

**SEP**

INSTITUTO TECNOLOGICO  
de parral

**“SYSTEM MOLDURAS”**

TRABAJO PROFESIONAL QUE PARA OBTENER  
EL TITULO DE

**LICENCIADO EN INFORMÁTICA**

P R E S E N T A

**GUSTAVO SANDOVAL MONTES**

ENERO 2008



SUBSECRETARÍA DE EDUCACIÓN SUPERIOR  
DIRECCIÓN GENERAL DE EDUCACIÓN  
SUPERIOR TECNOLÓGICA  
INSTITUTO TECNOLÓGICO DE PARRAL



Sistema Nacional de Educación Superior Tecnológica



Dirección General de Educación Superior Tecnológica



SECRETARÍA DE  
EDUCACIÓN PÚBLICA

**SEP**

**DIVISION DE ESTUDIOS PROFESIONALES**  
**OFICIO No. A338/08**  
**FECHA: 25 de feb/08**

**C. GUSTAVO SANDOVAL MONTES**  
**Presente.-**

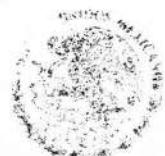
**Se le autoriza la impresión del trabajo profesional de la carrera de  
LIC. EN INFORMÁTICA**

**"SYSTEM MOLDURAS"**

**Con el proceder a su titulación bajo la opción "X MEMORIA DE  
RESIDENCIA PROFESIONAL" correspondiente al reglamento vigente.**

**Atentamente  
"POR UN ESPÍRITU CREADOR Y HUMANO"**

**LIC. SERGIO EDUARDO NUÑEZ CARAVEO  
Jefe División de Estudios Profesionales**



**SECRETARÍA DE  
EDUCACIÓN PÚBLICA  
INSTITUTO TECNOLÓGICO  
DE PARRAL  
DIVISIÓN DE ESTUDIOS  
PROFESIONALES**

SUBSECRETARÍA DE EDUCACIÓN SUPERIOR  
DIRECCIÓN GENERAL DE EDUCACIÓN  
SUPERIOR TECNOLÓGICA  
INSTITUTO TECNOLÓGICO DE PARRAL



SECRETARÍA DE  
EDUCACIÓN PÚBLICA



SEP

DIVISION DE ESTUDIOS PROFESIONALES  
OFICIO No. C338/08  
FECHA: 25 FEB/08

C. GUSTAVO SANDOVAL MONTES  
**Presente.-**

En respuesta a su solicitud de titulación bajo la opción “**X MEMORIA DE RESIDENCIA PROFESIONAL**” del correspondiente reglamento vigente, ésta División de Estudios autoriza el desarrollo del trabajo profesional de la carrera de **LIC. EN INFORMÁTICA** el siguiente:

CONTENIDO:

- I INTRODUCCIÓN
- II JUSTIFICACIÓN
- III OBJETIVOS GENERALES
- IV CARACTERIZACIÓN DEL ÁREA EN QUE SE PARTICIPÓ
- V PROBLEMAS A RESOLVER PRIORIZÁNDOLOS
- VI ALCANCES Y LIMITACIONES
- VII FUNDAMENTO TEÓRICO
- VIII PROCEDIMIENTO Y DESCRIPCIÓN DE ACTIVIDADES
- IX RESULTADOS
- X CONCLUSIÓN
- XI RECOMENDACIÓN
- XII BIBLIOGRAFÍA
- XIII ANEXOS



SECRETARÍA DE  
EDUCACIÓN PÚBLICA  
INSTITUTO TECNOLÓGICO  
DE PARRAL  
DIVISIÓN DE ESTUDIOS  
PROFESIONALES

Atentamente  
“POR UN ESPÍRITU CREADOR Y HUMANO”

~~LIC. SERGIO EDUARDO NUÑEZ CARAVEO~~  
Jefe División de Estudios Profesionales

Avenida Tecnológico # 57 C.P. 33850, Hidalgo del Parral, Chihuahua  
Tels. 627 52 3 03 36, 52 3 03 46, 52 3 02 12  
Fax 627 52 3 07 52



Molduras Ponderosa de Parral S. de R.L de C.V.  
Km. 2 Carr. A la Esmeralda  
Hgo. del Parral Chih., Mexico.

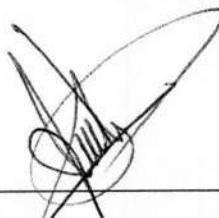
Asunto Carta de Terminación  
10 de Enero del 2008

**ING. JORGE LUIS GARCIA CANTON  
DIRECTOR DEL I.T. PARRAL  
P R E S E N T E.-**

Por medio de la presente me permito hacer constar que el **C. GUSTAVO SANDOVAL MONTES**, con No. De control **02410613**, de la carrera de **LIC. EN INFORMATICA**, realizo sus **RESIDENCIAS PROFESIONALES** en **MOLDURAS PONDEROSA DE PARRAL, S de R. L. de C.V.** con el proyecto de sistemas de inventarios llamado **"SYSTEM MOLDURAS"**. Realizadas durante el periodo de **Agosto 2007 a Enero 2008** del presente año acumulando un total de **640** horas.

Sin más por el momento me es grato quedar de usted

**A T E N T A M E N T E**

  
\_\_\_\_\_  
L.C. Jose Luis Chavira Pasillas  
Contador

RECIBIDO EN LA EMPRESA  
SISTEMAS DE INFORMATICA  
MOLDURAS PONDEROSA DE PARRAL S de R. L. de C.V.  
CARRERA 2 KM. 2 A LA ESMERALDA  
TEL. 01 (627) 522 80 67  
FAX 01 (627) 522 80 67

**INSTITUTO TECNOLOGICO DE PARRAL  
SISTEMA DE INVENTARIOS CON  
PÁGINA WEB**

**“SYSTEM MOLDURAS”**

ASESOR:

Lic. Jesús José Luna Carrete

gj

REVISORES:

Lic. Ever Ulysses Torres Carrillo

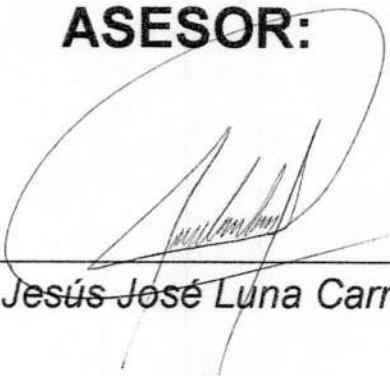
Lic. Gladis Patricia Moreno Hierro

# **INSTITUTO TECNOLOGICO DE PARRAL**

## **SISTEMA DE INVENTARIOS CON PAGINA WEB**

**“SYSTEM MOLDURAS”**

**ASESOR:**



Lic. Jesús José Luna Carrete

90

Calificación

Gustavo Sandoval Montes 02410613

# INSTITUTO TECNOLOGICO PARRAL

**RESIDENCIAS**

**“SYSTEM MOLDURAS”**

**ALMUNO:**

**GUSTAVO SANDOVAL MONTES**

**No. CONTROL**

**02410613**

**FECHA**

**25 de Febrero del 2008.**

# **INDICE**

<b>Introducción .....</b>	1
<b>Justificación .....</b>	2
<b>Objetivos Generales .....</b>	3
Objetivos Específicos .....	3
<b>Caracterización del área en que se participo.....</b>	4
Vista Interior .....	5
<b>Problemas a Resolver Priorizándolos .....</b>	6
<b>Alcances y limitaciones.....</b>	7
<b>Fundamento Teórico .....</b>	8
1. Lenguaje de Programación Delphi.....	11
2. Conceptos Básicos .....	13
3. Todo empieza con la madera.....	16
4. Clasificación según el aspecto visual de la madera.....	21
5. Organismos de clasificación de madera miembros del SEC.....	22
6. El proceso de secado de madera .....	25
<b>Procedimiento y Descripción de las Actividades .....</b>	27
Formato de Encuestas Aplicadas.....	28
Diagrama de Flujo .....	29
Diccionario de Datos.....	30
Pantallas Principales .....	31
<b>Resultados .....</b>	38
<b>Conclusión .....</b>	39
<b>Recomendación .....</b>	40
<b>Bibliografía.....</b>	41

**Anexo 1**

Manual de Usuario.....43

**Anexo 2**

Manual Técnico .....61

**Anexo 3**

Tablas de Diccionario de Datos.....146



## INTRODUCCIÓN

En este documento encontrará la realización de un sistema de información enfocado al manejo del control de inventarios para materia prima de una patio de madera en la empresa Molduras Ponderosa de Parral. Esto se dio al constatar la consecuente problemática que se presentaba en el mismo. En este documento se verá como se automatizó el control de la información mas indispensable. Dicha información era realizada a mano con un alto grado de dificultad para encontrarla y perdida de bastante tiempo muchas veces con el cual no se contaba.

El sistema permite llevar el control exacto sobre la mayoría de la información de cada carga de madera. También nos permite contar con información oportuna en el momento adecuado cada vez que el usuario lo requiera.

El sistema está pensado de forma que sea muy fácil de utilizar para el usuario final.

