

INSTITUTO TECNOLÓGICO SUPERIOR AERONÁUTICO

CARRERA DE TELEMÁTICA

**SISTEMA INFORMÁTICO PARA LA AUTOMATIZACIÓN DEL
INVENTARIO DE DOTACIÓN DE UNIFORMES, EQUIPOS DE LOS
CADETES DURANTE SU PERÍODO DE FORMACIÓN EN LA
ESCUELA SUPERIOR MILITAR DE AVIACIÓN
“COSME RENNELLA”**

POR:

SRTA. MARIANA ARACELI CORREA BÁEZ

**Proyecto de grado presentado como requisito parcial para la obtención del
título de:**

TECNÓLOGO EN TELEMÁTICA

2004

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por la Srta. Correa Báez Mariana Araceli como requerimiento parcial a la obtención del título de **TECNÓLOGO EN TELEMÁTICA.**

Ing. .Piedad Viera

DIRECTOR DEL PROYECTO

Latacunga 2004.

AGRADECIMIENTO

Mi sincero agradecimiento a Dios Todopoderoso creador y guía de nuestras vidas, si duda, esta tesis nunca hubiere podido ser concluida sin el apoyo, consejo y ánimo que he encontrado en muchas personas a lo largo de estos tres años. Quiero agradecer especialmente a mi directora Ing. Piedad Viera, por su ayuda, ideas y sugerencias, por no citar su amistad u ánimo en los momentos difíciles.

A la **FUERZA AEREA ECUATORIANA**, de manera especial al **INSTITUTO TECNOLÓGICO SUPERIOR AERONÁUTICO**, por haberme brindado la oportunidad de prepararme en sus aulas día a día, para de esta forma desarrollar y afianzar mis conocimientos.

Mi gratitud para los señores instructores académicos por sus enseñanzas y conocimientos impartidos durante el tiempo de permanencia en esta importante institución.

No me puedo olvidar de todos aquellos compañeros con quienes hemos intentado superarnos como profesionales y hacer que nuestras clases sean mejores, por el ánimo que me han dado y por su colaboración.

SRTA. MARIANA ARACELI CORREA BÁEZ

DEDICATORIA

La mejor dedicación de esfuerzo perseverancia y sacrificio es la recompensa que obtiene el ser humano al lograr el éxito y alcanzar la meta anhelada.

Este trabajo lo dedico a mis querida Madre, a mi hermana Edit quienes con su enorme amor, sacrificio y apoyo ayudaron a que se cumpla este sueño y también a Darwin Benavides y al todopoderoso a la Madre santísima que me han dado fuerza de voluntad y han iluminado mi vida.

A nuestros profesores que con su entrega total supieron transmitirnos todos sus conocimientos, para que nosotros seamos unas personas útiles a la Sociedad Ecuatoriana.

SRTA. MARIANA ARACELI CORREA BÁEZ

ÍNDICE GENERAL

Introducción.....	1
-------------------	---

EL PROBLEMA

Planteamiento del problema.....	3
Objetivos.....	4
○ Objetivo General.....	4
○ Objetivo Especifico.....	4
Justificación.....	4
Alcance.....	5

CAPÍTULO I

MARCO TEÓRICO

1.1.- Inventario de la Sección de Abastecimientos de la ESMA.....	6
1.1.1.-Definición de Inventario.....	6
1.1.2.-Importancia del Inventario.....	6
1.1.3.- Función de Inventarios.....	7
1.1.4.- Toma de Inventarios.....	7

1.1.5.- Relación entre la situación actual del inventario en la Sección de Abastecimientos del inventario a implementarse.....	9
1.2.- Conceptos básicos del lenguaje de programación de Microsoft Visual FoxPro 6.0.....	10
1.2.1.- La mecánica de programación de Visual fox Pro.....	10
1.2.2.- El proceso de la programación.....	14
1.2.3 Programación Orientada a objetos.....	15
1.2.4.- Crear una Base de Datos.....	15
1.2.5.- La interfaz de Visual fox Pro 6.0.....	28
1.3.- Comandos Fundamentales.....	29

CAPÍTULO II

DISEÑO DEL SISTEMA

2.-1 Estudio y evaluación de alternativas.....	36
2.2.- Requerimientos técnicos.....	38
2.3.- Elaboración de diagramas de flujo de datos	39
2.4.- Diseño de Pantallas.....	44
2.5.- Diseño de Menús.....	51

CAPÍTULO III

DESARROLLO DEL SISTEMA

3.1.- Almacenamiento de Datos.....	54
3.2.- Creación de Pantallas.....	56
3.3.- Programación.....	60
3.4.- Presentación e interacción de Pantallas.....	66

CAPÍTULO IV

PRUEBA Y ANÁLISIS DE RESULTADOS

4.1.- Implantación de sistema.....	67
4.2.- Pruebas del Sistema.....	69
4.3.- Evaluación.....	71

CAPÍTULO V

ESTUDIO ECONÓMICO

5.1.- Presupuesto.....	75
5.2.- Análisis Económico Financiero.....	75

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1.- Conclusiones.....	78
6.2.- Recomendaciones.....	79

Bibliografía

Anexos

Manual del Usuario

INDICE DE TABLAS

Tabla 1.1. Nombres cortos de Campo

Tabla 2.1. Visual FoxPro y otros lenguajes de Programación

Tabla 2.2. Símbolos utilizados para los diagramas de flujo

Tabla.5.1. Lista de costos y gastos de material de oficina

Tabla.5.2. Lista de costos y gastos de material de apoyo

Tabla.5.3. Lista de costos y gastos de varios

Tabla.5.4. Lista de costos total de la elaboración del sistema

INDICE DE FIGURAS

Figura.1.1. Ventana de Comandos

Figura.1.2. Ventana de Programación

Figura.1.3. Comandos de Ejecución

Figura.1.4. Creación de una nueva base de Datos

Figura.1.5. Utilización del Comando DataBase

Figura.1.6. Tablas en una Base de Datos

Figura.1.7. Tabla Libre

Figura.1.8. Agregar un Índice

Figura.1.9. Índice principal y normal

Figura.1.10. Administrar de Proyectos

Figura.1.11. Diseñador de formularios

Figura.2.1. Pantalla de clave de Ingreso

Figura.2.2. Pantalla de presentación

Figura.2.3. Pantalla de tarjetas individuales de vestuario

Figura.2.4. Pantalla de inventario

Figura.2.5. Pantalla de rubros

Figura.2.6. Pantalla de proveedores

Figura.2.7. Pantalla de usuarios

Figura.2.8. Menú parámetro activado

Figura.2.9. Menú parámetro desactivado

Figura 2.10. Menú Grupo

Figura.2.11. Menú inventario

Figura.2.12. Menú reportes

Figura.2.13. Menú Ayuda

Figura.3.1. Almacenamiento de datos en tiempo de diseño

Figura.3.2. Almacenamiento de datos en tiempo de ejecución

Figura.3.3. Base de datos del sistema

Figura.3.4. Pantalla de Presentación

Figura.3.5. Presentación para Formularios

Figura.3.6. Control de Formularios

Figura.3.7. Forma visual de distintos controles

Figura.3.8. Programación de la Pantalla de clave

Figura.3.9. Programación de la Pantalla de Usuarios

Figura.3.10. Programación para encriptar la clave

Figura.3.11. Presentación de reporte de nómina de alumnos.

Figura.3.12. Programación de la pantalla de inventario.

Figura.3.13. Presentación de Pantallas.

Figura.4.1. Funcionamiento del sistema automatizado

Figura.4.2. Pantalla de mantenimiento del sistema

Figura.4.3. Reporte desplegado

Figura.4.4. Aplicación en Microsoft Excel

INTRODUCCIÓN

En este trabajo se presenta, el desarrollado un proceso que tiene como eje la reflexión sobre la orientación que debe darse a la tecnología informática, en uno de sus paradigmas más actuales como son los sistemas informáticos de automatización.

Estos sistemas proporcionan a nuestros usuarios y a nosotros mismos la posibilidad de desarrollar nuevas experiencias, que pueden suponer una mejora de la práctica (proceso de enseñanza-aprendizaje).

Durante estos años, más que la persecución de un objetivo claro hemos experimentado una serie de situaciones entre aquello que creíamos que debía ser nuestra práctica educativa (y que ha variado y sigue variando con el tiempo) y la realidad que encontrábamos en el aula.

En esta situación se escogió el lenguaje de programación que se conoce como Visual FoxPro, que es un nuevo y potente lenguaje programación orientado a objetos, para la creación de bases de datos y la programación de aplicaciones. Este lenguaje proporciona todas las herramientas necesarias para administrar datos, tanto si va a organizar tablas de información y ejecutar consultas, como si va a crear un sistema de base de datos relacional integrado o programar una aplicación para la administración de datos de usuarios.

Por esta razón es que Microsoft Visual FoxPro, nos ofrece posibilidades expandidas, en cada área de programación de aplicaciones y bases de datos. Puede ver mejoras en el rendimiento, el uso de recursos del sistema y el diseño de entorno.

En la investigación se ha encontrado algo más que un método para desarrollar nuestra investigación, hemos visto una forma ideal de unir la teoría con la práctica y los diferentes aspectos que ocurren en nuestra práctica diaria, de mejorar o al menos justificar esa práctica y de analizar desde una nueva perspectiva nuestros problemas de mejora profesional.

CAPÍTULO I

MARCO TEÓRICO

1.1.- INVENTARIO EN LA SECCIÓN DE ABASTECIMIENTOS

1.1.1.- DEFINICIÓN DE INVENTARIO

Un sistema de inventarios es el set de políticas y procedimientos de control que monitorea los niveles de inventario y determina qué niveles deben mantenerse, cuándo debe reponerse, y de qué tamaño deben ser las órdenes.

El propósito del análisis de inventarios, es el de determinar cuándo ordenar y de qué tamaño debe ser.

1.1.2.- IMPORTANCIA DEL INVENTARIO

Proporcionan datos sobre los requerimientos del producto, que más tarde se utilizarán para desarrollar las descripciones de los productos y las especificaciones del mismo.

Los propósitos de mantener inventario son:

- Independizar o desacoplar operaciones.
- Proporcionar flexibilidad en la programación de la producción.
- Proporcionar seguridad ante la variación del tiempo de entrega de materiales.
- Llevar un plan de las futuras adquisiciones.

1.1.3.- FUNCIÓN DEL INVENTARIO

Se consideran dentro un inventario tres funciones importantes como son:

1. Compras.
2. Control de inventarios.
3. Manipulación de productos.

Esta forma coordinada permite efectivizar la adquisición de los productos, para luego seguir un largo proceso a partir de la planificación del servicio de entrega.

1.1.4.- TOMA DE INVENTARIOS

Se lo conoce como **Control de Inventarios**, este constata la existencia de un producto mediante el recuento de los artículos almacenados, establece una lista de todos los artículos existentes.

El proceso de toma de inventarios hace que todo el procedimiento sea más rápido y seguro pues registra, almacena y ordena la información en el punto mismo de inventariar con total movilidad, disminuyendo errores, eliminando el papeleo y entregando información oportuna y precisa para la administración.

Estas soluciones Permiten:

- Optimizar y flexibilizar el uso de los espacios de bodega.
- Incrementar la velocidad de rotación del inventario
- Obtener información estratégica para la gestión.
- Integrar el proceso en línea con los sistemas de gestión de bodega.

Formas de Controlar la Toma de Inventarios

Sistema de inventario periódico.- revisan los niveles de existencia a intervalos de tiempo predeterminados y los pedidos se basan en niveles actuales de existencia, el nivel de reserva de emergencia y un máximo establecido.

Mediante este método el reabastecimiento se hace sobre una base preestablecida, es decir, en una fecha fija. Sin embargo, la cantidad del pedido puede variar en cada ocasión.

Sistema de inventario perpetuo.- los niveles de existencia son revisados constantemente y los pedidos se hacen cuando la existencia baja o supera un nivel predeterminado.

En este sistema los reabastecimientos se hacen generalmente por cantidades estándares pero no se llevan a cabo en fechas establecidas de antemano.

Existe métodos para realizar inventarios de acuerdo al tiempo y a la forma, a continuación veremos algunos de ellos:

DE ACUERDO AL TIEMPO

Inventario continuo.- Se realiza poco a poco durante todo el año, cada artículo debe ser revisado por lo menos una vez al año.

Inventario Único.- Se paraliza todas las labores por dos o tres días, al final del año, es a corto plazo y su ventaja es que al suspender las operaciones se realiza inventario exacto y con pocos errores.

DE ACUERDO A LA FORMA

Último en entrar primero en Salir (UEPS).- Debe darle salida primero a la mercadería más reciente, por lo tanto el inventario queda valorado al precio más antiguo.

Primero en entrar primero en salir (PEPS).- Se valora los inventarios a los precios más recientes. Para los métodos de valoración del inventario, hay que tener presente que este método es para registrar costos y no para la salida del producto.

1.1.5.- RELACIÓN ENTRE LA SITUACIÓN ACTUAL DEL INVENTARIO EN LA SECCIÓN DE ABASTECIMIENTOS Y EL INVENTARIO A IMPLANTARSE

La Sección de Abastecimientos posee un sistema de inventario manual de tarjetas individuales de vestuario y de equipos, que se les entrega como dotación, durante sus cuatros años de formación en la Escuela Superior Militar “Cosme Rennella”.

Debido a esta particularidad es importante que en esta Sección, se lleve un control adecuado, pues se necesitará llevar un registro de las prendas entregadas, así como de los equipos.

De esta manera los productos que se encuentran en bodega irán disminuyendo y se necesitará comprar nuevos productos, de tal forma que no quede desabastecida la bodega.

Las personas encargadas, llevan un control de la mercadería comprada y entregada (*fecha, cantidad, valor y precio*), para mantener actualizado el inventario.

El sistema informático permitirá llevar estos registros de manera automática, prescindiendo así, del tedioso trabajo de manipular la información manualmente.



La tarea para llevar un adecuado control de inventario no termina en el registro como en tarjetas individuales, por ello es importante realizar un inventario físico de la dotación de uniformes y equipos, con el fin de determinar posibles errores en los registros, por ello el sistema facilitará la obtención de la información de manera rápida y segura.

Para asegurar el manejo del inventario, es importante dividir entre situaciones diferentes como productos, contratos y usuarios, etc. Entonces al implementar este sistema en la Sección de Abastecimientos, se tendrá la información física al momento, es decir, la cantidad en existencia, el valor, el precio, los pedidos y el presupuesto existente etc.

1.2.- CONCEPTOS BÁSICOS DEL LENGUAJE DE PROGRAMACIÓN DE MICROSOFT VISUAL FOXPRO 6.0

1.2.1.- MECÁNICA DE LA PROGRAMACIÓN EN VISUAL FOXPRO

Puede programar en Visual FoxPro escribiendo el código:

-  Instrucciones en forma de comandos
-  Funciones u operaciones que Visual FoxPro puede entender.

Usar la ventana comandos

Puede ejecutar un comando de Visual FoxPro si lo escribe en la ventana Comandos y presionar **ENTRAR**. Para volver a ejecutar el comando, lleve el cursor a la línea que contiene el comando y presione nuevamente **ENTRAR**.

