3 .OR. LASTKEY() = 27
        ENTREGA := NIL
        seggra := .F.
        KEYBOARD (27)
    ENDCASE
  ENDDO

```

```

ENDIF
RETURN(seggra)

/*****
* CAT_AY_H()
*
* Función que activa el ENTER
*****/
FUNCTION CAT_AY_H
MEMVAR getlist,lcgcatvar
lcgcatvar=SPACE(3)
KEYBOARD CHR(K_ENTER)
RETURN nil

/*****
* AY_CAT_H()
*
* Función que muestra un browse de las categorías
*****/
FUNCTION AY_CAT_H
MEMVAR getlist,seleant,lcgcatvar,lcgcatp,lcgrev
seleant:=SELE()
SELE cat
IF (LASTKEY()=K_F2 .OR. lcgcatvar=SPACE(3)) .AND. LASTKEY() <> 27
    lcgrev := .F.
    lcgcatvar := BROWPRO("CAT",LCGCATP)
ENDIF
SELE(seleant)
RETURN nil

/*****
* PROFCO()
*
* Función que envia el nombre de un profesor
*
*****/
FUNCTION PROFCO(RFC)
MEMVAR getlist,lcgrfc,lcgvnomb,seleold,vm_dir_sistema,vm_dir_carrera,;
entrega,vm_tit_carr,vm_anio_linea,vm_periодо_linea,lcgperiodo,;
lcgzeus,lcgcontinua,vm_inicial,lcgtinte,lcgphor,coloroid

PUBLIC vm_dir_sistema,
vm_dir_carrera,vm_tit_carr,vm_anio_linea,vm_inicial,;
vm_periодо_linea,lcgperiодо,entrega,lcgcontinua

entrega := {}
lcgrfc := rfc
lcgphor := SAVESCREEN(0,0,24,79)
coloroid := SETCOLOR()
seleold := SELE()
lcgtinte := "*/B"//AZUL
lcgzeus := vm_dir_sistema
USE &lcgzeus\profcaT INDEX &lcgzeus\prorfc ALIAS pro NEW SHARED

IF LASTKEY() <> 27
    SELE pro
    SEEK lcgrfc
    IF !FOUND()
        entrega := nil
    ELSE
        AADD(ENTREGA,{lcgrfc,lcgvnomb})
    ENDIF
ENDIF
SELE(seleold)

```

```

SETCOLOR(colorold)
RESTSCREEN(0,0,24,79,lcgphor)
IF LASTKEY() = 27
  entrega := nil
ENDIF
CLOSE pro
RETURN(entrega)

/*****
* PROF.DBF()
* Función que verifica que existan los archivos necesarios para
* el proceso
*****/
FUNCTION PROF.DBF()
MEMVAR getlist,lcgeva,lcgmat,lcgprocoo,vm_dir_sistema,vm_dir_carrera,
vm_tit_carr,vm_ano_linea,vm_periodo_linea,lcgperiodo,vm_inicial,
;
lcgzeus,lcgruta,lcgprocoo,lcgproaut,lcgproman,lcgprorev,lcgamy
FIELD
CVE MOV,CVE CAT,CVE MAT,CLAVE,GRUPO,PROFESOR,FLAG,RFC,TIPOMOV,NOM_PRO
LOCAL lcgnoe,thearray,lcgi,thefields,lcgano,lcganno,lcgsem,lcgany,;
lcgsem,lcgntx
PUBLIC vm_dir_sistema,vm_dir_carrera,vm_tit_carr,vm_ano_linea,lcgmat,;
vm_periodo_linea,lcgperiodo,vm_inicial,lcgzeus,lcgruta,lcgeva,;
lcgprocoo,lcgproaut,lcgproman,lcgprorev,lcgamy

/*****
* VARIABLES PUBLICAS
*****/
lcgperiodo := alltrim(vm_ano_linea)
lcgperiodo := substr(lcgperiodo,len(lcgperiodo)-1,2)
lcgperiodo := lcgperiodo + alltrim(vm_periodo_linea)
lcgruta := vm_dir_carrera
lcgzeus := vm_dir_sistema
lcgprocoo := 'PC'+vm_inicial+lcgperiodo
lcgproaut := 'PA'+vm_inicial+lcgperiodo
lcgproman := 'PM'+vm_inicial+lcgperiodo
lcgprorev := 'PR'+vm_inicial+lcgperiodo
lcgeva := 'PE'+vm_inicial+lcgperiodo
lcgmat := 'DMC'+vm_inicial
lcgnoe := 0
lcgano := ALLTRIM(vm_ano_linea)
lcganno := VAL(SUBSTR(lcgano,3,2))
lcgsem := VAL(ALLTRIM(vm_periodo_linea))
lcgany:=0; lcgsem:=0; lcgnoe=0
IF STR(lcgsem,1)='1'
  lcgany :=STR(lcganno-1,2)
  lcgsem:=STR(lcgsem+1,1)
  lcgany :='PC'+vm_inicial+lcgany+lcgsem
ELSE
  lcgsem:=STR(lcgsem-1,1)
  lcgany :='PC'+vm_inicial+STR(lcganno,2)+lcgsem
ENDIF

IF !FILE("&lcgzeus\profcats.dbf")
ALERT ("NO EXISTE*" + "&PROFCAT.DBF") ; lcgnoe=1;ENDIF
IF !FILE("&lcgruta&lcgprocoo*.dbf")
  USE &lcgruta&lcgamy ALIAS any NEW EXCL
  COPY TO &lcgruta&lcgprocoo
  CLOSE any
ENDIF
IF !FILE("&lcgruta&lcgproaut*.*.dbf")
TheFields:=PROF_AMR()
SET DEFA TO &lcgruta
DBCREATE("&lcgproaut",TheFields)

```

```

RELEASE TheFields
ENDIF
IF !FILE("&lcgruta\&lcgpproman"+"&.dbf")
  TheFields:=PROF_AMR()
  SET DEFA TO &lcgruta
  DCREATE("&lcgpproman",TheFields)
  RELEASE TheFields
ENDIF
IF !FILE("&lcgruta\&lcgprorev"+"&.dbf")
  TheFields:=PROF_AMR()
  SET DEFA TO &lcgruta
  DCREATE("&lcgprorev",TheFields)
  RELEASE TheFields
ENDIF
IF !FILE("&lcgruta\&lcgmat"+"&.dbf")
  ALERT ("NO EXISTE"+" "+"&lcgmat.DBF") ; lcgnoc=1 ; ENDIF
IF lcgnoc=1
  ALERT ("EL PROCESO NO CUENTA CON LA INFORMACION NECESARIA");RETURN
nil;ENDIF
IF !FILE("&lcgruta\&lcgeva"+"&.dbf")
  TheFields := {}
  AADD(TheFields, {"PROFESOR", "C", 30, 0}) && 1
  AADD(TheFields, {"RRC", "N", 13, 0}) && 2
  AADD(TheFields, {"GRUPO", "C", 4, 0}) && 3
  AADD(TheFields, {"MATERIA", "C", 20, 0}) && 4
  AADD(TheFields, {"CLAVE", "C", 4, 0}) && 5
  AADD(TheFields, {"SEMESTRE", "C", 4, 0}) && 6
  AADD(TheFields, {"A01", "N", 2, 0}) && 7
  AADD(TheFields, {"A02", "N", 2, 0}) && 8
  AADD(TheFields, {"A03", "N", 2, 0}) && 9
  AADD(TheFields, {"A04", "N", 2, 0}) && 10
  AADD(TheFields, {"A05", "N", 2, 0}) && 11
  AADD(TheFields, {"A06", "N", 2, 0}) && 12
  AADD(TheFields, {"A07", "N", 2, 0}) && 13
  AADD(TheFields, {"A08", "N", 2, 0}) && 14
  AADD(TheFields, {"A09", "N", 2, 0}) && 15
  AADD(TheFields, {"A10", "N", 2, 0}) && 16
  AADD(TheFields, {"A11", "N", 2, 0}) && 17
  AADD(TheFields, {"A12", "N", 2, 0}) && 18
  AADD(TheFields, {"A13", "N", 2, 0}) && 19
  AADD(TheFields, {"P01", "N", 2, 0}) && 20
  AADD(TheFields, {"P02", "N", 2, 0}) && 21
  AADD(TheFields, {"P03", "N", 2, 0}) && 22
  AADD(TheFields, {"P04", "N", 2, 0}) && 23
  AADD(TheFields, {"P05", "N", 2, 0}) && 24
  AADD(TheFields, {"P06", "N", 2, 0}) && 25
  AADD(TheFields, {"P07", "N", 2, 0}) && 26
  AADD(TheFields, {"P08", "N", 2, 0}) && 27
  AADD(TheFields, {"P09", "N", 2, 0}) && 28
  AADD(TheFields, {"P10", "N", 2, 0}) && 29
  AADD(TheFields, {"P11", "N", 2, 0}) && 30
  AADD(TheFields, {"P12", "N", 2, 0}) && 31
  AADD(TheFields, {"P13", "N", 2, 0}) && 32
  AADD(TheFields, {"P14", "N", 2, 0}) && 33
  AADD(TheFields, {"P15", "N", 2, 0}) && 34
  AADD(TheFields, {"P16", "N", 2, 0}) && 35
  AADD(TheFields, {"P17", "N", 2, 0}) && 36
  AADD(TheFields, {"COMENTARIO", "C", 60, 0}) && 37
  SET DEFA TO &lcgruta
  DCREATE("&lcgeva",TheFields)
  RELEASE TheFields
ENDIF
IF !FILE("&lcgruta\cat.dbf")

```

```

TheFields := {}
ADD(TheFields, {"CVE CAT", "C",03,0 } ) && 1
AADD(TheFields, {"CATEGORIA", "C",24,0 } ) && 2
SET DEFA TO &lcgruta
DBCREATE ("CAT",TheFields)
TheArray := {}
AADD(TheArray, {"ASA", "PROFESOR ASIGNATURA 'A'" } ) && 1
AADD(TheArray, {"ASB", "PROFESOR ASIGNATURA 'B'" } ) && 2
AADD(TheArray, {"AYA", "AYUDANTE DE PROFESOR 'A'" } ) && 3
AADD(TheArray, {"AYB", "AYUDANTE DE PROFESOR 'B'" } ) && 4
AADD(TheArray, {"CAA", "PROFESOR DE CARRERA 'A'" } ) && 5
AADD(TheArray, {"CAB", "PROFESOR DE CARRERA 'B'" } ) && 6
AADD(TheArray, {"CAC", "PROFESOR DE CARRERA 'C'" } ) && 7
AADD(TheArray, {"TA", "TECNICOS ACADEMICOS " } ) && 8
AADD(TheArray, {"PH", "PERSONAL POR HONORARIOS " } ) && 9
USE &lcgruta\cat ALIAS cat NEW EXCL
FOR lcgi = 1 TO LEN(TheArray)
APPEND BLANK
REPLA cve cat WITH TheArray[lcgi,1] , categoria WITH
TheArray[lcgi,2]
NEXT
CLOSE cat
RELEASE thefields,thearray
ENDIF

IF !FILE("&lcgruta\tipomov.dbf")
TheFields := {}
ADD(TheFields, {"CVE MOV", "C",02,0 } ) && 1
ADD(TheFields, {"MOVIMIENTO", "C",20,0 } ) && 2
SET DEFA TO &lcgruta
DBCREATES ("TIPOMOV",TheFields)
TheArray := {}
AADD(TheArray, {"01", "NUEVO INGRESO " } ) && 1
AADD(TheArray, {"02", "REINGRESO " } ) && 2
AADD(TheArray, {"03", "PRORROGA " } ) && 3
AADD(TheArray, {"04", "OTRO NOMBRAMIENTO " } ) && 4
AADD(TheArray, {"05", "AUMENTO DE HORAS " } ) && 5
AADD(TheArray, {"06", "PROMOCION " } ) && 6
AADD(TheArray, {"07", "FIN DE INTERINATO " } ) && 7
AADD(TheArray, {"08", "DISMINUCION DE HORAS " } ) && 8
AADD(TheArray, {"09", "RENUNCIA " } ) && 9
AADD(TheArray, {"10", "DEFUNCION " } ) && 10
AADD(TheArray, {"11", "RESCISION " } ) && 11
USE &lcgruta\tipomov ALIAS mov NEW EXCL
FOR lcgi = 1 TO LEN(TheArray)
APPEND BLANK
REPLA cve mov WITH TheArray[lcgi,1] , movimiento WITH
TheArray[lcgi,2]
NEXT
CLOSE mov
RELEASE thefields,thearray
ENDIF

lcgntx=0
IF !FILE("&lcgzeus\prorfc.ntx") ;lcgntx=1 ;ENDIF
IF !FILE("&lcgzeus\pronon.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
SET DEFA TO &lcgzeus
USE &lcgzeus\profcat NEW EXCL
INDEX ON rfc TO &lcgzeus\prorfc
INDEX ON nom_pro TO &lcgzeus\pronon
CLOSE DATAB
ENDIF
SET DEFA TO &lcgruta
lcgntx=0

```

```

IF !FILE("&lcgruta\coorrfc.ntx") ;lcgntx=1 ;ENDIF
IF !FILE("&lcgruta\coornom.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\&lcgprocoo NEW EXCL
  INDEX ON rfc TO &lcgruta\coorrfc
  INDEX ON nom_pro TO &lcgruta\coornom
  CLOSE DATAB
ENDIF
lcgntx = 0
IF !FILE("&lcgruta\proprfc.ntx") ;lcgntx=1 ;ENDIF
IF !FILE("&lcgruta\proptip.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\&lcgprorev ALIAS prop NEW EXCL
  INDEX ON rfc+flag TO &lcgruta\proprfc
  INDEX ON rfc+tipomov TO &lcgruta\proptip
  CLOSE prop
ENDIF
lcgntx = 0
IF !FILE("&lcgruta\autrfc.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\&lcgproaut ALIAS paut NEW EXCL
  INDEX ON rfc+flag TO &lcgruta\autrfc
  CLOSE paut
ENDIF
lcgntx = 0
IF !FILE("&lcgruta\manrfc.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\&lcgproman ALIAS pman NEW EXCL
  INDEX ON rfc+flag TO &lcgruta\manrfc
  CLOSE pman
ENDIF
lcgntx = 0
IF !FILE("&lcgruta\eva.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\&lcgeva ALIAS eva NEW EXCL
  INDEX ON profesor+grupo+clave TO &lcgruta\eva
  CLOSE eva
ENDIF
lcgntx = 0
IF !FILE("&lcgruta\mat.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\&lcgmat ALIAS mat NEW EXCL
  INDEX ON cve_mat TO &lcgruta\mat
  CLOSE mat
ENDIF
lcgntx = 0
IF !FILE("&lcgruta\cat.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\cat ALIAS cat NEW EXCL
  INDEX ON cve_cat TO &lcgruta\cat
  CLOSE cat
ENDIF
lcgntx = 0
IF !FILE("&lcgruta\mov.ntx") ;lcgntx=1 ;ENDIF
IF lcgntx = 1
  USE &lcgruta\&tipomov ALIAS mov NEW EXCL
  INDEX ON cve_mov TO &lcgruta\mov
  CLOSE mov
ENDIF
RELEASE lcgnoe,lcgi,lcgntx,lcgpcoo
RETURN nil

FUNCTION PROF_AMR()
LOCAL TheFields := {}
AADD(TheFields,{"RFC", "C",13,0} ) && 1

```

```

AADD(TheFields {"CVE ASIG", "C",4 ,0 } ) && 2
AADD(TheFields {"ASIGNATURA", "C",32,0 } ) && 3
AADD(TheFields {"GRUPO", "C",4 ,0 } ) && 4
AADD(TheFields {"TIPOMOV", "C",2 ,0 } ) && 5
AADD(TheFields {"TEORICA", "N",4 ,1 } ) && 6
AADD(TheFields {"PRACTICA", "N",4 ,1 } ) && 7
AADD(TheFields {"HORARIO", "C",40,0 } ) && 8
AADD(TheFields {"LUNES", "C",13,0 } ) && 9
AADD(TheFields {"MARTES", "C",13,0 } ) && 10
AADD(TheFields {"MIERCOLES", "C",13,0 } ) && 11
AADD(TheFields {"JUEVES", "C",13,0 } ) && 12
AADD(TheFields {"VIERNES", "C",13,0 } ) && 13
AADD(TheFields {"SABADO", "C",13,0 } ) && 14
AADD(TheFields {"FLAG", "C",2 ,0 } ) && 15
RETURN (TheFields)

/*****
* PROF_NTX()
* Función que regenera los indices de todos los archivos necesarios
* para el funcionamiento del sistema
*****/
FUNCTION PROF_NTX()
MEMVAR getlist,vm_dir_sistema,
vm_dir_carrera,vm_tit_carr,vm_anio_linea,
vm_periodo_linea,lcgperiodo,vm_inicial,lcgseus,lcgruta,lcgeva,lcg
mat,; lcgprocoo,lcgproaut,lcgproman,lcgprorev

FIELD
CVE_MAT,TIPOMOV,RFC,NOM_PRO,FLAG,CVE_MOV,CVE_CAT,CLAVE,GRUPO,PROFESOR

PUBLIC vm_dir_sistema, vm_dir_carrera,vm_tit_carr,vm_anio_linea,lcgmat,;
vm_periodo_linea,lcgperiodo,vm_inicial,lcgseus,lcgruta,lcgeva,;
lcgprocoo,lcgproaut,lcgproman,lcgprorev

lcgperiodo := alltrim(vm_anio_linea)
lcgperiodo := substr(lcgperiodo,len(lcgperiodo)-1,2)
lcgperiodo := lcgperiodo + alltrim(vm_periodo_linea)
lcgruta := vm_dir_carrera
lcgseus := vm_dir_sistema
lcgprocoo := 'PC'+vm_inicial+lcgperiodo
lcgproaut := 'PA'+vm_inicial+lcgperiodo
lcgproman := 'PM'+vm_inicial+lcgperiodo
lcgprorev := 'PR'+vm_inicial+lcgperiodo
lcgeva := 'PE'+vm_inicial+lcgperiodo
lcgmat := 'DMC'+vm_inicial
SET DEFA TO &lcgseus
CLOSE DATAB

USE &lcgseus\profcat NEW EXCL
INDEX ON rfc TO &lcgseus\profrfc
INDEX ON nom_pro TO &lcgseus\pronom
CLOSE DATAB

SET DEFA TO &lcgruta
CLOSE DATAB
USE &lcgruta\lcgprocoo NEW EXCL
INDEX ON rfc TO &lcgruta\coorffc
INDEX ON nom_pro TO &lcgruta\coonom
USE &lcgruta\lcgeva ALIAS eva NEW EXCL
INDEX ON profesor+grupo+clave TO &lcgruta\eva
USE &lcgruta\cat ALIAS cat NEW EXCL
INDEX ON cve_cat TO &lcgruta\cat
USE &lcgruta\tipomov ALIAS mov NEW EXCL

```

```

INDEX ON cve_mov TO &lcgruta\mov
USE &lcgruta\&lcgproaut ALIAS paut NEW EXCL
INDEX ON rfc+flag TO &lcgruta\autrfc
USE &lcgruta\&lcgproman ALIAS pman NEW EXCL
INDEX ON rfc+flag TO &lcgruta\manrfc
USE &lcgruta\&lcgprorev ALIAS prop NEW EXCL
INDEX ON rfc+flag TO &lcgruta\proprfc
INDEX ON rfc+tipomov TO &lcgruta\proptip
USE &lcgruta\&lcgmat ALIAS mat NEW EXCL
INDEX ON cve_mat TO &lcgruta\mat
CLOSE DATAB
RETURN nil

```

```

*****
**

```

```

FUNCTION HORA RFC HOEX(RFC,ANIO,HPERIODO)
MEMVAR getlist,vfileh,vm_dir_carrera,entrega,vm_inicial,year,seleold
PUBLIC ENTREGA:= {}
seleold := SELE()
year := substr(ano,3,2)
vfileh := 'HQ'+SUBSTR(vm_inicial,1,2)+year+hperido
USE &lcgruta\&vfileh ALIAS hor NEW SHARED
INDEX ON hor->rfc_tit TO hor
SEEK rfc
IF FOUND()
DO WHILE hor->rfc_tit = rfc
AADD(ENTREGA,{hor->SALON,hor->HORARIO,hor->CVE_MAT,hor->CUPREAL,;
hor->GRUPO,hor->HRB,hor->RFC_TIT,hor->RFC_SUP})
SKIP
ENDDO
ENDIF
SELE(seleold)
CLOSE HOR
RETURN(ENTREGA)

```

```

FUNCTION BAALTPR2(lcgrfc,lcgvnomb,lcgvfunc,lcgcat1,lcgcat2)
LOCAL x
@18,20 SAY lcgrfc
@19,20 SAY lcgvnomb
@20,20 SAY lcgvfunc
@21,20 SAY lcgcat1
@22,20 SAY lcgcat2
FOR x=1 TO 30000
NEXT
RETURN nil

```

```

FUNCTION BAPRHOR(RFC)
MEMVAR getlist,vfileb,ENTREGA,key,seleold
PUBLIC ENTREGA:= {}
/*seleold := SELE()
key := rfc
vfileb := 'BANCOOR'
USE &lcgruta\&vfileb ALIAS ban NEW EXCL
INDEX ON rfc TO ban
SEEK key
IF FOUND()
AADD(ENTREGA,{ban->banco_hora,ban->hor_o_jef,;
ban->area_o_jef,ban->cat_o_jef,;
ban->hor_teo_an,ban->hor_pra_an,;
ban->hor_teo,ban->hor_pra})
ENDIF

```

```

CLOSE ban
SELE (seleold)*/
ENTREGA:=NIL
RETURN(ENTREGA)

FUNCTION COPRTCA(DBF,NTX,COLOR,TIPO)
local linserta := set(29)
local ocapta := getnew()
local colant := col()
local rowant := row()
local caracter,punto,colorant,curant,ObjBrow,punto2,cadbus
local columna,region,carac,salgo,letrero,cvar,cont,saveins,panta
local r_f_c, OBJCOLOR
MEMVAR CC STANDAR,Vm_Dir_Carrera,Vm_Anio_Linea,Vm_Periodo_Linea
*****
CC STANDAR := 'w/B'
*****
setblink(.f.)
IF TIPO = NIL
TIPO := .T.
ENDIF
IF TIPO
IF DBF = NIL .OR. LEN(NTX) = 0
RETURN(NIL)
ENDIF
IF !ABRREDBF(DBF,NTX,.F.,'pro_fe')
RETURN(NIL)
ENDIF
ELSE
DBF := Vm_dir_Carrera+'\PRO'
NTX := Vm_dir_Carrera+'\MON'
if
file(alltrim(Vm_Dir_Carrera)+'\rac'+substr(Vm_Anio_Linea,Len(Vm_Anio_Linea)-1,2)+':
Vm_Periodo_Linea'.act')
DBF := DBF + '2'
NTX := NTX + '2'
endif
DBF := DBF + substr(Vm_Anio_Linea,Len(Vm_Anio_Linea)-1,2)
NTX := NTX + substr(Vm_Anio_Linea,Len(Vm_Anio_Linea)-1,2)
DBF := DBF + Vm_Periodo_Linea
NTX := NTX + Vm_Periodo_Linea
IF !ABRREDBF(DBF,dbf,.F.,'PRO_fe',.t.)
RETURN NIL
ENDIF
ENDIF
set order to 2

colorant := setcolor()
IF COLOR = NIL
OBJCOLOR = '15/s,8/15'
ELSE
OBJCOLOR = COLOR
ENDIF
panta := savescreen()
setcolor(OBJCOLOR)
saveins := set(29)
if linserta
curant := setcursor(3)
else
curant := setcursor(1)
endif
* 9,9 clear to maxrow()-2,maxcol()-9
* 9,9 to maxrow()-2,maxcol()-9 double color(OBJCOLOR)

```