Las limitaciones que encontramos fueron hechas por el mismo gerente. Ya que minimizo la información mas crucial sin excederse en algún tipo de dato que no se usara frecuentemente. El gerente pedía un sistema eficaz y nada complicado de usarse, sin información que él no necesitara, así que se limitó solo a organizar los datos requeridos. Proponiendo varias alternativas y reportes los cuales en su mayoría fueron rechazados pues excedían los requerimientos del sistema ya que otros sistemas con los que contaba pudieran proporcionar algunos de esos mismos datos.



## **JUSTIFICACIÓN**

La elaboración de este sistema se llevó a cabo por la clara necesidad que para la empresa Molduras Ponderosa de Parral significa llevar un adecuado control de inventarios para materias primas. La problemática que aquí existía era el descontrol de inventarios, la necesidad de llevar un control exacto y en forma de tipo reporte para el fácil acceso de información clara y confiable.



## **OBJETIVOS GENERALES**

Desarrollar un sistema de control de inventarios que permita automatizar los procesos manuales ahora existentes como lo son el control de las materias primas entradas y salidas, control de los proveedores, y una mejor clasificación de la madera en todo su proceso de entrada, secado y cepillado.

## **OBJETIVOS ESPECÍFICOS**

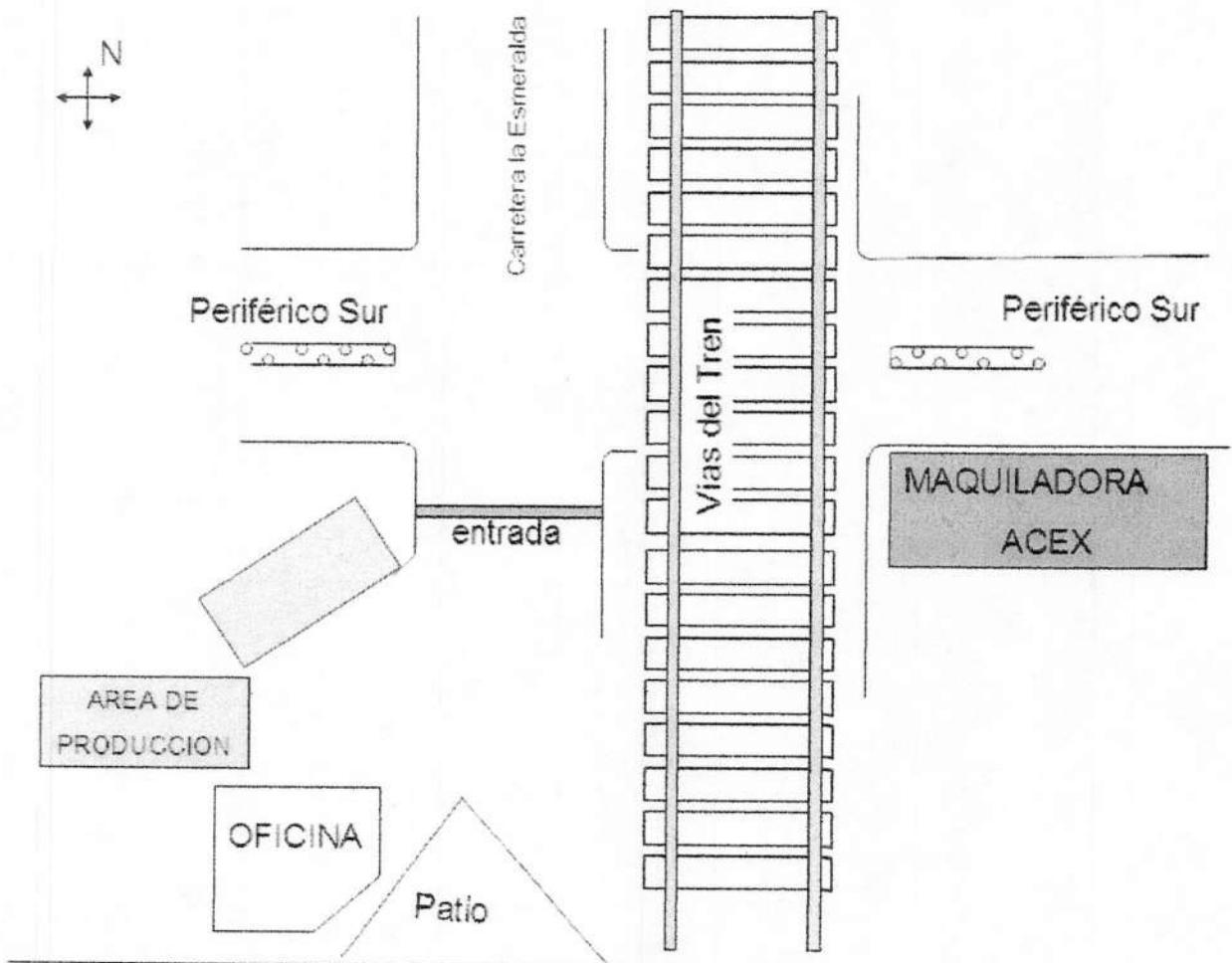
- Automatizar el inventario
- Controlar de manera mas exacta la materia prima
- Ayudar con la clasificación y reclasificación en el diferente proceso de la madera
- Reportar al gerente de la empresa el nombre de proveedores de la madera así como además datos personales
- Optimizar el proceso en cuestión de la información de la materia prima
- Llevar un control mas exacto en entradas y salidas de madera
- Crear un pagina Web de visita para dar a conocer a la empresa de forma mundial



## CARACTERIZACIÓN DEL ÁREA EN QUE SE PARTICIPO

### Descripción del área de trabajo

#### Vista Aérea





## Vista Interior:



## Descripción de la Empresa

### Giro de la empresa

MOLDURAS PONDEROSA DE PARRAL, S. DE R.L. es una empresa dedicada a la fabricación de moldura de diversos tipos de madera, principalmente de madera de pino ponderosa.

Forma parte de un grupo familiar de varias empresas dedicadas a varios giros del ramo forestal, que van desde la extracción y asierre de madera, fabricación de moldura de maderas selectas, elaboración de durmientes impregnados, elaboración de postes impregnados, entre otros.

### Ubicación de la Empresa

Molduras Ponderosa De Parral, se encuentra ubicada en la CD. de HIDALGO DEL PARRAL, en el estado de CHIHUAHUA con una ubicación estratégica, ya que se encuentra a tan solo 8 horas por carretera de distancia de la frontera con los ESTADOS UNIDOS DE NORTEAMÉRICA

Actualmente y desde hace muchos años, el sector forestal es la principal fuente de trabajo para la población, y que actualmente asciende aproximadamente a 120,000 habitantes



## **PROBLEMAS A RESOLVER PRIORIZÁNDOLOS**

1. Control de inventarios de la materia prima (madera) con la que se cuenta.
2. Control de clasificación y reclasificación de la materia prima
3. Control de entradas y salidas de materia prima mas exacto y automatizado
4. Optimizar el proceso en cuestión de la información de la materia prima
5. Reportar al gerente de la empresa el nombre de proveedores de la madera
6. Creación de una pagina Web en ingles y español para dar a conocer a esta empresa a nivel mundial



## ALCANCES Y LIMITACIONES

### Alcances

- Se logró automatizar el control de inventarios
- Se agilizaron los procesos de documentos, minimizando el tiempo de detección de información de la materia prima.
- Se redujo de manera considerable la carga de trabajo para la persona encargada de realizar dichas actividades
- Se crearon reportes de forma sistematizada para tener un mejor control de la información en su momento requerida

### Limitaciones

- Dentro de las limitaciones mas importantes que nos encontramos fue con ajustarnos en la mayoría de las pantallas a contener solo la información que el encargado creía sumamente necesaria, quería un sistema el cual no fuera muy complicado de entender y no tuviera información redundante la cual el pudiera manejar de forma sencilla
- El sistema es mono usuario



Aprobación de la solicitud.- No todos los proyectos son deseables o factibles sin embargo izo que si lo son deben incorporarse a los planes.

## 2.- Determinación de los requerimientos del sistema

El aspecto fundamental es comprender todas las facetas importantes de la parte de la empresa que se encuentra bajo estudio.

Al trabajar con los empleados y administradores se deben de estudiar los procesos de una empresa para dar respuesta a las siguientes preguntas:

1. Que es lo que se hace?
2. Como se hace?
3. Con que frecuencia se presenta?
4. Que tan grande es el volumen de transacciones o decisiones?
5. Cual es el grado de eficiencia con el que se detecto en las tareas?
6. Existe algún problema?
7. Si existe que tan serio es?
8. Cual es la causa que lo origina?

La investigación detallada requiere de estudio de manuales y reportes, la observación en condiciones reales con el fin de comprender el proceso en su totalidad.

## 3.-Diseño del sistema

Los analistas comienzan el proceso de diseño identificando reportes y salidas que debe producir el sistema. Es común que los diseñadores hagan un bosquejo de formato o pantallas que esperan que aparezcan cuando estén terminado el sistema.

## 4.-Desarrollo del software

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

## 5.- Prueba del sistema

Durante la fase de prueba el sistema se emplea de manera experimental para asegurarse que el software no tenga fallas en ocasiones se permite que los usuarios utilicen el sistema para observar si tratan de emplearlo en formas no previstas.