Puede ejecutar varias líneas de código en la ventana de Comandos como si constituyeran un programa. La ventana de Comandos se puede modificar comandos, insertar, eliminar, cortar, etc.

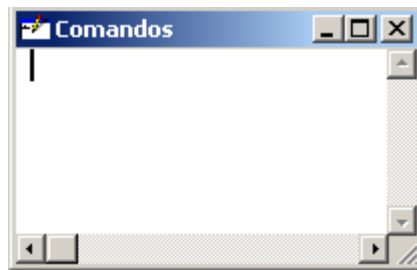


Figura 1. 1. Ventana de Comandos

La ventaja de escribir el código en la ventana Comandos radica en que las instrucciones se ejecutan de inmediatos.

Crear programas

Un programa de Visual FoxPro es un archivo de texto que contiene una serie de comandos. Puede crear un programa de la siguiente manera:

1. En el Administrador de proyectos, seleccione **Programas**.
2. Elija **Nuevo**.
 - O en el menú **Archivo**, elija **Nuevo**.
3. en el cuadro de diálogo **Nuevo**, seleccione **Programa**.
4. Elija **Nuevo archivo**.
 - O en la ventana de Comandos, escriba: ***“MODIFY COMMAND”***

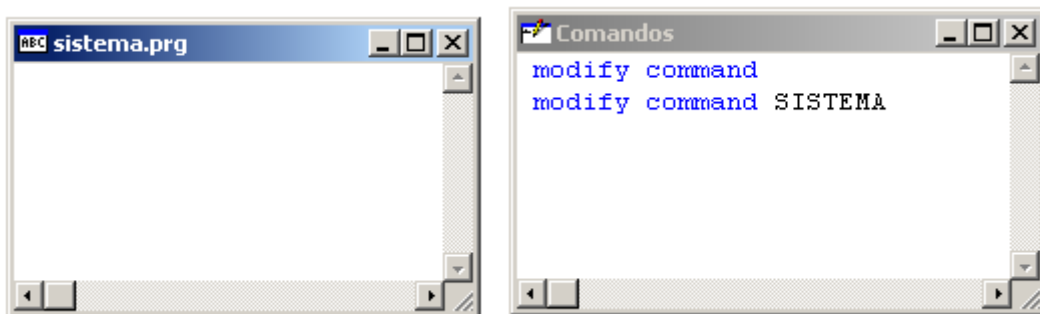


Figura 1. 2 Ventana de Programación

Guardar programas

Si intenta cerrar un programa sin antes guardarlo, aparecerá un cuadro de diálogo en el que se le preguntará si desea guardarlo o descartar los cambios realizados en el mismo.

Si guarda un programa al que todavía no ha asignado un nombre, se abrirá el cuadro de diálogo *Guardar como*, en el que podrá especificar el nombre del Programa.

Modificar programas

Si el programa forma parte de un proyecto, selecciónelo en el Administrador de proyectos y elija **Modificar**.

En la ventana **Comandos**, escriba “*el nombre del programa*” que desea modificar:

MODIFY COMMAND sistema.

Ejecutar programas

Si el programa forma parte de un proyecto, selecciónelo en el Administrador de proyectos y elija **Ejecutar**.

En la ventana de **Comandos**, escriba **DO** y el nombre del programa que se desea ejecutar:

DO sistema.

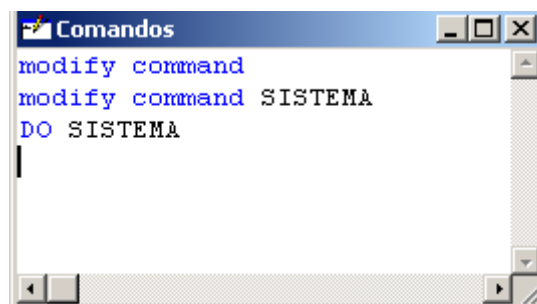


Figura 1. 3 Comando de Ejecución

1.2.2.- EL PROCESO DE LA PROGRAMACIÓN

La programación es un proceso repetitivo, los pasos se repiten para ir perfeccionando el código. Para empezar a programar se debe definir el problema. De no hacerlo puede terminar con un resultado que no esperaba.

También se debe desglosar el problema en pasos, en lugar de intentar resolver el problema. Se debe probar y depurar secciones del código a medida que avanza el programa. Con esto se encuentra y se soluciona problemas, que impiden que el código se ejecute correctamente.

Perfeccione los datos y el almacenamiento de datos para facilitar la manipulación de estos datos a través del código del programa.

1.2.3.- PROGRAMACIÓN ORIENTADA A OBJETOS

El diseño orientado a objetos y la programación orientada a objetos representa un cambio de perspectiva con respecto a la programación por procedimientos.

En lugar de pensar en el flujo del programa desde la primera hasta la última línea de código, se debe pensar en la creación de objetos:

Componentes auto contenidos de una aplicación que tienen funcionalidad privada además de la funcionalidad que se puede exponer al usuario.

1.2.4.- CREAR UNA BASE DE DATOS

La creación de una base de datos implica la creación de tablas. Al diseñar la base de datos especificó los campos de la tabla y las relaciones necesarias para su aplicación.

Ahora, al crear las tablas, deberá indicar con más detalle los tipos de datos, los títulos y los posibles valores predeterminados de cada campo y los índices que se utilizarán para establecer relaciones entre las tablas.

Crear tablas

A medida que crea la tabla puede crear nombres largos de tabla y de campo, y aprovechar las posibilidades del diccionario de datos para tablas de base de datos, nombres largos de campo, valores de campo predeterminados, reglas a nivel de campo y a nivel de registro, y desencadenantes.

Nueva Base de Datos

En Visual FoxPro se utilizan bases de datos para organizar y relacionar tablas y vistas. La base de datos proporciona una arquitectura necesaria para almacenar los datos y cuentas.

Para crear una nueva Base de Datos debe:

- En el Administrador de Proyectos, seleccione la ficha de **Datos**, luego **Base de Datos** y por último **Nuevo**.

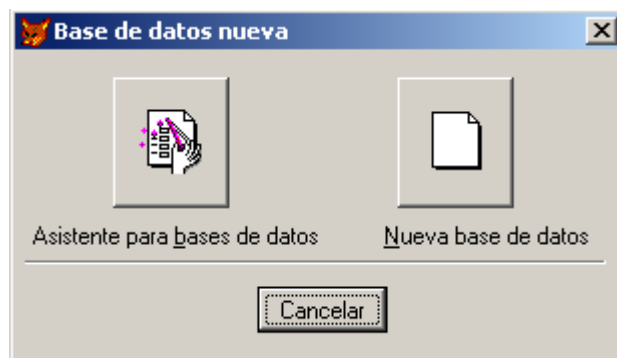


Figura 1. 4. Creación de nueva base de datos

- Otra alternativa es por medio de la ventana de comandos utilizando **CREATE DATABASE**.

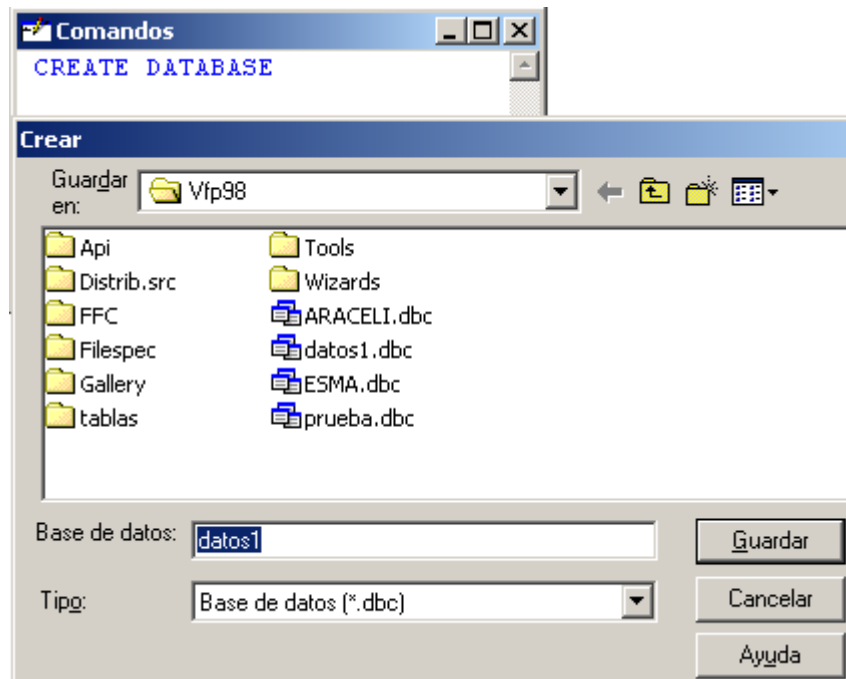


Figura 1. 5. Utilización de Comandos DATABASE

Proceso de diseño de base de datos

Si se usa un proceso de diseño de base de datos establecidos, se puede crear de una forma rápida y efectiva una base de datos bien diseñada que le proporciona acceso conveniente a la información que desea.

Visual FoxPro, almacena los datos, para luego ofrecer información de forma eficiente y precisa.

Diseño de una Base de Datos

Al diseñar una base de datos, en primer lugar se debe dividir la información que se desea almacenar como temas distintos y después indicar en Visual FoxPro cómo se relacionan estos temas.

Pasos que hay que seguir en el proceso de diseño de una base de datos:

1. **Determinar el propósito de la base de datos:** Este paso le ayudará a decidir los datos que desea que Visual FoxPro almacene.
2. **Determinar las tablas necesarias:** Cuando ya conoce claramente el propósito de la base de datos, puede dividir la información en temas distintos, como “uniformes” y “usuarios”. Cada tema será una tabla de la base de datos.
3. **Determinar los campos necesarios:** Tiene que decidir la información que desea incluir en cada tabla.
4. **Determinar las relaciones:** Observe cada tabla y decida cómo se relacionan sus datos con los de las tablas restantes.
5. **Perfeccionar el diseño:** Busque errores en el diseño.

Tablas en una Base de Datos

Para reunir tablas, se debe tener una base de datos para poder guardar todos los objetos (*vistas, conexiones y procedimientos almacenados*), asociados a las tablas que forman la base de datos.

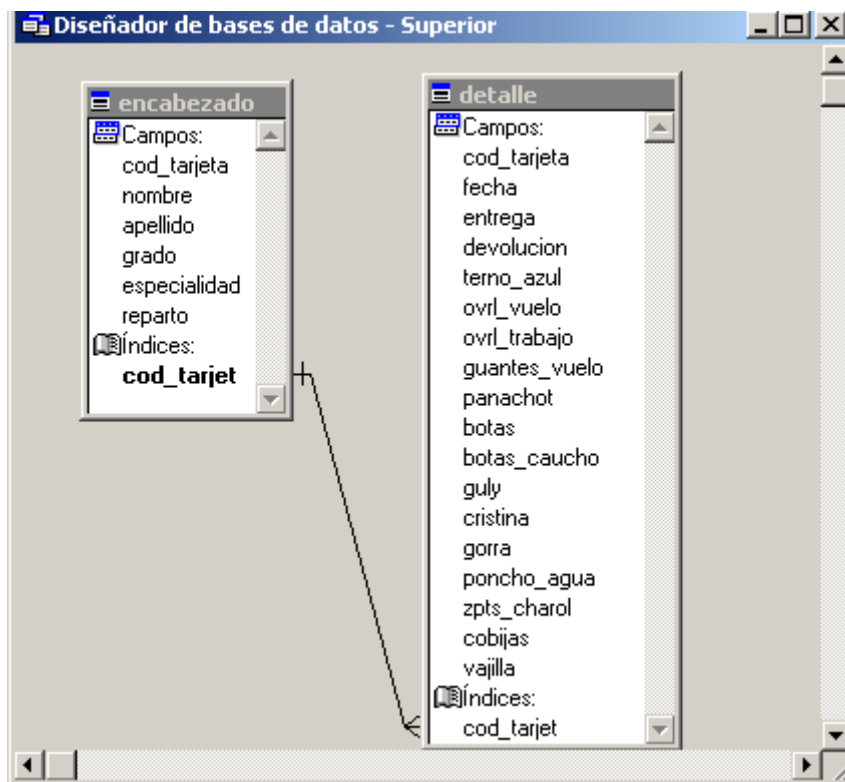


Figura 1. 6. Tablas en una base de datos

Agregar Tablas dentro de una Base de Datos

Las tablas, se las encuentra como:

1. **Tablas libres**
2. **Tablas de Base de Datos**

Las tablas se asocian a una base de datos, al crearlas desde dentro de una base de datos abierta o al agregar tablas existentes a una base de datos.

Para agregar una tabla libre a una base de datos, debe ir al Administrador de proyectos y seleccionar Tablas en la ficha Todos o en Datos, y agregue:

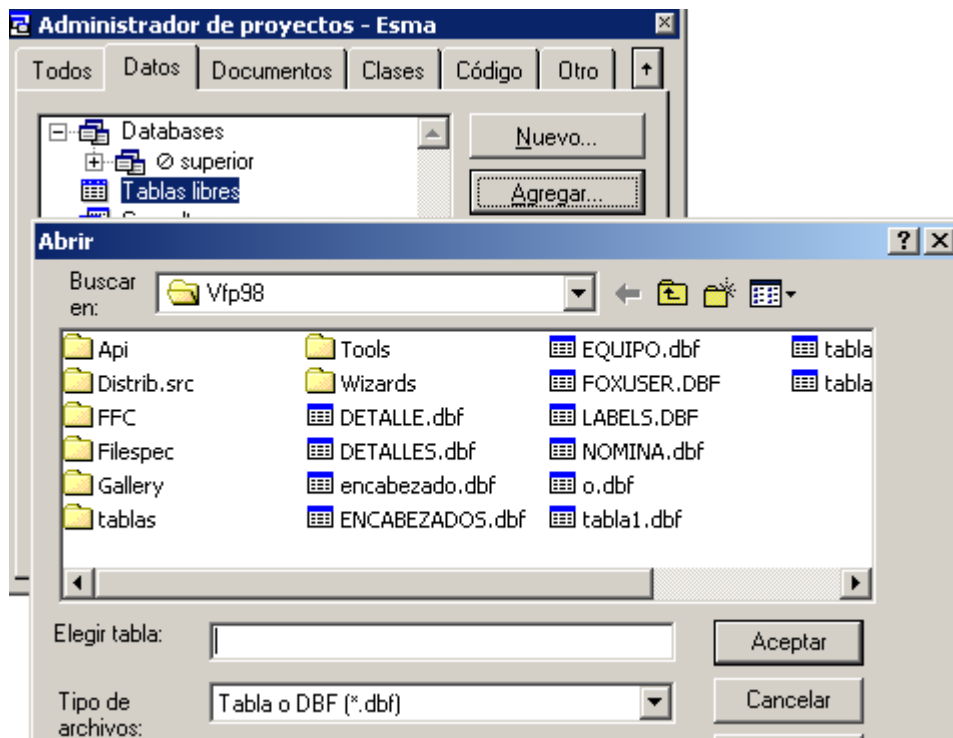


Figura 1. 7 Tabla Libre

Otra opción es con el Comando **ADD TABLE**.