```

ObjBrow:pageup()
ocapta:undo()
renglon := row()
columna := col()
case character = 6
ObjBrow:gotobottom()
ocapta:undo()
renglon := row()
columna := col()
case character = 1
ObjBrow:gotop()
ocapta:undo()
renglon := row()
columna := col()
case character = 27
salgo := .f.
punto := 0
case character = 21
ocapta:undo()
renglon := row()
columna := col()
case character = 22
linserta := .not. linserta
set(29, linserta)
if set(29)
setcursor(3)
else
setcursor(1)
endif
case character = 8
ocapta:backspace()
renglon := row()
columna := col()
case character = 7
ocapta:delete()
renglon := row()
columna := col()
otherwise
if character >= 32 .and. character <= 255
carac := chr(character)
if set(29)
ocapta:insert(carac)
else
ocapta:overstrike(carac)
endif
renglon := row()
columna := col()
cadbus := alltrim(ocapta:buffer)
ObjBrow:GoTopblock:={||salto(cadbus,ocapta,@renglon,@columna
)}
ObjBrow:gotop()
ObjBrow:GoTopblock:={||salto2()}
endif
endcase
enddo
ocapta:reset()
ocapta:killfocus()
restscreen(...,panta)
set(29,saveins)
setcursor(curant)
setcolor(colorant)
@ colant,rowant say ,,
if punto <> 0
goto punto
r_f_c := pro_fe->rfc

```

```

cuadro(9,9,maxrow()-2,maxcol()-9)
*sombra(9,9,maxrow()-2,maxcol()-9)

// DEFINICION DE TITULOS EN EL OBJETO BROWSE
@ 9.12 say " Catalogo de Profesores "
@ MAXROW()-2,MAXCOL()-60 SAY " AD: Seleccionar, Esc: Regresar "
@ MAXROW()-3,MAXCOL()-30 SAY " Busqueda Rapida "
sela pro_fe
ObjBrow := Throwsdb(10,10,maxrow()-4,maxcol()-10)
ObjBrow:adddcolumn(TBColumnNew('R.F.C.',{|| transform(pro_fe -> rfc,@R
!|!|-99999!|!|)})
if tipo
  ObjBrow:adddcolumn(TBColumnNew('Nombre',{|| pro_fe -> nom_pro}))
else
  ObjBrow:adddcolumn(TBColumnNew('Nombre',{|| pro_fe -> nombre}))
endif
ObjBrow:headsep := 'AAA'
ObjBrow:colsep := ' '
ObjBrow:footsep := 'AAA'
saigo := .t.
letrero := space(32)
ocapta:row := maxrow()-3
ocapta:col := 15

ocapta:picture := replicate('!',32)
ocapta:block := {|p| if(pcount()==0,letrero,cvar:=p)}
ocapta:setfocus()
for cont := ocapta:pos to 0
  ocapta:right()
next
renlon := row()
coluna := col()
ObjBrow:colorspec := OBJCOLOR
do while saigo
  ObjBrow:forcastable()
  @ renlon,coluna say ,,
  caracter := - aspera(1,maxcol()-9,30,.t.,CC_STANDAR)
  punto := recno()
  do case
    case caracter = 13
      saigo := .f.
    case caracter = 5
      ObjBrow:up()
      ocapta:undo()
      renlon := row()
      columna := col()
    case caracter = 24
      ObjBrow:down()
      ocapta:undo()
      renlon := row()
      columna := col()
    case caracter = 19
      ocapta:left()
      renlon := row()
      columna := col()
    case caracter = 4
      ocapta:right()
      renlon := row()
      columna := col()
    case caracter = 3
      ObjBrow:pagedown()
      ocapta:undo()
      renlon := row()
      columna := col()
    case caracter = 18

```

```
else
  r_f_c := nil
endif
*cierradb('pro_fe')
close pro_fe
if r_f_c <> nil
  keyboard chr(13)
endif
RETURN(r_f_c)

function salto2()
sele pro_fe
go top
return nil

function salto(cadbus,ocapta, renglon,columna)
local n_reg
sele pro_fe
n_reg := recno()
seek cadbus
if ifound()
  ocapta:backspace()
  renglon := row()
  columna := col()
  goto n_reg
endif
return nil
```

```

*****
* PROGRAM-ID:      Prof_For.prg ver 1.0      llamado por Prof_Pro.prg
* AUTHOR:         Leticia Cerd Garrido
* REMARKS:        Programa que Realiza el Formato de la Propuesta
* LIBERADO:       para la Contratacion de Profesores
*                 19 Enero, 1997.
*****
* MODIFICO : ERNESTO PEVALOZA ROMERO
* ULTIMA MODIFICACION: 23 DE ENERO DE 1997
*****
MEMVAR getlist, little, incia, vm tit_carr, lcgren, lcgccl, copia, lcgcopia, ;
lcgruta, lcgzseus, lcgrfcp, lcgjefe

LOCAL
lcgfecha, lcgcfecha, lcgmov, lcgccl, lcgccl, lcgccl, lcgccl, lcgmesc, lcgmesc, ;
lcgkey, lcgkey1, lcgmesc, lcgccl, lcgccl, lcgccl, lcgccl, lcgccl, ;
lcgmesc, lcgccl, lcgmesc, lcgccl, lcgccl, lcgccl, lcgccl, ;
lcgccl, cont, lcgccl

USE &lcgzseus\profcat INDEX &lcgzseus\profic ALIAS pro NEW SHARED
USE &lcgruta\cat INDEX &lcgruta\cat ALIAS cat NEW SHARED

lcgfecha=DATE()
lcgcfecha=DTOC(lcgfecha)
SELE prop
SET ORDER TO 2
GO TOP
lcgmov= ;lcgccl= ;lcgccl= ;lcgccl= ;lcgmesc= ;
lcgmesc= ;lcgmesc= ;
lcgjefe='ING. JUAN GASTALDI PEREZ'
SET CURSOR OFF
SET CONSOLE OFF
SET DEVICE TO PRINT
SET PRINT ON
BO, 0 SAY LITTLE
SEEK lcgrfcp
IF FOUND()
DO WHILE prop->rfc=lcgrfcp
lcgkey=prop->rfc
lcgkey1=prop->rfc+prop->tipomov
lcgmesc=substr(lcgfecha, 4, 2)
DO CASE
Febrero CASE lcgmesc='01';lcgmesc=' Enero' ;CASE lcgmesc='02';lcgmesc='
Abril' CASE lcgmesc='03';lcgmesc=' Marzo' ;CASE lcgmesc='04';lcgmesc='
Junio' CASE lcgmesc='05';lcgmesc=' Mayo' ;CASE lcgmesc='06';lcgmesc='
Agosto' CASE lcgmesc='07';lcgmesc=' Julio' ;CASE lcgmesc='08';lcgmesc='
Octubre' CASE lcgmesc='09';lcgmesc='Septiembre';CASE lcgmesc='10';lcgmesc='
Diciembre' CASE lcgmesc='11';lcgmesc=' Noviembre';CASE lcgmesc='12';lcgmesc='
ENDCASE
DO CASE
CASE prop->tipomov='01';lcgmov=' NUEVO INGRESO'
CASE prop->tipomov='02';lcgmov=' REINGRESO'
CASE prop->tipomov='03';lcgmov=' PRORROGA'
CASE prop->tipomov='04';lcgmov=' OTRO NOMBRAMIENTO'
CASE prop->tipomov='05';lcgmov=' AUMENTO DE HORAS'
CASE prop->tipomov='06';lcgmov=' PROMOCION'
CASE prop->tipomov='07';lcgmov='FIN DE INTERINATO'
CASE prop->tipomov='08';lcgmov='DISMINUCION DE HRS.'
CASE prop->tipomov='09';lcgmov='RENUNCIA'
CASE prop->tipomov='10';lcgmov='DEFUNCION'

```

```

CASE prop->tipomov="11";lcmov="RESCISION"
ENDCASE
SELE pro
SEEK lcgkey
IF FOUND()
  SELE coor
  SEEK lcgkey
  IF FOUND()
    lcgkcat=coor->categoria ; lcgkcatj=coor->cat_o_jef
    lcgkcatd=coor->cat_o_dep
    SELE cat
    SEEK lcgkcat
    IF FOUND() ; lcgcat=cat->categoria ; ENDIF
    SEEK lcgkcatj
    IF FOUND() ; lcgcatj=cat->categoria ; ENDIF
    SEEK lcgkcatd
    IF FOUND() ; lcgcatd=cat->categoria ; ENDIF
    SELE coor
    lcgfini=DTOC(coor->fe_inicio)
    lcgmesi=SUBSTR(lcgfini,4,2)
    DO CASE
      CASE lcgmesi='01';lcmesic='ENE';CASE
lcmesi='02';lcmesic='FEB'
      CASE lcgmesi='03';lcmesic='MAR';CASE
lcmesi='04';lcmesic='ABR'
      CASE lcgmesi='05';lcmesic='MAY';CASE
lcmesi='06';lcmesic='JUN'
      CASE lcgmesi='07';lcmesic='JUL';CASE
lcmesi='08';lcmesic='AGO'
      CASE lcgmesi='09';lcmesic='SEP';CASE
lcmesi='10';lcmesic='OCT'
      CASE lcgmesi='11';lcmesic='NOV';CASE
lcmesi='12';lcmesic='DIC'
    ENDCASE
    lcgflim=DTOC(coor->fe_limite)
    lcgmesl=SUBSTR(lcgflim,4,2)
    DO CASE
      CASE lcgmesl='01';lcmeslc='ENE';CASE
lcmesl='02';lcmeslc='FEB'
      CASE lcgmesl='03';lcmeslc='MAR';CASE
lcmesl='04';lcmeslc='ABR'
      CASE lcgmesl='05';lcmeslc='MAY';CASE
lcmesl='06';lcmeslc='JUN'
      CASE lcgmesl='07';lcmeslc='JUL';CASE
lcmesl='08';lcmeslc='AGO'
      CASE lcgmesl='09';lcmeslc='SEP';CASE
lcmesl='10';lcmeslc='OCT'
      CASE lcgmesl='11';lcmeslc='NOV';CASE
lcmesl='12';lcmeslc='DIC'
    ENDCASE
    FOR copia=1 TO lcgcopia
      @lcgren+1,lcgcol+45 SAY "ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES ARAGON"
      @lcgren+2,lcgcol+57 SAY "UNIDAD ADMINISTRATIVA"
      @lcgren+3,lcgcol+44 SAY "JEFATURA DE CARRERA DE "
      @lcgren+3,lcgcol+67 SAY vm tit carr
      @lcgren+3,lcgcol+44 SAY "JEFATURA DE CARRERA DE "
      @lcgren+3,lcgcol+67 SAY vm tit carr
      @lcgren+4,lcgcol+5 SAY "0"
      @lcgren+4,lcgcol+6 SAY REPLICATE("A",124)
      @lcgren+4,lcgcol+130 SAY "2" ; @lcgren+5,lcgcol+5 SAY ""
      @lcgren+5,lcgcol+8 SAY "NOMBRE: "
      @lcgren+5,lcgcol+8 SAY "NOMBRE: "
      @lcgren+5,lcgcol+80 SAY "R.F.C.:"
      @lcgren+5,lcgcol+80 SAY "R.F.C.:"
    
```

```

@lcgren+5.lcgcol+26 SAY pro->nom_pro
@lcgren+5.lcgcol+96 SAY pro->rfc
@lcgren+5.lcgcol+130 SAY " " ; @lcgren+6.lcgcol+5 SAY " "
@lcgren+6.lcgcol+8 SAY "DOMICILIO: "
@lcgren+6.lcgcol+8 SAY "DOMICILIO: "
@lcgren+6.lcgcol+80 SAY "TELEFONO: "
@lcgren+6.lcgcol+80 SAY "TELEFONO: "
@lcgren+6.lcgcol+20 SAY pro->domicilio
@lcgren+6.lcgcol+96 SAY pro->tel_casa
@lcgren+6.lcgcol+130 SAY " " ; @lcgren+7.lcgcol+5 SAY " "
@lcgren+7.lcgcol+5 SAY "0"
@lcgren+7.lcgcol+6 SAY REPLICATE("A",124)
@lcgren+7.lcgcol+130 SAY " " ; @lcgren+7.lcgcol+130 SAY " "
@lcgren+8.lcgcol+5 SAY " "
@lcgren+8.lcgcol+8 SAY "TIPO DE MOVIMIENTO: "
@lcgren+8.lcgcol+8 SAY "TIPO DE MOVIMIENTO: "
@lcgren+8.lcgcol+62 SAY "CATEGORIA: "
@lcgren+8.lcgcol+62 SAY "CATEGORIA: "
@lcgren+8.lcgcol+28 SAY legmov
@lcgren+8.lcgcol+73 SAY legcat
@lcgren+8.lcgcol+130 SAY " " ; @lcgren+9.lcgcol+5 SAY " "
@lcgren+9.lcgcol+5 SAY "0"
@lcgren+9.lcgcol+6 SAY REPLICATE("A",124)
@lcgren+9.lcgcol+50 SAY "0" ; @lcgren+9.lcgcol+90 SAY "0"
@lcgren+9.lcgcol+130 SAY " " ; @lcgren+9.lcgcol+130 SAY " "
@lcgren+10.lcgcol+5 SAY " "
@lcgren+10.lcgcol+15 SAY "FECHA INICIO: "
@lcgren+10.lcgcol+15 SAY "FECHA INICIO: "
@lcgren+10.lcgcol+58 SAY "FECHA LIMITE: "
@lcgren+10.lcgcol+58 SAY "FECHA LIMITE: "
@lcgren+10.lcgcol+100 SAY "NUMERO DE HORAS: "
@lcgren+10.lcgcol+100 SAY "NUMERO DE HORAS: "
@lcgren+10.lcgcol+34 SAY
SUBSTR(lcgfini,1,2)+'/'+lcgmeslc+'/' +SUBSTR(lcgfini,7,2)
@lcgren+10.lcgcol+50 SAY " "
IF SUBSTR(lcgmov,1,1) = "0" ; @lcgren+10.lcgcol+74 SAY
@lcgren+10.lcgcol+74 SAY
SUBSTR(lcgflim,1,2)+'/'+lcgmeslc+'/' +SUBSTR(lcgflim,7,3)
EMDIF
SELE prop
SEK lcgkey1
IF FOUND()
lcgttot=0
DO WHILE prop->rfc+prop->tipomov=legkey1
lcgttot=lcgttot+prop->teorica+prop->practica
SKIP
EMDDO
EMDIF
SELE coor
@lcgren+10.lcgcol+90 SAY " "
@lcgren+10.lcgcol+120 SAY lcgttot PICT '999.9'
@lcgren+10.lcgcol+130 SAY " " ; @lcgren+11.lcgcol+5 SAY " "
@lcgren+11.lcgcol+5 SAY "A" ; @lcgren+11.lcgcol+50 SAY "A"
@lcgren+11.lcgcol+6 SAY REPLICATE("A",124)
@lcgren+11.lcgcol+90 SAY "A"
@lcgren+11.lcgcol+130 SAY " " ; @lcgren+11.lcgcol+130 SAY
" "
" "
@lcgren+12.lcgcol+5 SAY " "
@lcgren+12.lcgcol+8 SAY "NUMERO DE HORAS DEL SEMESTRE
"
"
@lcgren+12.lcgcol+8 SAY "NUMERO DE HORAS DEL SEMESTRE
"
@lcgren+12.lcgcol+50 SAY "TEORICAS ( " )
@lcgren+12.lcgcol+60 SAY coor->hor_teo_an
@lcgren+12.lcgcol+78 SAY "PRACTICAS ( " )

```

```

@lcgren+12,lcgcol+89 SAY coor->hor_pra_an
lcgtot_an-coor->hor_teo_an-coor->hor_pra_an
@lcgren+12,lcgcol+109 SAY "TOTAL (
@lcgren+13,lcgcol+116 SAY lcgtot_an PICT '999.9'
@lcgren+12,lcgcol+130 SAY "3"
@lcgren+13,lcgcol+5 SAY "A"
@lcgren+13,lcgcol+6 SAY REPLICATE("A",124)
@lcgren+13,lcgcol+130 SAY "0" ; @lcgren+14,lcgcol+5 SAY "0"
@lcgren+14,lcgcol+6 SAY REPLICATE("A",124)
@lcgren+14,lcgcol+130 SAY "4"
@lcgren+15,lcgcol+5 SAY "3"
@lcgren+15,lcgcol+6 SAY "Imparte clases en otra(s)
Jefaturas(S) de Carrera de E.N.E.P. Aragon (Indique cual)"
@lcgren+15,lcgcol+130 SAY "3" ; @lcgren+16,lcgcol+5 SAY "3"
@lcgren+16,lcgcol+130 SAY "3" ; @lcgren+17,lcgcol+5 SAY "3"
@lcgren+17,lcgcol+25 SAY coor->area_o_jef
@lcgren+17,lcgcol+55 SAY lcgcattj
IF coor->hor_o_jef>0
@lcgren+17,lcgcol+120 SAY coor->hor_o_jef
.ENDIF
@lcgren+17,lcgcol+130 SAY "3" ; @lcgren+18,lcgcol+5 SAY "3"
@lcgren+18,lcgcol+11 SAY "A R E A : "
@lcgren+18,lcgcol+50 SAY "CATEGORIA: "
@lcgren+18,lcgcol+100 SAY "NUMERO DE HORAS: "
@lcgren+18,lcgcol+24 SAY REPLICATE("A",20)
@lcgren+18,lcgcol+61 SAY REPLICATE("A",30)
@lcgren+18,lcgcol+116 SAY REPLICATE("A",10)
@lcgren+18,lcgcol+130 SAY "3" ; @lcgren+19,lcgcol+5 SAY "3"
@lcgren+19,lcgcol+130 SAY "3" ; @lcgren+20,lcgcol+5 SAY "3"
@lcgren+20,lcgcol+6 SAY "Imparte clases en otra(s)
Dependencia(S) de la U.N.A.M. (Indique cual)"
@lcgren+20,lcgcol+130 SAY "3" ; @lcgren+21,lcgcol+5 SAY "3"
@lcgren+21,lcgcol+130 SAY "3" ; @lcgren+22,lcgcol+5 SAY "3"
@lcgren+22,lcgcol+28 SAY coor->dep_o_dep
@lcgren+22,lcgcol+65 SAY lcgcattd
IF coor->hor_o_dep>0
@lcgren+22,lcgcol+120 SAY coor->hor_o_dep
.ENDIF
@lcgren+22,lcgcol+130 SAY "3" ; @lcgren+23,lcgcol+5 SAY "3"
@lcgren+23,lcgcol+11 SAY "DEPENDENCIA: "
@lcgren+23,lcgcol+50 SAY "CATEGORIA: "
@lcgren+23,lcgcol+100 SAY "NUMERO DE HORAS: "
@lcgren+23,lcgcol+24 SAY REPLICATE("A",20)
@lcgren+23,lcgcol+61 SAY REPLICATE("A",30)
@lcgren+23,lcgcol+116 SAY REPLICATE("A",10)
@lcgren+23,lcgcol+130 SAY "3" ; @lcgren+24,lcgcol+5 SAY "3"
@lcgren+24,lcgcol+42 SAY "0"
@lcgren+24,lcgcol+43 SAY REPLICATE("A",50)
@lcgren+24,lcgcol+93 SAY "4" ; @lcgren+24,lcgcol+130 SAY
"3"
@lcgren+25,lcgcol+5 SAY "3" ; @lcgren+25,lcgcol+42 SAY
"3"
@lcgren+25,lcgcol+50 SAY "DATOS PARA LA FORMA DE SEGURO DE
VIDA"
@lcgren+25,lcgcol+93 SAY "3" ; @lcgren+25,lcgcol+130 SAY
"3"
@lcgren+26,lcgcol+5 SAY "3" ; @lcgren+26,lcgcol+5 SAY "A"
@lcgren+26,lcgcol+6 SAY REPLICATE("A",124)
@lcgren+26,lcgcol+42 SAY "3"
@lcgren+26,lcgcol+55 SAY "0"
@lcgren+26,lcgcol+93 SAY "3" ; @lcgren+26,lcgcol+130 SAY
"0"
@lcgren+26,lcgcol+130 SAY "3" ; @lcgren+27,lcgcol+5 SAY "3"
@lcgren+27,lcgcol+10 SAY "NOMBRE DE LOS BENEFICIARIOS"
@lcgren+27,lcgcol+42 SAY "3" ; @lcgren+27,lcgcol+47 SAY

```

"SEXO"

@lcgren+27,lcgcol+55 SAY ""
@lcgren+27,lcgcol+69 SAY "PARENTESCO"
@lcgren+27,lcgcol+93 SAY ""
@lcgren+27,lcgcol+105 SAY "PORCENTAJE %"
@lcgren+27,lcgcol+130 SAY "" ; @lcgren+28,lcgcol+5 SAY ""
@lcgren+28,lcgcol+5 SAY "A"
@lcgren+28,lcgcol+6 SAY REPLICATE("A",124)
@lcgren+28,lcgcol+42 SAY "" ; @lcgren+28,lcgcol+55 SAY

""

@lcgren+28,lcgcol+93 SAY ""
@lcgren+28,lcgcol+130 SAY "0" ; @lcgren+28,lcgcol+130 SAY

""

@lcgren+29,lcgcol+5 SAY "" ; @lcgren+29,lcgcol+42 SAY

""

@lcgren+29,lcgcol+55 SAY ""
@lcgren+29,lcgcol+93 SAY ""
@lcgren+29,lcgcol+130 SAY ""
IF lcgmov="NUEVO INGRESO"
@lcgren+29,lcgcol+6 SAY pro->beneficial
@lcgren+29,lcgcol+44 SAY pro->sexb1
@lcgren+29,lcgcol+56 SAY pro->parb1
ENDIF

@lcgren+30,lcgcol+5 SAY "" ; @lcgren+30,lcgcol+42 SAY ""
@lcgren+30,lcgcol+55 SAY ""
@lcgren+30,lcgcol+93 SAY ""
@lcgren+30,lcgcol+130 SAY ""
IF lcgmov="NUEVO INGRESO"
@lcgren+30,lcgcol+6 SAY pro->beneficia2
@lcgren+30,lcgcol+44 SAY pro->sexb2
@lcgren+30,lcgcol+56 SAY pro->parb2
ENDIF

"A"

@lcgren+31,lcgcol+5 SAY "A"
@lcgren+31,lcgcol+6 SAY REPLICATE("A",124)
@lcgren+31,lcgcol+42 SAY "A" ; @lcgren+31,lcgcol+55 SAY

@lcgren+31,lcgcol+93 SAY "A"
@lcgren+31,lcgcol+130 SAY "0" ; @lcgren+32,lcgcol+5 SAY "0"
@lcgren+32,lcgcol+6 SAY REPLICATE("A",124)
@lcgren+32,lcgcol+11 SAY "0"
@lcgren+32,lcgcol+42 SAY "0" ; @lcgren+32,lcgcol+48 SAY

"0"

@lcgren+32,lcgcol+99 SAY "0" ; @lcgren+32,lcgcol+110 SAY

"0"

@lcgren+32,lcgcol+122 SAY "0" ; @lcgren+32,lcgcol+130 SAY

"1"

@lcgren+33,lcgcol+5 SAY "" ; @lcgren+33,lcgcol+6 SAY

"CLAVE"

@lcgren+33,lcgcol+11 SAY ""
@lcgren+33,lcgcol+17 SAY "A S I G N A T U R A"
@lcgren+33,lcgcol+42 SAY "" ; @lcgren+33,lcgcol+43 SAY

"GRUPO"

@lcgren+33,lcgcol+48 SAY ""
@lcgren+33,lcgcol+62 SAY "H O R A R I O"
@lcgren+33,lcgcol+99 SAY ""
@lcgren+33,lcgcol+100 SAY "H-TEORICAS"
@lcgren+33,lcgcol+110 SAY ""
@lcgren+33,lcgcol+111 SAY "H-PRACTICAS"
@lcgren+33,lcgcol+122 SAY ""
@lcgren+33,lcgcol+123 SAY "H-TOTAL"
@lcgren+33,lcgcol+130 SAY ""
@lcgren+34,lcgcol+5 SAY ""
@lcgren+34,lcgcol+5 SAY "A"
@lcgren+34,lcgcol+6 SAY REPLICATE("A",124)
@lcgren+34,lcgcol+11 SAY "" ; @lcgren+34,lcgcol+42 SAY