## 6.-Implantación y evaluación

Es el proceso de verificar e instalar nuevo equipo, entrenar a los usuarios, instalar la aplicación y concluir todos los archivos de datos necesarios para utilizarla.

Unas vez instaladas las aplicaciones se emplean durante muchos años; pero es indudable que debe darse mantenimiento a las aplicaciones; dado que los sistemas de las organizaciones junto con el ambiente de las empresas experimentan cambios.

(*Análisis y sistemas de información, segunda edición, James A: SENN año 1992*)

### Diagrama de flujo de datos

Los diagramas de flujo de datos muestran el panorama más amplio y posible de entradas, procesos y salidas de un sistema. El enfoque de flujo de datos enfatiza la lógica subyacente del sistema, mediante el uso de solamente cuatro símbolos.

La mayor ventaja se encuentra en la libertad conceptual que se obtiene con el uso de los cuatro símbolos básicos para diagramar el movimiento de datos.

Los cuatro símbolos para diagramar el movimiento de datos son:

Símbolo	Significado	Ejemplo
	Entidad	
	Flujo de datos	Cliente 
	Proceso	
	Almacén de datos	D3 base de datos

(*Análisis y diseño de sistemas, tercera edición, KENDALL & KENDALL año 1997*)



## 1. LENGUAJE DE PROGRAMACION DELPHI

### ¿Qué es Delphi?

Para hacernos una idea de lo que es, y de lo que se puede conseguir con Delphi, vamos a ver de forma breve las propiedades más importantes y destacadas de que consta.

- Delphi 1 es una potente herramienta de desarrollo de programas que permite la creación de aplicaciones para Windows 3.x, Windows95 y Windows NT. De hecho, aunque el programa ANALOGIA.EXE corre perfectamente en cualquier tipo de Windows, fue desarrollado sobre una plataforma Windows NT Workstation.
- Las aplicaciones pueden colocarse de forma muy sencilla en la pantalla según el principio de módulos. Para ello se dispone de una paleta dotada de una gran variedad de componentes, algo así como los bloques de construcción de cada programa. Esta paleta es denominada por Borland VCL (Visual Component Library), o biblioteca de componentes visuales. Tiene un aspecto similar a Visual Basic, pero aunque el aspecto externo indica la misma facilidad de uso que Visual Basic, el corazón del sistema Delphi es mucho más potente.
- Esta VCL es mucho más amplia que la de Turbo Pascal para Windows o la conocida OWL 1.0 (Object Windows Library) de Borland Pascal, y ofrece además una abstracción mucho más alta del sistema operativo. El programador es totalmente independiente de las particularidades de Windows, tales como manejadores (Handlers), punteros y funciones del API de Windows (Application Programming Interface). La programación se realiza con los cómodos componentes de Delphi y no con las complejas llamadas al sistema de Windows. Esto simplifica enormemente la hasta ahora poco clara programación bajo Windows.
- En realidad el hecho de que no necesitemos (normalmente) usar llamadas al API, no quiere decir que Delphi no lo permita. En casos muy concretos es necesario y no se puede hacer de otra manera. Un ejemplo concreto en ANALOGIA.EXE es la rutina que escribe las letras verticales de las etiquetas de los componentes del sistema mecánico (si el sistema es vertical). Esta rutina hace uso de varias llamadas al API de Windows puesto que Delphi no admite la posibilidad de escribir en vertical directamente. Pero como se ve, es un caso muy particular para una necesidad muy concreta.
- A diferencia de otras herramientas de desarrollo visuales (Visual Basic, Toolbook, etc) con Delphi es posible crear nuevos componentes que pueden entonces incorporarse en la paleta con los componentes ya existentes y que pueden ser utilizados de la misma forma. La VCL puede



estructurarse libremente y así adaptarse totalmente a las situaciones propias de programación.

- Delphi no solo dispone de componentes para la sencilla creación de entornos de aplicaciones como cuadros de lista, conmutadores o cuadros de diálogo terminados, sino que cubre con sus componentes muchos temas de la programación bajo Windows: se incluye entre los mismos un completo centro de control para la creación de aplicaciones multimedia, así como una gran variedad de componentes que actúan "debajo" del entorno, como tipos de listado muy variados y contenedores generales de datos. También hay herramientas de comunicación para DDE y OLE a través de las que se pueden crear vínculos de datos y comandos con otras aplicaciones. Uno de los aspectos más destacados lo constituyen los componentes que Borland ha incluido en Delphi para el desarrollo de completas aplicaciones de bases de datos. No se está limitado a un formato de datos determinado, sino que se tiene acceso a 50 formatos de datos diferentes a través de controladores suministrados por terceros (IDAPI y ODBC). Entre éstos se encuentran todos los estándares importantes de bases de datos en el área del PC como XBase, Paradox, Access, etc. Pero también es posible acceder de forma muy cómoda a servidores de bases de datos de otros sistemas (por ejemplo UNIX) por medio del SQL (Structured Query Language) que constituye un estándar de lenguaje de uso general para consultar y modificar datos administrados por servidores especiales de bases de datos como Oracle, Sybase, Informix o Adabas.
- Delphi dispone del Object Pascal, un lenguaje de programación muy poderoso que está sin dudas a la altura del C++ y que incluso lo supera en algunos aspectos. Este lenguaje surge a partir del desarrollo del Borland Pascal 7.0, un lenguaje que ocupa un lugar muy importante en la programación de ordenadores personales. El Object Pascal es totalmente compatible con el Borland Pascal 7.0, lo que permite que programas desarrollados con este último puedan ser convertidos a Delphi. Incluso la biblioteca de clases OWL 1.0 se incluye con el paquete de Delphi. Aspectos nuevos en el Object Pascal en relación a sus predecesores son el Exception-Handling (tratamiento y canalización de errores en run-time), un manejo más sencillo de los punteros con reconocimiento automático y referenciación, las llamadas propiedades de objetos que pueden ser asignadas como las variables, etc.
- Las aplicaciones terminadas quedan disponibles como archivos ejecutables (.EXE) que pueden utilizarse solos y sin bibliotecas adicionales, tal y como se ha hecho en ANALOGIA.EXE. Consecuentemente la velocidad con la que pueden ejecutarse los programas creados es muy alta. Excepcionalmente, si se incluyen llamadas a VBX, o DLLs, éstas se deben incluir junto con el ejecutable. También es necesario incluir el BDE (Borland Database Engine) en las aplicaciones de bases de datos, por lo cual no se incluye con ANALOGIA.EXE.



## 2. Conceptos básicos en la programación con Delphi

No es el objeto del presente proyecto hacer un repaso exhaustivo de la programación en Pascal, por eso vamos a ver de forma somera solo algunas de las características de que dispone Delphi, principalmente aquellas que le diferencian más del Pascal estándar. Para ello, vamos a introducir una serie de conceptos:

### Objetos

Como su propio nombre indica, el Object Pascal (usado por Delphi) es un lenguaje totalmente orientado a objetos. Prácticamente todos los elementos del entorno Delphi son objetos, que unas veces se encuentran definidos de antemano (los forms, los componentes de la paleta, objetos no visuales, etc), y otras se definen en nuestra propia aplicación (por ejemplo el componente TPostIt de ANALOGIA.EXE definido en la unit U\_Misc).

Todo el desarrollo de aplicaciones en Delphi está íntimamente ligado con la definición y uso de objetos, por lo que es fundamental conocer la mecánica que Object Pascal utiliza para describir un objeto, sus características y su funcionamiento, sobre todo a la hora de que el programador cree sus propios componentes. En muchos casos no es necesario conocer el funcionamiento de la programación orientada a objetos para programar con Delphi, puesto que en la mayoría de los casos existen controles ya creados sin necesidad de tener que programarlos. Por ejemplo en ANALOGIA.EXE la mayoría de los objetos forman parte de la VCL de Delphi y otros dos componentes han sido escritos por terceros (RChart y TFormulaParser), sin embargo ha sido útil desarrollar un componente propio (TPostIt) para mostrar la tabla de valores de los componentes en el programa.

La propia programación visual de Delphi, que escribe parte del código automáticamente, hace posible utilizar objetos de forma práctica sin tener por qué comprender al cien por cien su funcionamiento.

### Componentes

Un componente es cualquiera de los elementos que podemos insertar en una ficha, tanto si su función es visual como si no lo es (por supuesto un componente es también un objeto). Un ejemplo de componente puede ser RChart, usado en ANALOGIA.EXE para dibujar las gráficas. Sin conocer exactamente el cómo realiza su función, el programador manipula una serie de propiedades, métodos y eventos que caracterizan al componente, a través de los cuales se "maneja" el componente en la forma deseada. Por supuesto el usuario puede crear sus propios componentes y usarlos en distintas aplicaciones, de forma que la reusabilidad del código es máxima. A los



componentes que cuentan con una parte visual, como puede ser un botón, se les denomina controles

### Propiedades

Los componentes, y de hecho todos los objetos de Delphi son de uso general, por lo que a la hora de usarlos de alguna forma deberemos adecuarlos a nuestras necesidades. Para ello nos serviremos de las propiedades de cada objeto, mediante las cuales podremos establecer el título de una ventana, el tipo de letra de una etiqueta de texto o el color en el que aparecen los distintos controles.