Quitar una tabla de una Base de Datos

En el proceso de quitar una tabla de una Base de Datos, se suprime la tabla y la información del archivo de la base de datos, también actualiza la información del vínculo anterior para reflejar el nuevo estado de la tabla.

Para quitar una tabla de una Base de Datos, puede utilizar la interfaz, o el comando **REMOVE TABLE**. También dentro del Administrador de Proyectos, seleccione el nombre de la tabla y luego elija Quitar.

Al quitar una tabla de la Base de Datos, el archivo de la tabla no se elimina automáticamente.

Copiar y modificar la estructura de una tabla

Para modificar la estructura de una tabla puede abrir el Diseñador de tablas o bien puede realizar los cambios mediante programación con el comando ALTER TABLE

Crear campos

Al crear campos de tabla debe especificar un nombre de campo, un tipo de datos y un ancho de campo. También puede indicar si el campo debe permitir valores nulos y especificar el valor predeterminado del campo.

Para asignar nombre a un campo de la tabla

- En el Diseñador de tablas, escriba un nombre de campo en el cuadro de texto **Nombre**.

–O bien–
- Utilice el comando **CREATE TABLE** o el comando **ALTER TABLE**.

Usar nombres cortos de campo

Al crear una tabla en una base de datos, Visual FoxPro almacena el nombre largo de los campos de la tabla en un registro del archivo.

Si los 10 primeros caracteres del nombre largo de campo no son únicos en la tabla, Visual FoxPro generará un nombre formado por los n primeros caracteres del nombre largo y un número secuencial a continuación, de forma que el nombre del campo tenga 10 caracteres:

Tabla 1.1 Nombres cortos de Campos

Nombre largo	Nombre corto
cliente_contacto_nombre	Cliente_co
cliente_contacto_dirección	Cliente_c2
cliente_contacto_city	Cliente_c3

Para agregar un campo a una tabla

- En el Diseñador de tablas, elija **Insertar**.

–O bien–

- Utilice la cláusula ADD COLUMN del comando ALTER TABLE.

Eliminar campos

Puede eliminar un campo existente de una tabla mediante el Diseñador de tablas.

Para eliminar un campo de una tabla

- En el Diseñador de tablas, seleccione el campo y elija **Eliminar**.

Trabajar con registros

Una vez diseñada y creada la estructura de una tabla, puede almacenar datos en ella agregándole registros. Después podrá modificar y eliminar los registros existentes. Cada una de estas tareas puede realizarse a través de la interfaz o mediante comandos.

Agregar registros

Cuando cree una tabla de Visual FoxPro, la tabla estará abierta, pero vacía. Si intenta almacenar datos en una tabla sin crear primero un registro en ella, no sucederá nada. El primer paso para agregar registros a una tabla nueva es agregar filas en las que almacenar los nuevos datos.

En el ejemplo siguiente se utiliza el comando APPEND BLANK para añadir registros al final de la tabla.

Para mostrar los registros y modificarlos

EDIT.- permite modificar los datos dentro de la misma tabla.

- Utilice el comando EDIT.

Marcar registros para su eliminación

DELETE.- Para marcar registros de una base de datos para ser borrados posteriormente.

PACK.- Borra definitivamente los registros marcados con el comando DELETE.

GO 2

PACK

BROWSE

Indexar tablas

Para desplazarse por los registros de una tabla, verlos o manipularlos en un orden determinado, debe utilizar un índice. La flexibilidad supone crear y utilizar múltiples claves de índice distintas para la misma tabla, lo que permite trabajar con los registros ordenados de formas diferentes, según las necesidades de la aplicación.

Un índice de Visual FoxPro es un archivo que contiene punteros ordenados lógicamente según los valores de una clave de índice.

Movimiento de Punteros

GO.- Ubica el puntero del registro en el número de registro especificado.

GO TO.- Ubica el puntero en el primer registro de la base de datos.

GO BOTTON.- Ubica el puntero en el último registre de la base de datos.

INDEX.- Ordena los datos de la base de datos de acuerdo al campo especificado este alterna en forma ascendente.

CLOSE INDEX.- Cierra los índices que hayan sido abiertos y comandos de vista.

Crear un índice

- En el Diseñador de tablas, elija la ficha Índices e introduzca la información de una clave de índice. Elija **Normal** como tipo de índice.

–O bien–

- Utilice el comando INDEX.

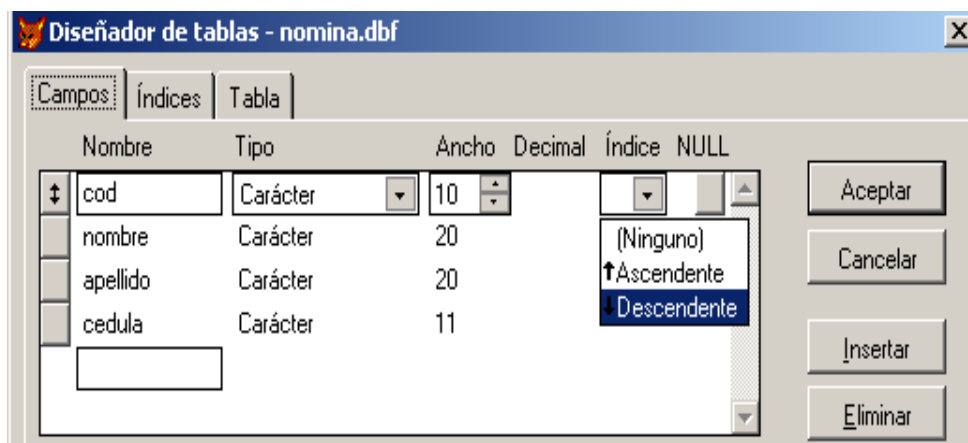


Figura 1.8. Agregar un Índice

Controlar los valores duplicados

Visual FoxPro cuenta con varios tipos de índices: principal, candidato y normal. Estos tipos determinan si se admiten o no valores duplicados en los campos y registros de la tabla.

Un **índice normal**.- se puede definir un índice normal, si especifica ascendente o descendente en la columna Índice del Diseñador de tablas.

Un **índice principal**.- Los índices principales se utilizan sobre todo en la tabla principal o “referenciada” para establecer la integridad referencial en una relación persistente.

Un **índice candidato** El nombre “candidato” hace referencia al estado del índice, se convierten en “candidatos” para ser elegidos como índice principal de la tabla. Puede crear múltiples índices candidatos para una tabla.

Para crear un índice principal o candidato

- En el Diseñador de tablas, elija la ficha **Índices** y cree un índice, seleccionando como tipo **Principal** o **Candidato**.

–O bien–

- Utilice el comando ALTER TABLE.

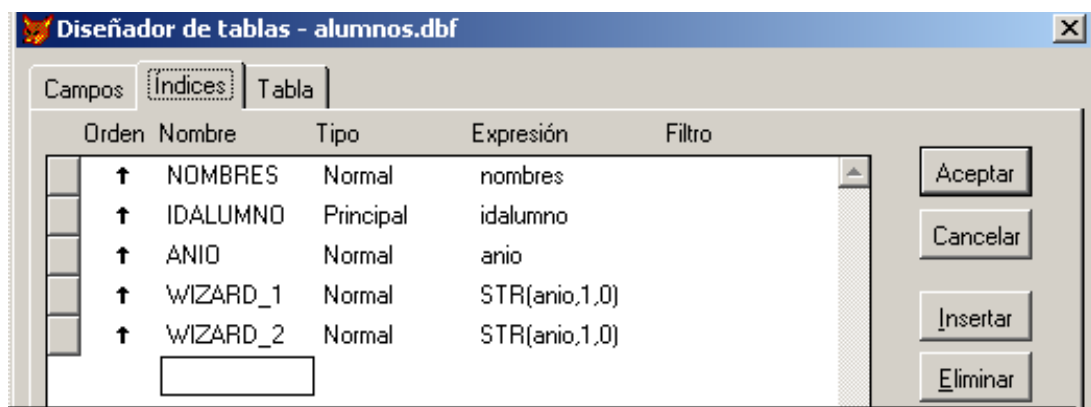


Figura 1.9. Índice Principal y Normal.

1.2.6.- INTERFAZ DE VISUAL FOXPRO

Administrador de Proyectos

Se puede agregar archivos gráficos a un proyecto, esto puede ser desde las fichas **Todos** y **Otros**, del Administrador de Proyectos.

Luego seleccione **Otros Archivos** y luego **Agregar**.

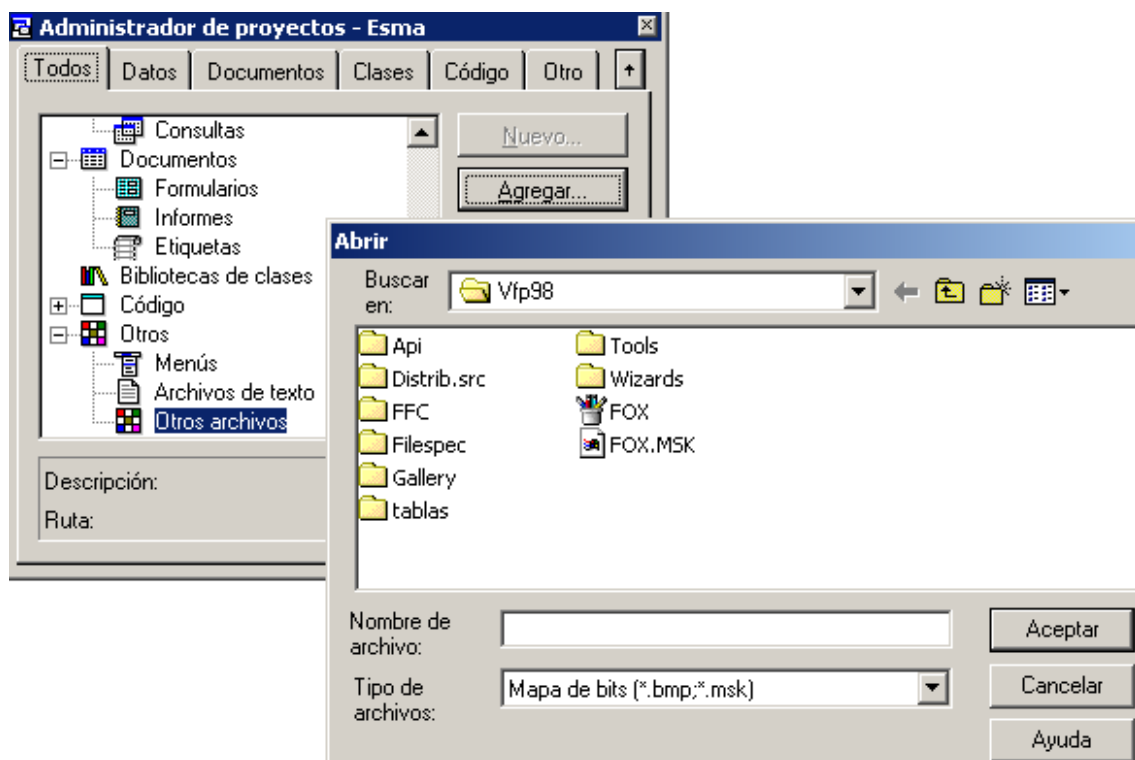


Figura 1. 10. Administrador de Proyectos

Diseñador de Formularios

En la ventana **Propiedades**, se da doble clic en la propiedad o se elige el botón del cuadro de diálogo de la propiedad, puede mostrar el cuadro de diálogo **Abrir** para las propiedades que admiten archivos gráficos.



Figura 1.11. Diseñador de Formularios

1.3.- COMANDOS FUNDAMENTALES DE VISUAL FOXPRO

Los Comandos y las Funciones de Visual FoxPro, sirven para realizar una acción o para devolver valores, que se utilizan en otros comandos y funciones. Un Comando, es una instrucción que cuando se la ejecuta el usuario, hace que realice una acción determinada. Un Comando se puede escribir desde el teclado, o se puede seleccionar en un menú, o incluir en código de programa.

Las Funciones terminan con dos paréntesis, que los distingue de los comandos. Las Funciones no se utilizan solas, siempre se combinan con un comando de Visual FoxPro. Las Funciones y Comandos se comportan de forma distintas como vemos:

1. Los **Comandos** realizan una acción.
2. Las **Funciones** devuelven un valor de un tipo de datos determinado.

ALLTRIM () (Función)

Elimina los espacios en blanco iniciales y finales de la expresión de caracteres especificada y devuelve la expresión recortada como una cadena de caracteres.

Sintaxis: *ALLTRIM (Expresión)*

Tipo devuelto: Character

Argumentos: Especifica la expresión de caracteres de la que se eliminan los espacios en blanco iniciales y finales.

Comentarios: ALLTRIM () se puede utilizar para asegurar que los espacios en blanco se eliminan de los datos escritos por un usuario.

ASC () (Función)

Devuelve el código ANSI del carácter situado más a la izquierda en una expresión de caracteres.

Sintaxis: *ASC (Expresión)*

Tipo devuelto: Numeric

Argumentos: Especifica la expresión de caracteres que contiene el carácter cuyo valor ANSI devuelve ASC (°) ASC () pasa por alto los caracteres que aparezcan después del primer carácter de *Expresión*.

Comentarios: ASC () devuelve la posición del carácter en la tabla de caracteres de la página de códigos actual. Cada carácter tiene un valor ANSI único entre 0 y 255.

EOF () (Función)

Determina si el puntero del registro está situado o no más allá del último registro de la tabla actual o especificada.

Sintaxis: EOF (nÁreaTrabajo cAliasTabla)

Tipo devuelto: Logical

Argumentos: *nÁreaTrabajo*; Especifica el número de área de trabajo de la tabla.

cAliasTabla; Especifica el alias de la tabla.

Comentarios:

EOF () devuelve falso (.F.) si no está abierta una tabla en un área de trabajo que especifique.

EOF () devuelve verdadero (.T.) sí el puntero de registro alcanza el final del archivo de tabla (EOF). El final de la tabla se alcanza cuando el puntero del registro transfiere el último registro de la tabla.

IF... ENDIF (Comando)

Ejecuta condicionalmente un conjunto de comandos dependiendo del resultado de una expresión lógica.

Sintaxis:

IF *expresión* **THEN**

Comandos

ELSE *Comandos* **ENDIF**

Argumentos: *Expresión*; Especifica la expresión lógica evaluada.

Si *Expresión* da como resultado verdadero (.T.), se ejecutarán cualesquiera instrucciones posteriores a IF y anteriores a ENDIF.

Si la *Expresión* es (.F.) y se incluye ELSE, se ejecutan los comandos situados después de ELSE y antes de ENDIF.

Si la *Expresión* es falsa (.F.) y no se incluye ELSE, se pasarán por alto todas las instrucciones entre IF y ENDIF.

Comentarios: Puede anidar **IF ... ENDIF** dentro de otro bloque IF ... ENDIF.

Los comentarios precedidos de && se pueden colocar en la misma línea después de **IF, ELSE y ENDIF**. Estos comentarios se pasan por alto durante la compilación y la ejecución del programa.