```

**
** @lcgren+34,lcgcol+48 SAY "" ; @lcgren+34,lcgcol+99 SAY
** @lcgren+34,lcgcol+110 SAY "" ; @lcgren+34,lcgcol+122 SAY
** @lcgren+34,lcgcol+130 SAY "" ; @lcgren+34,lcgcol+130 SAY
"0"
SELE prop
SEEK lcgkey1
IF FOUND()
  lcgtteo=0 ; lcgtpra=0 ; lcgttot=0
  cont=0
  DO WHILE prop->rfc-prop->stipomov=lcgkey1
    @PROW()+1,lcgcol+5 SAY ""
    @PROW(),lcgcol+07 SAY prop->cve_asig
    @PROW(),lcgcol+11 SAY ""
    @PROW(),lcgcol+12 SAY prop->asignatura
    @PROW(),lcgcol+42 SAY ""
    @PROW(),lcgcol+43 SAY prop->grupo
    @PROW(),lcgcol+48 SAY ""
    @PROW(),lcgcol+50 SAY prop->horario
    @PROW(),lcgcol+99 SAY ""
    lcgtteo=lcgtteo+prop->teorica
    @PROW(),lcgcol+102 SAY prop->teorica
    @PROW(),lcgcol+110 SAY "" ; lcgtpra=lcgtpra+prop-
>practica
    @PROW(),lcgcol+113 SAY prop->practica
    @PROW(),lcgcol+122 SAY ""
    lcgttot=lcgttot+prop->teorica+prop->practica
    lcgtotal=prop->teorica+prop->practica
    @PROW(),lcgcol+124 SAY lcgtotal PICT '999.9'
    @PROW(),lcgcol+130 SAY ""
    cont=cont+1
    SKIP
  ENDDO
ENDIF
DO WHILE cont<11
  @PROW()+1,lcgcol+5 SAY "" ; @PROW(),lcgcol+11 SAY ""
  @PROW(),lcgcol+42 SAY "" ; @PROW(),lcgcol+48 SAY ""
  @PROW(),lcgcol+99 SAY "" ; @PROW(),lcgcol+110 SAY ""
  @PROW(),lcgcol+122 SAY "" ; @PROW(),lcgcol+130 SAY ""
  cont=cont+1
  ENDDO
  @PROW()+1,lcgcol+5 SAY "" ; @PROW(),lcgcol+5 SAY "A"
  @PROW(),lcgcol+11 SAY "A" ; @PROW(),lcgcol+6 SAY
REPLICATE("A",124)
  @PROW(),lcgcol+42 SAY "A" ; @PROW(),lcgcol+48 SAY "0"
  @PROW(),lcgcol+99 SAY "" ; @PROW(),lcgcol+110 SAY ""
  @PROW(),lcgcol+122 SAY "" ; @PROW(),lcgcol+130 SAY ""
  @PROW()+1,lcgcol+5 SAY ""
  @PROW(),lcgcol+82 SAY "TOTAL DE HORAS: "
  @PROW(),lcgcol+99 SAY ""
  @PROW(),lcgcol+101 SAY lcgtteo PICT '999.9'
  @PROW(),lcgcol+110 SAY ""
  @PROW(),lcgcol+112 SAY lcgtpra PICT '999.9'
  @PROW(),lcgcol+122 SAY ""
  @PROW(),lcgcol+124 SAY lcgttot PICT '999.9'
  @PROW(),lcgcol+130 SAY "" ; @PROW()+1,lcgcol+5 SAY "A"
  @PROW(),lcgcol+6 SAY REPLICATE("A",124);@PROW(),lcgcol+99
SAY "0"
  @PROW(),lcgcol+110 SAY "0" ; @PROW(),lcgcol+122 SAY "0"
  @PROW(),lcgcol+130 SAY "0"
  @lcgren+51,lcgcol+35 SAY "San Juan de Aragen, Edo. de M.x.,
a "
  @PROW(),lcgcol+72 SAY SUBSTR(lcgcfecha,1,2)

```

```

@PROW(),lccol+75 SAY "de"
@PROW(),lccol+78 SAY lcgmesc
@PROW(),lccol+91 SAY "de 19"
@PROW(),lccol+97 SAY SUBSTR(lcgfecha,7,2)
@PROW()+3,lccol+18 SAY "JEFE DE CARRERA"
@PROW(),lccol+60 SAY "INTERESADO"
*@PROW()+2,lccol+55 SAY PRO->nom_pro
@PROW()+3,lccol+10 SAY REPLICATE("A",30)
@PROW(),lccol+50 SAY REPLICATE("A",30)
@PROW(),lccol+90 SAY REPLICATE("A",30)
@PROW()+1,lccol+14 SAY lcgjeje
@PROW(),lccol+58 SAY PRO->NOM_PRO
@PROW(),lccol+92 SAY 'DEPARTAMENTO DE PERSONAL'
EJEC
NEXT
ENDIF
ENDIF
ENDDO
ENDIF
@0,0 SAY INICIA
SET PRINT OFF
SET CONSOLE ON
SET DEVICE TO SCREEN
CLOSE pro
CLOSE cat
RETURN
*/ KOF()
```

```
* PROGRAMA...: PROF_FUN.PRG
* USO.....: LIBRERIA DE FUNCIONES
* FECHA.....: 11 DE NOVIEMBRE DE 1993
* MODIFICADO.: 19 DE MAYO DE 1996
```

```
* DIBUJA UN CUADRO CON O SIN SOMBRA
```

```
FUNCTION CUADRO(FONDO,C1,C2,C3,C4,SOMBRA)
IF SOMBRA="S" ; SOMLCG(C1,C2,C3,C4) ; ENDF
SET COLOR TO (FONDO)
@C1,C2,C3,C4 BOX "O";DAA; "
RETURN nil
```

```
*****
*****
```

```
* REALIZA LA SOMBRA TENUE
FUNCTION SOMLCG( NTOP, NLEFT, NBOTTOM, NRIGHT )
LOCAL NSHADTOP ; LOCAL NSHADLEFT ; LOCAL NSHADBOTTOM ; LOCAL NSHADRIGHT
NSHADTOP := NSHADBOTTOM := MIN( NBOTTOM + 1, MAXROW() )
NSHADLEFT := NLEFT + 1
NSHADRIGHT := MIN( NRIGHT + 1, MAXCOL() )
RESTSCREEN( NSHADTOP, NSHADLEFT, NSHADBOTTOM, NSHADRIGHT,
;
; TRANSFORM( SAVESCREEN( NSHADTOP, NSHADLEFT, NSHADBOTTOM,
NSHADRIGHT),
; REPLICATE("X " ; NSHADRIGHT - NSHADLEFT + 1 ) ) )
NSHADTOP := NTOP + 1
NSHADLEFT := NSHADRIGHT := MIN( NRIGHT + 1, MAXCOL() )
NSHADBOTTOM := NBOTTOM
RESTSCREEN( NSHADTOP, NSHADLEFT, NSHADBOTTOM, NSHADRIGHT,
; TRANSFORM( SAVESCREEN( NSHADTOP, NSHADLEFT, NSHADBOTTOM, NSHADRIGHT),
; REPLICATE("X ", NSHADBOTTOM - NSHADTOP + 1 ) ) )
RETURN nil
```

```
*****
*****
```

```
* FORMA EL MENU DE OPCIONES VERTICAL
FUNCTION MENUV(LINEA,COLUMNA,TITULO,TEXTO)
LOCAL veces
@23,03 CLEAR TO 23,75
FOR VECES=1 TO LEN(TITULO)
@LINEA+1,COLUMNA+2 PROMPT TITULO[VECES] MESSAGE TEXTO[VECES]
LINEA+=1
NEXT
RETURN nil
```

```
*****
*****
```

```
* FORMA EL MENU DE OPCIONES HORIZONTAL
FUNCTION MENUH(LINEA,TITULO,TEXTO,COLUMNA)
LOCAL veces
@23,03 CLEAR TO 23,75
FOR VECES=1 TO LEN(TITULO)
@LINEA,COLUMNA[VECES] PROMPT TITULO[VECES] MESSAGE TEXTO[VECES]
NEXT
RETURN nil
```

```
*****
*****
```

```
* ENVIA CUADRO CON MENSAJE DE UN RENGLON CON O SIN SONIDO Y RETARDO
```

FUNCTION MENSAJEE (M1,C1,C2,RETARDO,FONDO)

@C1+1,C2+2 SAY M1
IF RETARDO="R"
 TONE(300,10)
 TONE(200,10)
 INKEY(1)

ENDIF
SET COLOR TO (FONDO)
RETURN n11

*AVISA SI UNA CONDICION NO ES CORRECTA Y PERMITE REINTENTAR
FUNCTION FILTRO(C1,C2,C3,C4,AVISO,FONDO)
MEMVAR getlist.pantallaf,lcgerror,opc
PANTALLAF=SAVESCREEN(C1,C2,C3+1,C4+1) ; CUADRO(LCGERROR,C1,C2,C3,C4,"S")
@C1+1,C2+2 SAY AVISO ; OPC=-SPACE(1)
@C1+2,((C4-C2-24)/2)+C2+1 SAY "DESEA REINTENTAR S/N ?" GET OPC PICT "1"
VALID(OPC5="SN")
READ ; RESTSCREEN(C1,C2,C3+1,C4+1,PANTALLAF) ; SET COLOR TO (FONDO)
RETURN (OPC)

* RUTINA DE ENVIO DE REPORTE Y CHEQUEO DE IMPRESORA

FUNCTION REPORTE()
MEMVAR getlist,lcgtinte2,opc,lcgtinte
LOCAL qr1,qr2,qr3,qr4,panreporte
PUBLIC lcgtinte2,opc,lcgtinte
QR1=10 ; QR2=22 ; QR3=13 ; QR4=58
PANREPORTE=SAVESCREEN(QR1,QR2,QR3+1,QR4+1) ;
CUADRO(lcgtinte2,QR1,QR2,QR3,QR4,"S")
@QR1+1,QR2+6 SAY "COLOQUE PAPEL CARTA"; INKEY(3)
@QR1+2,QR2+8 SAY "ENCIENDA LA IMPRESORA" ; READ
INKEY (3)
RESTSCREEN(QR1,QR2,QR3+1,QR4+1,PANREPORTE) ; OPC=-SPACE(1)
DO WHILE .T.
 IF .NOT. ISPRINTER()
 FILTRO(10,17,13,64,"* LA IMPRESORA NO SE ENCUENTRA PREPARADA
***,lcgtinte)
 IF OPC="S" ; LOOP
 ENDIF
 ENDIF
 EXIT
ENDDO
SETCOLOR(lcgtinte)
RETURN (OPC)

FUNCTION REPORTE2()
MEMVAR getlist,lcgtinte2,opc,lcgtinte
LOCAL qr1,qr2,qr3,qr4,panreporte
PUBLIC lcgtinte2,opc,lcgtinte
QR1=10 ; QR2=22 ; QR3=13 ; QR4=58
PANREPORTE=SAVESCREEN(QR1,QR2,QR3+1,QR4+1) ;
CUADRO(lcgtinte2,QR1,QR2,QR3,QR4,"S")
@QR1+1,QR2+6 SAY "COLOQUE PAPEL STOCK CARTA"; INKEY(3)
@QR1+2,QR2+8 SAY "ENCIENDA LA IMPRESORA" ; READ
INKEY (3)
RESTSCREEN(QR1,QR2,QR3+1,QR4+1,PANREPORTE) ; OPC=-SPACE(1)
DO WHILE .T.
 IF .NOT. ISPRINTER()
 FILTRO(10,17,13,64,"* LA IMPRESORA NO SE ENCUENTRA PREPARADA

```

***.lsgtinte)
  IF OPC="S" ; LOOP
  ENDF
  ENDF
  EXIT

```

```

ENDDO
SETCOLOR(lsgtinte)
RETURN(OPC)

```

```

* PERMITE SUSPENDER LA IMPRESION
FUNCTION SUSPENDER(C1,C2,C3,C4,AVISO,FONDO)
MEMVAR getlist,lcgerror,opc
LOCAL pantallaf
PUBLIC lcgerror,opc
PANTALLAF=SAVESCREEN(C1,C2,C3+1,C4+1) ; CUADRO(LCGERROR,C1,C2,C3,C4,"S")
@C1+1,C2+2 SAY AVISO ; OPC=SPACE(1)
@C1+2,((C4-C2-24)/2)+C2+2 SAY "DESEA CONTINUAR S/N ?" GET OPC PICT "I"
VALID(OPCS"SN")
READ ; RESTSCREEN(C1,C2,C3+1,C4+1,PANTALLAF) ; SET COLOR TO (FONDO)
RETURN (OPC)

```

```

***** FUNCION LIMPIA PANTALLA *****

```

```

#include "grump.ch"
#command CLEARIT(<t>, <l>, <b>, <r>) => scroll(<t>, <l>, <b>, <r>, 0) ;
                                for yy := 1 TO ndelay; next
#define MIDPOINT int((maxcol + 1) / 2)

function clrscr(ntype, ndelay)
local xx, yy, maxrow := maxrow(), maxcol := maxcol(), buffer, mrow, ;
    ndelays := {100, 100, 40, 40, 40, 40, 150, 40, 15}
default ntype to 1
default ndelay to ndelays[ntype]
do case
case ntype == WINDOWDOWN
for xx = 0 to maxrow
CLEARIT(xx, 0, xx, maxcol)
next
case ntype == WINDOWUP
for xx = maxrow to 0 step -1
CLEARIT(xx, 0, xx, maxcol)
next
case ntype == DOORRIGHT
for xx = 0 to maxcol
CLEARIT(0, xx, maxrow, xx)
next
case ntype == DOORLEFT
for xx = maxcol to 0 step -1
CLEARIT(0, xx, maxrow, xx)
next
case ntype == CURTAINOPEN
for xx = MIDPOINT to maxcol
scroll(00, maxcol-xx, maxrow, maxcol-xx, 0)
CLEARIT(0, xx, maxrow, xx)
next
case ntype == CURTAINSHUT
for xx = maxcol to MIDPOINT step -1
scroll(00, maxcol-xx, maxrow, maxcol-xx, 0)
CLEARIT(0, xx, maxrow, xx)
next
case ntype == CAMERALENS // still set up for 25 x 80 mode
mrow := 0
for xx = 0 to 33 step 3
scroll(mrow, xx, mrow, maxcol - xx, 0)
scroll(maxrow - mrow, xx, maxrow - mrow, maxcol - xx, 0)

```

```

scroll(mrow, xx, maxrow - mrow, xx + 3, 0)
CLEARIT(mrow, 76 - xx, maxrow - mrow, maxcol - xx)
mrow++
next
scroll(mrow, 37, mrow, 42, 0)
case ntype == DIAGONAL
for xx = 0 to maxcol // still set up for 25 x 80 mode
scroll(int(xx/3.2), xx, maxrow, xx, 0)
scroll(0, maxcol - xx, maxrow - int(xx/3.2), maxcol - xx, 0)
for yy = 1 to ndelay
NEXT
NEXT
otherwise // HALVES
for xx = 0 to maxrow
buffer := savescreeen(xx, 0, maxrow - 1, MIDPOINT - 1)
scroll(xx, 0, xx, MIDPOINT - 1, 0)
IF xx < maxrow
restscreen(xx+1, 0, maxrow, MIDPOINT - 1, buffer)
ENDIF
buffer := savescreeen(1, MIDPOINT, maxrow-xx, maxcol)
scroll(maxrow-xx, MIDPOINT, maxrow-xx, maxcol, 0)
IF xx < maxrow
restscreen(0, MIDPOINT, maxrow - 1 - xx, maxcol, buffer)
ENDIF
for yy = 1 to ndelay
NEXT
NEXT
ENDCASE
RETURN NIL

```

```

* END FUNCTION LIMPIA()
-----

```

```

*****
* Si un campo requiere de un codigo, al pulsar enter se puede
* llamar a esta funcion para que el usuario elija en una ventana
* la opcion que desea
*
* Sintaxis: ayuda_cod
* (alias,campo,formato,color_caja,tipo_caja,efecto)
*
* Retorna el valor del resultado de la operacion de lectura
* 0 : asignación exitosa
* 2 : no habia registros
* 1 : Se pulso ESCAPE
* 3 :

```

```

FUNCTION BRONPRO(alias,campo,formato,color_caja,tipo_caja,efecto)
NEWVAR
getlist,kf12,k_enter,lcgaux,lcgrav,lcginte7,lcgkey,esta,aquella,;
banderin
LOCAL
correcto,v_reg,cont,aux,nohay,ccolor,ancho,v_area,longi,rensupsq,;
colsupisq,reninfder,colinfder,stat_cursor,pantalla,dato
PUBLIC
vcadnum,lcginte,lcginte7,lcgrav,lcgaux,lcgkey,k_f12,k_enter,esta,;
aquella,banderin
DATO := NIL
CCOLOR=SETCOLOR()
SETCOLOR(color_caja)
* COORDENADAS DE LA VENTANA
CORRECTO := T
CORRECTO := T
V_AREA=SELECT()
V_REG := RECNO()
SELECT &ALIAS

```

```

ANCHO := 0
IF (PCOUNT() > 2)
  FOR CONT = 1 TO LEN(FORMATO)
    ancho := ANCHO + LEN(FORMATO[CONT]) + 2
  NEXT CONT
ELSE
  FOR CONT = 1 TO LEN(CAMPO)
    AUX := CAMPO[CONT]
    DATO := &AUX
    DO CASE
      CASE TYPE (CAMPO[CONT]) = "C"
        IF LEN(AUX) < LEN(DATO)
          ancho := ANCHO + len(DATO) + 2
        ELSE
          ancho := ANCHO + len(AUX) + 2
        ENDIF
      CASE TYPE (CAMPO[CONT]) = "N"
        IF LEN(AUX) < LENNUM(DATO)
          ancho := ANCHO + lenNUM(DATO) + 2
        ELSE
          ancho := ANCHO + len(AUX) + 2
        ENDIF
      CASE TYPE (CAMPO[CONT]) = "D"
        IF LEN(AUX) < LENNUM(DATO)
          ANCHO := ANCHO + 12
        ELSE
          ancho := ANCHO + len(AUX) + 2
        ENDIF
      CASE TYPE (CAMPO[CONT]) = "U"
        CORRECTO := .F.
    ENDCASE
  NEXT CONT
ENDIF
IF ( CORRECTO )
* UN RECONUT ( ) NO FUNCIONA PORQUE PUEDE HABER UN SET FILTER
GO TOP
LONGI := 0
DO WHILE (.NOT.EOF())
  LONGI ++
  skip
ENDDO
GO TOP
IF (LONGI < 1)
  CORRECTO := .F.
  ALERT("NO EXISTE INFORMACION DISPONIBLE")
  lcgkey := SPACE(15)
  RETURN (lgkey)
ELSE
  GO TOP
  IF ( LONGI > MAXROW() - 6 )
    RENSUPIZO := 3
    RENINFDER := MAXROW() - 3
  ELSE
    RENSUPIZO := INT(MAXROW()/2 - longi / 2 - 2)
    RENINFDER := INT(MAXROW()/2 + LONGI / 2 + 2)
  ENDIF
  IF (ANCHO > MAXCOL() - 6 )
    COLSUPIZO := 3
    COLINFDER := MAXCOL() - 3
  ELSE
    COLSUPIZO := INT (MAXCOL()/2 - ANCHO / 2 - 2)
    COLINFDER := INT (MAXCOL()/2 + ANCHO / 2 + 2)
  ENDIF
  * PARAMETROS POR OMISSION
  IF PCOUNT() < 4

```

```

COLOR CAJA = "N/W,W/N"
ENDIF
IF PCOUNT() < 5
TIPO CAJA := 3
ENDIF
IF PCOUNT() < 6
EFECTO := .F.
ENDIF
STAT CURSOR = SETCURSOR()
PANTALLA := SAVESCREEN(RENSUPIZQ, COLSUPIZQ, RENINFDER+2,
COLINFDER+2)
SET KEY 13 TO SOL127
setcolor(COLOR_CAJA)
CUADRO (lcsint@7,RENSUPIZQ, COLSUPIZQ, RENINFDER, COLINFDER, "S")
ESTA := RENINFDER
AQUELLA:=COLSUPIZQ
BANDERIN = .T.
DEBIT (RENSUPIZQ+1, COLSUPIZQ+1, RENINFDER-1, COLINFDER-
1, CAMPO, "AYU DATO", format@)
RESTSCREEN(RENSUPIZQ, COLSUPIZQ, RENINFDER+2,
COLINFDER+2, PANTALLA)
SET KEY 13 TO
ENDIF
IF LASTKEY() = 27
DATO := SPACE(13)
ELSEIF lcsrev
AUX := CAMPO[1]
lcsaux:= CAMPO[2]
DATO := &AUX+&lcsaux
ELSE
AUX := CAMPO[1]
DATO := &AUX
ENDIF
ENDIF
* RESTAURA CONDICIONES INICIALES
SELECT &V AREA
SETCOLOR(CCOLOR)
SETCURSOR(STAT CURSOR)
CLEAR TYPEHEAD
RETURN (DATO)
*****
FUNCTION AYU_DATO
RENVAR c@11:1:1 k_f12 esta, aquella, banderin, vcdnum
PARAMETERS MODO, POSIC
LOCAL valor, tecl, ren, col
PUBLIC k_f12
VALOR := 1
REN = ESTA+1
COL = AQUELLA+2
DO CASE
CASE LASTKEY() = 27
VALOR = 0
CASE LASTKEY() = 0
VALOR = 0
CASE (LASTKEY() >= 65 .AND. LASTKEY() <= 90) .OR.;
(LASTKEY() >= 97 .AND. LASTKEY() <= 122)
TECL := UPPER(CHR(LASTKEY()))
IF BANDERIN
VCDNUM := TECL
ELSE
VCDNUM := VCDNUM + TECL
ENDIF
BANDERIN := .F.

```

```
@ REN, COL SAY VCADNUM
SEEK VCADNUM
KEYBOARD CHR(K_F12)
VALOR=2
CASE (LASTKEY() >= 48 .AND. LASTKEY() <= 57)
  TECL := CHR(LASTKEY())
  IF BANDERIN
    VCADNUM := TECL
  ELSE
    VCADNUM := VCADNUM + TECL
  ENDF
  BANDERIN := .F.
@ REN, COL SAY VCADNUM
SEEK VAL(VCADNUM)
KEYBOARD CHR(K_F12)
VALOR=2
CASE LASTKEY() = 32
  @REN, COL CLEAR TO REN, COL+LEN(VCADNUM)
  VCADNUM = SPACE(0)
ENDCASE
RETURN (VALOR)

FUNCTION COD_AYUDA
#include "inKey.ch"
MEMVAR getlist, k_enter
KEYBOARD CHR(K_ENTER)
RETURN nil

FUNCTION SOL127
KEYBOARD CHR(00)
RETURN nil
```

```

*****
* PROGRAM-ID:      Prof. Gen. prg ver 1.0      llamado por Prof_Pro.prg
* AUTHOR:         Leticia Cerd Garrido
* REMARKS:        que Prepara la Informacion para Generar
*                 las Propuestas de Profesores
* MODIFIED:       07 Julio, 1996.
*****
MEMVAR getlist,lcgthoract,lcgrfcpc,vm_ano_linea,vm_periodo_linea,;
entrega,i,actrfc_tit,actrfc_sup,acthorario,actcve_mat,actgrupo,ac
thrs,;
heteorica,htpractica,hteorica,hpractica
ENTREGA := HORA_RFC HOEX (lcgrfcpc,vm_ano_linea,vm_periodo_linea)
IF ENTREGA=NIL .AND. LEN(ENTREGA)>0
  FOR I = 1 TO LEN(ENTREGA)
    actrfc_tit := ENTREGA[I,7]
    actrfc_sup := ENTREGA[I,8]
    acthorario := ENTREGA[I,2]
    actcve_mat := ENTREGA[I,3]
    actgrupo := ENTREGA[I,5]
    acthrs := ENTREGA[I,6]
    PROF_PP()
    hteorica=hteorica+hteorica
    htpractica=htpractica+hpractica
  NEXT
ENDIF
RELEASE entrega,i,actrfc_tit,actrfc_sup,acthorario,actcve_mat,actgrupo,;
acthrs,lcgkey1,lcgkeyhor,lcghorpro,y,lcglun,lcgmar,lcgmie,lcgvie
,;
lcgvie,lcgsab,lcgvln,lcgvmar,lcgvmaie,lcgvjue,lcgvvie,lcgvsab,x,
x,;
lcgmateria,lcgvtipo
RETURN