Se puede pensar en las propiedades como si fuesen variables pertenecientes a un objeto (veremos que no es así exactamente), de tal forma que para acceder a ellas generalmente habrá que indicar no sólo el nombre de la propiedad, sino también a qué objeto pertenece.

La modificación o consulta del valor de una propiedad puede diferir según intentemos acceder a ella mientras estamos diseñando un form, en tiempo de diseño, o bien mediante el código del programa, en tiempo de ejecución. Ciertas propiedades están sólo accesibles en tiempo de ejecución, por lo que mientras estamos diseñando la ficha no aparecerán. Además hay algunas propiedades que son de sólo lectura, por lo que su valor puede ser consultado, pero no modificado, y otras que son sólo de escritura.

Aunque a primera vista si usamos un componente prefabricado, las propiedades pueden parecer simples variables, en la mayoría de las ocasiones una propiedad no es una variable, y la modificación o consulta de su valor puede conllevar que internamente el componente ejecute un cierto código. Si por ejemplo tuviésemos un componente para comunicaciones serie que se encargase de enviar mensajes a un ordenador remoto, podríamos tal vez asignar una cadena de caracteres a una hipotética propiedad "Envia". De esta forma, con una simple asignación (Envia:='Mensaje a mandar') se pondría en marcha todo el mecanismo implícito en el componente para enviar el mensaje al ordenador remoto. Esta propiedad podría ser un ejemplo de propiedad de sólo escritura, ya que sólo interesa mandar el mensaje al ordenador remoto, sin conservar información acerca del mensaje, por lo que sería inapropiada su lectura.

### Eventos

Como dije anteriormente, la programación en el entorno Windows se caracteriza por estar dirigida por eventos, de tal forma que un programa no tiene por qué ejecutarse necesariamente de forma secuencial, sino que ciertas porciones de código se ejecutarán cuando ocurra un cierto evento.



Los eventos son señales que el entorno recibe desde distintos elementos, como puedan ser el ratón, el teclado o un temporizador. Estos eventos son redirigidos a las aplicaciones, que en caso de aceptarlos deberán responder adecuadamente de ellos. Ciertos eventos pueden ser gestionados por el propio Windows, otros quedarán a cargo del propio lenguaje que estemos usando, y un tercer grupo serán los que lleguen hasta nuestro programa. En Delphi prácticamente todo el código que escribimos irá asociado a algún evento. Si retomamos el ejemplo del componente para comunicaciones serie, podría interesarnos que se ejecutara un evento cada vez que se recibiese un carácter por el puerto serie, de forma que podríamos escribir el código necesario para guardar el carácter en un archivo cada vez que se produjese el evento. Normalmente los eventos a los que reaccionarán los componentes serán las pulsaciones del teclado o el ratón, activaciones de los componentes, etc.

## Métodos

Los componentes Delphi además de disponer de propiedades y poder responder a ciertos eventos, habitualmente también disponen de métodos. Un método es un procedimiento o función que nos permite realizar una determinada acción en el componente, pudiendo necesitar o no el paso de algún parámetro.

Al igual que ocurre en las propiedades, a la hora de usar un cierto método normalmente tendremos que indicar primero el objeto o componente al que pertenece, de tal forma que la acción del método recaiga sobre él, y no sobre cualquier otro. El hecho de que cada objeto sea propietario de una serie de propiedades y métodos, variables y código, que no son accesibles a ningún otro objeto externo, recibe el nombre de encapsulación, aunque también es posible definirlos de tal modo que sean accesibles a otros objetos.



### 3. TODO EMPIEZA CON LA MADERA

#### Introducción a la madera

Existen más de 1,000 especies de árboles, de las cuales sólo unas 100 se utilizan para construir y fabricar productos de madera. De este grupo, vamos a seleccionar las más fáciles de trabajar y las que resultan más agradables a la vista.

Existen, básicamente, dos tipos de madera: madera dura y madera blanda. Además, hay ciertas características que son comunes a todos los tipos de madera. A continuación, le explicamos algunos de los términos más comunes y definiciones que debe conocer:

#### MADERA DURA

Esta madera se obtiene de los árboles que pierden las hojas en otoño (caducifolios). De toda esta gran variedad de árboles, sólo 200 existen en cantidad suficiente y son lo bastante flexibles para la carpintería. Las maderas duras, como nuestra piel, tienen poros microscópicos en la superficie. El tamaño de estos poros es lo que determina el dibujo de la veta y la textura. Debido a estas características, las maderas duras se clasifican según la apertura del poro en: maderas de poros cerrados (poros pequeños), entre las cuales las más usadas son el cerezo y el arce, y maderas de poros circulares (poros más grandes), entre las cuales las más usadas son el roble, el fresno y el álamo.

#### BLANDA

La madera blanda se obtiene de los árboles de hoja perenne (coníferas). En carpintería sólo se usa el 25 % de todas las maderas blandas. Todas las maderas blandas tienen poros cerrados (poros pequeños) que apenas se perciben en el producto acabado. Las maderas blandas más usadas son el cedro, el abeto, el pino y la picea.

#### DENSIDAD

Las maderas duras, como el roble, se reconocen por su peso y por el dibujo apretado de la veta. Son más resistentes al desgaste, a las mellas y a las ralladuras que las maderas blandas.

#### TEXTURA

Es la propiedad de la madera que determina el estado de la superficie y la estabilidad. Se trata de un factor decisivo a la hora de elegir un acabado para la madera.

#### DEFECTOS

Los defectos en la madera son algo natural y muy apreciado por muchos carpinteros, porque le confieren a la madera un aspecto único.

#### COLOR

El color aporta personalidad a la madera. El cedro rojo, por ejemplo, posee un aspecto y un carácter muy diferentes de los del pino blanco.

#### VETA

La veta es la característica más conocida de la madera. La veta es producto de la orientación de las fibras de las células de la madera. El proyecto que vaya a realizar será el que determine cuál es el tipo de veta más apropiado.



## CLASIFICACIÓN

La clasificación de la madera se basa en el número, la ubicación y el tamaño de los defectos del tablero, no en su resistencia. Cuanto más clara sea la madera, mayor es su clasificación.

## ESTABILIDAD

La estabilidad es la capacidad de la madera para contraerse o dilatarse antes o después de ser trabajada.

## DURABILIDAD

Las maderas duraderas resisten mejor el exceso de humedad y el contacto con la tierra, donde las posibilidades de que se pudran son mayores. Recuerde que ninguna madera se pudre si se mantiene seca.

## Explicación detallada de la veta de la madera

Dos tableros de la misma especie de árbol pueden presentar aspectos muy diferentes. En cada árbol el dibujo de la veta es distinto. Este dibujo se debe a la dirección en la que crecen las fibras de las células de la madera. Las variaciones en la dirección de la veta pueden influir de forma significativa en su proyecto.

La dirección de la veta es un factor importante que se debe tener en cuenta cuando se construyen proyectos estructurales o decorativos, como muebles o manualidades. Por ejemplo, cuando se trabaja en un objeto estructural, los tableros con veteado recto son, por regla general, los más adecuados por su resistencia. En proyectos más decorativos, una veta de características variadas realza la belleza y personalidad del proyecto.

Existen seis tipos generales de veta:

La densidad del dibujo de la veta determina la resistencia. Como cabría esperar, una pieza de madera con una veta apretada es más dura que otra con la veta más suelta. Hay que añadir que, en la construcción, la resistencia de un tablero aumenta cuando se colocan las demás piezas perpendiculares a la veta y no en paralelo.

## Clasificación de la madera

En el momento de escoger los tableros, ¿sabe realmente lo que significa el sello que llevan en la madera? ¡No se asuste! Nosotros hemos descifrado esos jeroglíficos para que los entienda. Así, la próxima vez que vaya al almacén de maderas, hablará como todo un profesional.



Veta de la madera.



## Clasificación de las maderas duras

La madera se clasifica en función del número de defectos que haya en una sección dada del largo y el ancho del tablero. Al igual que en las maderas blandas, una madera de clase inferior puede ser perfectamente aceptable dependiendo del lugar donde se vaya a colocar y el uso que se le vaya a dar. El organismo encargado de clasificar las maderas duras es la National Hardwood Lumber Association. Aquí tiene una tabla que le ayudará a entender el sistema de clasificación. Las distintas clases están ordenadas de la clase más alta a la más baja.

Clase	Abreviatura	Tamaño mínimo	% de material útil de la tabla por una cara
Firsts and Seconds (primeras y segundas)	FAS	6" x 8'	83
Select (selecta)	Sel	4" x 6'	83
#1 Common (común n.º 1)	#1 Com	3" x 4'	66
#2 Common (común n.º 2)	#2 Com	3" x 4'	50

## Clasificación de las maderas blandas

Las maderas blandas se dividen en dos categorías: madera dimensional, clasificada en función de la resistencia, y paneles aparentes, que se utilizan habitualmente en proyectos de carpintería. La clasificación de las maderas blandas es obra de varias agencias, así que encontrará algunas variaciones en la terminología. Las distintas clases están ordenadas de la clase más alta a la más baja.