PACK (Comando)

Borra definitivamente todos los registros marcados para eliminar de la tabla actual.

Sintaxis: PACK [MEMO] [DBF]

Argumentos: *MEMO*; Elimina el espacio no utilizado del archivo memo, pero no elimina los registros marcados para eliminar de la tabla.

Comentarios: Cuando se ejecuta PACK, Microsoft Visual FoxPro copiará todos los registros que no están marcados para borrar a una tabla temporal. Cuando se termine PACK, Visual FoxPro eliminará la tabla original del disco y cambiará el nombre de la tabla temporal por el nombre de la tabla original.

PACK necesita utilizar de forma exclusiva la tabla. Si la tabla activa tiene uno o varios índices abiertos, PACK regenera los archivos de índice.

Precaución: Tenga cuidado de marcar solamente los registros que no va a necesitar más.

RECNO () (Función)

Devuelve el número del registro actual de la tabla actual o la especificada.

Sintaxis: RECNO (*nÁreaTrabajo* *cAliasTabla*)

Tipos devueltos: Numeric

Argumentos: *nAreaTrabajo*; Especifica el número del área de trabajo de una tabla abierta en otra área de trabajo..

nAliasTabla; Especifica el alias de una tabla abierta en otra área de trabajo.

Comentarios: El registro actual es el registro en el cual se encuentra el puntero de registro. RECNO () devuelve números negativos para registros anexados en un búfer de tablas.

Ejecutar: RECNO () sin los argumentos opcionales *nÁreaTrabajo* o *cAliasTabla* devuelve número de registro actual de la tabla situada en el área de trabajo seleccionada actualmente.

RELEASE (Método)

Libera de la memoria un formulario o un conjunto de formularios.

Sintaxis: *Objeto.Release*

Comentarios: El método Reléase es muy útil cuando se crea un formulario o un conjunto de formularios con el comando DO FORM y no haya variables que hagan referencia al formulario o al conjunto de formularios.

SET FILTER (Comando)

Especifica una condición que deben cumplir los registros de la tabla actual para estar accesibles.

Sintaxis: SET FILTER TO { *Expresión* }

Argumentos: *Expresión*; Especifica la condición que deben cumplir los registros.

Comentarios: Una vez ejecutado SET FILTER, solamente están disponibles en la tabla los registros que satisfacen la condición especificada con la expresión lógica *Expresión*.

SUBSTR () (Función)

Devuelve un número de caracteres específico de una expresión de caracteres o un campo memo.

Sintaxis: SUBSTR (cExpresión, nPosiciónInicial [, nCaracteresDevueltos])

Tipos devueltos: Character

Argumentos: *Expresión*; Especifica la expresión de caracteres o el campo memo desde el cual se devuelve la cadena de caracteres.

Posicióninicial; Especifica la posición en la expresión de caracteres o en la *Expresión* del campo memo desde la que se devuelve la cadena de caracteres. El primer carácter de *Expresión* es la posición 1.

Comentarios: SUBSTR () devuelve una cadena de caracteres de una expresión de tipo Character o de un campo memo, a partir de la posición especificada de la expresión de caracteres y hasta el número especificado de caracteres.

SET RELATION (Comando)

Establece una relación entre dos tablas abiertas.

Sintaxis: *Expresión1*

Tipo Devuelto: Numérico

Argumentos: *Expresión1* Especifica la expresión relacional que establece una relación entre las tablas primaria y secundaria.

Comentarios: Antes de que pueda establecer una relación, una tabla primaria debe estar abierta y la otra tabla secundaria debe estar abierta en otra área de trabajo.

CAPÍTULO II

DISEÑO DEL SISTEMA

2.1.- ESTUDIO Y EVALUACIÓN DE LAS ALTERNATIVAS.

Hoy en día tenemos una gran variedad de lenguajes de programación de alto y bajo nivel, entre los cuales se destacan Microsoft Visual Basic, Microsoft Visual C++, Lenguaje C de Programación y Microsoft Visual FoxPro.

Al realizar un estudio de las alternativas que ofrecen todos estos lenguajes de programación, nos hemos decidido por Microsoft Visual FoxPro 6.0, porque nos ofrece las siguientes ventajas:

- El Almacenamientos de datos.
- Este dispone de las características más avanzadas en el diseño de clases orientada a objetos.
- Puede suministrar la interfase de usuario a través del uso de formularios.
- Permite un suave aprendizaje a los programadores de base de datos.

La única desventaja que se puede observar en este lenguaje de programación es la siguiente:

- La necesidad de archivos adicionales.

Normalmente, cualquier función que pueda realizar con un programa podrá realizarla también a mano, si dispone de suficiente tiempo. Por ejemplo, si desea consultar información sobre un alumno en una tabla denominada nomina, como por ejemplo la de la Escuela Superior Militar de Aviación “Cosme Rennella”, podría hacerlo manualmente si sigue una secuencia concreta de instrucciones.

Visual FoxPro y otros lenguajes de programación

La mayoría de los lenguajes de programación tienen características básicas similares. Si está familiarizado con un lenguaje de programación, puede traducir fácilmente estos conocimientos a otro lenguaje de programación. .

Tabla 2.1 Visual FoxPro y otros lenguajes de Programación

Visual FoxPro	Basic
No distingue mayúsculas de minúsculas	<ul style="list-style-type: none"> • No distingue mayúsculas de minúsculas • Sencillo de programar. • Diseño con interfaz gráfica.

Pascal	C/C++
Distingue mayúsculas de minúsculas	<ul style="list-style-type: none"> • Distingue mayúsculas de minúsculas • Mas complicado de aprender. • No tiene un entorno visual de acuerdo al requerimiento del usuario.

Aunque Visual FoxPro admite la programación estándar por procedimientos, se ha ampliado la capacidad del lenguaje para proporcionar la potencia y la flexibilidad propias de la programación orientada a objetos.

El diseño orientado a objetos y la programación orientada a objetos representan un cambio de perspectiva con respecto a la programación estándar por procedimientos.

En lugar de pensar en el flujo del programa desde la primera hasta la última línea de código, se debe pensar en la creación de objetos: componentes auto contenidos de una aplicación que tienen funcionalidad privada además de la funcionalidad que se puede exponer al usuario.

2.2.- REQUERIMIENTOS TÉCNICOS.

Para la instalación de Microsoft Visual FoxPro se debe tener en cuenta lo siguiente:

Preparar la instalación

Antes de comenzar a instalar Visual FoxPro, compruebe lo siguiente:

- Requisitos de instalación de Visual FoxPro
- Requisitos para instalar el Asistente para upsizing

Requisitos para el uso de Visual FoxPro

Puede ejecutar Visual FoxPro con:

- Microsoft Windows 95
- Microsoft Windows 98
- Microsoft Windows NT 4.0 o posterior.
- Microsoft Windows 2000

Sistemas Operativos:

A continuación se exponen los requisitos mínimos para ejecutar Visual FoxPro con los sistemas operativos Windows:

- Un PC compatible con IBM con un procesador 486 a 66 MHz (o superior).
- Un *Mouse*.
- 16 MB de RAM.

Se recomienda un monitor VGA o de mayor resolución.

2.3.- ELABORACIÓN DE DIAGRAMAS DE FLUJO DE DATOS.


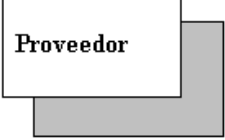

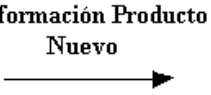



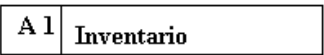
Un diagrama de Flujo se lo conoce también como organigrama de sistemas, son una descripción gráfica y marcan la secuencia de operaciones y/o programas para resolver un problema.

Los diagramas de flujos son muy utilizados en los niveles básicos de enseñanza y aprendizaje de algoritmos. Un organigrama es un diagrama de símbolos geométricos, la mayor parte de ellos cuadriláteros, conectados entre sí, por medio de flechas, el diagrama debe mostrar el camino que toman los datos, desde la entrada al programa hasta su salida.

Pero los diagramas de flujo tienen cuatro ventajas importantes:

1. Permiten describir cada componente usado en el diagrama.
2. Mayor comprensión de las interrelaciones de los sistemas y subsistemas.
3. Libertad conceptual que le brinda los cuatro símbolos básicos.
4. Herramienta usada para interactuar con los usuarios.

Tabla 2.2 Símbolos utilizados para los diagramas de flujo

SIMBOLO	SIGNIFICADO	EJEMPLO
	ENTIDAD	
	FLUJO DE DATOS	
	PROCESO	
	ALMACEN DE DATOS	

2.4.- DISEÑO DE PANTALLAS.

Pantalla de Clave de Ingreso

Con esta pantalla, tenemos la posibilidad de ingresar una clave determinada por cada usuario del sistema. Así podremos impedir el ingreso sin autorización, y proteger la información que se encuentra en el sistema.

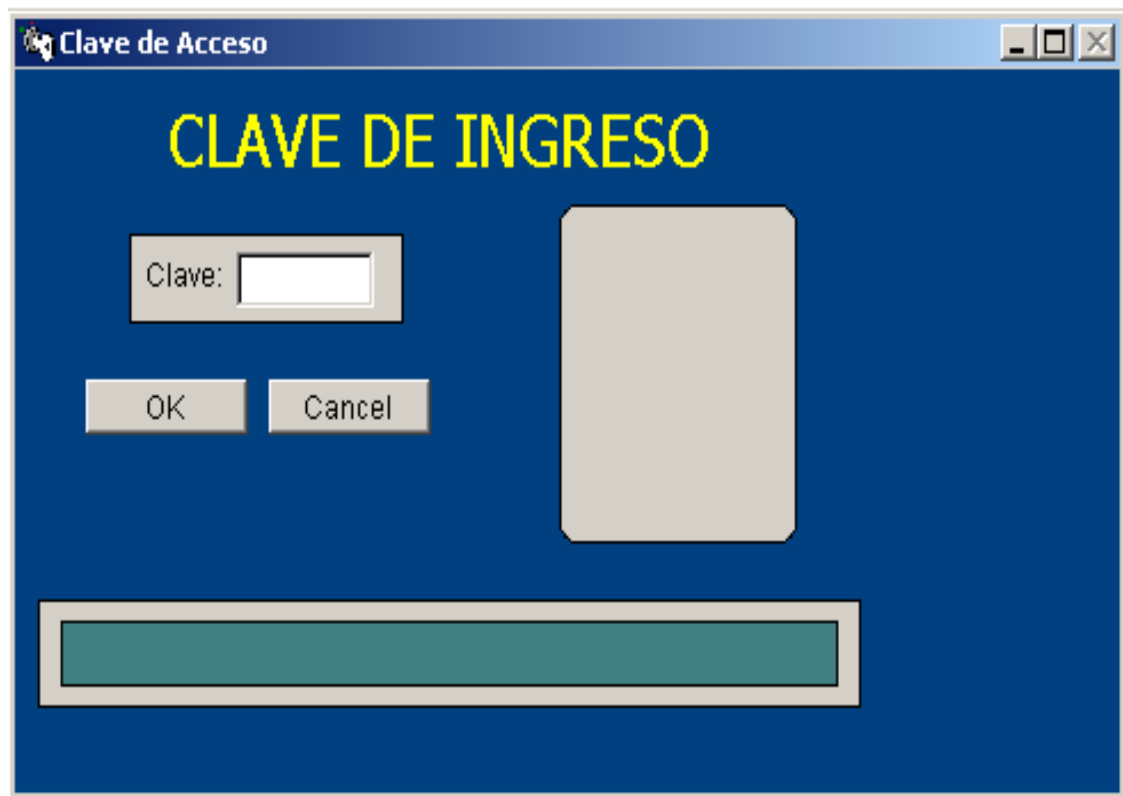


Figura 2.1. Pantalla de Clave de Ingreso

Pantalla de Presentación

Luego de haber ingresado la clave del usuario se despliega la pantalla de presentación, aquí hay que tomar en cuenta que se activarán las opciones del menú, dependiendo del nivel de acceso que tenga el usuario.

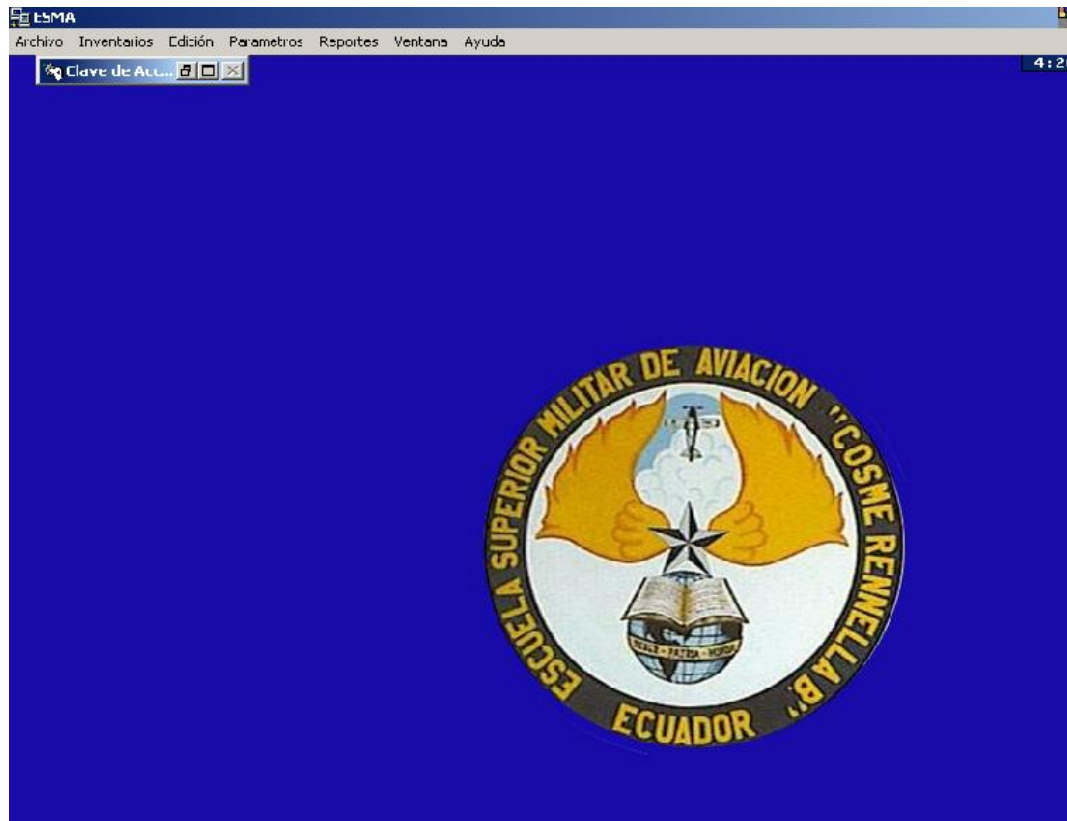


Figura 2.2. Pantalla de Presentación

Pantalla de Tarjetas Individuales de Vestuario

Esta pantalla nos permite elaborar las tarjetas de vestuario de los Cadetes, con esto se puede llevar un control de las prendas militares que se les va entregando, para no tener dificultad luego, además es posible ingresar sus datos personales así como su especialidad, grado, reparto al que pertenecen también cuentan con una identificación o código de cada Cadete. Con esta pantalla se logra tener un sistema de información de las tallas de su vestuario y de su calzado.