/*****
* PROF_PP()
* Funcion que genera la informacion para las propuestas de profesores
* y las tarjetas de control de asistencia
*****/
FUNCTION PROF_PP()
MEMVAR getlist,lcgrfcpc,lcgthoract,lcgthorant,lcgflag,actrfc_tit,;
actrfc_sup,acthorario,actcve_mat,actgrupo,acthrs,hteorica,hpracti
ca
LOCAL lcgkey1,lcgkeyhor,lcghorpro,y,lcglun,lcgmar,lcgmie,lcgvie,lcgvsab,lcgvln,lcgvmar,lcgvmaie,lcgvjue,lcgvvie,lcgvsab,x,s,lcgmate
ria,;
lcgvtipo

lcgkey1 := actcve_mat
lcgkeyhor := acthorario
lcghorpro := y
lcglun:= ; lcgmar:= ; lcgmaie:= ; lcgjue:= ; lgvvie:= ; lcgvsab:=
lcgvln:= ; lgvmar:= ; lgvmaie:= ; lgvjue:= ; lgvvie:= ; lgvvsab:=
FOR x=1 TO LEN(lcgkeyhor)
  IF SUBSTR(lcgkeyhor,x,1)='L'; y='LUNES' ; lgvln=1 ; ENDDIF
  IF SUBSTR(lcgkeyhor,x,1)='M'; y='MARTES' ; lgvmar=1 ; ENDDIF
  IF SUBSTR(lcgkeyhor,x,1)='M'; y='MIERCOLES' ; lgvmaie=1 ; ENDDIF
  IF SUBSTR(lcgkeyhor,x,1)='J'; y='JUEVES' ; lgvjue=1 ; ENDDIF
  IF SUBSTR(lcgkeyhor,x,1)='V'; y='VIERNES' ; lgvvie=1 ; ENDDIF
  IF SUBSTR(lcgkeyhor,x,1)='S'; y='SABADO' ; lgvvsab=1 ; ENDDIF
  IF
SUBSTR(lcgkeyhor,x,1)='0'.OR.SUBSTR(lcgkeyhor,x,1)='1'.OR.SUBSTR(lcgkeyh
or,x,1)='2'
y='(+SUBSTR(lcgkeyhor,x,5)+'-'+SUBSTR(lcgkeyhor,x+5,5)+')'

```

```

z=SUBSTR(lcgkeyhor,x,5)+' a '+SUBSTR(lcgkeyhor,x+5,5)
x=x+9
IF lcgvlun=1 ; lcgln=z ; lcgvlun=0; ENDIF
IF lcgvmar=1 ; lcgmar=z ; lcgvmar=0; ENDIF
IF lcgvmie=1 ; lcgmie=z ; lcgvmie=0; ENDIF
IF lcgvjue=1 ; lcgjue=z ; lcgvjue=0; ENDIF
IF lcgvie=1 ; lcgvie=z ; lcgvie=0; ENDIF
IF lcgvsab=1 ; lcgvsab=z ; lcgvsab=0; ENDIF
ENDIF
IF y='(.OR.y=' '.OR.x=1
lcghorpro=lcghorpro+y
ELSE
lcghorpro=lcghorpro+'.'y
ENDIF
y=' '
NEXT
SELE mat.
SEEK lcgkey1
IF FOUND()
lcgmateria=mat->des_mat
lcgvtipo=mat->tipo
ELSE
lcgmateria=space(32)
lcgvtipo=' '
ENDIF
IF lcgthorant = lcgthoract
/*****
* PROPUESTAS AUTOMATICAS
*****/
SELE paut
APPEND BLANK
lcgflag=lcgflag+1
IF actrfc_tit = lcgrefcp
REPLA paut->rfc WITH actrfc_tit
ELSE
REPLA paut->rfc WITH actrfc_sup
ENDIF
REPLA paut->asignatura WITH lcgmateria,paut->grupo WITH actgrupo
REPLA paut->tipomov WITH '03',paut->cve_asig WITH actcve_mat
REPLA paut->horario WITH lcghorpro
REPLA paut->lunes WITH lcgln, paut->martes WITH lcgmar
REPLA paut->miercoles WITH lcgmie, paut->jueves WITH lcgjue
REPLA paut->viernes WITH lcgvie, paut->sabado WITH lcgvsab
REPLA paut->flag WITH STR(lcgflag,2)
IF lcgvtipo='*'
REPLA paut->practica WITH acthrs
hpractica := acthrs
ELSE
REPLA paut->teorica WITH acthrs
hteorica := acthrs
ENDIF
ELSE
/*****
* PROPUESTAS MANUALES
*****/
SELE pman
APPEND BLANK
lcgflag=lcgflag+1
IF actrfc_tit = lcgrefcp
REPLA pman->rfc WITH actrfc_tit
ELSE
REPLA pman->rfc WITH actrfc_sup
ENDIF
REPLA pman->asignatura WITH lcgmateria, pman->grupo WITH actgrupo
REPLA pman->cve_asig WITH actcve_mat

```

```
REPLA pman->horario WITH lcghorpro
REPLA pman->lunes WITH lcglun, pman->martes WITH lcgmart
REPLA pman->miercoles WITH lcgmie, pman->jueves WITH lcgmie
REPLA pman->viernes WITH lcgvie, pman->sabado WITH lcgsab
REPLA pman->flag WITH STR(lcgflag,2)
IF lcgvtipo='*'
  REPLA pman->practica WITH acthrs
  hpractica := acthrs
ELSE
  REPLA pman->teorica WITH acthrs
  hteorica := acthrs
ENDIF
ENDIF
RETURN nil
*/ EOF()
```

```

*****
* PROGRAM-ID: Prof_Man.prg ver 1.0 llamado por Prof_Rev.prg *
* AUTHOR: Leticia Cerd Garrido *
* REMARKS: Programa que Permite Revizar la Informacion antes *
* de Imprimir las Propuestas. *
* MODIFIED: 08 Julio, 1996. *
*****
#include "inkey.ch"
#define T_INSERT 22
MEMVAR geElisr.lcgruta
USE lcgruta\lcgprorev ALIAS prop NEW SHARED
SET INDEX TO lcgruta\proprfc,lcgruta\proptip
USE lcgruta\tipomov ALIAS mov NEW SHARED
SET INDEX TO lcgruta\mov

SELE pman
GO TOP
DO WHILE LASTKEY() <> 27
MOV_PMAN()
ENDDO
SET KEY K_F3 TO
SET KEY K_CTRL_DEL TO
SET KEY K_CTRL_INS TO
CLOSE prop
CLOSE mov
RELEASE lcgcontl,lcgrfc,lcgkey,lcgrev,lcgalt,lcgbanh,lcghrs,lcgvea,;
lcgvaais,lcgvgrupo,lcgvtmov,lcgvteo,lcgvpra,lcgvttot,lcgvhor,lcgh
old,;
lcgvln,lcgvmar,lcgvmie,lcgvjue,lcgvvie,lcgvsab,lcgvvea,lcgflag
;
lcgconta,lcgpalt,ptr,x,lcgres,lcgpan,lcgtot,lcgbajp,seguir,r_baj
a,;
lcgcmov,lcgvmov,seleold,lcgrev

RETURN

/*****
* MOV_PMAN()
*
* Función que realiza los movimientos al archivo de propuestas manuales
*****/
FUNCTION MOV_PMAN()
MEMVAR getlist,lcgcontl,lcgenpana,lcgtinte,lcgpunam,lcgcampo_p,lcgrfc,;
lcgkey,lcgrev,lcgalt,lcgbanh,lcghrs
lcgcontl := .T.

SET KEY 19 TO ; SET KEY 4 TO

DO WHILE (lcgcontl)
lcgalt := 0
SET KEY K_CTRL_INS TO ALT_PMAN()
SET CURSOR ON
SET KEY K_F3 TO
SET KEY K_CTRL_DEL TO
SETCOLOR (lcgenpana)
CLRSCR (5, 130)
RESTDSCREEN(0,0,24,79,lcgpunam)
CUADRO (lcgtinte,8,11,21,76,"5")
LCGSCREM()
DECLARE LCGCAMPO_P[3]
LCGCAMPO_P[1] := "RFC"
LCGCAMPO_P[2] := "FLAG"
LCGCAMPO_P[3] := "HORARIO"
SET KEY K_F2 TO COD_AYUDA
lcgrfc := SPACE(11)
lcgkey := SPACE(15)

```

```

MENSAJEE (" <CTR-INS> DAR DE ALTA UN REGISTRO ",22,20)
#9,19 SAY "RFC DEL PROFESOR:" GET lcgrfc PICT "@!"
READ
IF (LASTKEY() = K_F2 .OR. lcgrfc = SPACE(13)).AND. LASTKEY() <> 27
    lcgreve := .T.
    lcgkey := BROWPRO("PMAN",LCGCAMPO_P)
    lcgrfc := SUBSTR(lcgkey,1,13)
ENDIF
RELEASE LCGCAMPO_P
SET KEY K_F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgrfc) <> 0)
    SELE coor
    SEEK lcgrfc
    IF FOUND()
        lcgbanh := coor->bancos_hora
        lcghrs := coor->hor_teo + coor->hor_pra
        SELE pman
        SEEK lcgkey
        IF FOUND()
            lcgrfc := pman->rfc
            @9,17 SAY lcgrfc PICT "@!"
            CAM_PMAN()
        ENDIF
    ELSE
        lcgconti = .F.
    ENDIF
ENDIF
ENDDO
COMMIT
RETURN nil

/*****
* LCGSCREM()
*
* Función que despliega la pantalla de revisión
*****/
FUNCTION LCGSCREM()
@ 11, 14 SAY "Clave de Asignatura: "
@ 12, 14 SAY "Asignatura: "
@ 13, 14 SAY "Tipo de Movimiento: "
@ 14, 14 SAY "Grupo: "
@ 15, 14 SAY "Teoria: "
@ 16, 14 SAY "Practica: "
@ 17, 14 SAY "Horario: "
RETURN nil

/*****
* CAM_PMAN()
*
* Función que realiza los cambios al archivo de propuestas manuales
*****/
FUNCTION CAM_PMAN()
MEMVAR
getlist .lcgvcea,lcgvasig,lcgvgrupo,lcgvtmov,lcgvteo,lcgvpra,lcgvttot,;
lcgvhor,lcghold,lcgvlun,lcgvmar,lcgvmie,lcgvjue,lcgvvie,lcgvsab,;
lcgvcea,lcgvlag

LOCAL seguir
lcgvcea := pman->cve_asig
lcgvasig := pman->asignatura
lcgvgrupo := pman->grupo
lcgvtmov := pman->tipomov
lcgvteo := pman->teorica
lcgvpra := pman->practica
lcgvttot := pman->teorica+pman->practica
lcgvhor := pman->horario

```

```

lcghold := lcgvhor
lcgvlnun := pman->lunes
lcgvmar := pman->martes
lcgvmie := pman->miercoles
lcgvjue := pman->jueves
lcgvvie := pman->viernes
lcgvsab := pman->sabado
lcgflag := pman->flag
seguir = .T.
DO WHILE (seguir)
  @ 11,36 SAY lcgvvea PICT "@!"
  @ 12,36 SAY lcgvvasig PICT "@!"
  @ 13,36 SAY lcgvtmov PICT "@!"
  @ 14,36 SAY lcgvgrupo PICT "@!"
  @ 15,36 SAY lcgvteo PICT "@!"
  @ 16,36 SAY lcgvpra PICT "@!"
  @ 17,36 SAY lcgvhor PICT "@!"
  GRA_PMAN()
  seguir := .F.
  IF LASTKEY() = K_ESC
    RETURN nil
ENDIF
ENDDO
RETURN nil

/*****
* ALT_PMAN()
*
* Función que realiza las altas al archivo de propuestas manuales
*****/
FUNCTION ALT_PMAN()
MEMVAR getlist,lcgvtot,lcgvpra,lcgvteo,lcgvsab,lcgvvie,lcgvjue,lcgvmie,;
lcgvmar,lcgvlnun,lcgvhor,lcgvtmov,lcgvgrupo,lcgvvasig,lcgvvea,;
lcgrfc,lcgflag,lcgkey,lcghold,lcgconta,lcgpalt,lcgalt,lcgbanh,lcg
hra
PUBLIC lcgvvea := SPACE(4),lcgvvasig := SPACE(32),;
lcgvgrupo := SPACE(4),lcgvtmov := SPACE(2),lcgvhor :=
SPACE(40),;
lcgvlnun := SPACE(13),lcgvmar := SPACE(13),lcgmie :=
SPACE(13),;
lcgvjue := SPACE(13),lcgvvie := SPACE(13),lcgvsab :=
SPACE(13),;
lcgvteo := 0,lcgvpra := 0,lcgvtot := 0,lcgflag :=
0,lcghold:= ' '
lcgconta:=.T.
DO WHILE (lcgconta)
  lcgpalt = SAVESCREEN(0,0,24,79)
  @9,37 GET lcgrfc PICT "@!"
  IF lcgalt = 0
    COD_AYUDA()
  ENDIF
  READ
  @ 9,37 SAY lcgrfc PICT "@!"
  SELE coor
  SEEK lcgrfc
  IF !FOUND()
    ALERT("NO EXISTE EN EL CATALOGO DE PROFESORES")
    lcgrfc:=SPACE(13)
    lcgalt = 1
    RETURN nil
  ELSE
    SET KEY K_CTRL_INS TO
    lcgbanh := coor->banco_hora
    lcghra := coor->hor_teo + coor->hor_pra

```

```

SELE pman
SEEK lcgRFC
IF FOUND()
DO WHILE pman->rfc=lcgRFC
SKIP
ENDDO
SKIP-1
lcgflag:=VAL(pman->flag)+1
ELSE
SELE paut
SEEK lcgRFC
IF FOUND()
DO WHILE paut->rfc=lcgRFC
SKIP
ENDDO
SKIP-1
lcgflag:=VAL(paut->flag)+1
ELSE
lcgflag:=1
ENDIF
ENDIF
lcgflag:=STR(lcgflag,2)
lcgkey :=lcgRFC+lcgflag
GRA_PMAN()
lcgconta:=-.F.
ENDIF
ENDDO
RESTSCREEN(0,0,24,79,lcgpalt)
SET KEY K_CTRL_INS TO ALT_PMAN()
lcgRFC:=SPACE(13)
RETURN nil

/*****
* GRA_PMAN()
*
* Función que graba los movimientos al archivo de propuestas manuales
*****/
FUNCTION GRA_PMAN()
#define T BORRAR 127
MEMVAR geList,ptr,lcgRFC,lcgvvea,x,lcgvasig,lcgvhor,lcghold,lcgtinte,;
lcgres,lcgpan,lcgtinte2,lcgvgrupo,lcgvtmov,lcgvteo,lcgvpra,lcgvto
t,;
lcgvln,lcgvmar,lcgvmie,lcgvjue,lcgvvie,lcgvsab,lcgkq,lcgflag,;
lcgbanh,lcghrs,lcgtot,lcgcmov,lcgvmmov,seleoid,lcgrev
LOCAL seguir
SET KEY K_CTRL_INS TO
SET KEY K_F3 TO BAJ_PMAN()
SET KEY K_CTRL_DEL TO BAJ_PMAN()
seguir = .T.
ptr := 1
* MIENTRAS NO ESTEN CORRECTOS LOS DATOS o NO SE BORRE EL REGISTRO
DO WHILE (seguir)
MENSAJEE (" <F3> e <CTR-DEL> BORRAR ",22,20)
@ 9,37 SAY lcgRFC PICT "@!"
@ 9,62 SAY "HRS AUT : "
@ 9,71 SAY lcgbanh PICT "##.##"
@10,62 SAY "HRS ASIG: "
@10,71 SAY lcghrs PICT "##.##"
x=subs(lcgvvea,1,1)
IF TYPE(x)="#N"
@ 11,36 GET lcgvvea PICT "@!"
@ 12,36 GET lcgvasig PICT "@!"
ELSE
@ 11,36 SAY lcgvvea PICT "@!"
@ 12,36 SAY lcgvasig PICT "@!"

```

```

ENDIF
IF (LASTKEY() <> T_BORRAR .AND. LASTKEY() <> K_ESC)
  DECLARE LCGCMOV[2]
  LCGCMOV[1] := "CVE MOV"
  LCGCMOV[2] := "MOVIMIENTO"
  SET KEY K_F2 TO MOV_AYUDM
  lcgvmov:=lcgvtmov
  @ 13,36 CLEAR TO 13,50
  @ 13,36 GET lcgvmov PICT "@!" VALID lcgvmov $
*01,02,03,04,05,06,07,08,09,10,11, "
  READ
  Seleold:=SELE()
  SELE mov
  IF (LASTKEY()=K_F2 .OR. lcgvmov=SPACE(2)) .AND. LASTKEY() <> 27
    lcgrev := "F"
    lcgvmov := BROWPRO("MOV",LCGCMOV)
  ENDIF
  SELE (seleold)
  lcgvtmov:=lcgvmov
  seleold := SELE()
  SELE mov
  SEEK lcgvtmov
  IF FOUND()
    @13,36 SAY mov->movimiento
  ENDIF
  SELE (seleold)
ENDIF
@ 14,36 GET lcgvgrupo PICT "@!"
@ 15,36 GET lcgvteo PICT "####"
@ 16,36 GET lcgvpra PICT "####"
@ 17,36 GET lcgvhor PICT "@!"
READ
IF lcgvhor <> lcghold
  @ 18,14 SAY "Lunes:" " GET lcgvlun PICT "@!"
  @ 18,45 SAY "Martes:" GET lcgvmar PICT "@!"
  @ 19,14 SAY "Miercoles:" GET lcgvmie PICT "@!"
  @ 19,45 SAY "Jueves:" GET lcgvjue PICT "@!"
  @ 20,14 SAY "Viernes:" GET lcgvvie PICT "@!"
  @ 20,45 SAY "Sabado:" GET lcgvsab PICT "@!"
  READ
ENDIF
IF (LASTKEY() <> T_BORRAR .AND. LASTKEY() <> K_ESC)
  lcgres = 1
  lcgspan = SAVESCREEN(0,0,24,79)
  CUADRO(lcgtinte2,15,30,18,50,"s")
  @16,34 SAY " CORRECTO ?"
  @17,32 PROMPT "SI"
  @17,36 PROMPT "NO"
  @17,39 PROMPT "CANCELAR"
  MENU TO lcgres
  RESTSCREEN(0,0,24,79,lcgspan)
  SETCOLOR(lcgtinte)
  DO CASE
  CASE lcgres = 1
    SELE prop
    SET ORDER TO 1
    SEEK lcgRFC
    lcgtot = lcgvteo + lcgvpra
    IF FOUND()
      DO WHILE prop->rfc=lcgRFC
        IF prop->rfc + prop->flag = lcgkey
          lcgtot = lcgtot
        ELSE
          lcgtot = lcgtot + prop->teorica + prop->practica
        ENDIF
      ENDIF
    ENDIF
  ENDIF

```

```

                SKIP
            ENDDO
        ENDIF
        IF lcgctot > lcgbanh
        DO WHILE lcgctot > lcgbanh .AND. LASTKEY() <= 27
            sequir = .T.
            lcghrs = lcgctot
            ALERT("LAS HORAS ASIGNADAS SON MAYORES QUE LAS
AUTORIZADAS")
            *
            ELIMINAR LA SIGUIENTE LINEA
            @ 9,72 GET lcgbanh      PICT "####N"
            READ
            ENDDO
            /* BORRRAR LAS OCHO LINEAS SIGUIENTES*/
            SELE coor
            SEEK lcgkey
            IF FOUND()
            IF RLOCK()
                REPLACE coor->banco_hora WITH lcgbanh
            ENDIF
            ENDIF
            ELSE
            SELE prop
            SET ORDER TO 1
            SEEK lcgkey
            IF !FOUND()
                APPEND BLANK
            ENDIF
            IF RLOCK()
                REPLACE prop->rfc WITH lcgRFC
                REPLACE prop->cve_asig WITH lcgvcvea
                REPLACE prop->asignatura WITH lcgvasig
                REPLACE prop->grupo WITH lcgvgrupo
                REPLACE prop->tipomov WITH lcgvtmov
                REPLACE prop->teorica WITH lcgvteo
                REPLACE prop->practica WITH lcgvpra
                REPLACE prop->horario WITH lcgvhor
                REPLACE prop->lunes WITH lcgvlun
                REPLACE prop->martes WITH lcgvmar
                REPLACE prop->miercoles WITH lcgvmie
                REPLACE prop->jueves WITH lcgvjue
                REPLACE prop->viernes WITH lcgvvie
                REPLACE prop->sabado WITH lcgvsab
                REPLACE prop->flag WITH lcgflag
            ENDIF
            SELE pman
            SEEK lcgkey
            IF !FOUND()
                APPEND BLANK
            ENDIF
            IF RLOCK()
                REPLACE pman->rfc WITH lcgRFC
                REPLACE pman->cve_asig WITH lcgvcvea
                REPLACE pman->asignatura WITH lcgvasig
                REPLACE pman->grupo WITH lcgvgrupo
                REPLACE pman->tipomov WITH lcgvtmov
                REPLACE pman->teorica WITH lcgvteo
                REPLACE pman->practica WITH lcgvpra
                REPLACE pman->horario WITH lcgvhor
                REPLACE pman->lunes WITH lcgvlun
                REPLACE pman->martes WITH lcgvmar
                REPLACE pman->miercoles WITH lcgvmie
                REPLACE pman->jueves WITH lcgvjue
                REPLACE pman->viernes WITH lcgvvie
                REPLACE pman->sabado WITH lcgvsab
            ENDIF
        
```



```
KEYBOARD CHR(27)+CHR(T_BORRAR)
RETURN nil
```

```
*/ EOF()
```

```
/******
```

```
* MOV_AYUDM()
```

```
*
```

```
* Funcion que activa el ENTER
```

```
*****/
```

```
FUNCTION MOV_AYUDM
```

```
MEMVAR getlist.lcgmovvar
```

```
lcgmovvar=SPACE(2)
```

```
KEYBOARD CHR(K_ENTER)
```

```
RETURN nil
```

```

*****
* PROGRAM-ID:      Prof_Rev.prg ver 1.0      llamado por Prof_Pro.prg
* AUTHOR:         Leticia Cerd Garrido
* REMARKS:        Programa que Muestra el Menú para Revisar la
*                 Informacion de Profesores
* MODIFIED:       08 Junio, 1996.
*****

```

```

MEMVAR getlist,legtinte,lcgopr,titulo,texto,lcgpcntr,lcgpcnpl,lcgruta

```

```

CUADRO(legtinte,11,39,17,56,"S")

```

```

lcgopr=1

```

```

DO WHILE lcgopr#0

```

```

  TITULO:={"PROP.AUTOMATICAS"," PROP.MANUALES " ,"AUMENTO DE HORAS",;
          "PROP.SIN HORARIO"," S A L I R " }

```

```

  TEXTO :={"REVISAS LAS PROPUESTAS AUTOMATIZADAS",;
          "REVISAS LAS PROPUESTAS QUE SE DEBEN HACER MANUALMENTE",;
          "REVISAS LOS PROFESORES QUE TIENEN AUMENTO DE HORAS",;
          "REVISAS LOS PROFESORES QUE NO TIENEN HORARIO",;
          "SALIR AL MENU ANTERIOR"}

```

```

  MENUV(11,38,TITULO,TEXTO)

```

```

  SET KEY 19 TO

```

```

  SET KEY 4 TO

```

```

  MENU TO lcgopr

```

```

  lcgpcntr=SAVSCREEN(0,0,24,79)

```

```

  DO CASE

```

```

    CASE lcgopr=1

```

```

      DO PROF_AUT

```

```

    CASE lcgopr=2

```

```

      *ALERT("LAS SIGUIENTES PROPUESTAS DEBEN REALIZARSE
MANUALMENTE")

```

```

    DO PROF_MAN

```

```

    CASE lcgopr=3

```

```

      ALERT("LOS SIGUIENTES PROFESORES EXCEDEN SUS HORAS
AUTORIZADAS")

```

```

      PROF_EXC()

```

```

    CASE lcgopr=4

```

```

      ALERT("LOS SIGUIENTES PROFESORES NO TIENEN HORARIO ASIGNADO")

```

```

      PROF_SHR()

```

```

    CASE lcgopr=5.OR.LASTKEY()=27

```

```

      RESTSCREEN(0,0,24,79,lcgpcnpl)

```

```

      RETURN

```

```

  ENDCASE

```

```

  RESTSCREEN(0,0,24,79,lcgpcntr)

```

```

ENDDO

```

```

RESTSCREEN(0,0,24,79,lcgpcnpl)

```

```

RETURN

```

```

/*****

```

```

* PROF_EXC()

```

```

* Función que realiza la búsqueda de los profesores que tienen
* mas horas asignadas que sus horas autorizadas

```