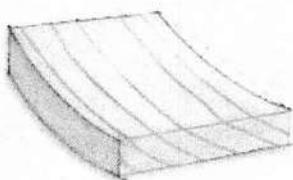
### Clase Significado

C Select	Casi libre de defectos. Muy utilizada en molduras y gabinetes para interiores.
D Select	Aspecto bonito, similar a C Select. Puede tener pequeños nudos.
1 Common (común n.º 1)	La mejor si busca pino de alta calidad con aspecto nudoso. Los nudos están apretados, lo que significa que no se van a desprender y que suelen ser pequeños.

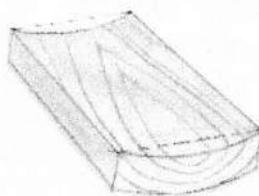


2	Common (común n.º 2)	Nudos apretados, pero más grandes que los de 1 Common. Se utiliza con frecuencia para hacer paneles y estantes. Muy apropiada para proyectos de carpintería generales.
---	----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------

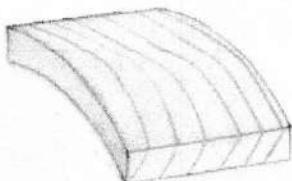
### Defectos comunes de la madera



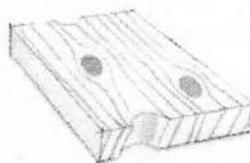
**ALABEADO:** comba de la cara del tablero en sentido longitudinal.



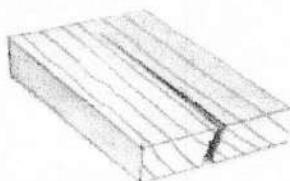
**ABARQUILLAMIENTO:** concavidad de la cara del tablero en sentido transversal.



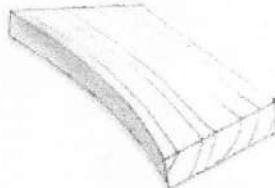
**ARQUEAMIENTO:** comba del canto, conocido también como corona.



**NUDO o AGUJERO DE NUDO:** un nudo apretado, por regla general, no es problemático. Un nudo suelto o muerto, rodeado de un anillo oscuro, puede desprenderse o puede haber dejado ya un agujero.



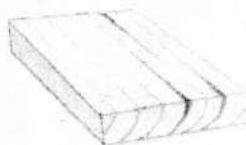
**HENDIDURA:** grieta que atraviesa toda la pieza de madera, generalmente en los extremos.



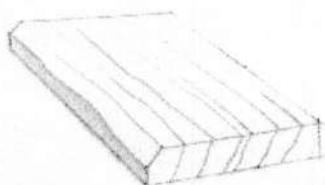
**RETORCIMIENTO:** el tablero está combado por muchos lugares.



**GRIETA EN CABECERA:** grieta paralela a los anillos de crecimiento anuales que no atraviesa toda la madera.



**RAJADURA:** separación de las fibras entre los anillos de crecimiento, que frecuentemente se extiende a lo largo de la cara del tablero y a veces por debajo de su superficie.



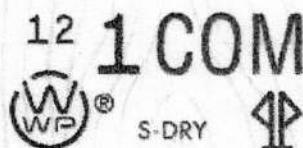
**CANTO REDONDEADO:** falta de madera o corteza no recortada a lo largo del canto o las esquinas de la pieza.

#### Sello de clasificación de la madera

Cuando la madera de la misma especie y tamaño aún está en el aserradero, se examina y se distribuye según su clase. Luego, se identifica con un sello y suele hacerse un inventario por clases y especies. Cuando seleccione la madera, asegúrese de leer la marca de clasificación, porque a veces los almacenes de madera utilizan nombres diferentes para la misma clase. (Y recuerde pedir ayuda si tiene problemas para desentrañar todo esto.) La clasificación se realiza en función de determinados defectos, como los nudos o los cantos redondeados. Tenga en mente el resultado final de su proyecto cuando seleccione una clase u otra de madera. La clase no indica el grado de uniformidad del color ni del dibujo de la veta.

**Fabricante:** número del aserradero, nombre o logotipo (por ejemplo, 12)

**Sello de certificación:** símbolo de la agencia que ha supervisado el control de calidad (por ejemplo, WWP®)



Sello de clasificación típico



**Clase:** Generalmente abreviada. 1 Common en el dibujo (por ejemplo, 1COM)

**Contenido de humedad (MC):** MC es la abreviatura de "moisture content", o contenido de humedad en el momento en que se desbastó el tablero. MC 15 significa un 15 por ciento o menos KD o S-DRY significa un 19 por ciento o menos, S-GRN significa madera verde con más del 19 por ciento de humedad (por ejemplo, S-DRY)

**Sello de la especie:** símbolo o abreviatura del tipo de árbol. El ejemplo que se muestra es de pino Ponderosa (por ejemplo, PP)

#### 4. CLASIFICACIÓN SEGÚN EL ASPECTO (VISUAL) DE LA MADERA

##### Los aserraderos Sélection

Vosges clasifican las piezas en función de la presencia, tamaño y repartición de las singularidades en las distintas caras.

También se tienen en cuenta las alteraciones naturales provocadas por los hongos y los insectos.

Las particularidades de las piezas cortadas y de la madera aserrada también son registradas.

##### NORMAS

Nuestra madera se clasifica según la norma europea de clasificación por aspecto visual EN 1611-1

##### CINCO CATEGORÍAS

Nuestra madera se clasifica en cinco categorías:

- Calidad 0

- Carpintería/Ebanistería

- Calidad 1 Estructura y cepillable
- Calidad 2 Construcción de armaduras
- Calidad 3 Embalaje – construcción de cajas
- Calidad 4 Encofrado

##### CONSULTAR

##### CALIDAD

0

CHOIX 0

CHOIX 3

##### CALIDAD

1

CHOIX 1

CHOIX 4

##### CALIDAD

2

##### CALIDAD

3

##### CALIDAD

4

##### INFORMACIÓN

Sélection Vosges proporciona a sus clientes un documento técnico detallado y completo sobre la clasificación de la madera aserrada resina.



MÉTODO DE MEDICIÓN	OTRAS	CLASIFICACIONES
Los métodos de medición utilizados son conformes con la norma europea EN 1310 para la medición de la singularidad y la norma EN 1311 para la medición de las alteraciones biológicas.	Para la madera destinada a armazones o estructuras, existe un método visual de clasificación según la norma NFB 52-001. En este caso, hay tres categorías o calidades: ST1-C30 / ST2-C24 / ST3-C18.	

## 5. ORGANISMOS DE CLASIFICACIÓN DE MADERA MIEMBROS DEL SEC

### *Organismos de clasificación de madera miembros del SEC*

En esta publicación se ofrece una muestra representativa de las diversas clases de madera de coníferas occidentales proporcionadas por los organismos miembros del Consejo de Exportación de Madera de Coníferas (Softwood Export Council - SEC):

- Oficina de Inspección de Madera del Pacífico (Pacific Lumber Inspection Bureau - PLIB),
- Servicio de Inspección de Madera de Secoya (Redwood Inspection Service - RIS), una división de la Asociación de Productores de Madera de Secoya de California (California Redwood Association - CRA),
- Oficina de Inspección de Madera de la Costa Oeste (West Coast Lumber Inspection Bureau - WCLIB) y
- Asociación de Fabricantes de Productos de Madera del Oeste (Western Wood Products Association - WWPA).

Todos estos organismos están acreditados por el Comité Norteamericano de Normas de la Madera (American Lumber Standard Committee, Inc.), subordinado al Departamento de Comercio de EE.UU.

RIS, WCLIB y WWPA son organismos acreditados de redacción de reglas y representan tres, de los únicos seis organismos de EE.UU. cuyas reglas de clasificación están certificadas que cumplen con la Norma PS-20 de Madera de Coníferas de EE.UU. Esta norma de productos de EE.UU. constituye un mecanismo para proporcionar los valores de tamaño, clasificación y diseño indicados en la etiqueta.

Todos los cuatro organismos están certificados para proporcionar servicios de clasificación e inspección de productos de uso estructural según la Regla de Clasificación Nacional de Madera para Construcción (NGR). PLIB, WCLIB y WWPA también están acreditados por la Junta Canadiense de Acreditación de Normas para la Madera (Canadian Lumber Standards Accreditation Board) para suministrar servicios de inspección en la clasificación según las



Reglas de Clasificación Estandarizada de Madera Canadiense publicadas por la Dirección Nacional de Clasificación de Madera de Canadá (National Lumber Grades Authority - NLGA).

Las compañías miembros de los organismos de clasificación miembros del SEC manufacturan y embarcan aproximadamente un 85% de su producción total de madera de coníferas de la región occidental de EE.UU.

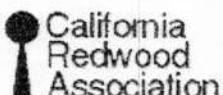
**La Oficina de Inspección de Madera del Pacífico (Pacific Lumber Inspection Bureau - PLIB)**, establecida desde 1903, es un organismo privado no lucrativo de inspección de control de calidad de madera. Quizás se le conoce mejor por sus reglas de clasificación, las *Reglas de Clasificación y Cepillado de la Lista "R" de Exportación de Madera de Coníferas de la Costa Oeste (Export "R" List Grading and Dressing Rules for West Coast Softwood Lumber)*, una de las más antiguas y más extensamente usadas fuentes de referencia en el comercio internacional de madera de coníferas.