Dentro de esta pantalla se cuenta con dos botones que permiten el **Pase de Año del Cadete**, al ejecutarlo saldrá un mensaje el cual debe leer y estar seguro si lo hace o no. El

otro botón permite **Dar de Baja** al Cadete que se retire de la ESMA. Para realizar estos dos procedimientos de debe estar completamente seguros, caso contrario el procedimiento es irre recuperable, y puede causar problemas a los usuarios.

TARJETA INDIVIDUAL DE VESTUARIO

Id. Alumno: 001 **Año:** 2 **Apellidos y Nombres:** CASA MARCELO

Grado: **Especialidad:** TECNICO **Reparto:** SALIJAS

Camisa: **Pantalón:** **Overol:** **Chompa:** **Panacho:**

Calzado: **Gorra:** **Guantes:** **Otro:** **Otro:**

Fecha	No. Docume	Id. Rubro	Descripción	Cantidad	Devo
19/05/04	01	00003	COLONIA	2	<input type="checkbox"/>

Figura 2.3. Pantalla de Tarjetas Individuales de Vestuario

Pantalla del Inventario

Esta pantalla nos permite realizar el inventario de la cantidad existente en bodega, nos permite llevar un control de la dotación que se entrega cada año a los Cadetes durante

su permanencia en la ESMA. Permitiendo llevar un control de lo entregado y de la existencia de la dotación, así como un control del valor unitario de del valor total.

También permite llevar un control de la cantidad máxima que deberíamos entregar y una cantidad mínima con la que deberíamos contar en bodega para no sufrir de un desabastecimiento total.

Id. Rubro:	Id. Grupo:	Descripción:
00001	00001	AFEITADORA
Cantidad:	Primer Año:	Entregado:
1.000	2	6
Canmax:	Segundo Año:	Existencia:
1.000	2	994
Canmin:	Tercer Año:	Valor Unitario:
10	1	0,05
	Cuarto Año:	Valor Total:
	1	0,30

Figura 2.4. Pantalla del Inventario

Pantalla de Rubros

En esta pantalla se generan las diferentes cantidades de los productos que se van a entregar a los cadetes durante los cuatro años, se podrá ver la existencia en bodega y la

cantidad entregada. Mientras se ingresa en la pantalla de Inventario en esta pantalla de rubros se genera automáticamente, presenta la misma apariencia de las hojas de Microsoft Excel, aquí se van generando en una lista automáticamente, y si nos equivocamos en algo nos permite realizar el cambio correspondiente.

Id. Rubro	Id. Grupo	Descripción	Prim	Segu	Terc	Cua	Canti	Entre	Exist	Valor Ur	Valor Tot	Des
00001	00001	AFEITADORA	1	2	1	1	20	5	15	1,00	5,00	<input type="checkbox"/>
00001	00003	UNIFORMES	4	7	9	5	100	25	75	9,00	10,00	<input type="checkbox"/>
00003	00002	COLONIA			4		0			1,00	10,00	<input type="checkbox"/>
00004	00005	uniformes	1				5	0	0		0,00	<input type="checkbox"/>
00005	0004	LAPIZ										<input type="checkbox"/>
00006												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>
												<input type="checkbox"/>

Figura 2.5. Pantalla de Rubros

Pantalla de Proveedores

En esta pantalla se ingresan todos los datos de los proveedores, de esta manera se los puede identificar por Empresas o por el Nombre del Contacto.

The screenshot shows a window titled 'PROVEEDORES' with a yellow background. The form contains the following fields and values:

Id. Proveedor:	Empresa:		
0000000002	COMPUTONER		
Nombre de Contacto:		Cargo del Contacto:	
LIC.OLGA VERA		VENDEDORA	
Dirección:		Ciudad:	
SALINAS		GUAYAQUIL	
Código Postal	Región:	País:	Telefono:
33333	COSTA	ECUADOR	04528963
Fax:	Célular:	Terminos:	
0000000000000000	0502339854	CONTRATO	
Nota:	DEBE TRAER PROFORMAS DE COMPUTADORAS PORTÁTILES.		

At the bottom of the window, there is a navigation bar with the following buttons: **Primero**, **Anterior**, **Siguiente** (highlighted), **Último**, **Buscar**, **Imprimir**, **Agregar**, **Modificar**, **Eliminar**, and **Salir**.

Figura 2.6. Pantalla de Proveedores

Pantalla de Usuarios

Esta pantalla está relacionada con las personas que manipularán el sistema, aquí es donde se almacena la información necesaria para que un determinado usuario pueda ingresar o no al sistema.

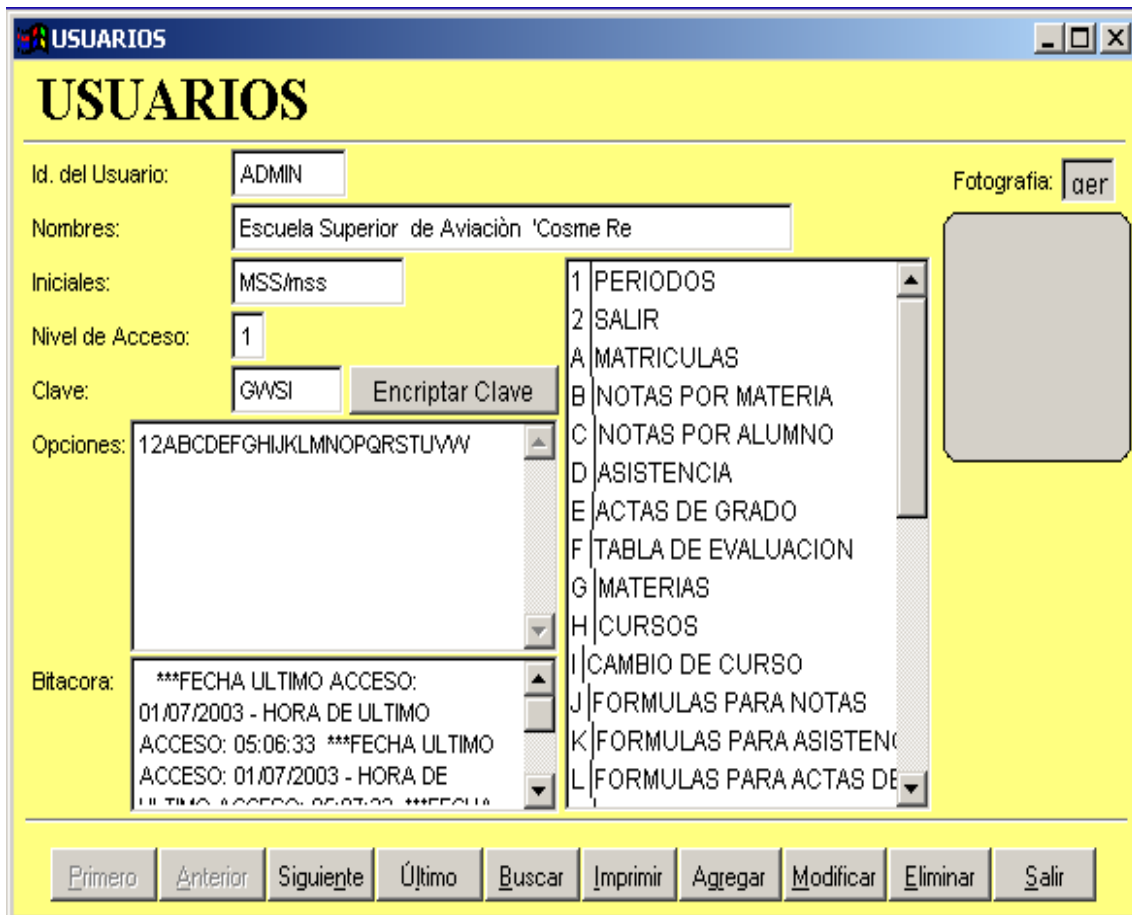


Figura 2.7. Pantalla de Usuarios.

2.5.- DISEÑO DE MENÚS

Si el usuario pertenece al nivel 1 podrá acceder a todas las opciones, principalmente a la de parámetros, que permite crear nuevos usuarios y realizar el mantenimiento del sistema.

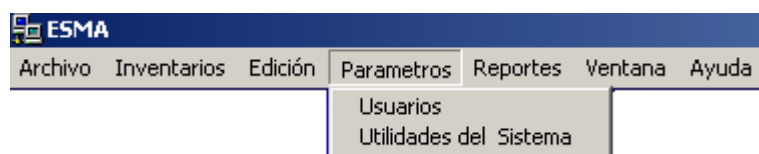


Figura 2.8. Menú Parámetros Activado

En el nivel de acceso 2 se desactiva el menú Parámetros.

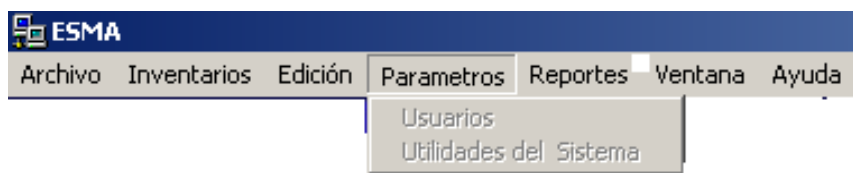


Figura 2.9. Menú Parámetros Desactivado

En el menú inventarios se encuentran los siguientes submenús:

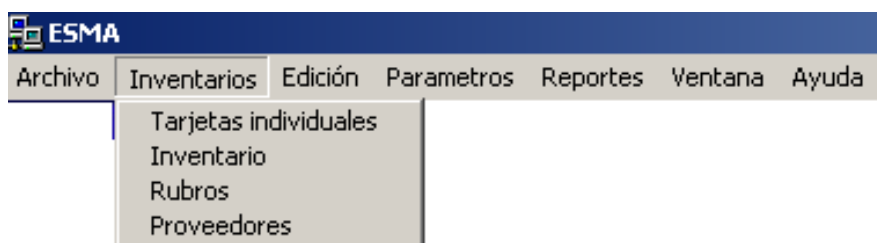


Figura 2.10. Menú Inventario

En el menú Reportes se encuentran los siguientes submenús:

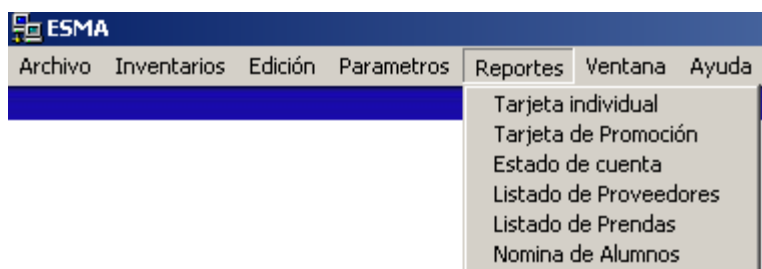


Figura 2.11. Menú Reportes

MENÚ AYUDA

Al dar clic en el menú ayuda se visualiza una pantalla, esta contiene una dirección de Internet, al momento que damos clic en aceptar se visualizará el manual de usuario y se podrá revisar la información que esta contiene.

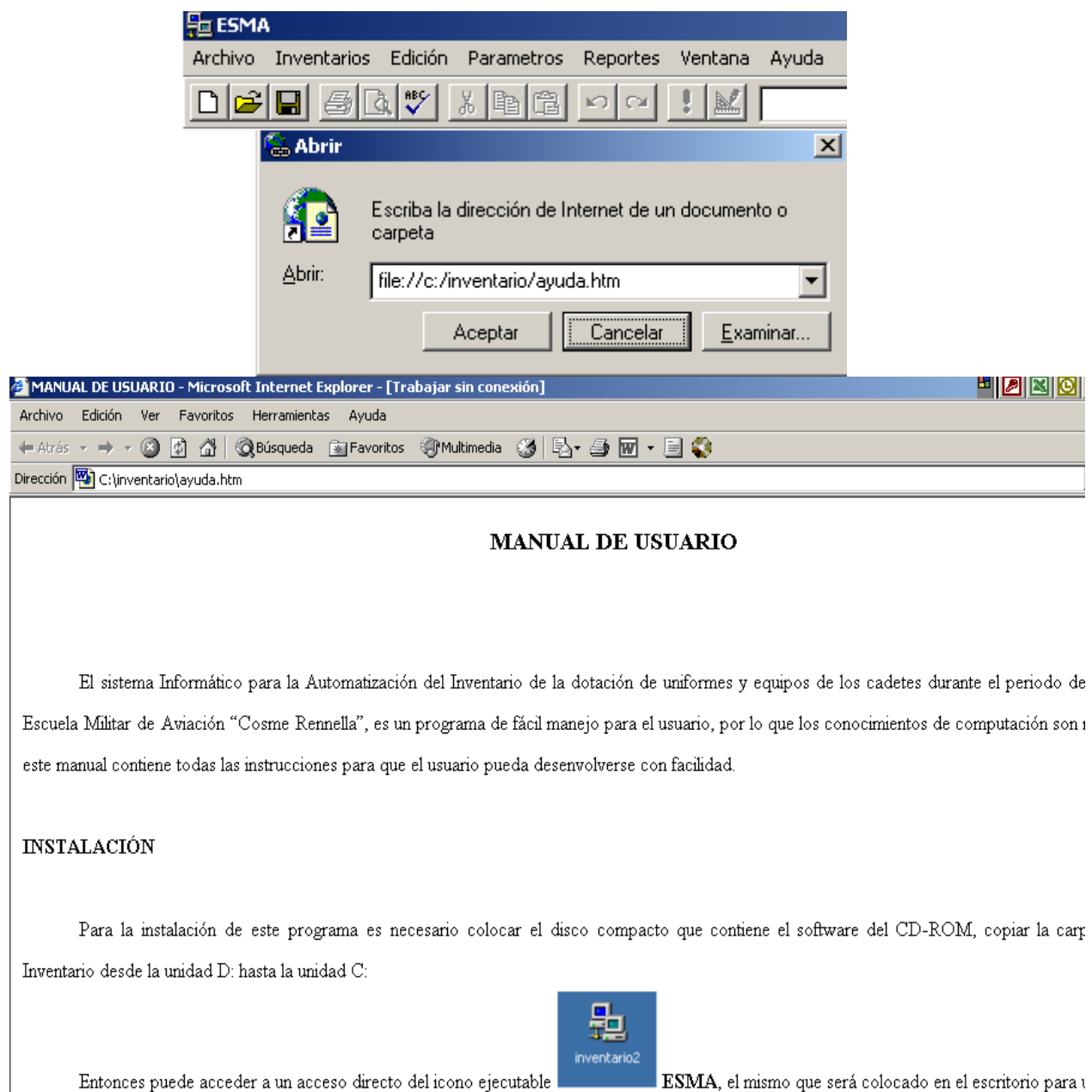


Figura 2.12. Menú Ayuda

CAPÍTULO III

DESARROLLO DEL SISTEMA

3.1.- ALMACENAMIENTO DE DATOS

Se puede almacenar los datos de dos formas:

1. Directamente en tiempo de diseño, en las tablas a las que pertenezca la información, así por ejemplo

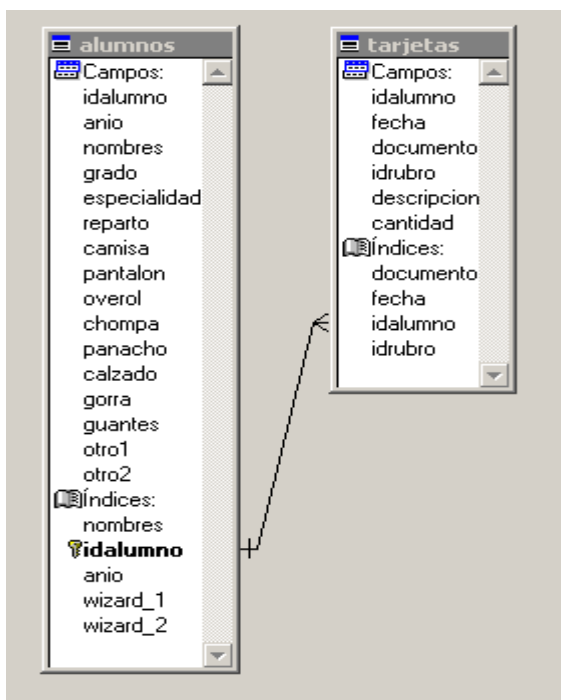


Figura 3.1. Almacenamiento de datos en tiempo de diseño

2. En tiempo de ejecución a través de los formularios creados, así:

TARJETA INDIVIDUAL VESTUARIO

Id. Alumno: 0000000001 **Año:** 1 **Apellidos y Nombres:** ASCASUBI BOHORQUEZ ANDRES

Grado: **Especialidad:** PILOTO **Reparto:**

Camisa: **Pantalón:** **Overol:** **Chompa:** **Panacho:**

Calzado: **Gorra:** **Guantes:** **Otro:** **Otro:**

Figura 3.2. Almacenamiento de datos en tiempo de ejecución

Bases de datos.