```

*****/

```

```

FUNCTION PROF_EXC()

```

```

MEMVAR get11E,lcgruta,lcgrfc,lcgrev,lcgcampo

```

```

LOCAL thefield,banentry,lcgnom,lcgbanhor,lcghoract,;

```

```

lcgvteor,lcgvprac,lcgbanh

```

```

IF !FILE("%lcgruta\EXC_HRS.DBF")

```

```

  TheField := {}

```

```

  ADD(TheField,{"RFC","C",13,0}) && 1

```

```

  ADD(TheField,{"NOM_PRO","C",32,0}) && 2

```

```

  ADD(TheField,{"HRS_AUT","N",04,1}) && 3

```

```

  ADD(TheField,{"HRS_ASIG","N",04,1}) && 4

```

```

  SET DEFA TO lcgruta

```

```

DBCREATE("EXC_HRS",TheField)
RELEASE thefield
ENDIF
USE &lcgruta\exc_hrs ALIAS exc NEW EXCL ; ZAP
SELE coor
SET ORDER TO 1
GO TOP
DO WHILE (EOF())
  lcgrfc=coor->rfc
  legnom=coor->nom_pro
  BANENTRY := BAPRHOR(lcgrfc)
  SELE coor :
  IF banentry:=NIL .AND. LEN(banentry)>0
    lcgbanh := BANENTRY[1]
    lcgvteor := BANENTRY[7]
    lcgvprac := BANENTRY[8]
  ELSE
    lcgbanh := 0
    lcgvteor := 0
    lcgvprac := 0
  ENDIF
  RELEASE BANENTRY
  lcghoract := lcgvteor + lcgvprac
  lcgbanhor := lcgbanh
  IF lcghoract > lcgbanhor
    SELE exc
    APPEND BLANK
    REPLA exc->rfc WITH lcgrfc ,exc->nom_pro WITH legnom
    REPLA exc->hrs_aut WITH lcgbanhor,exc->hrs_asig WITH lcghoract
  ENDIF
  SELE coor
  SKIP
ENDDO
SELE exc
DECLARE LCGCAMPO[4]
LCGCAMPO[1] := "RFC"
LCGCAMPO[2] := "NOM_PRO"
LCGCAMPO[3] := "HRS_AUT"
LCGCAMPO[4] := "HRS_ASIG"
lcgrev := .F.
BROWPRO("EXC",LCGCAMPO)
CLOSE exc
RELEASE thefield,lcgcampo,banentry,legnom,lcgbanhor,lcghoract,;
lcgvteor,lcgvprac,lcgbanh
RETURN nil

```

```

/*****
* PROF_SHR()
* Función que realiza la búsqueda de los profesores que no tienen
* horario asignado
*****/
FUNCTION PROF_SHR()
MEMVAR
getlist,lcgruta,lcgrfc,lcgrev,legnom,vm_ano_linea,vm_periodo_linea,;
lcgcampo
LOCAL thefield,entrega
IF !FILE("&lcgruta\SIN_HOR.DBF")
  TheField := {}
  AADD(TheField,{"RFC", "C",13,0} ) && 1
  AADD(TheField,{"NOM_PRO", "C",32,0} ) && 2
  SET DEFA TO &lcgruta
  DBCREATE("SIN_HOR",TheField)
  RELEASE thefield
ENDIF

```

```
USE &lcgruta\sin_hor ALIAS sinh NEW EXCL ; ZAP
SELE coor
SET ORDER TO 1
GO TOP
DO WHILE !EOF()
  lcgrfc=coor->rfc
  lcgnom=coor->nom_pro
  ENTREGA := HORA_RFC_HOEX(lcgrfc,vm_anio_linea,vm_periodo_linea)
  IF ENTREGA=NIL .AND. LEN(ENTREGA)<=0
    SELE sinh
    APPEND BLANK
    REPLA sinh->rfc WITH lcgrfc,sinh->nom_pro WITH lcgnom
  ENDIF
  RELEASE ENTREGA
SELE coor
SKIP
ENDDO
SELE sinh
DECLARE LCGCAMPO[2]
LCGCAMPO[1] := "RFC"
LCGCAMPO[2] := "NOM_PRO"
lcgrev := .F.
BROWPRO("SINH",LCGCAMPO)
RELEASE LCGCAMPO
CLOSE sinh
RETURN nil

//EOF()
```

```

* .....
* PROGRAM-ID:   Prof Tar.prg ver 1.0      llamado por Prof_Imp.prg
* AUTHOR:      LetiCia Cerd Garrido
* REMARKS:     Programa que Muestra el Mené para Imprimir
*              las Tarjetas de Profesores
* MODIFIED:    07 Julio, 1996
* .....

```

MEMVAR

```

getlist,lcgtinte,lcgopi2,lcgopic,titulo,texto,lcgpponi2,lcgpponi,;
lcgren,lcgcol,lcgmesi,lcgmes1,lcgtinte2,colorold,lcgmes,lcgmes2,
;
lcgmes3,lcgmes4,lcgmes5,lcgmes6,lcgruta

```

```

PUBLIC
lcgren,lcgcol,lcgmesi,lcgmes1,lcgmes2,lcgmes3,lcgmes4,lcgmes5,lcgmes6

```

```

CUADRO(lcgtinte,12,48,16,76,"S") lcgren=0 ;lcgcol=0 ;lcgmesi=SPACE(10)
lcgmes1=1

```

SET CURSOR ON

```

@13,49 SAY "APARTIR DEL MES:"

```

```

@13,66 GET lcgmes1 PICT "@" VALID lcgmes1 $ 'ENERO      ,FEBRERO
,MARZO      ,ABRIL      ,MAYO      ,JUNIO     ,JULIO     ,AGOSTO
,SEPTIEMBRE,OCTUBRE  ,NOVIEMBRE ,DICIEMBRE ,

```

READ

```

lcgppmes=SAVESCREEN(0,0,24,79)

```

```

colorold=SETCOLOR()

```

```

DO WHILE lcgmesi=SPACE(10).AND.LASTKEY()<>27

```

```

CUADRO(lcgtinte2,08,66,21,77)

```

```

@09,67 PROMPT "ENERO      " ;@10,67 PROMPT "FEBRERO  "
@11,67 PROMPT "MARZO      " ;@12,67 PROMPT "ABRIL    "
@13,67 PROMPT "MAYO       " ;@14,67 PROMPT "JUNIO    "
@15,67 PROMPT "JULIO      " ;@16,67 PROMPT "AGOSTO   "
@17,67 PROMPT "SEPTIEMBRE" ;@18,67 PROMPT "OCTUBRE  "
@19,67 PROMPT "NOVIEMBRE " ;@20,67 PROMPT "DICIEMBRE "

```

```

SET KEY 19 TO ; SET KEY 4 TO

```

```

MENU TO lcgmesi

```

DO CASE

```

CASE lcgmesi=1 ;lcgmes1='ENERO      '
CASE lcgmesi=2 ;lcgmes1='FEBRERO   '
CASE lcgmesi=3 ;lcgmes1='MARZO     '
CASE lcgmesi=4 ;lcgmes1='ABRIL     '
CASE lcgmesi=5 ;lcgmes1='MAYO      '
CASE lcgmesi=6 ;lcgmes1='JUNIO     '
CASE lcgmesi=7 ;lcgmes1='JULIO     '
CASE lcgmesi=8 ;lcgmes1='AGOSTO    '
CASE lcgmesi=9 ;lcgmes1='SEPTIEMBRE'
CASE lcgmesi=10;lcgmes1='OCTUBRE  '
CASE lcgmesi=11;lcgmes1='NOVIEMBRE '
CASE lcgmesi=12;lcgmes1='DICIEMBRE '

```

ENDCASE

ENDDO

DO CASE

```

CASE lcgmes1='ENERO      ' ;lcgmes2='FEBRERO   ' ;lcgmes3='MARZO     '
lcgmes4='ABRIL     ' ;lcgmes5='MAYO      ' ;lcgmes6='JUNIO     '
CASE lcgmes1='FEBRERO   ' ;lcgmes2='MARZO     ' ;lcgmes3='ABRIL     '
lcgmes4='MAYO      ' ;lcgmes5='JUNIO     ' ;lcgmes6='JULIO     '
CASE lcgmes1='MARZO     ' ;lcgmes2='ABRIL     ' ;lcgmes3='MAYO      '
lcgmes4='JUNIO     ' ;lcgmes5='JULIO     ' ;lcgmes6='AGOSTO    '
CASE lcgmes1='ABRIL     ' ;lcgmes2='MAYO      ' ;lcgmes3='JUNIO     '
lcgmes4='JULIO     ' ;lcgmes5='AGOSTO    ' ;lcgmes6='SEPTIEMBRE'
CASE lcgmes1='MAYO      ' ;lcgmes2='JUNIO     ' ;lcgmes3='JULIO     '
lcgmes4='AGOSTO    ' ;lcgmes5='SEPTIEMBRE' ;lcgmes6='OCTUBRE  '
CASE lcgmes1='JUNIO     ' ;lcgmes2='JULIO     ' ;lcgmes3='AGOSTO    '
lcgmes4='SEPTIEMBRE' ;lcgmes5='OCTUBRE  ' ;lcgmes6='NOVIEMBRE '
CASE lcgmes1='JULIO     ' ;lcgmes2='AGOSTO    ' ;lcgmes3='SEPTIEMBRE'
lcgmes4='OCTUBRE  ' ;lcgmes5='NOVIEMBRE ' ;lcgmes6='DICIEMBRE '

```

```

CASE lcgmes1='AGOSTO' ;;lcgmes2='SEPTIEMBRE';lcgmes3='OCTUBRE' ;
lcgmes4='NOVIEMBRE';lcgmes5='DICIEMBRE' ;;lcgmes6='ENERO'
CASE lcgmes1='SEPTIEMBRE';lcgmes2='OCTUBRE' ;;lcgmes3='NOVIEMBRE'
lcgmes4='DICIEMBRE' ;;lcgmes5='ENERO' ;;lcgmes6='FEBRERO'
CASE lcgmes1='OCTUBRE' ;;lcgmes2='NOVIEMBRE' ;;lcgmes3='DICIEMBRE'
lcgmes4='ENERO' ;;lcgmes5='FEBRERO' ;;lcgmes6='MARZO'
CASE lcgmes1='NOVIEMBRE' ;;lcgmes2='DICIEMBRE' ;;lcgmes3='ENERO'
lcgmes4='FEBRERO' ;;lcgmes5='MARZO' ;;lcgmes6='ABRIL'
CASE lcgmes1='DICIEMBRE' ;;lcgmes2='ENERO' ;;lcgmes3='FEBRERO'
lcgmes4='MARZO' ;;lcgmes5='ABRIL' ;;lcgmes6='MAYO'

ENDCASE
SETCOLOR(colorold)
RESTSCREEN(0,0,24,79,lcgpmes)
@13,66 SAY lcgmes1
@14,49 SAY "REGLON INICIAL;" GET lcgren PICT "99" RANGE -9,10
@15,49 SAY "COLUMNA INICIAL;" GET lcgcol PICT "99" RANGE -9,10
READ
RESTSCREEN(0,0,24,79,lcgpponi)
SET CURSOR OFF
CUADRO(lcginte,12,63,16,76,"S") ;lcgopi2=1
@13,70 SAY "CARA A";@14,70 SAY "CARA B";@15,70 SAY "AMBAS "
DO WHILE lcgopi2<=0
TITULO:={"TODOS","RANGO","SALIR"}
TEXTO :={"IMPRIMIR TODOS","IMPRIMIR UN RANGO","SALIR AL MENU
ANTERIOR"}
MENUV(12,62,TITULO,TEXTO)
SET KEY 19 TO ; SET KEY 4 TO
MENU TO lcgopi2
lcgopic=1
DO WHILE lcgopic<=0.AND.LASTKEY()<=27
IF lcgopi2<=3
TITULO:={"CARA A","CARA B","AMBAS "}
TEXTO :={"IMPRIMIR CARA A DE LAS TARJETAS";
"IMPRIMIR CARA B DE LAS TARJETAS";
"IMPRIMIR AMBAS CARAS DE LAS TARJETAS"}
MENUV(12,68,TITULO,TEXTO)
SET KEY 19 TO ; SET KEY 4 TO
MENU TO lcgopic
ENDIF
lcgpponi2=SAVESCREEN(0,0,24,79)
DO CASE
CASE lcgopi2=1.AND.lcgopic=1 ; TARAIMPT()
CASE lcgopi2=1.AND.lcgopic=2 ; TARBIMPT()
CASE lcgopi2=1.AND.lcgopic=3 ; TARDIMPT()
CASE lcgopi2=2.AND.lcgopic=1 ; TARAIMPR()
CASE lcgopi2=2.AND.lcgopic=2 ; TARBIMPR()
CASE lcgopi2=2.AND.lcgopic=3 ; TARDIMPR()
CASE lcgopi2=3.OR.LASTKEY()=27
RESTSCREEN(0,0,24,79,lcgpponi)
RELEASE
lcgopi2,lcgopic,titulo,texto,lcgpponi2,lcgpponi,lcgren;;
lcgcol,lcgmes1,lcgmes1,colorold,lcgpmes,lcgmes2,lcgm
es3;;
lcgmes4,lcgmes5,lcgmes6,lcgrest,lcgpold,lcgban,lcgre
v,;
selectold,lcg_scm,lcg_any
RETURN
ENDCASE
lcgopic=0
RESTSCREEN(0,0,24,79,lcgpponi2)
ENDDO
ENDC
RESTSCREEN(0,0,24,79,lcgpponi)
RELEASE lcgopi2,lcgopic,titulo,texto,lcgpponi2,lcgpponi,lcgren,lcgcol;;
lcgmes1,lcgmes1,colorold,lcgpmes,lcgmes2,lcgmes3,lcgmes4,lcgmes5

```

```
..
      lcgmes6,lcgrest,lcgpold,lcgban,lcgrev,seleold,lcg_scm,lcg_any
RETURN
```

```
/******
 * TARAIMPT()
 *
 * Funci3n que imprime todas las tarjetas de asistencia cara a
 *****/
FUNCTION TARAIMPT()
MEMVAR getlist,lcgpunam,lcgunam,lcgrfcp,lcgtinte,lcgpconi,lcgopic
CLRSCR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
CUADRO(lcgunam,11,9,14,33,"S")
@12,10 SAY "CONTROL DE ASISTENCIA" ; @13,11 SAY "IMPRIMIENDO
TARJETAS"
CUADRO(lcgunam,16,40,19,70,"S")
@17,41 SAY "ESPERE...PROCESO EN EJECUCION" ; @18,46 SAY "GRACIAS POR
ESPERAR"
SELE coor
GO TOP
DO WHILE !EOF()
  lcgrfcp=coor->rfc
  PROFTARA()
  SELE coor
  SKIP
ENDDO
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpconi)
RETURN nil
```

```
/******
 * TARBIMPT()
 *
 * Funci3n que imprime todas las tarjetas de asistencia cara b
 *****/
FUNCTION TARBIMPT()
MEMVAR getlist,lcgpunam,lcgunam,lcgrfcp,lcgtinte,lcgpconi,lcgopic
CLRSCR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
CUADRO(lcgunam,11,9,14,33,"S")
@12,10 SAY "CONTROL DE ASISTENCIA" ; @13,11 SAY "IMPRIMIENDO
TARJETAS"
CUADRO(lcgunam,16,40,19,70,"S")
@17,41 SAY "ESPERE...PROCESO EN EJECUCION" ; @18,46 SAY "GRACIAS POR
ESPERAR"
SELE coor
GO TOP
DO WHILE !EOF()
  lcgrfcp=coor->rfc
  PROFTARB()
  SELE coor
  SKIP
ENDDO
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpconi)
RETURN nil
```

```
/******
 * TARDIMPT()
 *
 * Funci3n que imprime todas las tarjetas de asistencia ambas caras
 *****/
FUNCTION TARDIMPT()
MEMVAR getlist,lcgpunam,lcgunam,lcgrfcp,lcgtinte,lcgpconi,lcgopic ;
```

```

lcgntinte2,lcgrest,lcgpold
CLRSKR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
CUADRO(lcgunam,11,9,14,33,"S")
@12,10 SAY "CONTROL DE ASISTENCIA" ; @13,11 SAY "IMPRIMIENDO
TARJETAS"
CUADRO(lcgunam,16,40,19,70,"S")
@17,41 SAY "ESPERE...PROCESO EN EJECUCION" ; @18,46 SAY "GRACIAS POR
ESPERAR"
SELE coor
GO TOP
DO WHILE !EOF()
  lcgrfcp=coor->rfc
  PROFTARA()
  SELE coor
  SKIP
ENDDO
lcgrest = 1
lcgpold = SAVESCREEN(0,0,24,79)
CUADRO(lcgntinte2,10,13,14,65,"S")
@11,14 SAY 'IMPRESION DE LA CARA "A" DE LAS TARJETAS CONCLUIDA'
@12,14 SAY 'PREPARE LA CARA "B" DE LAS TARJETAS PARA CONTINUAR'
@13,24 PROMPT 'CONTINUAR'
@13,46 PROMPT 'CANCELAR'
MENU TO lcgrest
RESTSCREEN(0,0,24,79,lcgpold)
DO CASE
  CASE lcgrest = 1
    GO TOP
    DO WHILE !EOF()
      lcgrfcp=coor->rfc
      PROFTARB()
      SELE coor
      SKIP
    ENDDO
  CASE lcgrest =2
    SETCOLOR(lcgntinte)
    RESTSCREEN(0,0,24,79,lcgpconi)
    RETURN nil
ENDCASE
SETCOLOR(lcgntinte)
RESTSCREEN(0,0,24,79,lcgpconi)
RETURN nil

/*****
* TARAIMPR()
* Función que imprime un rango de tarjetas de asistencia cara a
*****/
FUNCTION TARAIMPR()
MEMVAR
getlist,lcgpunam,lcgrfci,lcgrfcf,lcgban,lcgunam,lcgntinte,lcgrfcp,;
lcgpconi
RAN TARI()
SELE coor
SEEK lcgrfci
IF FOUND()
  DO WHILE coor->rfc=lcgrfci.AND.coor->rfc<=lcgrfcf
    lcgrfcp=coor->rfc
    PROFTARA()
    SELE coor
    SKIP
  ENDDO
ENDIF
SETCOLOR(lcgntinte)

```

```

RESTSCREEN(0,0,24,79,lcgpconi)
RETURN nil

/*****
* TARBIMPR()
* Función que imprime un rango de tarjetas de asistencia cara b
*****/
FUNCTION TARBIMPR()
MEMVAR
getlist,lcgpunam,lcgrfci,lcgrfcf,lcgban,lcgunam,lcgtinte,lcgrfcf,;
lcgpconi
RAN TARI()
SELE coor
SEEK lcgrfci
IF FOUND()
    DO WHILE coor->rfc=>lcgrfci.AND.coor->rfc<=>lcgrfcf
        lcgrfcf=coor->rfc
        PROFTARB()
        SELE coor
        SKIP
    ENDDO
ENDIF
SETCOLOR(lcgtinte)
RESTSCREEN(0,0,24,79,lcgpconi)
RETURN nil

/*****
* TARDIMPR()
* Función que imprime un rango de tarjetas de asistencia ambas caras
*****/
FUNCTION TARDIMPR()
MEMVAR
getlist,lcgpunam,lcgrfci,lcgrfcf,lcgban,lcgunam,lcgtinte,lcgrfcf,;
lcgpconi,lcgrest,lcgpold,lcgtinte2
RAN TARI()
SELE coor
SEEK lcgrfci
IF FOUND()
    DO WHILE coor->rfc=>lcgrfci.AND.coor->rfc<=>lcgrfcf
        lcgrfcf=coor->rfc
        PROFTARA()
        SELE coor
        SKIP
    ENDDO
ENDIF
lcgrest = 1
lcgpold = SAVESCREEN(0,0,24,79)
CUADRO(lcgtinte2,10,13,14,65,"S")
@11,14 SAY 'IMPRESION DE LA CARA "A" DE LAS TARJETAS CONCLUIDA'
@12,14 SAY 'PREPARE LA CARA "B" DE LAS TARJETAS PARA CONTINUAR'
@13,24 PROMPT 'CONTINUAR'
@13,46 PROMPT 'CANCELAR'
MENU TO lcgrest
RESTSCREEN(0,0,24,79,lcgpold)
DO CASE
CASE lcgrest = 1
SEEK lcgrfci
IF FOUND()
    DO WHILE coor->rfc=>lcgrfci.AND.coor->rfc<=>lcgrfcf
        lcgrfcf=coor->rfc
        PROFTARB()
        SELE coor
        SKIP
    ENDDO

```

```

        ENDDO
    ENDIF
    CASE lcgrstat =2
        SETCOLOR(lcgrtinte)
        RESTSCREEN(0,0,24,79,lcgpconi)
        RETURN nil
    ENDCASE
    SETCOLOR(lcgrtinte)
    RESTSCREEN(0,0,24,79,lcgpconi)
    RETURN nil

/*****
* RAN_TARI()
* Función que selecciona un rango de tarjetas
*
*****/
FUNCTION RAN_TARI()
MEMVAR
getlist,lcgpunam,lcgrfci,lcgrfcf,lcgban,lcgunam,lcgtinte,lcgrfcf,
lcgpconi,lcgrev,lcgcampo,seleold
PUBLIC lcgrfci,lcgrfcf

#include "inkey.ch"
CLRSCR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
lcgrfci:=SPACE(13); lcgrfcf:=SPACE(13); lcgban=0
CUADRO(lcgunam,9,10,12,40,"S")
@10,14 SAY "CONTROL DE ASISTENCIA" ; @11,15 SAY "IMPRIENDO
TARJETAS"
CUADRO(lcgunam,13,25,17,55,"S")
DO WHILE lcgban=0.AND.LASTKEY()<>27
    DECLARE LCGCAMPO[2]
    LCGCAMPO[1] := "RFC"
    LCGCAMPO[2] := "NOM PRO"
    SET KEY K_F2 TO COD_AYUDA
    SET CURSOR ON
    @14,26 SAY "R.F.C. INICIAL:" GET lcgrfci PICT "@" ; READ
    IF (LASTKEY() = K_F2 .OR. lcgrfci = SPACE(13)).AND. LASTKEY() <> 27
        lcgrfcf := "F"
        seleold:= SELE()
        SELE coor
        SET FILTER TO coor->propuesta="*"
        GO TOP
        lcgrfcf := BROWPRO("COOR",LCGCAMPO)
        SET FILTER TO
        SELE(seleold)
        @14,42 SAY lcgrfci PICT "@"
    ENDF
    RELEASE LCGCAMPO
    SET KEY K_F2 TO
    IF (LASTKEY() <> 27 .AND. LEN(lcgrfcf) <> 0)
        SELE prop
        SET ORDER TO 1
        SEEK lcgrfci
        IF !FOUND() ; ALERT("NO EXISTE EN EL ARCHIVO DE PROPUESTAS")
            lcgrfci:=SPACE(13)
        ELSE ; lcgban=1 ;ENDIF
    ENDF
ENDDO
lcgban=0
DO WHILE lcgban=0.AND.LASTKEY()<>27
    DECLARE LCGCAMPO[2]
    LCGCAMPO[1] := "RFC"
    LCGCAMPO[2] := "NOM PRO"
    SET KEY K_F2 TO COD_AYUDA

```