La PLIB suministra servicios de clasificación e inspección según especificaciones contractuales entre clientes y proveedores, a más de 40 productores y remanufactureros de madera de coníferas de la región occidental de Oregón, Washington y en la parte sur del interior de Columbia Británica, Canadá.

La PLIB está aprobada, tanto por Comité Norteamericano de Normas de la Madera (American Lumber Standard Committee - ALSC) como por la Junta Canadiense de Acreditación de Normas para la Madera (Canadian Lumber Standards Accreditation Board), para suministrar servicios de supervisión de aserraderos de conformidad con las reglas NGR, RIS, WCLIB, WWPA y NLGA, así como de supervisión de madera pegada y clasificada por máquina.

Las reglas de clasificación de madera de secoya, las *Especificaciones Estándar de las Clases de Madera de Secoya de California*, han sido establecidas y publicadas por el **Servicio de Inspección de Madera de Secoya (Redwood Inspection Service - RIS)**. Este organismo de inspección está aprobado por el ALSC para proporcionar servicios de supervisión de aserraderos según la regla NGR, sus propias reglas y las reglas de clasificación de la WCLIB y de la WWPA, y para proporcionar servicios de supervisión de madera clasificada con máquina.



El servicio RIS es la división de servicios de clasificación e inspección de la **Asociación de Productores de Madera de Secoya de California (California Redwood Association -**



CRA), la cual, a los 80 años de haber sido establecida, es una de las asociaciones comerciales privadas no lucrativas más antiguas de la industria de productos forestales. La principal responsabilidad de la CRA es mantener la integridad y la alta calidad de las diversas clases de madera de secoya según las ha establecido el servicio RIS. Además, la CRA fomenta el uso de madera de secoya y proporciona información y servicios de apoyo a los fabricantes de productos dicha madera, técnicos de especificaciones arquitectónicas y constructores.

El nacimiento de la **Oficina de Inspección de Madera de la Costa Oeste (West Coast Lumber Inspection Bureau - WCLIB)** data de 1911, cuando cuatro asociaciones de la región noroeste de EE.UU. se unieron para formar la Asociación de Madereros de la Costa Oeste (West Coast Lumberman's Association). Cuarenta años después se estableció la división de servicios de clasificación como subsidiaria independiente de la WCLA, y en 1968 se registró oficialmente la WCLIB en calidad de organismo independiente. En la actualidad, la WCLIB proporciona servicios a más de 110 aserraderos y remanufactureros situados principalmente en la región noroeste de EE.UU. Además de su objetivo principal de establecer y actualizar normas uniformes de clasificación de la madera mediante la publicación de las *Reglas de Clasificación Estandarizada para Madera de la Costa Oeste (Standard Grading Rules for West Coast Lumber)*, la WCLIB proporciona servicios de supervisión de clasificación, servicios de reinspección e inspección transitoria, servicios de capacitación de clasificadores y servicios de asistencia general y técnica de productos. La WCLIB está aprobada, tanto por Comité Norteamericano de Normas de la Madera (American Lumber Standard Committee - ALSC) como por la Junta Canadiense de Acreditación de Normas para la Madera (Canadian Lumber Standards Accreditation Board), para suministrar servicios de supervisión de aserraderos de conformidad con la regla NGR, sus propias reglas, y las reglas de RIS, WWPA y NLGA, así como servicios de supervisión de madera pegada y clasificada por máquina.



Datando de 1906 uno de sus organismos predecesores, la **Asociación de Fabricantes de Productos de Madera del Oeste (Western Wood Products Association - WWPA)** se encuentra entre las asociaciones comerciales privadas no lucrativas más antiguas y más grandes de la industria de productos forestales del mundo, y representa a más de 130 aserraderos en los 12 estados contiguos de la región occidental de EE.UU. y Alaska. Además de su más importante función de control de calidad como organismo





certificado de redacción de reglas, clasificación e inspección, la WWPA proporciona análisis económicos y asistencia técnica a los productores miembros de la misma, servicios de asistencia técnica de productos e información a ingenieros, arquitectos y constructores, así como servicios de información a la industria.

La WWPA está aprobada, tanto por Comité Norteamericano de Normas de la Madera (American Lumber Standard Committee - ALSC) como por la Junta Canadiense de Acreditación de Normas para la Madera (Canadian Lumber Standards Accreditation Board), para suministrar servicios de supervisión de aserraderos de conformidad con la regla NGR, sus propias reglas, y las reglas de RIS, WCLIB y NLGA, así como servicios de supervisión de madera pegada y clasificada por máquina.

## 6. EL PROCESO DE SECADO DE MADERA

- La madera es un material higroscópico, gana ó pierde humedad, de acuerdo al medio que la rodea.
- El secado de la madera en hornos depende de 3 factores principales: La velocidad del aire, la humedad Relativa y la temperatura del aire que rodea al horno de secado de madera.
- Controlando éstos tres factores apropiadamente secamos madera.

### EL PROCESO DE SECADO CONTROLA

- La velocidad del aire
- La Humedad relativa del aire
- La temperatura

#### La Velocidad del Aire.

- El aire es el medio encargado de transportar la humedad que despiden la superficie de la madera. A mayor velocidad aumenta la capacidad de arrastre, pero exceso de aire provocaría un resecamiento de la superficie de la madera generando esfuerzos que pueden dañar la madera o interrumpir en algunos casos el proceso de secado de madera. En la actualidad se utilizan controladores que varía la velocidad del aire de acuerdo a la etapa de secado de la madera.

#### La Humedad Relativa del Aire.

- Es la relación entre la cantidad de vapor de agua contenida en el aire y la máxima cantidad que puede contener a la misma temperatura. Si aumentamos la temperatura, aumenta la capacidad del aire de contener vapor de agua, lo que quiere decir que puede absorber mayor cantidad



de vapor de agua de la madera. Igualmente si extraemos vapor de agua del aire, bajamos su humedad relativa y aumenta la capacidad de sacar vapor de la madera. A una humedad relativa dada, hay un valor de equilibrio de la humedad de la madera. De allí ya no pierde más humedad. Por eso es que tenemos que ir variando las condiciones del ambiente a medida que la madera se va secando.

#### La Temperatura

- Como dijimos la temperatura varía el valor de la humedad relativa del aire y por lo tanto la capacidad del mismo de extraer humedad de la madera.

#### Control del Proceso

- Como ya hemos comentado para controlar el proceso en los hornos de secado de madera debemos controlar La circulación del aire, la humedad relativa del Aire y la temperatura dentro de las cámaras u hornos secaderos de madera.
- Para un adecuado proceso de secado de madera deben usarse los valores de la temperatura y humedad relativa que experimentalmente ya se han determinado para cada especie de madera.
- En la mayoría de los secaderos de madera, la humedad relativa se mide a través del "Bulbo Húmedo" que es una medida termodinámica de la humedad contenida en el aire. Este parámetro se mide con un aparato llamado Psicómetro que es sencillo y por eso su uso.
- Todas las tablas de secado están basadas en los valores de "Bulbo Seco" ó temperatura y "Bulbo Húmedo" ó humedad.

#### Gradiente de Secado de Maderas

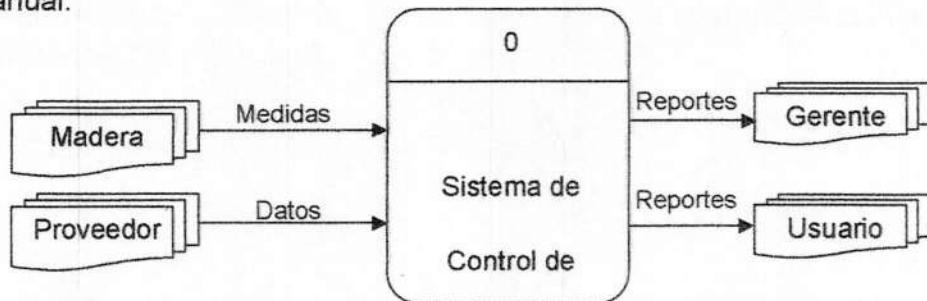
- El gradiente de secado de madera es la relación entre la humedad contenida en la madera y la humedad de equilibrio en ese ambiente. Este es el criterio más importante para un adecuado proceso de secado de madera en hornos.
- Un gradiente muy bajo puede prolongar el tiempo de secado de la madera excesivamente, mientras que un gradiente muy alto aceleraría mucho el proceso pero seguramente traería consecuencias graves en la calidad de la madera ó la interrupción del secado de la madera.

Los valores normales están entre 1,5 y 2,5 dependiendo de la especie. En piezas muy gruesas no se pueden usar gradientes altos ya que la diferencia entre el centro de la pieza, que está húmedo, y la superficie, que tiene una humedad de equilibrio de acuerdo al ambiente, es muy alta y provoca esfuerzos que dañan la madera ó interrumpen



## PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES

Se llevó a cabo un análisis con la finalidad de ver los requerimientos que se tenían en la empresa Molduras Ponderosa de Parral para dar paso a la automatización de los procesos que anteriormente se realizaban de forma manual.