Una base de datos es una fuente central de datos que está diseñada para que sea compartida por muchos usuarios.

La parte medular de la base de datos es el DBMS (sistema de manejo de base de datos) que permite la creación, modificación y actualización de la base de datos, la recuperación de datos y la generación de reportes.

Los objetivos de efectividad de la base de datos incluyen:

- Mantener datos que sean precisos y consistentes.
- Asegurarse de que la base de datos pueda ser compartida entre los usuarios de una diversidad de aplicaciones.

- Permitir que la base de datos evolucione y que las necesidades de los usuarios crezcan.

Una base de datos bien diseñada también puede ser más flexible que archivos separados, es decir, una base de datos puede evolucionar conforme cambian las necesidades de los usuarios y las aplicaciones.

Por último, el enfoque de base de datos tiene la ventaja de permitir que los usuarios tengan su propia vista de los datos. Los usuarios no necesitan preocuparse de la estructura actual de la base de datos o de su almacenamiento físico.



Figura 3. 3 Base de Datos del Sistema.

3.2.- CREACIÓN DE PANTALLAS

Las pantallas vienen a ser la interfaz con el usuario. Por lo tanto una interfaz bien diseñada puede guiar a los usuarios por el sistema sin complicaciones.

Los formularios, clases, controles, menús y barras de herramientas ofrecen un conjunto variado de herramientas para diseñar una excelente interfaz de usuario.

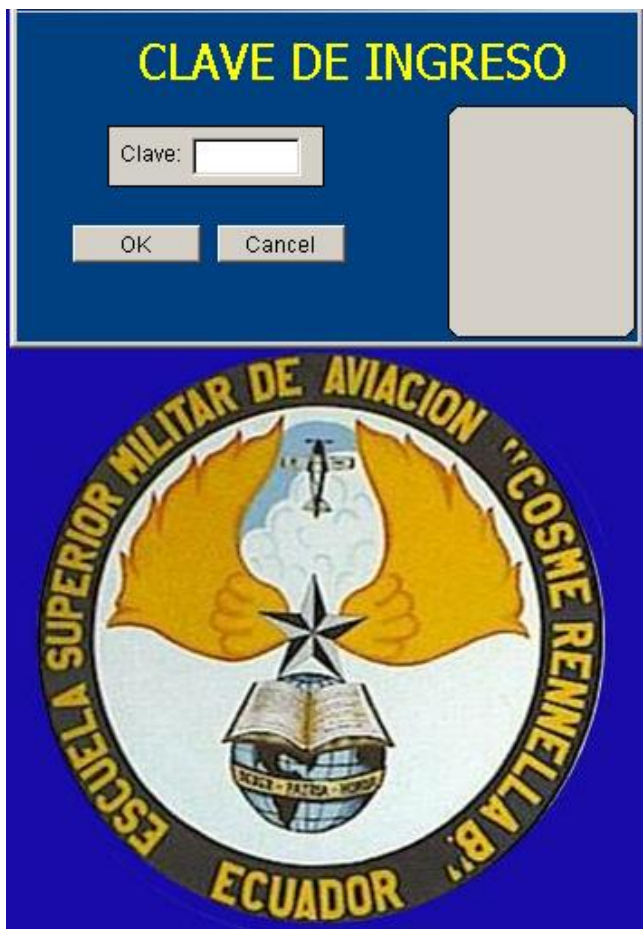


Figura 3.3. Pantalla de Presentación

Crear formularios

Los formularios no sólo sirven para ofrecer a los usuarios una interfaz familiar para ver e introducir datos en una base de datos, sino que también ofrecen un amplio conjunto de objetos que pueden responder a los eventos del usuario (o del sistema) permitiéndoles realizar las tareas de administración de información de la forma más sencilla e intuitiva posible.

Para crear un formulario nuevo

- En el Administrador de proyectos, seleccione **Formularios** y elija **Nuevo**.

-O bien-

En el menú **Archivo**, elija **Nuevo**, seleccione **Formulario** y, a continuación, elija **Nuevo archivo**.

-O bien-

- Utilice el comando CREATE FORM.

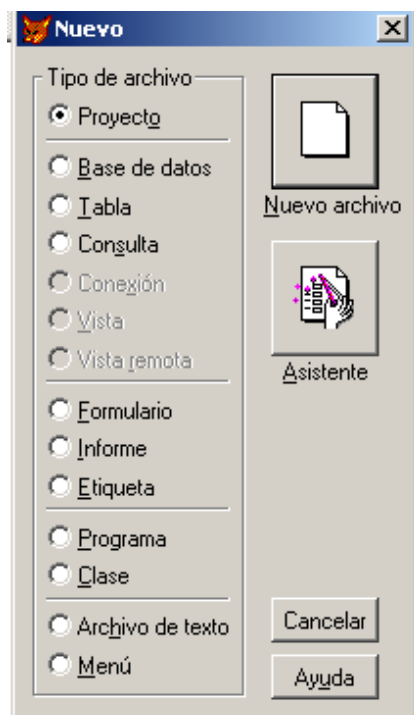


Figura 3.4. Presentación para Formularios

Por medio del menú llegamos al diseñador de formularios.

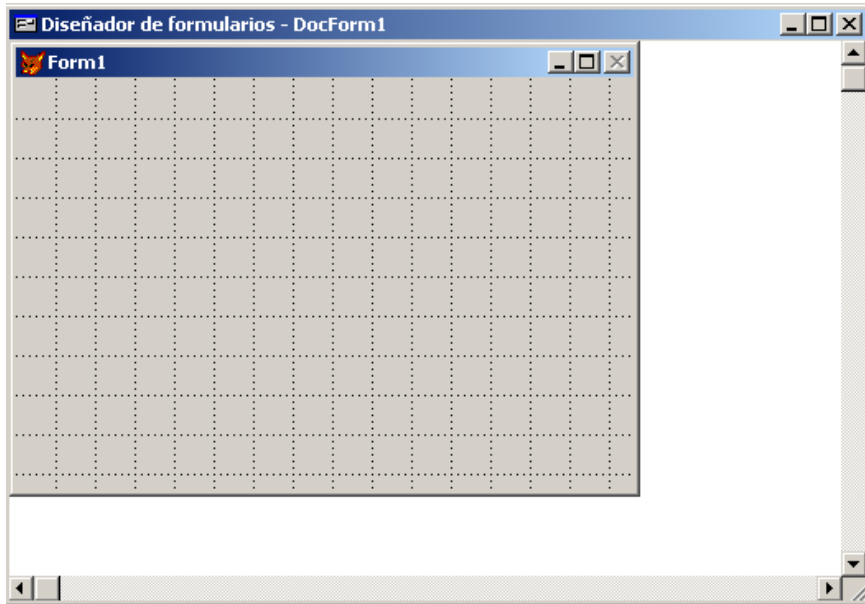


Figura 3.5 Diseñador de formularios

Para poder cambiar o manipular las propiedades, tanto del formulario como de los objetos que puede contener, tenemos una ventana específica.

Los formularios y conjuntos de formularios son objetos con sus propias propiedades, eventos y métodos

Qué son los Controles

Los controles son el medio fundamental de interacción de los usuarios. Se utilizan para manipular datos que se encuentran en ellos y llevar a cabo distintas tareas.

Se colocan controles en el formulario, de acuerdo a la necesidad del usuario.

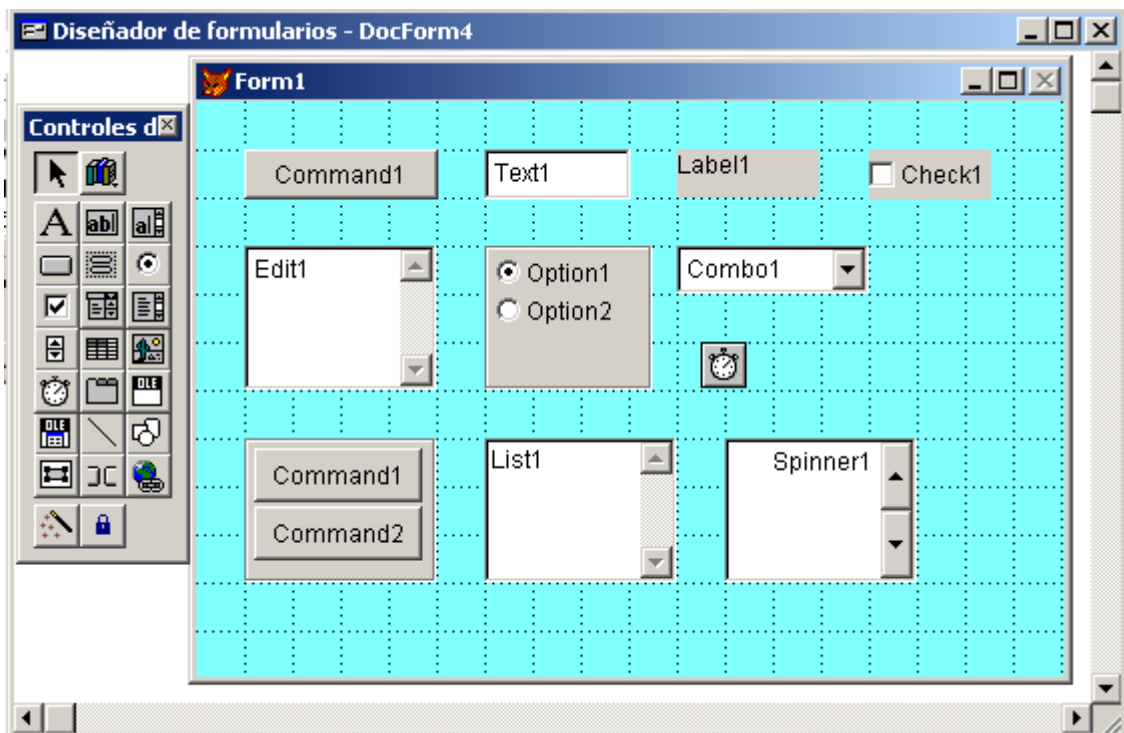


Figura 3.6. Forma Visual de distintos controles

Diseñar Menús y Barras de Herramientas

Cada diseñador dispone de una o varias barras de herramientas que proporcionan acceso rápido a las características o acciones más habituales de la herramienta. Por ejemplo, el Diseñador de formularios tiene barras de herramientas distintas para controles y diseño de controles, y una paleta de colores.

Puede situar en la pantalla tantas barras de herramientas como necesite mientras trabaja. También puede acoplar barras de herramientas en la parte superior, inferior o en los laterales de la pantalla para personalizar el entorno de trabajo

Para mostrar barras de herramientas

1. En el menú **Ver**, elija **Barras de herramientas**.
2. En el cuadro de diálogo **Barras de herramientas**, seleccione las barras de herramientas que desea utilizar.
3. Elija **Aceptar**

Para acoplar una barra de herramientas

- Arrastre la barra de herramientas hasta la parte superior, inferior o a los laterales de la pantalla.

Diseñadores de menús

Utilice el **Diseñador de menús** para crear menús, elementos de menú, submenús de elementos de menú, líneas que separen grupos de elementos de menú relacionados, etc.

Debido a que a menudo, los usuarios examinan los menús antes de buscar información sobre la aplicación en cualquier otro lugar.

3.3 PROGRAMACIÓN

La programación orientada a objetos representa un cambio de perspectiva con respecto a la programación estándar por procedimientos.