```

@16,26 SAY "R.F.C. FINAL: " GET lcgrfcf PICT "@!"; READ
IF (LASTKEY() = K_F2 .OR. lcgrfcf = SPACE(13)) .AND. LASTKEY() <> 27
  lcgrrev := .F.
  seleold := SELE()
  SELE coor
  SET FILTER TO coor->propuesta-' '
  GO TOP
  lcgrfcf := BROWPRO("COOR",LCGCAMPO)
  SET FILTER TO
  SELE(seleold)
  @16,42 SAY lcgrfcf PICT "@!"
ENDIF
RELEASE LCGCAMPO
SET KEY K_F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgrfcf) <> 0)
  SELE prop
  SEEK lcgrfcf
  IF (FOUND())
    ALERT("NO EXISTE EN EL ARCHIVO DE PROPUESTAS")
;lcgrfcf=SPACE(13)
    ELSEIF lcgrfcf < lcgrfcf
      lcgrfcf = SPACE(13)
    ALERT("EL R.F.C. FINAL DEBE SER MAYOR QUE EL INICIAL")
  ELSE
    lcgban=1
  ENDIF
ENDIF
ENDDO
CUADRO(lcgunam,18,40,21,70,"S")
@19,41 SAY "ESPERE...PROCESO EN EJECUCION" ; @20,46 SAY "GRACIAS POR
ESPERAR"
RETURN nil

/*****
* PROPTARA()
* Función que realiza el formato del registro de asistencia
* del personal docente CARA A
*****/
FUNCTION PROPTARA()
MEMVAR
getlist,vm_ano_linea,lcg_sem,lcg_ano,lcgrfcf,vm_tit_carr,lcgperiodo.;
  vm_periodo_linea,lcgren,lcgcol,lcgmes1,lcgmes2,lcgmes3,incia
SET CURSOR OFF
SET CONSOLE OFF
SET DEVICE TO PRINT
SET PRINT ON
PUBLIC lcgperiodo := alltrim(vm_ano_linea)
PUBLIC lcgperiodo := substr(lcgperiodo,len(lcgperiodo) -1,2)
PUBLIC lcgperiodo := lcgperiodo + alltrim(vm_periodo_linea)
lcg_sem = " " + substr(lcgperiodo,len(lcgperiodo) -2,2) + "-" +
vm_periodo_linea + " "
lcg_ano = substr(lcgperiodo,len(lcgperiodo) -2,2)
SELE prop
SET ORDER TO 1
GO TOP
SEEK lcgrfcf
IF FOUND()
  DO WHILE prop->rfc=lcgrfcf
    @0,0 SAY INICIA
    @lcgren+05,lcgcol+22 SAY vm_tit_carr
    @lcgren+05,lcgcol+65 SAY lcg_sem
    @lcgren+07,lcgcol+57 SAY ALLTRIM(lcgmes1)+'/'+lcg_ano
    @lcgren+22,lcgcol+35 SAY prop->lunes
    @lcgren+23,lcgcol+35 SAY prop->martes
    @lcgren+24,lcgcol+35 SAY prop->miercoles
  
```

```

@lcgren+25,lcgcol+35 SAY prop->jueves
@lcgren+26,lcgcol+35 SAY prop->viernes
@lcgren+27,lcgcol+35 SAY prop->sabado
@lcgren+29,lcgcol+08 SAY coor->nom_pro
@lcgren+29,lcgcol+39 SAY prop->asignatura
@lcgren+29,lcgcol+72 SAY prop->grupo
@lcgren+30,lcgcol+08 SAY lcgrfcp
@lcgren+30,lcgcol+39 SAY prop->horario
@lcgren+32,lcgcol+57 SAY ALLTRIM(lcgmes2)+'/'+lcg_any
@lcgren+46,lcgcol+57 SAY ALLTRIM(lcgmes3)+'/'+lcg_any
@60,lcgcol+08 SAY coor->nom_pro
@60,lcgcol+39 SAY prop->asignatura
@60,lcgcol+72 SAY prop->grupo
EJEC
SKIP
ENDDO
ENDIF
SET PRINT OFF
SET CONSOLE ON
SET DEVICE TO SCREEN
RETURN nil

/*****
* PROFTARB()
* Función que realiza el formato del registro de asistencia
* del personal docente CARA B
*****/
FUNCTION PROFTARB()
MEMVAR
getlist,vm_anio linea,lcg_sem,lcg_any,lcgrfcp,vm_tit_carr,lcgperiodo,;
vm_periodo_linea,lcgren,lcgcol,lcgmes4,lcgmes5,lcgmes6,incia
SET CURSOR OFF
SET CONSOLE OFF
SET DEVICE TO PRINT
SET PRINT ON
PUBLIC lcgperiodo := alltrim(vm_anio linea)
PUBLIC lcgperiodo := substr(lcgperiodo,len(lcgperiodo)-1,2)
PUBLIC lcgperiodo := lcgperiodo + alltrim(vm_periodo linea)
lcg_sem = " " + substr(lcgperiodo,len(lcgperiodo)-2,2) + "-" +
vm_periodo_linea + " "
lcg_any = substr(lcgperiodo,len(lcgperiodo)-2,2)
SELE prop
SET ORDER TO 1
GO TOP
SEEK lcgrfcp
IF FOUND()
DO WHILE prop->rfc=lcgrfcp
@0,0 SAY INICIA
@lcgren+05,lcgcol+22 SAY vm_tit_carr -
@lcgren+05,lcgcol+65 SAY lcg_sem
@lcgren+07,lcgcol+57 SAY ALLTRIM(lcgmes4)+'/'+lcg_any
@lcgren+22,lcgcol+35 SAY prop->lunes
@lcgren+23,lcgcol+35 SAY prop->martes
@lcgren+24,lcgcol+35 SAY prop->miercoles
@lcgren+25,lcgcol+35 SAY prop->jueves
@lcgren+26,lcgcol+35 SAY prop->viernes
@lcgren+27,lcgcol+35 SAY prop->sabado
@lcgren+29,lcgcol+08 SAY coor->nom_pro
@lcgren+29,lcgcol+39 SAY prop->asignatura
@lcgren+29,lcgcol+72 SAY prop->grupo
@lcgren+27,lcgcol+35 SAY prop->sabado
@lcgren+30,lcgcol+39 SAY prop->horario
@lcgren+32,lcgcol+57 SAY ALLTRIM(lcgmes5)+'/'+lcg_any
@lcgren+46,lcgcol+57 SAY ALLTRIM(lcgmes6)+'/'+lcg_any
@60,lcgcol+08 SAY coor->nom_pro

```

```
@60.lcgcol+39 SAY prop->signatura
@60.lcgcol+72 SAY prop->grupo
EJEC
SKIP
ENDDC
ENDIF
SET PRINT OFF
SET CONSOLE ON
SET DEVICE TO SCREEN
RETURN nil
//EOF
```

```

*****
* PROGRAM-ID:      Prof_Rep.prg ver 1.0      llamado por Prof_Imp.prg
* AUTHOR:         Leticia Cerd Garrido
* REMARKS:        Programa que muestra el Menf de Impresión del
*                 reporte de la evaluación
* MODIFIED:       07 Junio, 1997.
*****
#include "inkey.ch"
MEMVAR getlist,lcgtinte,lcgopp,lcgpcol,lcgpcoe,lcgprev,lcgruta,lcgeva,;
lcgperiodo
LOCAL titulo,texto,lcgrsup,lcgcizq,lcgrinf,lcgder
USE &lcgruta&lcgeva INDEX &lcgruta\eva ALIAS eva NEW SHARED

lcgpcoe=SAVESCREEN(0,0,24,79)
CUADRO(lcgtinte,10,65,14,73,"S") ;lcgopp=1
DO WHILE lcgopp#0
  TITULO:={"TODOS","RANGO","SALIR"}
  TEXTO:={"IMPRESION DE TODOS","IMPRESION DE UN RANGO","REGRESAR AL
MENU ANTERIOR"}
  MENUV(10,65,TITULO,TEXTO)
  SET KEY 19 TO ; SET KEY 4 TO
  MENU TO lcgopp
  lcgpcol=SAVESCREEN(0,0,24,79)
  DO CASE
    CASE lcgopp=1; PROF_RET()
    CASE lcgopp=2; PROF_RER()
    CASE lcgopp=3.OR.LASTKEY()=27
      EXIT
    ENDCASE
  RESTSCREEN(0,0,24,79,lcgpcol)
ENDDO
RESTSCREEN(0,0,24,79,lcgpcoe)
CLOSE eva
RELEASE
lcgopp,lcgpcol,lcgpcoe,lcgprev,titulo,texto,lcgrsup,lcgcizq,lcgrinf,;
lcgder,titulo0,titulo00,titulo01,lcgpcoe,lcgban,lcgpro
fi,;
lcgproff,lcgrev,fecha,pag,lcgkey,lcgkey1,pta01,pta02,pta03,pta04
,;
pta05,ptp01,ptp02,ptp03,ptp04,ptp05,ptp06,ptp07,ptp08,ptp09,ptp1
0,;
tea1,tea2,tea3,tea4,tea5,tep1,tep2,tep3,tep4,tep5,tep6,tep7,tep8
,;
tep9,tep10,tcu,clave1,material,profesol,grupol,semestrl,pep,pea
RETURN

/*****
* PROF_RET()
*
* Función de impresión del reporte de la evaluación de todos los
profesores
*****/
FUNCTION PROF_RET
MEMVAR
getlist,lcgpunam,lcgunam,titulo0,titulo00,titulo000,titulo1,lcgtinte,;
lcgpcoe,inicia
SELE eva
GO TOP
CLRSCR (5, 130)
RESTSCREEN(0,0,24,79,lcgpunam)
CUADRO(lcgunam,11,9,15,39,"S")
@12,13 SAY "EVALUACION A PROFESORES"
@14,12 SAY "REPORTE POR GRUPO/MATERIA"
CUADRO(lcgunam,16,40,19,70,"S")

```

```

@17,41 SAY "ESPERE...IMPRESION EN PROCESO"
@18,46 SAY "GRACIAS POR ESPERAR"
SET CURSOR OFF
SET CONSOLE OFF
SET DEVICE TO PRINT
SET PRINT ON
DO WHILE !EOF()
  @0,0 SAY INICIA
  REPO_EVA()
ENDDO
SETCOLOR(1cgtinte)
SET PRINT OFF
SET CONSOLE ON
SET DEVICE TO SCREEN
RESTSCREEN(0,0,24,79,1cgpcoe)
RETURN nil

/*****
* PROF_RER()
*
* Función que imprime el reporte de la evaluación para un rango de
* profesores
*****/
FUNCTION PROF_RER()
MEMVAR
getlist, lcgpunam, lcgban, lcgprofi, lcgproff, lcgnum, lcgtime, titulo0, :
        titulo00, titulo000, titulo1, lcgpcol, lcgcampo, lcgrev, inicia
RESTSCREEN(0,0,24,79,1cgpunam)
lcgban=0 ; lcgprofi=SPACE(30) ; lcgproff=SPACE(30)
CUADRO(lcgnum,9,10,12,40,"S")
@10,13 SAY "EVALUACION A PROFESORES "
@11,14 SAY "IMPRESION DE UN RANGO"
CUADRO(lcgnum,13,25,17,75,"S")
DO WHILE lcgban=0.AND.LASTKEY() <>27
  DECLARE LCGCAMPO[2]
  LCGCAMPO[1] := "PROFESOR"
  LCGCAMPO[2] := "RFC"
  SET KEY K F2 TO COD_AYUDA
  SET CURSOR ON
  @14,26 SAY "PROFESOR INICIAL:" GET lcgprofi PICT "@!" ; READ
  IF (LASTKEY() = K_F2 .OR. lcgprofi = SPACE(30)).AND. LASTKEY() <> 27
    lcgrev := .F.
    lcgprofi := BROWPRO("EVA",LCGCAMPO)
    @14,44 SAY lcgprofi PICT "@!"
  ENDIF
  RELEASE LCGCAMPO
  SET CURSOR OFF
  SET KEY K F2 TO
  IF (LASTKEY() <> 27 .AND. LEN(lcgprofi) <> 0)
    SELE eva
    SEEK lcgprofi
    IF !FOUND() ; ALERT("NO EXISTE EN EL ARCHIVO DE EVALUACIONES")
      lcgprofi=SPACE(30)
    ELSE ; lcgban=1 ;ENDIF
  ENDIF
ENDIF
ENDDO
lcgban=0
DO WHILE lcgban=0.AND.LASTKEY() <>27
  DECLARE LCGCAMPO[2]
  LCGCAMPO[1] := "PROFESOR"
  LCGCAMPO[2] := "RFC"
  SET KEY K F2 TO COD_AYUDA
  SET CURSOR ON
  @16,26 SAY "PROFESOR FINAL: " GET lcgproff PICT "@!" ; READ

```

```

IF (LASTKEY() = K_F2 .OR. lcgproff = SPACE(30)).AND. LASTKEY() <> 27
  lcgrev := .F.
  lcgproff := BROWPRO("EVA",LCGCAMPO)
  @16,44 SAY lcgproff PICT "@!"
ENDIF
RELEASE LCGCAMPO
SET CURSOR OFF
SET KEY K_F2 TO
IF (LASTKEY() <> 27 .AND. LEN(lcgproff) <> 0)
  SELE eva
  SEEK lcgproff
  IF !FOUND()
    ALERT("NO EXISTE EN EL ARCHIVO DE EVALUACIONES")
  ;lcgproff=SPACE(30)
  ELSEIF lcgproff < lcgprofi
    lcgproff = SPACE(30)
    ALERT("EL PROFESOR FINAL DEBE SER MAYOR QUE EL INICIAL")
  ELSE
    lcgban=1
  ENDIF
ENDIF
ENDDO
CUADRO(lcguram, 18,40,21,70,"S")
@19,41 SAY "ESPERE...PROCESO EN EJECUCION"
@20,46 SAY "GRACIAS POR ESPERAR"
SELE eva
GO TOP
SET CURSOR OFF
SET CONSOLE OFF
SET DEVICE TO PRINT
SET PRINT ON
DO WHILE eva->profesor=>lcgprofi.AND.eva->profesor<=>lcgproff
  @0,0 SAY INICIA
  REPO_EVA()
ENDDO
SETCOLOR(lcgtinte)
SET PRINT OFF
SET CONSOLE ON
SET DEVICE TO SCREEN
RESTSCREEN(0,0,24,79,lcgpcol)
RETURN nil

```

```

/*****
* TIT_EVA()
*
* Funcion de encabezados del reporte de la evaluaci3n
*****/
FUNCION TIT_EVA()
MEMVAR getlist, fecha, titulo0, titulo00, titulo000, titulo1, pag
pag:=1
@PROW(), PCOL() SAY CHR(15)
TITULO0
-*****
TITULO1 -** UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
**
TITULO2 -** CAMPUS ARAGON UNIDAD DE PLANEACION
**
TITULO3 -** DEP. DE INFORMATICA EN COLABORACION CON ING. EN
COMPUTACION **
TITULO4 -** REPORTE DE EVALUACION POR PROFESOR
**
@ 1,(132-(LEN (TITULO0)))/2 SAY titulo0
@ 2,(132-(LEN (TITULO1)))/2 SAY titulo1
@ 3,(132-(LEN (TITULO2)))/2 SAY titulo2

```

```

@ 4, (132-(LEN (TITULO3)))/2 SAY titulo3
@ 5, (132-(LEN (TITULO4)))/2 SAY titulo4
@ 6, (132-(LEN (TITULO6)))/2 SAY titulo6
RETURN nil

/*****
* REPO_EVA()
*
* Función que realiza el reporte de la evaluación
*****/
FUNCION REPO EVA()
MEMVAR getlist.lcgkey, lcgkey1, lcgpcoe
LOCAL
pta01, pta02, pta03, pta04, pta05, ptp01, ptp02, ptp03, ptp04, ptp05, ptp06, pep, ;
pcp07, pcp08, pcp09, ptp10, teal, tea2, tea3, tea4, tea5, tep1, tep2, tep3, te
p4, ; tep5, tep6, tep7, tep8, tep9, tep10, tcu, clavel, material, profesol, grupol
, pea
lcgkey=eva->profesor
pta01=0;pta02=0;pta03=0;pta04=0;pta05=0;ptp01=0;ptp02=0;ptp03=0
ptp04=0;ptp05=0;ptp06=0;ptp07=0;ptp08=0;ptp09=0;ptp10=0
DO WHILE eva->profesor.lcgkey.AND.1.EOF()
teal=0; mina01=9.85; maxa01=5.45; tea2=0; mina02=9.85; maxa02=5.45
tea3=0; mina03=9.85; maxa03=5.45; tea4=0; mina04=9.85; maxa04=5.45
tea5=0; mina05=9.85; maxa05=5.45; tep1=0; minp01=9.85; maxp01=5.45
tep2=0; minp02=9.85; maxp02=5.45; tep3=0; minp03=9.85; maxp03=5.45
tep4=0; minp04=9.85; maxp04=5.45; tep5=0; minp05=9.85; maxp05=5.45
tep6=0; minp06=9.85; maxp06=5.45; tep7=0; minp07=9.85; maxp07=5.45
tep8=0; minp08=9.85; maxp08=5.45; tep9=0; minp09=9.85; maxp09=5.45
tep10=0; minp10=9.85; maxp10=5.45; tcu=0
lcgkey1=eva->grupo+eva->clave
DO WHILE eva->grupo+eva->clave.lcgkey1.AND.1.EOF()
clave1=eva->clave ; material=eva->materia
profesol=eva->profesor ; grupol=eva->grupo ; tcu = tcu+1
tea1 = teal+eva->a01 ; tea2 = tea2+eva->a02 ; tea3 = tea3+eva-
>a03
tea4 = tea4+eva->a04 ; tea5 = tea5+eva->a05 ; tep1 = tep1+eva-
>p01
tep2 = tep2+eva->p02 ; tep3 = tep3+eva->p03 ; tep4 = tep4+eva-
>p04
tep5 = tep5+eva->p05 ; tep6 = tep6+eva->p06 ; tep7 = tep7+eva-
>p07
tep8 = tep8+eva->p08 ; tep9 = tep9+eva->p09 ; tep10=tep10+eva-
>p10
IF mina01 > eva->a01 ; mina01 = eva->a01 ; ENDF
IF maxa01 < eva->a01 ; maxa01 = eva->a01 ; ENDF
IF mina02 > eva->a02 ; mina02 = eva->a02 ; ENDF
IF maxa02 < eva->a02 ; maxa02 = eva->a02 ; ENDF
IF mina03 > eva->a03 ; mina03 = eva->a03 ; ENDF
IF maxa03 < eva->a03 ; maxa03 = eva->a03 ; ENDF
IF mina04 > eva->a04 ; mina04 = eva->a04 ; ENDF
IF maxa04 < eva->a04 ; maxa04 = eva->a04 ; ENDF
IF mina05 > eva->a05 ; mina05 = eva->a05 ; ENDF
IF maxa05 < eva->a05 ; maxa05 = eva->a05 ; ENDF
IF minp01 > eva->p01 ; minp01 = eva->p01 ; ENDF
IF maxp01 < eva->p01 ; maxp01 = eva->p01 ; ENDF
IF minp02 > eva->p02 ; minp02 = eva->p02 ; ENDF
IF maxp02 < eva->p02 ; maxp02 = eva->p02 ; ENDF
IF minp03 > eva->p03 ; minp03 = eva->p03 ; ENDF
IF maxp03 < eva->p03 ; maxp03 = eva->p03 ; ENDF
IF minp04 > eva->p04 ; minp04 = eva->p04 ; ENDF
IF maxp04 < eva->p04 ; maxp04 = eva->p04 ; ENDF
IF minp05 > eva->p05 ; minp05 = eva->p05 ; ENDF
IF maxp05 < eva->p05 ; maxp05 = eva->p05 ; ENDF
IF minp06 > eva->p06 ; minp06 = eva->p06 ; ENDF

```