Para ello se optó por la entrevista la cual permitió la relación de manera personal con el gerente y a no limitarse a determinadas preguntas, esto brinda un vínculo de confianza, para dar paso a un mejor trabajo. Se recopilo toda la información necesaria para la construcción del sistema; dicha entrevista ayudó a entender como se hacían los procesos de control de inventarios, quien los hacia, cada cuando los hacia, se presentaron unos cuestionarios a los trabajadores y al mismo gerente para que dieran su opinión de cómo seria la forma de realizar el control de los inventarios de una mejor manera, dicha información fue recopilada.



## FORMATO DE ENCUESTAS APLICADAS

Objetivo de la encuesta:

Conocer el punto de vista del gerente acerca de la implantación de un nuevo sistema el cual les ayudaría a agilizar los determinados procesos.

¿Cuál es el giro de la empresa?

¿Todos tienen acceso a la información?

¿Con cuantos proveedores cuenta actualmente?

¿Cuáles son las actividades que realizan entorno a los inventarios?

¿Cuáles son los datos que mas batalla de buscar entre toda la papelería y cuales son los que mas usa?

¿Cada cuando se realizan este tipo de búsquedas?

¿Ha utilizado alguna vez una computadora?

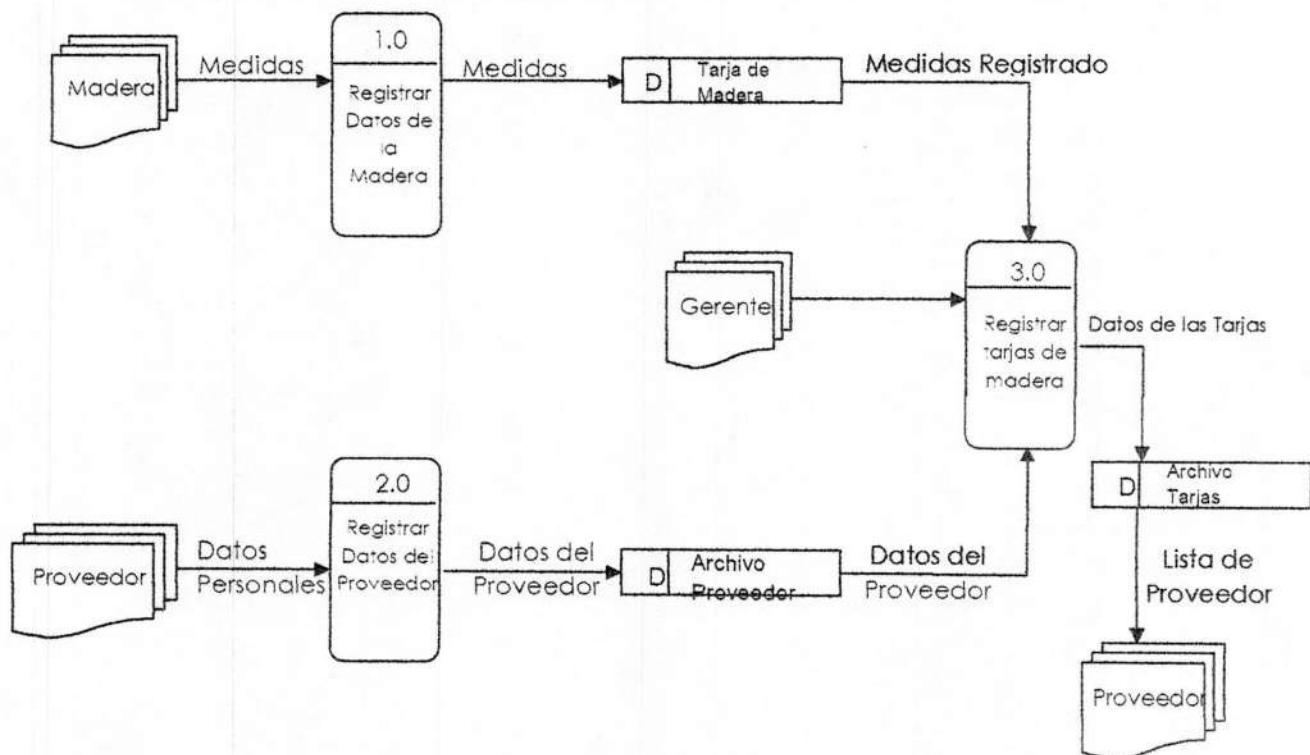
¿Sabe usted lo que es un sistema?

¿Considera los sistemas como una ayuda?



## DIAGRAMA DE FLUJO

Posteriormente con dicha información se hizo un diseño de cómo quedaría el sistema; se diseñaron las entradas, salidas, así como la interfaz y todo lo relacionado con el manejo del sistema.



A continuación se empezó a desarrollar el sistema mediante el lenguaje de programación, manejando solo la información indispensable y conservando la integridad de la misma. En la etapa de la prueba se aplicó el sistema para ver que en realidad el sistema funcionaba de acuerdo a las especificaciones requeridas y de manera adecuada. Esto se hizo introduciendo datos al sistema y comprobando que el sistema era valido y ver todas las posibles fallas que pueda tener y si así fuera darles solución.

Después de hacer pruebas al sistema se procede a la etapa de implantación esto es el proceso de instalar la aplicación en el equipo que se trabajara, construir todos los archivos necesarios para utilizarlo, entrenar al usuario. Y finalmente se procede a evaluarlo observando el tiempo de ejecución, el tiempo de respuesta y como lo manejaba el usuario final, saber si le era entendible, si tiene un ambiente agradable para usarlo etc. Por lo tanto se considera que todo el proceso se hizo de una manera eficiente.



## DICCIONARIO DE DATOS

Las tablas principales para el desarrollo de la aplicación denominada System Molduras son las siguientes:

- Cliente.
- Devolución.
- Temporal Devolución.
- Proveedor.
- Maderas
- Pino
- Folio.
- Radiata

**Ver Tablas con sus características en Anexo 3.**



## Pantalla Principal

Molduras Ponderosa De Parral  
Compras Clientes Proveedores Madera Inventario Salir



Esta pantalla es de recepción en la cual podemos dar de alta la materia prima con sus características así como nombre del proveedor, consignatario y quien recibió.

Recepción					
Madera	Nacional	Precio	19		
		Ancho	16		
		Largo	12		
Grueso				Total	
1	/	2			
		Cantidad	120		
Clave de Proveedor			2120.120.1	Proveedor	Ivan Garfio
Consignatario				Direccion	Hiereniz 12
Chofer				Camion	
Fecha			1/16/2008	Guia	
Rem. Ftel.					
Recibio					
<input type="button" value="Cancelar"/>			<input type="button" value="Imprimir"/>		



Pantalla devoluciones donde podemos dar de baja de inventarios algún registro

Detalle de Devolución									
Folio	MADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	FECHA	TOTAL	REMITENTE
H1	Pino Nacional	2 Y M	1.25	4	10	190	12/11/2007	792	Ivan Garfio
J1	Pino Nacional	2 Y M	0.625	6	10	120	12/11/2007	375	Ivan Garfio

En la pantalla podemos ver la entrada de madera ya sea por reporte o por proveedor del día en curso

Reporte de Entrada									
Entrada por Proveedor									
	MADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE	
	Pino Nacional		1.25	6	16	450	4500	Lizbeth Meraz	
	Pino Nacional	2 Y M	0.625	8	16	500	7467	Yop	
	Radiata	Shops	0.5	8	10	100	332	Ivan Garfio	



En esta pantalla para dar altas a los clientes

Registro Clientes

Clave	000
Cliente	
Direccion	
Ciudad	

Esta es la pantalla para dar de baja o eliminar a los Clientes

Bajas Clientes

Cliente	
Clave	0000
Direccion	
Ciudad	

Con esta pantalla podemos actualizar cambios en la información de nuestros clientes

Cambios Clientes

Cliente	
Clave	0000
Direccion	
Ciudad	



En esta pantalla se consultan e imprimen los clientes de la empresa

Consultas Clientes

**Cliente**

CLAVE	CLIENTE	DIRECCION	CIUDAD	RFC
► 1	ANA	AQUI	Parral	CACL850506
1	Karina	Independencia	Parral	SAGK851219
1	Tu	Lejos	53	JSHG904520
000	Yo	Alla	Chi	SAMG820406

Cerrar

Imprimir

En esta pantalla para dar altas a los proveedores

Registro de Proveedores

Cuenta

Proveedor

Direccion

Ciudad

Telefono

Cancelar



Esta es la pantalla para dar de baja o eliminar a los Proveedores

Baja Proveedores

Cuenta	<input type="text"/>
Proveedor	<input type="text"/>
Direccion	<input type="text"/>
Cd	<input type="text"/>
Telefono	<input type="text"/>

Cancelar       Eliminar

Con esta pantalla podemos actualizar cambios en la información de proveedores

Cambios Proveedores

Cuenta	<input type="text"/>
Proveedor	<input type="text"/>
Direccion	<input type="text"/>
Cludad	<input type="text"/>
Telefono	<input type="text"/>

Cancelar       Modificar

En esta pantalla se consultan e imprimen los proveedores con los que trabaja la empresa