Las herramientas utilizadas son: **comandos, funciones y operadores.**



```
STORE .F. TO ingresa
```

```
mclave = ALLTRIM(THISFORM.text1.TEXT)
```

```
STORE LEN(mclave) TO N
```

```

STORE " TO emcriptada

FOR i=1 TO N

    STORE emcriptada + CHR(ASC(SUBSTR(mclave,i,1))+i*2)TO emcriptada

ENDFOR

LOCATE FOR ALLTRIM(usuarios.clave) = ALLTRIM(emcriptada)

IF EOF()

    THISFORM.label4.CAPTION = 'ILEGAL ...'

    thisform.olecontrol1.visible=.t.

    WAIT " TIMEOUT 0.5

    thisform.olecontrol1.visible=.f.

    thisform.olecontrol2.visible=.t.

    WAIT " TIMEOUT 3

    thisform.olecontrol2.visible=.f.

    THISFORM.RELEASE

    CLEAR EVENTS

ELSE

    STORE .T. TO ingresa

    mbitacora = ' ***FECHA ULTIMO ACCESO: '+DTC(DATE())+' - HORA DE ULTIMO ACCESO:
'+TIME()

    xiniciales=usuarios.iniciales

    xopciones=usuarios.opciones

    REPLACE usuarios.bitacora WITH mbitacora ADDITIVE

    thisform.oleboundcontrol1.visible=.t.

    THISFORM.label4.CAPTION = '* BIENVENIDO *'

    THISFORM.label3.CAPTION = usuarios.nomusu

    * ThisForm.Oleboundcontrol1.Refresh

    THISFORM.REFRESH

    WAIT " TIMEOUT 1

```

```

IF TYPE("THISFORM.Parent") = "O"

    THISFORMSET.RELEASE

ELSE

    THISFORM.RELEASE

ENDIF

```

```

public accesos
dimension accesos(30,2)
accesos(1,1)="A"
accesos(1,2)="SALIR"
accesos(2,1)="B"
accesos(2,2)="TARJETAS INDIVIDUALES"
accesos(3,1)="C"
accesos(3,2)="INVENTARIO"
accesos(4,1)="D"
accesos(4,2)="RUBROS"
accesos(5,1)="E"
accesos(5,2)="PROVEEDORES"
accesos(6,1)="F"
accesos(6,2)="USUARIOS"
accesos(7,1)="G"
accesos(7,2)="UTILIDADES DEL SISTEMA"

```

```

accesos(8,1)="H"
accesos(8,2)="TARJETA INDIVIDUAL"
accesos(9,1)="I"
accesos(9,2)="TARJETA DE PROMOCION"
accesos(10,1)="J"
accesos(10,2)="ESTADO DE CUENTA"
accesos(11,1)="K"
accesos(11,2)="LISTADO DE PROVEEDORES"
accesos(12,1)="L"
accesos(12,2)="LISTADO DE PRENDAS"
accesos(13,1)="M"
accesos(13,2)="NOMINA DE ALUMNOS"
accesos(13,1)="N"
accesos(13,2)="GRUPOS"

```

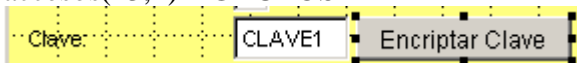


Figura 3.9. Programación para Encriptar la Clave

```

MESSAGEBOX("¿Se va a Encriptar la Clave?")
MCLAVE = ALLTRIM(ThisForm.CLAVE1.TEXT)
STORE LEN(MCLAVE) TO N
STORE " TO emcriptada
FOR i=1 TO n
    STORE emcriptada + CHR(ASC(SUBSTR(MCLAVE,i,1))+i*2)TO emcriptada
ENDFOR
ThisForm.CLAVE1.Value=emcriptada

```

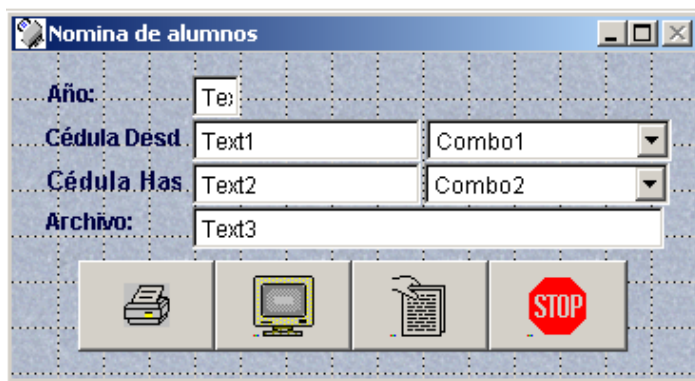


Figura 3.10 Programación de Reporte de Nómina de Alumnos

```

Public manio,mdesde,mhasta,marchivo
store " to mdesde,mhasta,marchivo
store 0 to manio
SET DATE short

```



```
manio=val(ThisForm.Text4.value)
```

```
if manio>0 then
```

```
Report Form nomina NOCONSOLE For alumnos.anio=manio .and.  
alumnos.idalumno>=mdesde .and. alumnos.idalumno<=mhasta Preview
```

```
else
```

```
Report Form nomina NOCONSOLE For alumnos.idalumno>=mdesde .and.  
alumnos.idalumno<=mhasta Preview
```

```
Endif
```

INVENTARIO

Id. Rubro: Text1 **Id. Grupo:** Text1 **Descripción:** Text1

Cantidad: Text1 **Primer Año:** Text1 **Entregado:** Text1

Canmax: Text1 **Segundo Año:** Text1 **Existencia:** Text1

Canmin: Text1 **Tercer Año:** Text1 **Valor Unitario:** Text1

Cuarto Año: Text1 **Valor Total:** Text1

Primero Anterior Siguiente Último Buscar Imprimir Agregar Modificar Eliminar Salir

```
SET SAFETY OFF
```

```
SET SCOREBOARD OFF
```

```
SET CENTURY ON
```

```
SET DECIMAL TO 3
```

```
SET ECHO OFF
```

```
SET TALK OFF
```

```
SET CLOCK ON
```

```
SET DELETE ON
```

```
SET ESCAPE OFF
```

```
SET DATE DMY&&LONG
```

```

SET STATUS OFF

SET EXCLUSIVE OFF

CLEAR WINDOWS ALL

RELEASE WINDOWS ALL

DEACTIVATE WINDOW "Comandos"

DEACTIVATE WINDOW "Administrador de proyectos"

PUBLIC ingresa,diract, xiniciales, xopciones, mdescrip
mdescrip=""

dimension matmat(20,2)

STORE .F. TO ingresa

STORE SYS(5)+SYS(2003)+'\ ' TO diract

SET DEFA TO &DIRACT

_SCREEN.PICTURE=diract+"ENTRADA.JPG"

_SCREEN.CAPTION="ESMA"

_SCREEN.WindowState = 2

DO FORM clave

quit

```

3.4.- PRESENTACIÓN E INTERACCIÓN DE PANTALLAS

La presentación de las pantallas se hace al llamar al formulario desde el menú, en el cual se puede encontrar varias opciones de interacción entre el sistema y el usuario.

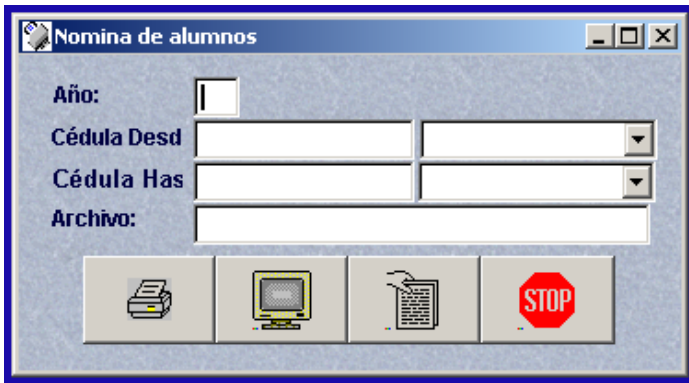


Figura 3.11 Presentación de Pantalla

De manera que el usuario no tenga problemas para trabajar con estas pantallas, de son desarrolladas de una forma gráfica, agradable para el usuario final y entendible.

CAPÍTULO IV

PRUEBAS Y ANÁLISIS DE RESULTADOS

4.1.- IMPLANTACIÓN DEL SISTEMA

Cuanto más conozca el lenguaje, mayor será la rapidez con que pueda programar y podrá realizar más pruebas preliminares mentalmente.

Una vez que se tiene la habilidad de manipular este lenguaje, se procede a la fase de desarrollo del sistema, adicionando el entrenamiento a los usuarios, quienes van a tener contacto con el sistema, siempre bajo la supervisión de los programadores.

El sistema creado se puede implantar de dos maneras:

- Manera paralela donde se corren o ejecutan los dos sistemas a la vez, el manual y el nuevo. Mientras transcurre el tiempo se van acoplando al sistema nuevo y el anterior se va dejándolo en el pasado.
- La siguiente manera es de forma independiente donde se trabaja solo con el sistema actual, dejando de trabajar totalmente con el sistema manual.

Al realizar esto, se debe realizar una conversión total de archivos de formatos antiguos a los nuevos, la construcción de una base de datos, la instalación del equipo y ponerlo en funcionamiento el nuevo sistema.

TARJETA INDIVIDUAL VESTUARIO

TARJETA INDIVIDUAL DE VESTUARIO

Id. Alumno: 001 Año: 2 Apellidos y Nombres: CASA MARCELO

Grado: Especialidad: TECNICO Reparto: SALIJAS

Camisa: Pantalón: Overol: Chompa: Panacho:

Calzado: Gorra: Guantes: Otro: Otro:

Fecha	No. Docume	Id. Rubro	Descripción	Cantidad	Devoluc
19/05/04	01	00003	COLONIA	2	<input type="checkbox"/>

Figura 4.1. Funcionamiento del Sistema Automatizado

Con esto se debe lograr lo siguiente:

- Tener acceso directo al otro programa, incluyendo todos sus objetos y comandos.
- Puede compartir datos directamente con el otro programa sin tener que exportarlos o convertirlos a otro formato.
- Puede controlar el otro programa con el modelo familiar de propiedades y métodos.
- El otro programa no tiene que estar visible necesariamente cuando lo llame. Por ejemplo, puede llamar a Microsoft Excel, colocar algunos datos en celdas, ejecutar un cálculo complejo en los datos, leer el resultado y, a continuación, mostrarlo en Visual FoxPro, todo sin mostrar Microsoft Excel.

- Los comandos (métodos y propiedades) para controlar el otro programa están incrustados en programas de Visual FoxPro familiares. No tiene que aprender un lenguaje de programación diferente para poder controlar el otro programa.

4.2 PRUEBAS Y MANTENIMIENTO DEL SISTEMA

Antes de poner en funcionamiento este o cualquier sistema, debe ser chequeado y probado.

Es más barato encontrar problemas antes de que el sistema sea entregado a los usuarios. Muchas de las pruebas son realizadas por los programadores y en ocasiones con ayuda de analistas.

Por ejemplo, probablemente concibe la interfaz de usuario de su aplicación en términos de formularios, menús e informes de Visual FoxPro. Además, cuando programa aplicaciones en Visual FoxPro, pensará almacenar los datos de la aplicación en tablas de Visual FoxPro. Por esto el mantenimiento del sistema y su documentación, comienza aquí, y se lo debe hacer durante toda la vida del sistema de manera rutinaria.

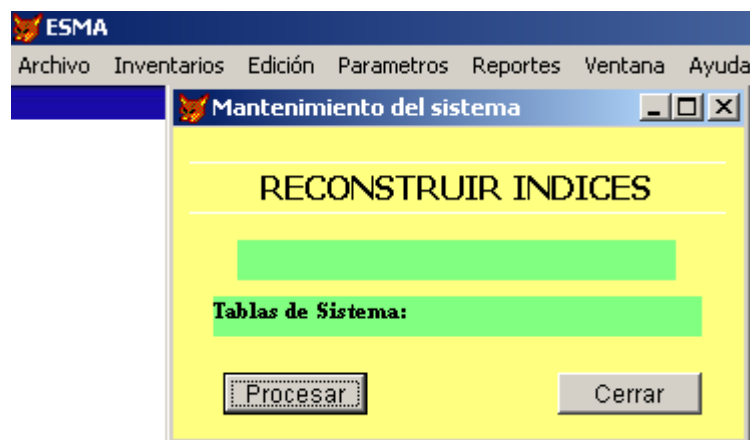


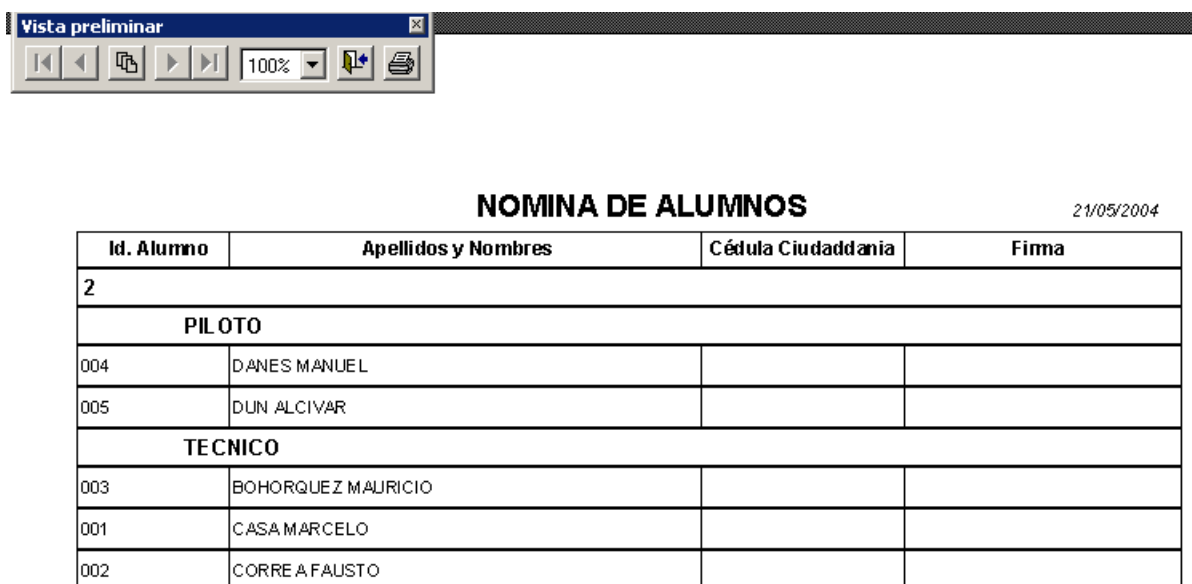
Figura 4.2 Pantalla de Mantenimiento del Sistema

Los procedimientos sistemáticos que se emplean, pueden ayudar a asegurar que el mantenimiento se mantenga al mínimo. De aquí que este procedimiento se rutinario, pues los negocios gastan una gran cantidad de tiempo y de dinero realizando este procedimiento.

El mantenimiento se lo realiza por dos razones:

- ✚ La primera de estas es para corregir errores de software
- ✚ La segunda es para realizar el mantenimiento del sistema, con esto se mejora la capacidad del software en respuesta a las necesidades organizacionales.

Los usuarios con regularidad solicitan características adicionales, mucho después de familiarizarse con el sistema, muchas veces son simples, como el desplegado de datos adicionales en un reporte. O demasiadas complicadas como realizar en su totalidad un nuevo sistema.



NOMINA DE ALUMNOS			
			21/05/2004
Id. Alumno	Apellidos y Nombres	Cédula Ciudadania	Firma
2			
PILOTO			
004	DANES MANUEL		
005	DUN ALCIVAR		
TECNICO			
003	BOHORQUEZ MAURICIO		
001	CASA MARCELO		
002	CORREA FAUSTO		

Figura 4.3. Reporte Desplegado

El hardware y software están cambiando a un ritmo acelerado. Un sistema que usa tecnología antigua puede ser modificado para usar las capacidades de una tecnología

nueva, esto es un poco más costoso y complicado y toma un poco más de tiempo, pero se lo puede realizar.

Un ejemplo de esto es el cambio o reemplazo de una terminal de microcomputadora con una estación de trabajo de una microcomputadora o una computadora de escritorio, etc.

Después de que el sistema está instalado se le debe dar mantenimiento preventivo y correctivo en caso de ser necesario, esto significa que los programas de computadora deben ser modificados o mantenerlos actualizados.

Conforme aumenta la cantidad de programas escritos, también aumenta la cantidad de mantenimiento que requieren.

Es por ello que es más conveniente, realizar un nuevo estudio del sistema, debido al costo que esto genera.

El costo del mantenimiento es mayor que la creación de un sistema de información completamente nuevo.

4.3.- EVALUACIÓN

La evaluación debe ser considerada como parte final del ciclo de vida del desarrollo del sistema, la evaluación se la debe realizar en cada fase del desarrollo del sistema.