```

IF maxp06 < eva->p06 ; maxp06 = eva->p06 ; ENDIF
IF minp07 > eva->p07 ; minp07 = eva->p07 ; ENDIF
IF maxp07 < eva->p07 ; maxp07 = eva->p07 ; ENDIF
IF minp08 > eva->p08 ; minp08 = eva->p08 ; ENDIF
IF maxp08 < eva->p08 ; maxp08 = eva->p08 ; ENDIF
IF minp09 > eva->p09 ; minp09 = eva->p09 ; ENDIF
IF maxp09 < eva->p09 ; maxp09 = eva->p09 ; ENDIF
IF minp10 > eva->p10 ; minp10 = eva->p10 ; ENDIF
IF maxp10 < eva->p10 ; maxp10 = eva->p10 ; ENDIF
SKIP
ENDDO
pta01=ROUND((tea1/tcu),2) ;pta02=ROUND((tea2/tcu),2)
pta03=ROUND((tea3/tcu),2) ;pta04=ROUND((tea4/tcu),2)
pta05=ROUND((tea5/tcu),2) ;pta01=ROUND((tep1/tcu),2)
ptp02=ROUND((tep2/tcu),2) ;ptp03=ROUND((tep3/tcu),2)
ptp04=ROUND((tep4/tcu),2) ;ptp05=ROUND((tep5/tcu),2)
ptp06=ROUND((tep6/tcu),2) ;ptp07=ROUND((tep7/tcu),2)
ptp08=ROUND((tep8/tcu),2) ;ptp09=ROUND((tep9/tcu),2)
ptp10=ROUND((tep10/tcu),2)
pea=ROUND(((pta01+pta02+pta03+pta04+pta05)/5),2)
pep=ROUND(((ptp01+ptp02+ptp03+ptp04+ptp05+ptp06+ptp07+ptp08+ptp09+ptp
10)/10),2)
TIT. EVA()
@08.15 SAY "GRUPO: " ; @08.23 SAY grup01
@08.30 SAY "CLAVE DE LA MATERIA: " ; @08.51 SAY Clavel
@08.55 SAY "Semestre: " ; @08.65 SAY lcgperiodo
@09.15 SAY "MATERIA: " ; @09.23 SAY material
@09.35 SAY "No. De Cuest.: " ; @09.53 SAY tcu
@09.55 SAY "No. Pag.: "-str(pag,3)
@10.15 SAY "PROFESOR: " ; @10.25 SAY profes01
@12.25 SAY "PART. I: AUTOEVALUACION DEL ALUMNO"
@12.68 SAY "EVALUACION EVALUACION EVALUACION EVALUACION"
@13.25 SAY "PREGUNTAS"
@13.67 SAY "CUANTITATIVA CUALITATIVA MINIMA MAXIMA"
@15.15 SAY "A) PLANEACION DEL CURSO"
@15.75 SAY pta01 PICT "99.99"
lcgcuali := cuali(pta01)
@15.83 SAY lcgcuali PICT "1@"
@15.98 SAY mina01 PICT "99.99"
@15.115 SAY maxa01 PICT "99.99"
@16.17 SAY "1.-EN QUE MOMENTO DEL SEMESTRE TE INTERESASTE POR
CONOCER"
@20.17 SAY " EL PROGRAMA DE LA MATERIAL ?"
@21.15 SAY "B) CONOCIMIENTO DE LA MANTERIA"
@20.75 SAY pta02 PICT "99.99"
lcgcuali := cuali(pta02)
@20.83 SAY lcgcuali PICT "1@"
@20.98 SAY mina02 PICT "99.99"
@20.115 SAY maxa02 PICT "99.99"
@21.17 SAY "2.-QUE NIVEL DE COMPRENSION CONSIDERAS QUE ALCANZASTE
SOBRE"
@22.17 SAY " LOS CONTENIDOS REVIZADOS EN EL CURSO ?"
@23.15 SAY "C) METODOLOGIAS Y ESTRATEGIAS DIDACTICAS"
@23.75 SAY pta03 PICT "99.99"
lcgcuali := cuali(pta03)
@23.83 SAY lcgcuali PICT "1@"
@23.98 SAY mina03 PICT "99.99"
@23.115 SAY maxa03 PICT "99.99"
@24.17 SAY "3.-PARTICIPASTE EN LAS DIVERSAS ACTIVIDADES DE ENSEANZA"
@25.17 SAY " APRENDIZAJE DESARROLLADAS EN CLASE ?"
@26.15 SAY "D) SEGUIMIENTO Y EVALUACION"
@26.75 SAY pta04 PICT "99.99"
lcgcuali := cuali(pta04)
@26.83 SAY lcgcuali PICT "1@"
@26.98 SAY mina04 PICT "99.99"

```

@26,115 SAY maxa04 PICT "99.99"
 @27,17 SAY "4.-CUMPLISTE CON LOS REQUISITOS ESTABLECIDOS PARA TU"
 @28,17 SAY " EVALUACION ?"
 @29,15 SAY "E) DISCIPLINA Y PUNTUALIDAD"
 @29,75 SAY pta05 PICT "99.99"
 lcgcuall := cuall(pta05)
 @30,83 SAY lcgcuall PICT "10"
 @30,98 SAY mina05 PICT "99.99"
 @30,115 SAY maxa05 PICT "99.99"
 @31,17 SAY "5.-ASISTE CON PUNTUALIDAD Y CONSTANCIA AL CURSO ?"
 @32,(132-(LEN (TITULO0)))/2 SAY titulo0
 @33,19 SAY "PROMEDIO DE LA EVALUACION DEL ALUMNO:"
 @33,75 SAY pea PICT "99.99"
 lcgcuall := cuall(pea)
 @33,83 SAY lcgcuall PICT "10"
 @34,(132-(LEN (TITULO0)))/2 SAY titulo0
 @36,25 SAY "PARTE II: EVALUACION DEL ALUMNO HACIA EL PROFESOR"
 @38,15 SAY "A) PLANEACION DEL CURSO"
 @38,75 SAY ptp01 PICT "99.99"
 lcgcuall := cuall(ptp01)
 @38,83 SAY lcgcuall PICT "10"
 @38,98 SAY minp01 PICT "99.99"
 @30,115 SAY maxa05 PICT "99.99"

 @41,15 SAY " 2.- HAY CONTINUIDAD EN EL DESARROLLO DEL CURSO:"
 @41,100 SAY ptp02 PICT "99.99"
 @42,100 SAY ptp03 PICT "99.99"
 @42,100 SAY ptp03 PICT "99.99"
 @43,15 SAY " 4.- SE CUMPLEN LAS FECHAS ESTABLECIDAS EN EL CALENDARIO
 ESCOLAR:"
 @43,100 SAY ptp04 PICT "99.99"
 @44,15 SAY " 5.- SE RESPETAN LOS HORARIOS DE CLASE:"
 @44,100 SAY ptp05 PICT "99.99"
 @45,15 SAY " 6.- CUENTA CON LOS CONOCIMIENTOS DE LA MATERIA
 NECESARIOS:"
 @45,100 SAY ptp06 PICT "99.99"
 @46,15 SAY " 7.- HAY UNA SECUENCIA COHERENTE DE LOS CONTENIDOS:"
 @46,100 SAY ptp07 PICT "99.99"
 @47,15 SAY " 8.- LOS CONTENIDOS ESTAN ACTUALIZADOS:"
 @47,100 SAY ptp08 PICT "99.99"
 @48,15 SAY " 9.- SE CONCLUYERON LAS TEMATICAS DEL CURSO:"
 @48,100 SAY ptp09 PICT "99.99"
 @49,15 SAY "10.- SE DESARROLLO EL CURSO EN FORMA VARIADA Y DINAMICA:"
 @49,100 SAY ptp10 PICT "99.99"
 @50,15 SAY "11.- LAS ACTIVIDADES DESARROLLADAS PROPICIAN LA
 COMUNICACION GRUPAL:"
 @50,100 SAY ptp11 PICT "99.99"
 @51,15 SAY "12.- LA RELACION CON LOS ALUMNOS (AS) SE ESTABLECE EN
 MARCOS ETICOS"
 @51,100 SAY ptp12 PICT "99.99"
 @52,15 SAY "13.- SE PROPICIA EN LOS ALUMNOS UNA ACTIVIDAD
 PARTICIPATIVA, ACTIVA Y REFLEXIVA:"
 @52,100 SAY ptp13 PICT "99.99"
 @53,15 SAY "14.- SE PROMUEVE LA PARTICIPACION EXTRACLASE:"
 @53,100 SAY ptp14 PICT "99.99"
 @54,15 SAY "15.- SE REALIZA UNA EVALUACION CONTINUA Y GRADUAL:"
 @54,100 SAY ptp15 PICT "99.99"
 @55,15 SAY "16.- LA EVALUACION PERMITE CONOCER CONTINUAMENTE TU
 AVANCE:"
 @55,100 SAY ptp16 PICT "99.99"
 @56,15 SAY "17.- EL CURSO CONTRIBUYE A TU FORMACION PROFESIONAL:"
 @56,100 SAY ptp17 PICT "99.99"
 @58,50 SAY "PROMEDIO EVALUACION DEL PROFESOR:"
 @58,100 SAY pep PICT "99.99"
 ETEC

ENDDO
RESTSCREEN(0,0,24,79,1cgpcoe)
RETURN nil

//EOF()

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGÓN

COORDINACIÓN DE INGENIERÍA EN COMPUTACIÓN

MANUAL DE USUARIO

SISTEMA DE CONTROL DE PROFESORES

VER. 1.0

MÉXICO, 1997

ÍNDICE

I.- INTRODUCCIÓN

II.- DESCRIPCIÓN DEL SISTEMA

- II.1.- Objetivo
- II.2.- Descripción de los Módulos del Sistema
- II.3.- Manejo de Pantallas del Sistema
- II.4.- Visualización General del Sistema

III.- MODULO DE CONSULTAS

- III.1.- Consulta de Datos Personales
- III.2.- Consulta de Datos Administrativos

IV.- MODULO DE MOVIMIENTOS

V.- MODULO DE PROPUESTAS

- V.1.- Generación de Propuestas
- V.2.- Revisión de Propuestas
 - V.2.1.- Profesores con Aumento de Horas
 - V.2.2.- Profesores sin Horario

VI.- MODULO DE EVALUACIONES

VII.- MODULO DE IMPRESIÓN

- VII.1.- Impresión de Evaluaciones
- VII.2.- Impresión de Propuestas
- VII.3.- Impresión de Tarjetas de Control de Asistencia

I. INTRODUCCIÓN

Este manual se realizó con el objeto de que el usuario del Sistema de Control de Profesores, en cada una de las Coordinaciones que integran el Plantel, tenga una herramienta de apoyo para la operación del mismo, ya que las dudas que tenga al momento de realizar alguna de las tareas, las podrá aclarar consultando el manual en el capítulo del módulo indicado.

El usuario deberá leer este manual con atención, para que en caso de requerir alguna aclaración sobre la forma de operar el Sistema, ubique inmediatamente el capítulo a consultar.

El Sistema de Control de Profesores está realizado en forma modular para que su operación resulte sumamente sencilla, y sin necesidad de tener conocimientos sobre informática ó de ser un técnico especializado pueda usted operarlo, mediante una capacitación muy sencilla y el apoyo del presente manual.

En el contenido de este manual encontrará muestras de Menús, Pantallas de Captura de Datos, Consulta de Información, generación de las impresiones de las Propuestas, Diagnostico de Evaluación y Tarjetas de Control de Asistencia. A lo largo del presente, se explica en forma detallada cada uno de los pasos que debe seguir el usuario para operar con éxito cada uno de los módulos que componen el Sistema de Control de Profesores.

II.- DESCRIPCIÓN DEL SISTEMA

II.1.- Objetivo

El Sistema de Control de Profesores tiene por objetivo emitir las Propuestas de Contratación del Personal Académico, Realizar el Diagnostico de la Evaluación del mismo e Imprimir las Tarjetas de Control de Asistencia de dicho personal.

II.2.- Descripción de los Módulos del Sistema

- Módulo de Consultas

Este modulo se encuentra dividido en dos partes en una de las cuales se podrán consultar los datos personales de cada profesor como son su domicilio, teléfono particular, si es funcionario, beneficiarios del seguro de vida, si esta titulado, etc. En la otra se puede consultar la información administrativa de cada profesor es decir, su categoría, si esta activo, antigüedad, horas autorizadas, horas asignadas tanto teóricas como practicas, así como si imparte clases en otra jefatura del plantel u otra dependencia de la Universidad.

-Módulo de Movimientos

Se diseño para efectuar básicamente el registro (captura de datos) de los profesores que conforman el personal académico del plantel, además ofrece la alternativa de modificar los datos en caso de errores de captura, o cambios de información.

-Módulo de Propuestas

Usted encontrará en éstas opciones, una serie de procesos de apoyo para la generación de las propuestas y la revisión de las mismas, también se puede identificar los profesores que no tienen horario asignado y aquellos cuyas horas asignadas son mayores a sus horas autorizadas.

-Módulo de Evaluación

En este módulo se puede capturar la información contenida en los cuestionarios de evaluación a profesores la cual será utilizada para realizar el diagnostico de la evaluación.

Módulo de Impresión

En esta opción el usuario podrá imprimir las propuestas de contratación de profesores, los reportes de la evaluación y las tarjetas de control de asistencia.

II.3.- Manejo de Pantallas del Sistema

Las pantallas del Sistema se componen por: Menús, Captura de Datos, Consulta de Información, Ejecución de procesos y parámetros, las cuales están desarrolladas de una manera sencilla y amigable, para que el usuario tenga confianza en los pasos que va efectuando en la operación del Módulo del Sistema que éste trabajando.

Los Menús del Sistema están formados por Títulos, opciones del módulo y Mensajes de Explicación de cada una de las Opciones que tiene el mismo.

- Los Títulos de los Menús del Sistema contienen la siguiente información:

- Fecha Actual (Ángulo superior izquierdo)
- Nombre de la Empresa (Centro Primer Renglón)
- Hora Actual (Ángulo superior derecho)
- Nombre del Sistema(Centro Segundo renglón)
- Coordinación que esta operando el Sistema(Tercer renglón Ángulo izquierdo)
- Periodo de trabajo(Tercer renglón Ángulo derecho)

Con lo cual usted está informado sobre la parte del sistema que este trabajando.

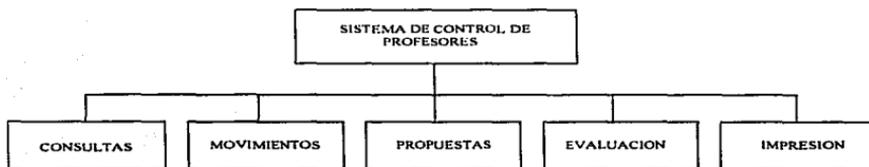
- Opciones del Menú.- En ésta parte de la pantalla, están indicadas todas las opciones que puede acceder y conforme se vaya posicionando en ellas con las teclas de movimiento, estas se irán iluminando, cuando oprimamos la tecla <Enter> la opción activada será la que en ése momento esté iluminada.

- Mensajes de Explicación de las Opciones.- En la parte inferior de la pantalla, irá apareciendo en una forma más descriptiva la opción en que estemos situados.

II.4.- Visualización General del Sistema

- El Sistema de Control de Profesores está conformado por cinco módulos principales, y nueve submódulos en total, sin tomar en cuenta los procesos de ayuda en cada Módulo.

En seguida se presenta el diagrama a bloques del Sistema con todos sus módulos explicados anteriormente:



En primera instancia describire las pantallas iniciales del Sistema ZEUS debido a que el Sistema que nos ocupa forma parte del **SISTEMA INTEGRAL DE ADMINISTRACIÓN ESCOLAR**.

Al principio del Sistema se presenta una pantalla inicial de clave de usuario y clave de acceso, las cuales fueron incluidas con el propósito de restringir su uso a personal no autorizado, teniendo que teclear las claves que permitan el acceso a dicho sistema.



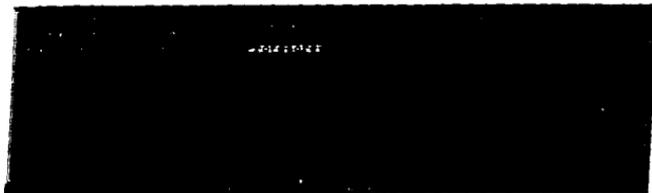
Este Sistema cuenta con un módulo de utileras en el cual se permite la creación o cambio de semestre, cuenta también con las opciones de reindexación y respaldo, así como la selección de tipo de impresora.



Cuando ha concluido su trabajo y desea salir del Sistema debe irse al módulo de salida y dar <Enter> en la opción Confirmar si en este momento decide que desea continuar en el Sistema seleccione la opción Cancelar.



Ahora bien para poder trabajar con el Sistema de Control de Profesores debe seleccionar la opción **Profesores** del menú principal del Sistema Administrativo ZEUS.



III.- MODULO DE CONSULTAS

Este modulo se encuentra dividido en dos partes en una de las cuales se podrán consultar los datos personales de cada profesor como son su domicilio, teléfono particular, si es funcionario, beneficiarios del seguro de vida, si esta titulado, etc. En la otra se puede consultar la información administrativa de cada profesor es decir, su categoría, si esta activo, antigüedad, horas autorizadas, horas asignadas tanto teóricas como practicas, así como si imparte clases en otra jefatura del plantel u otra dependencia de la Universidad.



III.1.- Consulta de Datos Personales

Una de las opciones de consulta es **GENÉRICO** y se refiere a los datos personales de cada profesor si este es el tipo de consulta que requiere efectuar debe seleccionar esta opción.



Para poder realizar la consulta debe indicar el RFC del profesor que desea consultar este puede proporcionarlo tecleándolo directamente en el recuadro o bien haciendo uso del catalogo de ayuda el cual se despliega oprmiendo la tecla de función <F2> o <Enter>.

III.2.- Consulta de Datos Administrativos

Si lo que desea es consultar la informacion administrativa, es decir, la categoria del profesor, cuantas horas tiene asignadas, su antiguedad, y si da clases en otra jefatura del plantel o alguna otra dependencia de la UNAM debera seleccionar la opcion COORDINACION.



Para poder realizar la consulta debe indicar el RFC del profesor que desea consultar este puede proporcionarlo tecleandolo directamente en el recuadro o bien haciendo uso del catalogo de ayuda el cual se despliega oprimiendo la tecla de funcion <F2> o <Enter>.



Una vez que es desplegado el catalogo de ayuda debe seleccionar mediante las teclas de movimiento al profesor que desea consultar, cuando dicho profesor esta seleccionado debe oprimir la tecla <Enter>

| ESC | NOV 83 |
|----------|-------------------------|
| 10000000 | ALFONSO HERRERA MORALES |
| 10000001 | ALFONSO HERRERA MORALES |
| 10000002 | ALFONSO HERRERA MORALES |
| 10000003 | ALFONSO HERRERA MORALES |
| 10000004 | ALFONSO HERRERA MORALES |
| 10000005 | ALFONSO HERRERA MORALES |
| 10000006 | ALFONSO HERRERA MORALES |
| 10000007 | ALFONSO HERRERA MORALES |
| 10000008 | ALFONSO HERRERA MORALES |
| 10000009 | ALFONSO HERRERA MORALES |
| 10000010 | ALFONSO HERRERA MORALES |
| 10000011 | ALFONSO HERRERA MORALES |
| 10000012 | ALFONSO HERRERA MORALES |
| 10000013 | ALFONSO HERRERA MORALES |
| 10000014 | ALFONSO HERRERA MORALES |
| 10000015 | ALFONSO HERRERA MORALES |
| 10000016 | ALFONSO HERRERA MORALES |
| 10000017 | ALFONSO HERRERA MORALES |
| 10000018 | ALFONSO HERRERA MORALES |
| 10000019 | ALFONSO HERRERA MORALES |
| 10000020 | ALFONSO HERRERA MORALES |
| 10000021 | ALFONSO HERRERA MORALES |
| 10000022 | ALFONSO HERRERA MORALES |
| 10000023 | ALFONSO HERRERA MORALES |
| 10000024 | ALFONSO HERRERA MORALES |
| 10000025 | ALFONSO HERRERA MORALES |
| 10000026 | ALFONSO HERRERA MORALES |
| 10000027 | ALFONSO HERRERA MORALES |
| 10000028 | ALFONSO HERRERA MORALES |
| 10000029 | ALFONSO HERRERA MORALES |
| 10000030 | ALFONSO HERRERA MORALES |
| 10000031 | ALFONSO HERRERA MORALES |
| 10000032 | ALFONSO HERRERA MORALES |
| 10000033 | ALFONSO HERRERA MORALES |
| 10000034 | ALFONSO HERRERA MORALES |
| 10000035 | ALFONSO HERRERA MORALES |
| 10000036 | ALFONSO HERRERA MORALES |
| 10000037 | ALFONSO HERRERA MORALES |
| 10000038 | ALFONSO HERRERA MORALES |
| 10000039 | ALFONSO HERRERA MORALES |
| 10000040 | ALFONSO HERRERA MORALES |
| 10000041 | ALFONSO HERRERA MORALES |
| 10000042 | ALFONSO HERRERA MORALES |
| 10000043 | ALFONSO HERRERA MORALES |
| 10000044 | ALFONSO HERRERA MORALES |
| 10000045 | ALFONSO HERRERA MORALES |
| 10000046 | ALFONSO HERRERA MORALES |
| 10000047 | ALFONSO HERRERA MORALES |
| 10000048 | ALFONSO HERRERA MORALES |
| 10000049 | ALFONSO HERRERA MORALES |
| 10000050 | ALFONSO HERRERA MORALES |
| 10000051 | ALFONSO HERRERA MORALES |
| 10000052 | ALFONSO HERRERA MORALES |
| 10000053 | ALFONSO HERRERA MORALES |
| 10000054 | ALFONSO HERRERA MORALES |
| 10000055 | ALFONSO HERRERA MORALES |
| 10000056 | ALFONSO HERRERA MORALES |
| 10000057 | ALFONSO HERRERA MORALES |
| 10000058 | ALFONSO HERRERA MORALES |
| 10000059 | ALFONSO HERRERA MORALES |
| 10000060 | ALFONSO HERRERA MORALES |
| 10000061 | ALFONSO HERRERA MORALES |
| 10000062 | ALFONSO HERRERA MORALES |
| 10000063 | ALFONSO HERRERA MORALES |
| 10000064 | ALFONSO HERRERA MORALES |
| 10000065 | ALFONSO HERRERA MORALES |
| 10000066 | ALFONSO HERRERA MORALES |
| 10000067 | ALFONSO HERRERA MORALES |
| 10000068 | ALFONSO HERRERA MORALES |
| 10000069 | ALFONSO HERRERA MORALES |
| 10000070 | ALFONSO HERRERA MORALES |
| 10000071 | ALFONSO HERRERA MORALES |
| 10000072 | ALFONSO HERRERA MORALES |
| 10000073 | ALFONSO HERRERA MORALES |
| 10000074 | ALFONSO HERRERA MORALES |
| 10000075 | ALFONSO HERRERA MORALES |
| 10000076 | ALFONSO HERRERA MORALES |
| 10000077 | ALFONSO HERRERA MORALES |
| 10000078 | ALFONSO HERRERA MORALES |
| 10000079 | ALFONSO HERRERA MORALES |
| 10000080 | ALFONSO HERRERA MORALES |
| 10000081 | ALFONSO HERRERA MORALES |
| 10000082 | ALFONSO HERRERA MORALES |
| 10000083 | ALFONSO HERRERA MORALES |
| 10000084 | ALFONSO HERRERA MORALES |
| 10000085 | ALFONSO HERRERA MORALES |
| 10000086 | ALFONSO HERRERA MORALES |
| 10000087 | ALFONSO HERRERA MORALES |
| 10000088 | ALFONSO HERRERA MORALES |
| 10000089 | ALFONSO HERRERA MORALES |
| 10000090 | ALFONSO HERRERA MORALES |
| 10000091 | ALFONSO HERRERA MORALES |
| 10000092 | ALFONSO HERRERA MORALES |
| 10000093 | ALFONSO HERRERA MORALES |
| 10000094 | ALFONSO HERRERA MORALES |
| 10000095 | ALFONSO HERRERA MORALES |
| 10000096 | ALFONSO HERRERA MORALES |
| 10000097 | ALFONSO HERRERA MORALES |
| 10000098 | ALFONSO HERRERA MORALES |
| 10000099 | ALFONSO HERRERA MORALES |
| 10000100 | ALFONSO HERRERA MORALES |

Ahora se presenta en la pantalla los datos administrativos del profesor, en el momento que haya concluido la consulta deberá oprimir la tecla <Esc> para regresar al menú.



Cuando haya realizado todas las consultas deseadas podrá salir de este módulo mediante la opción SALIR con la cual regresara al menú principal del Sistema de Control de Profesores.



IV.- MODULO DE MOVIMIENTOS

Se diseñó para efectuar básicamente el registro (captura de datos) de los profesores que conforman el personal académico del plantel, además ofrece la alternativa de modificar los datos en caso de errores de captura, o cambios de información

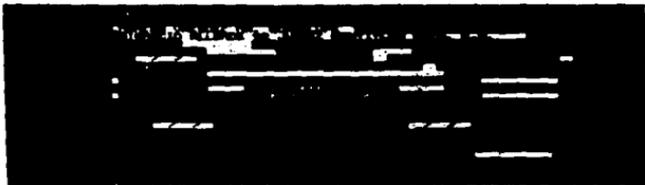


Para poder realizar algún movimiento debe indicar el RFC del profesor que desea modificar, este puede proporcionarlo tecleándolo directamente en el recuadro o bien haciendo uso del catalogo de ayuda el cual se despliega oprimiendo la tecla de función <F2> o <Enter>.

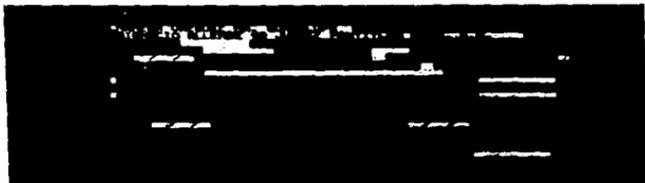


En caso de tratarse de un profesor que no se encuentra registrado en el catalogo se presentara la pantalla que permitirá la captura de la información para poder dar de alta al profesor.

En el momento que se ha terminado la actualización aparece un cuadro de confirmación en el cual si se indica que la información no es correcta regresa al modo de edición para corregir nuevamente la información si por el contrario le indica que la información es correcta graba los cambios en los archivos correspondientes y presenta nuevamente la pantalla para actualizar otro registro, en caso de no desear mas movimientos debe dar <Esc> para regresar al menú principal del Sistema de Control de Profesores.



Si por alguna razón desea cancelar los cambios hechos deberá contestar con la opción CANCELAR con lo cual se pierden los cambios hechos y presenta nuevamente la pantalla para actualizar otro registro, en caso de no desear mas movimientos debe dar <Esc> para regresar al menú principal del Sistema de Control de Profesores.



V.- MODULO DE PROPUESTAS

Usted encontrará en éstas opciones, una serie de procesos de apoyo para la generación de las propuestas y la revisión de las mismas, también se puede identificar los profesores que no tienen horario asignado y aquellos cuyas horas asignadas son mayores a sus horas autorizadas.



V.1.- Generación de Propuestas

Una vez que ha cerrado su Banco de Horas mediante el Modulo de Banco de Horas del Sistema Administrativo ZEUS, ha registrado los horarios correspondientes en el Modulo de Horarios del mismo Sistema y ha actualizado debidamente la información de los profesores en la opción MOVIMIENTOS del Modulo de profesores del mencionado Sistema, está en condiciones de generar la información para las Propuestas de los Profesores, esto lo logra ejecutando la opción GENERAR del módulo PROPUESTAS.



Al momento de dar esta opción despliega un mensaje en el que advierte que la información será modificada dando las alternativas de continuar o cancelar esto es debido a que al ejecutar esta opción los archivos de propuestas se reinician perdiendo la información anterior, es decir, que si usted ya genero sus propuestas debe estar seguro de querer volver a generarlas ya que perderá las modificaciones que les haya hecho.



La generación de propuestas puede hacerse de manera global lo cual se logra cuando seleccionamos la opción **TODOS** del menú **GENERAR**.



Si su selección fue **TODOS** el sistema envía el siguiente mensaje para indicar que la generación de información de propuestas de profesores está en proceso

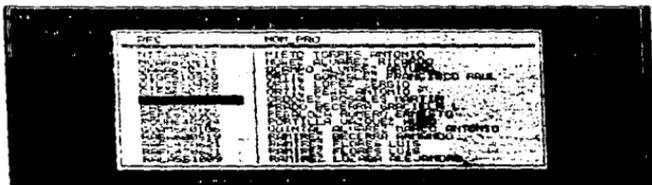


Si únicamente desea la generación de propuestas para un rango de profesores deberá seleccionar la opción **RANGO** del menú **GENERAR**.

Posteriormente el sistema le solicitara el RFC final del rango, es decir, hasta que profesor desea la generacion de informacion, este puede proporcionarlo tecleandolo directamente en el recuadro o bien haciendo uso del catalogo de ayuda el cual se despliega oprimiendo la tecla de funcion <F2> o <Enter>



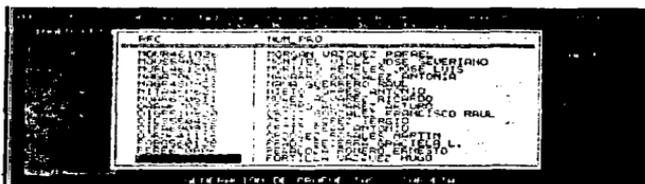
Estando desplegado el catalogo de ayuda debe seleccionar mediante las teclas de movimiento el profesor final del rango, cuando dicho profesor esta seleccionado debe oprimir la tecla <Enter>



Debe tener cuidado de seleccionar un profesor que se encuentre despues del profesor inicial o incluso puede tratarse del mismo profesor si solo deseamos la propuesta de un solo profesor de lo contrario aparecera el siguiente mensaje:



En caso de haber cometido el error de seleccionar un profesor menor al inicial deberá volver a desplegar el catalogo de ayuda y seleccionar mediante las teclas de movimiento el profesor final del rango, cuando dicho profesor esta seleccionado debe oprimir la tecla <Enter>



Si su selección fue RANGO el sistema envía el siguiente mensaje para indicar que la generación de información de propuestas de profesores de un rango determinado esta en proceso



En caso de haber concluido con la generación de propuestas ya sea global o de un rango, o si simplemente desea salir de este menu podra hacerlo mediante la opcion SALIR con la cual regresara al menu de PROPOSTAS.



V.2.- Revisión de Propuestas

Esta opción presenta cuatro alternativas las cuales son revisión de propuestas automáticas, revisión de propuestas manuales, aumento de horas y profesores sin horario las cuales detallaremos a continuación:



Ya que ha generado la información para las propuestas es necesario que está sea revisada, ya que solo se imprimirán las propuestas cuya información haya sido revisada, como los procesos de revisión de propuestas automáticas y manuales son idénticos solamente lo explicare una vez para evitar ser redundante.

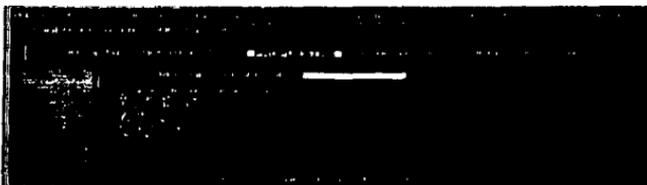
Propuestas Automáticas - Son las propuestas de profesores que tanto el semestre anterior como el actual impartieron el mismo número de horas, para revisarlas debe seleccionar la opción PROP AUTOMATICAS del menú REVISAR.



Propuestas Manuales - Son las propuestas de profesores que el semestre anterior impartieron diferente número de horas en comparación con el semestre actual, para revisarlas debe seleccionar la opción PROP MANUALES del menú REVISAR.



Como ya mencione el procedimiento para revisar las propuestas automáticas y las manuales es el mismo por tal motivo únicamente lo explicare una vez, al seleccionar cualquiera de las dos opciones aparecerá la siguiente pantalla en la cual se pide el RFC del Profesor el cual debemos seleccionar del catalogo de ayuda que se desplegara al oprimir la tecla de función F2 o <Enter>, es preferible que se seleccione al profesor del catalogo porque el RFC esta relacionado con un campo de control para identificar cada uno de los registros de un profesor.



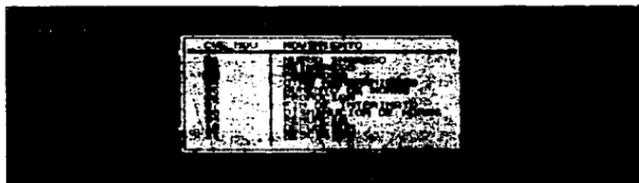
Estando desplegado el catalogo de ayuda debe seleccionar mediante las teclas de movimiento el registro que desea revisar oprimiendo la tecla <Enter>



Una vez que ha sido seleccionado el registro a revisar el sistema muestra una pantalla con la información que se reflejara en la propuesta permitiendo su edición.



También se cuenta con catalogo de ayuda de tipos de movimiento del cual debe seleccionar mediante las teclas de movimiento el tipo de movimiento valga la redundancia. Cuando dicho tipo esta seleccionado debe oprimir la tecla <Enter>



En el caso de requerir modificar las horas asignadas en algún registro el sistema contabiliza automáticamente las horas totales asignadas para ese profesor y lo compara con el total de horas autorizadas, enviando un mensaje de error cuando las horas asignadas excedan a las horas autorizadas, si esto sucede el sistema permite la modificación de las horas autorizadas cuando así corresponda.



En caso de haber modificado el horario se mostrara adicionalmente la información que se reflejara en las tarjetas de control de asistencia para que sea modificada en función del horario.



En el momento que se ha terminado la revisión aparece un cuadro de confirmación en el cual si le indica que la información no es correcta regresa al modo de edición para corregir nuevamente la información si por el contrario le indica que la información es correcta graba los cambios en los archivos correspondientes y presenta nuevamente la pantalla para revisar otra propuesta, en caso de no desear más revisiones debe dar <Esc> para regresar al menú de PROPUESTA.



Si en el último momento usted decide cancelar la información debe seleccionar la opción CANCELAR del cuadro de confirmación y quedaran cancelados todos los cambios.



Si es necesario insertar un nuevo registro en la propuesta puede hacerlo con la combinación de teclas CTRL-INS con lo cual nos aparecerá una pantalla de captura donde nos solicita desde el rfc del profesor, en este caso no se cuenta con catalogo de ayuda por tal motivo deberá teclearlo directamente en el recuadro.



Al momento de proporcionar el rfc el sistema verifica que este exista en el catalogo de profesores de no ser así envía el siguiente mensaje y nos devuelve la pantalla anterior para volver a intentarlo.



Si el rfc teclado existe en el catalogo de profesores nos permite dar de alta el registro solicitando la información necesaria para completar el registro. cuando dicha información ha sido capturada aparece el cuadro de confirmación descrito anteriormente por lo tanto no considero necesario volver a explicarlo..



Si lo que desea es dar de baja un registro puede lograrlo oprimiendo la combinación de teclas CTR-DEL o la tecla de función F3 al momento de ejecutar esta acción el sistema pide confirmación para poder eliminar el registro.



Si da <Enter> en la opción SI el registro se borra del archivo, ahora bien si su respuesta fue no nos muestra la siguiente pantalla donde indica que la operación fue cancelada.



V.2.1.- Profesores con Aumento de Horas

Al momento de asignar los horarios a cada profesor puede ocurrir que las horas asignadas sean mayores a las horas autorizadas, lo cual es una situación incorrecta, por lo tanto la finalidad de este proceso es identificar a dichos profesores para corregir esta irregularidad.



Hecha la debida revisión nos despliega una pantalla con los profesores que excedieron sus horas asignadas en relación con las horas autorizadas.



V.2.2.- Profesores sin Horario

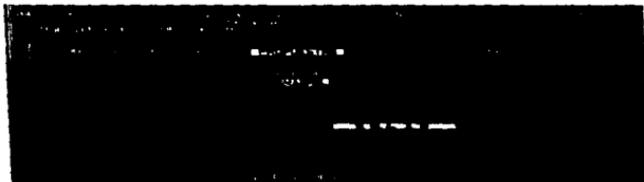
La opción PROF. SIN HORARIO hace una revisión de los profesores que no tienen horario asignado, esto le servirá al usuario para identificar a dichos profesores y así corregir esta irregularidad si así corresponde.



Hecha la debida revisión nos despliega una pantalla con los profesores que no tienen horario asignado.



En caso de haber concluido con la revisión de propuestas ya sea automáticas o manuales, y después de identificar los profesores con horas excedentes o sin horario, o simplemente si desear salir de este menú podrá hacerlo mediante la opción SALIR con la cual regresara al menú de PROPUESTAS.



Una vez que se han generado las propuestas y concluido la revisión de las mismas o en caso de desear de este menú podrá hacerlo mediante la opción SALIR con la cual regresara al menú de principal del Sistema de Control de Profesores.

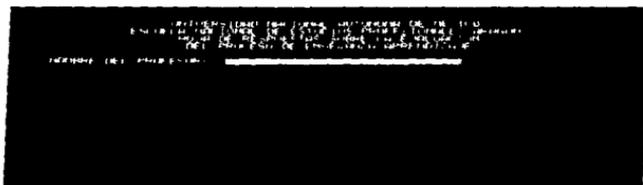


VI- MODULO DE EVALUACIONES

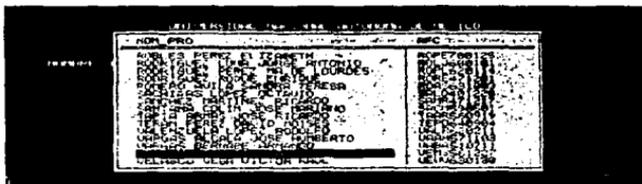
En este módulo se puede capturar la información contenida en los cuestionarios de evaluación a profesores la cual será utilizada para realizar el diagnóstico de la evaluación.



Al seleccionar el módulo EVALUACION del menú principal se muestra la siguiente pantalla en donde solicita el nombre del profesor que será evaluado, este dato lo puede teclear directamente en el recuadro iniciando con Apellido paterno, materno y nombre (s) o bien hacer uso del catalogo de ayuda mediante la tecla de función F2 o <Enter>.



Estando desplegado el catalogo de ayuda debe seleccionar mediante las teclas de movimiento el registro que desea evaluar oprimiendo la tecla <Enter>



Ahora se presenta la pantalla de captura del grupo, materia, clave de la misma, las calificaciones de cada una de las preguntas, tanto de autoevaluación del alumno como de la evaluación del alumno hacia el profesor y por ultimo los comentarios que pudiera haber hecho el alumno.



Al término de la captura el sistema pregunta si la información es correcta, de no ser así nos regresa a la edición de la información para corregir lo que no este correcto, en caso contrario se graba la información en el archivo correspondiente



VII.- MODULO DE IMPRESION

En esta opción el usuario podrá imprimir las propuestas de contratación de profesores, los reportes de la evaluación y las tarjetas de control de asistencia.



En primera instancia, cuando desea imprimir, debe especificar el tipo de impresora que utilizara para tal efecto, es decir, si trabajara con una impresora remota laser, remota matriz o una local laser o de matriz segun sea el caso



En este momento el sistema le indica que coloque papel tamaño carta y que encienda la impresora.



Posteriormente el sistema verifica que la impresora se encuentre lista para imprimir, de no ser así envía un mensaje de error indicando que la impresora no se encuentra preparada en el cual pregunta si desea reiniciar si le indica que no el sistema nos regresará al menú IMPRESION. Una vez que hemos verificado que la impresora este en línea le podemos indicar que si deseamos reiniciar para de esta manera poder continuar con la impresion.



VII.1.- Impresión de Evaluaciones

La primera opción del menú IMPRESION es EVALUACION con la cual se imprimen los diagnosticos de evaluacion



La impresión del diagnostico de evaluación puede hacerse de manera global lo cual se logra cuando seleccionamos la opción TODOS del submenú EVALUACION.



Si su selección fue **TODOS** el sistema envía el siguiente mensaje para indicar que la impresión del diagnóstico de evaluación de profesores está en proceso.



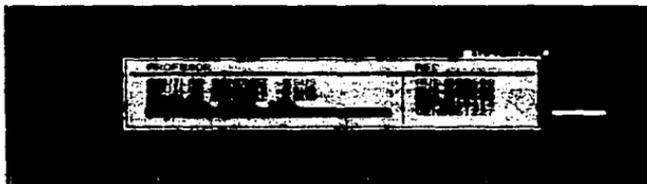
Si únicamente desea la impresión del diagnóstico de evaluación para un rango de profesores deberá seleccionar la opción **RANGO** del submenú **EVALUACION**.



Si va a imprimir el diagnóstico de evaluación solo para un rango de profesores el sistema le solicitará el RFC inicial del rango, es decir, a partir de que profesor desea la impresión de información, este puede proporcionarlo tecleándolo directamente en el recuadro o bien haciendo uso del catálogo de ayuda el cual se despliega oprimiendo la tecla de función <F2> o <Enter>.



Una vez que es desplegado el catalogo de ayuda debe seleccionar mediante las teclas de movimiento el profesor inicial del rango. cuando dicho profesor esta seleccionado oprima la tecla <Enter>



Posteriormente el sistema le solicitara el RFC final del rango. es decir. hasta que profesor desea la impresion de informacion. este puede proporcionarlo tecleandolo directamente en el recuadro o bien haciendo uso del catalogo de ayuda el cual se despliega oprimiendo la tecla de funcion <F2> o <Enter>.



Debe tener cuidado de seleccionar un profesor que se encuentre después del profesor inicial o incluso puede tratarse del mismo profesor si solo deseamos la evaluación de un solo profesor de lo contrario aparecerá el siguiente mensaje:



Si su selección fue RANGO el sistema envía el siguiente mensaje para indicar que la impresión de información de evaluación de profesores de un rango determinado esta en proceso:



En caso de haber concluido con la impresión del diagnostico de evaluación ya sea global o de un rango, o si simplemente desea salir de este menú podrá hacerlo mediante la opción SALIR con la cual regresara al submenú de EVALUACION.



VII.2.- Impresión de Propuestas

La siguiente opción del menú IMPRESION es PROPUESTAS con la cual se imprimen las propuestas para contratación del personal académico.



Al seleccionar esta opción nos aparece un pequeño recuadro en el que se solicita el número de copias que deseamos de cada propuesta y tanto el renglón como la columna inicial, estos últimos dos datos nos sirven para ajustar la impresión en cuanto a margen.



La impresión de las propuestas puede hacerse de manera global lo cual se logra cuando seleccionamos la opción TODOS del submenú PROPUESTAS.



Si su selección fue **TODOS** el sistema envía el siguiente mensaje para indicar que la impresión de propuestas de profesores esta en proceso.



Si únicamente desea la impresión de propuestas para un rango de profesores deberá seleccionar la opción **RANGO** del submenú **PROPUESTAS**.



Si va a imprimir las propuestas para un rango de profesores el sistema le solicitará el RFC inicial del rango, es decir, a partir de que profesor desea la impresión de información, este puede proporcionarlo tecleándolo directamente en el

Debe tener cuidado de seleccionar un profesor que se encuentre después del profesor inicial o incluso puede tratarse del mismo profesor si solo deseamos la propuesta de un solo profesor de lo contrario aparecerá el siguiente mensaje:



Si su selección fue RANGO el sistema envía el siguiente mensaje para indicar que la impresion de propuestas de profesores de un rango determinado esta en proceso:

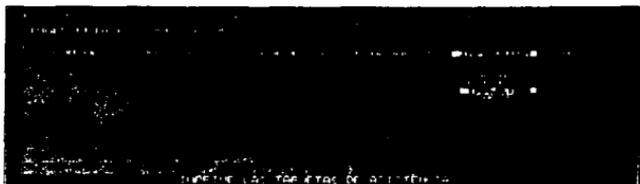


En caso de haber concluido con la impresion de propuestas ya sea global o de un rango, o si simplemente desea salir de este menu podrá hacerlo mediante la opcion SALIR con la cual regresara al submenu de PROPUESTAS.



VII.3.- Impresión de Tarjetas de Control de Asistencia

La última opción del menú IMPRESION ES TARJETAS con la cual se imprimen las tarjetas de control de asistencia del personal académico.



En el momento de seleccionar esta opción el sistema nos solicita a partir de que mes se iniciará la impresión de las tarjetas de control de asistencia, a partir del cual se toma un período de seis meses.



Se cuenta con una tabla auxiliar de la cual se puede seleccionar el mes de inicio, esta tabla se activa con la tecla <Enter>.



De igual manera que en la impresión de propuestas se solicita tanto el renglón como la columna inicial para efecto de ajuste de márgenes.



La impresión de las tarjetas de control de asistencia puede hacerse de manera global lo cual se logra cuando seleccionamos la opción TODAS del submenú TARJETAS



Se debe indicar al sistema que cara deseamos imprimir, es decir, la cara A, la cara B o Ambas.



Si su seleccion fue **TODOS** el sistema envia el siguiente mensaje para indicar que la impresion de tarjetas de control de asistencia esta en proceso



Si su seleccion fue **TODOS y ambas caras** el sistema envia el siguiente mensaje para indicar que la impresion de la cara A ha concluido y le pide que prepare la cara B de las tarjetas para continuar con el proceso, presentando la alternativas de **CONTINUAR** O **CANCELAR**.



Si únicamente desea la impresion de tarjetas de control de asistencia para un rango de profesores deberá seleccionar la opción **RANGO** del submenú **TARJETAS**.



Si va a imprimir las tarjetas de control de asistencia para un rango de profesores el sistema le solicitara el RFC inicial del rango, es decir, a partir de que profesor desea la impresion de informacion, este puede proporcionarlo tecleandolo directamente en el recuadro o bien haciendo uso del catalogo de ayuda el cual se despliega oprimiendo la tecla de funcion <F2> o <Enter>



Una vez que es desplegado el catalogo de ayuda debe seleccionar mediante las teclas de movimiento el profesor inicial del rango, cuando dicho profesor esta seleccionado oprima la tecla <Enter>

Posteriormente el sistema le solicitara el RFC final del rango, es decir, hasta que profesor desea la impresion de informacion, este puede proporcionarlo tecleandolo directamente en el recuadro o bien haciendo uso del catalogo de ayuda el cual se despliega oprimiendo la tecla de funcion <F2> o <Enter>



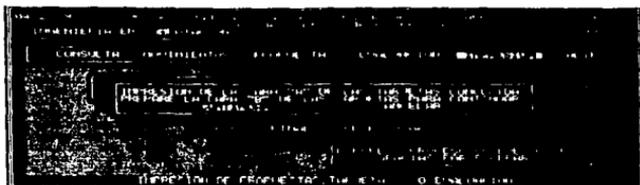
Debe tener cuidado de seleccionar un profesor que se encuentre después del profesor inicial o incluso puede tratarse del mismo profesor si solo descamos la tarjeta de control de asistencia de un solo profesor de lo contrario aparecerá el siguiente mensaje:



Si su selección fue RANGO el sistema envía el siguiente mensaje para indicar que la impresión de tarjetas de control de asistencia de profesores de un rango determinado esta en proceso:



Si su selección fue RANGO y ambas caras el sistema envía el siguiente mensaje para indicar que la impresión de la cara A ha concluido y le pide que prepare la cara B de las tarjetas para continuar con el proceso, presentando la alternativas de CONTINUAR Ó CANCELAR.



En caso de haber concluido con la impresión de tarjetas de control de asistencia ya sea global o de un rango, o si simplemente desea salir de este menú podrá hacerlo mediante la opción SALIR con la cual regresará al submenú de TARJETAS.



En caso de haber concluido con el proceso de impresión ya sea de evaluación, de propuestas o de tarjetas, o si simplemente desea salir de este menú podrá hacerlo mediante la opción SALIR con la cual regresará al menú principal del Sistema de Control de Profesores.



Por último si ya realizó el proceso correspondiente tanto de consulta como de movimientos, en su caso generación de propuestas, captura de evaluación a profesores o impresión, o si simplemente desea salir de este Sistema podrá hacerlo mediante la opción SALIR con la cual regresara al menú principal del Sistema Administrativo ZEUS.



MANUAL DE PROCEDIMIENTOS

Dentro de las funciones que desempeña el diseñador de sistemas se encuentra la de establecer los métodos y procedimientos bajo los cuales trabajará el sistema.

El delinear los procedimientos a seguir durante la vida del sistema es parte fundamental para su buen funcionamiento. Este apéndice está dedicado a indicar los pasos a seguir para actualizar la información con la que se trabajará en semestres posteriores.

-Los nombre de archivos de datos se forman de la siguiente manera:

Las dos primeras letras corresponden a las iniciales de la información que contienen. PC (Profesores de la Coordinación), PA (Propuestas Automáticas), PM(Propuestas Manuales), PR (Propuestas Revisadas), PE (Profesores Evaluación).

Las tres letras siguientes corresponden a las siglas de la coordinación. ICO (Ingeniería en Computación), IME (Ingeniería Mecánica Eléctrica), etc.

Los dos dígitos siguientes corresponden al año. 96 (1996), 97 (1997).

El último dígito corresponde al semestre del periodo 1 ó 2.

- Un profesor que es funcionario conserva su cargo en cualquier coordinación.
- Un profesor puede ser titular y suplente al mismo tiempo.
- También debe haber propuestas para profesores suplentes.
- Para saber que profesores no tienen horario debe hacerse en el archivo de la coordinación y chequear también suplentes.
- Se manejan subdirectorios.
- Variables de Sistema

```
carrera:=vm_dir_carrera
periodo:=alltrim(vm_anio_linea)
periodo:=substr(periodo)-1,2)
periodo:= periodo+alltrim(vm_periodo_linea)
```

- No pedir año y semestre, tomarlo de las variables de sistema al igual que la carrera.
- Manejar MEMVAR y todas las variables deben estar declaradas.
- Para generar información de la propuesta se debe checar con el total de horas si $hor_ant = hor_act$ se genera la propuesta como prorroga de forma automática, de lo contrario se genera la propuesta manualmente presentando el RFC con toda la información contenida en hor_act además de las horas autorizadas en banco de horas, indicando mediante un mensaje que la propuesta debe generarse en forma manual.
- Para revisar las propuestas al dar un RFC se presentara la información permitiendo su modificación, únicamente cuando la materia sea de descarga se permitirá modificar la clave de materia y el nombre de asignatura.
- Solo se imprimirán aquellas propuestas que hayan sido previamente revisadas.
- Para poder dar de baja un registro se debe consultar con horarios, banco de horas y extraordinarios para verificar que no tenga movimientos, solo en el caso de que movimientos sea nulo se permitirá borrar el registro.
- El dato de horas en otras coordinaciones lo proporciona banco de horas, únicamente se debe registrar el total de horas en otras jefaturas y se pondrá como base la coordinación en la que se tenga mas horas y un asterisco por cada una de las otras coordinaciones como parte del nombre de la jefatura.

BIBLIOGRAFÍA

- | | |
|---|--|
| Gorki Starlin C. Oliveira | Clipper 5.2 ED. Ventura, S.A. de C.V. |
| José A. Ramalho | Clipper 5.01 Avanzado ED. Mc Graw Hill |
| Enrique Pérez Villaseñor/ Pedro Pablo Torija Rojas | Manual de referencia rápida del Clipper 5.01 ED. Ventura, S.A. de C.V. |
| James Martin IBM Systems Research Institute | Organización de las Bases de Datos Editorial Prentice/Hall Internacional PHI |
| Luis Lizasoain | Bases de Datos en CD-ROM Editorial Paraninfo, S. A. |
| Henry C. Lucas, Jr | Sistemas de Información Análisis, Diseño y Puesta a Punto Editorial Paraninfo, S.A. |
| Peter Abrams Walter Corvin | Elementos de Procesos de Datos Editorial C.E.C.S.A. |
| Popkin Pike | Introducción al Proceso de Datos Editorial Paraninfo, S.A. |
| John G. Burch Jr Felix R. Strater Jr | Sistemas de información Teoría y Práctica Editorial Limusa |

TESIS CONSULTADAS

JUANA OLIVA BARRIOS ESTRADA, DESARROLLO DE UN SISTEMA EN PROCESO DISTRIBUIDO PARA PETRÓLEOS MEXICANOS SOPORTADO EN ARQUITECTURA CLIENTE/SERVIDOR, E.N.E.P. ARAGÓN, ASESOR ING JUAN GASTALDI PÉREZ, SAN JUAN DE ARAGÓN 1996.

GUSTAVO LIMON CALDERON, "SISTEMA INTEGRAL DE ADMINISTRACIÓN ESCOLAR MODULO BANCO DE HORAS", E.N.E.P. ARAGÓN, ASESOR ING. ERNESTO PEÑALOZA ROMERO, SAN JUAN DE ARAGÓN 1996.