El principal criterio como programadores debe ser, si el usuario ya está manejando el sistema y de cómo este responde.

Una forma más eficaz de interactuar con otras aplicaciones es usar la Automatización. Con programas de Visual FoxPro, puede tener acceso a los objetos

Pues los sistemas trabajan de forma cíclica, termina una fase del desarrollo del sistema y se pasa a la siguiente, logrando con esto determinar la existencia de errores en el sistema, obligando a regresar a la fase anterior y modificarlo.

Solo así se logra que el usuario final no tenga complicaciones al momento de poner en funcionamiento el sistema realizado.

CAPÍTULO V

ESTUDIO ECONÓMICO

Este capítulo se detallaran los costos que se invirtieron en el desarrollo del proyecto.

Aprobado el perfil de Tesis del Proyecto de Gado, se realizó un estudio económico para la realización del Sistema Informático para la Automatización del Inventario de Dotación de Uniformes y Equipos de los Cadetes durante su periodo de formación en la Escuela Superior Militar de Aviación “Cosme Rennella”.

Debido a que el objetivo de este proyecto, es aportar con algo al desarrollo intelectual y técnico del personal del Departamento de Abastecimientos de la Escuela Superior Militar de Aviación “Cosme Rennella”.

Con esto se persigue poder satisfacer la necesidad institucional, ya que con este sistema informático lograremos reducir el gasto de tiempo y dinero, que conlleva el tener un sistema de inventario manual, pues provoca descontento en las personas que desempeñan estas labores.

5.1.- PRESUPUESTO

Realizado un estudio antes de concretar este proyecto, se llego a la conclusión de que el sistema informático costaba 700.00 USD.

5.2.- ANÁLISIS ECONOMICO-FINANCIERO.

Existen 2 rubros en la construcción del sistema informático que son:

- **MATERIAL DE OFICINA**

Este rubro comprende los materiales de oficina utilizados en la elaboración del proyecto, alquiler de computadora, consultas en Internet, impresiones, empastados, etc.

Tabla 5.1. Lista de costos de gastos de material de oficina.

MATERIAL DE OFICINA	
DETALLE	VALOR
Material de oficina, Internet, alquiler de computadora, impresiones, escáner, anillado, empastado, etc.	400.00
COSTO TOTAL	400.00

- **MATERIAL DE APOYO**

Comprende todos los insumos de computación que se utilizaron para realizar este sistema informático.

Tabla 5.2. Lista de costos de gastos de material de apoyo.

MATERIAL DE APOYO	
DETALLE	VALOR
Manual de Usuario de Microsoft Visual FoxPro.	100.00
Disquetes	30.00
Cartuchos	60.00
Resmas de Hojas	20.00
Cd's	30.00
COSTO TOTAL	240.00

- **VARIOS**

Comprende todos los materiales que se necesitaron como fotocopias, imprevistos, etc.

Tabla 5.3. Lista de costos de gastos de varios.

VARIOS	
DETALLE	VALOR
Fotocopias	10.00
Imprevistos	50.00
COSTO TOTAL	60.00

Por lo tanto, el costo total de la elaboración del sistema informático se detalla a continuación:

Tabla 5.4. Lista de costo total de la elaboración del sistema informático

COSTO TOTAL DE ELABORACIÓN	
DETALLE	VALOR
Materiales de Oficina	400.00
Material de Apoyo	240.00
Varios	60.00
TOTAL	700.00

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1.- CONCLUSIONES

- Al concluir con este sistema, seremos capaces de identificar y aplicar las estructuras básicas de programación y conocer aspectos sobre la programación Orientada a Objetos.
- El sistema fue elaborado en Microsoft Visual FoxPro, que es un lenguaje de programación sencilla, de fácil entendimiento para programadores principiantes como programadores expertos.
- Al elaborar este sistema informático, se familiarizó con el entorno de desarrollo de aplicaciones para base de datos.
- Aplicamos conocimientos para la integración de una interfaz con el usuario y la base de datos.
- Microsoft Visual FoxPro, trata sobre el diseño, creación y manipulación de tablas libres o tablas con integridad referencial (base de datos).
- El sistema informático fue desarrollado con Microsoft Visual FoxPro, porque es un gestor de base de datos, orientado a la programación de Objetos y a la gestión de base de datos y desarrollo de aplicaciones.

6.2.- RECOMENDACIONES

- Se recomienda trabajar con Microsoft Visual FoxPro, porque con ello aprendemos a diseñar las salidas de los sistemas de información, haciendo uso de las herramientas gráficas que el programa ofrece.

- Se ha elegido Microsoft Visual FoxPro, por ser un lenguaje de mucha rapidez, lo que lo convierte en uno de los lenguajes más rápidos del mercado, así como contiene mucha potencia en el manejo de las bases de datos.

- También se recomienda realizar una copia de respaldo por lo menos cada mes, para evitar la pérdida de información en caso de algún accidente.

- El software debe instalarse en un computador Pentium 4, porque tiene mayor nivel de operación en el ambiente militar.

BIBLIOGRAFÍA

MANUALES

Manual de Usuario Visual FoxPro 6.0

Manual MSDN Library Visual Studio 6.0

INTERNET

www.lafacu.com

www.monografias.com

www.bibliotecavirtual.com

DOCUMENTOS

Documentación de la ESMA.

A

N

E

X

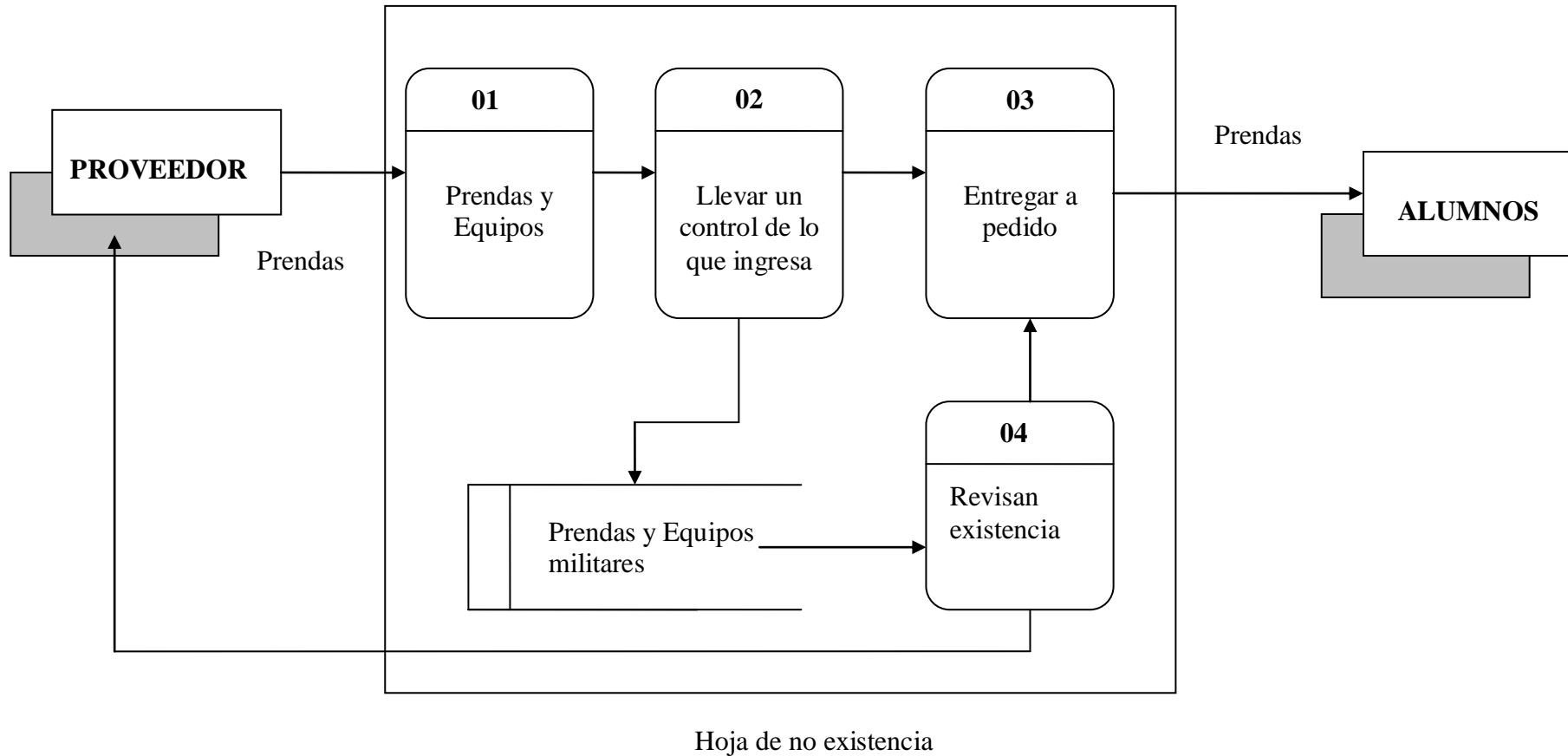
O

S

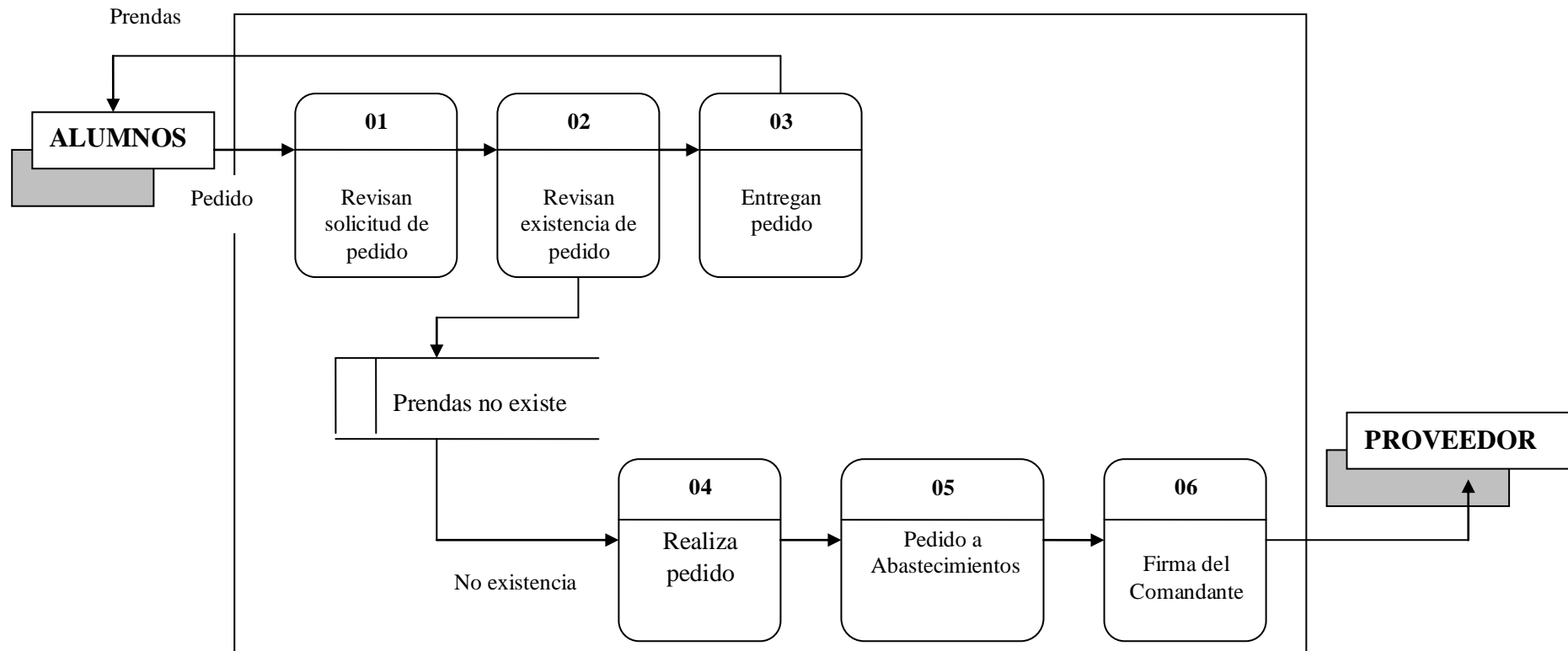
FUERZA AEREA ECUATORIANA

INSTITUTO TECNOLÓGICO SUPERIOR AERONÁUTICO

DIAGRAMA DE FLUJO DE LA SECCIÓN DE ABASTECIMIENTOS

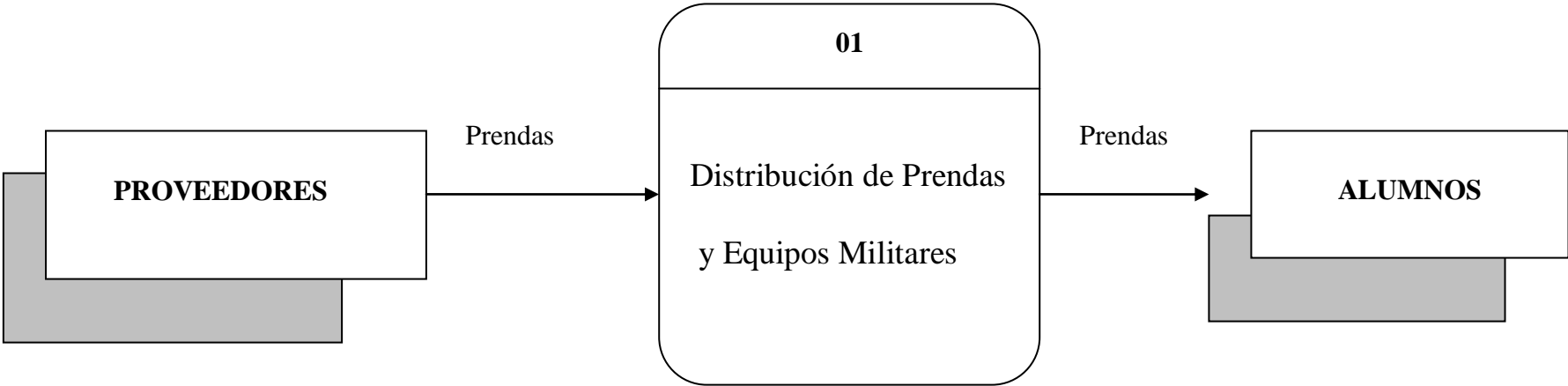


FUERZA AEREA ECUATORIANA
INSTITUTO TECNÓLOGICO SUPERIOR AERONÁUTICO
DIAGRAMA DE FLUJO DEL PEDIDO DE LOS ALUMNOS



FUERZA AEREA ECUATORIANA
INSTITUTO TECNOLÓGICO SUPERIOR AERONÁUTICO

DIAGRAMA DE FLUJO DE CONTEXTO



HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

Correa Báez Mariana Araceli

DIRECTOR DE ESCUELAS DEL INSTITUTO TECNOLÓGICO
SUPERIOR AERONÁUTICO

Ing. Guillermo Trujillo J.

Director de Carreras del ITSA

Lugar y fecha: Latacunga a, 16 de Junio del 2004