Consulta Proveedores

Proveedor					
CUENTA	PROVEEDOR	DIRECCION	CIUDAD	TEL	
2120.123.5	Gabina Isela	Juarez	Parral	2901	
2120.128.1	Ivan Garfio	Hiervanz 12	Parral	3371	
2514.236.6	Lizbeth Meraz	Hiervanz 14	Parral	360:	
123	Tava	Rio	Parral	523	

Cancelar       Imprimir



## Hoja de reporte de entrada de madera o recepción de tarjas



## Molduras Ponderosas de Parral

Tarja de Recepcion

Folio

Remitente \_\_\_\_\_

Consignatario \_\_\_\_\_ Dirección \_\_\_\_\_

Chofer \_\_\_\_\_ Camion \_\_\_\_\_

Fecha \_\_\_\_\_ Guia \_\_\_\_\_

Rem. Ftal. \_\_\_\_\_

Madera \_\_\_\_\_ Tipo \_\_\_\_\_ Grueso \_\_\_\_\_

Recibio \_\_\_\_\_

Ancho	Largo	Cantidad	Precio	Total
-------	-------	----------	--------	-------

**Total**

En otra expresión

## Reporte de Devolución de Madera



## Molduras Ponderosas de Parral

Tarja de Devolucion

Folio

Remitente \_\_\_\_\_

Consignatario \_\_\_\_\_ Dirección \_\_\_\_\_

Chofer \_\_\_\_\_ Camion \_\_\_\_\_

Fecha \_\_\_\_\_ Guia \_\_\_\_\_

Rem. Ftal. \_\_\_\_\_

Madera \_\_\_\_\_ Tipo \_\_\_\_\_ Grueso \_\_\_\_\_

Recibio \_\_\_\_\_

Ancho	Largo	Cantidad	Precio	Total
-------	-------	----------	--------	-------

**Total**



## Reporte por Tarja de Proveedores

### Resumen de tarjas

#### Proveedor

Periodo Del 28/2/2008 Al 28/2/2008

Madera	Tipo	Grueso	Ancho	Largo	Cantidad	Piezaje	Precio	Fecha	Importe
Rodaja	Sheet	0.5	8	10	100	333	15	11/17/2008	220
Pino Nacional		1.25	8	15	450	4500	12	12/5/2008	540
Pino Nacional 2 YM		0.875	8	15	600	7467	10	12/6/2008	60

Piezaje total 7467

## Reporte por Tarja por tipo

### Resumen de tarjas por tipo

Madera	Grueso	Ancho	Largo	Cantidad	Total	Precio	Proveedor
Pino Nacional 10	1	15	450	4500	4500	12	Lisseth Nerey
Pino Nacional 15	1	15	100	4000	4000	10	Yop
Pino Nacional 0.875	8	15	600	7467	7467	10	Yop
Pino Nacional 25	1	15	60	560	560	10	Lisseth Nerey
Rodaja	0.5	8	10	100	200	15	Ivan Gardo

## Reporte por Tarja por fecha

### Resumen de tarjas por Fechas:

Del 11/5/2007 Al 11/5/2007

Madera	Tipo	Grueso	Ancho	Largo	Cantidad	Total	Precio	Proveedor
Rodaja	Sheet	0.5	8	10	100	200	15	Ivan Gardo
Pino Nacional		1.25	8	15	450	4500	12	Lisseth Nerey
Pino Nacional	2 YM	0.875	8	15	600	7467	10	Yop



## **RESULTADOS**

Una vez instalado el sistema todo indica que se han logrado satisfacer las necesidades especificadas al inicio del proyecto en la empresa Molduras Ponderosa de Parral en el área administrativa permitiendo agilizar los procesos antes realizados de manera manual.

Dentro de los beneficios hoy existen se encuentran:

- Una mayor rapidez a la hora de contar con información oportuna y confiable.
- Un control exacto y fiable.
- Una entrega de la información efectiva
- La seguridad de que el sistema nos ayuda a tomar decisiones rápidas y reales en la empresa Molduras Ponderosas de Parral.

Todo lo anterior permite corroborar un adecuado control de inventarios pero sobre todo el ahorro de tiempo y rapidez en la forma de contar con la información. Así mismo se brindo rapidez no solo al personal que ahí labora sino también a la hora de solicitar algún reporte.



## **CONCLUSION**

Una vez concluido el proyecto, basándonos en un espíritu creador y de ayuda solo queda decir que se culminó con un gran éxito cumpliéndose todos los objetivos planteados desde un principio, esto demuestra que se han alcanzado dichos objetivos establecidos en el inicio de la investigación.

Cabe mencionar que el trato dado por parte del personal de la empresa Molduras Ponderosa de Parral fue excepcional, dentro de todo este tiempo que se estuvo laborando en los diferentes departamentos que brindaron un apoyo incondicional el cual se les agradece ya que hicieron más fácil la estadía y la realización de dicho proyecto.



## **RECOMENDACION**

Se recomiendan una serie de indicaciones para el óptimo funcionamiento del sistema y a su vez mantenerlo y conservar más su vida útil.

- De principio se recomienda usarlo correctamente; Esto para darle cumplimiento a la seguridad de los datos y no vaya a ocurrir complicaciones de filtración de información.
- Darle un adecuado mantenimiento al equipo al igual que al sistema, que se elimine información que vaya dejando de ser útil para evitar en cierto momento la perdida de datos o el entorpecimiento del sistema.
- Y se recomienda que el usuario antes de empezar a usar el sistema se le facilite el manual de usuario para que se familiarice más rápido con el mismo.



## BIBLIOGRÁFIA

- KENDALL & KENDAL  
Análisis y diseño de sistemas, tercera edición, año 1997.
- Lic. Diana Estela Parra Flores  
Instructivo para la elaboración de reportes de residencias  
Profesionales
- Diana Alicia Campos Gallegos  
Trabajo profesional para obtener el título de lic. en informática
- Nancy Hasell Lara Chaparro  
Trabajo profesional para obtener el título de lic. en informática

# ANEXO

1

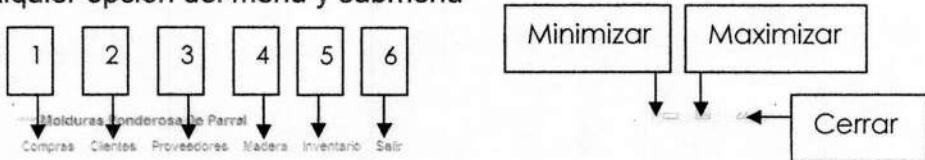


## MANUAL DE USUARIO

*Splash* esta pantalla es de entrada y el usuario debe esperar a que cargue en un 100% para entrar al sistema.



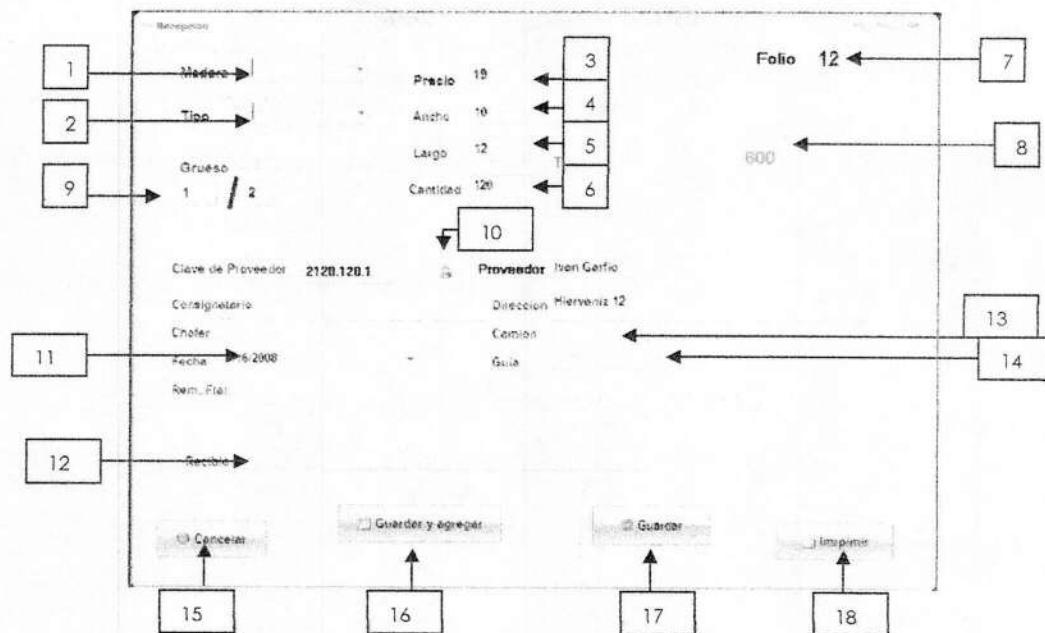
*Pantalla principal* donde se ubica el usuario cuando entra el usuario y puede utilizar cualquier opción del menú y submenú



- Este botón de compras nos despliega un submenú donde podemos tomar una de las siguientes opciones que veremos a continuación.
- El botón clientes nos manda a las opciones de dar altas bajas y consultas de clientes
- El botón proveedores nos manda a las opciones de dar altas bajas y consultas de proveedores
- Pantalla de madera donde se puede dar consultas cambios y bajas
- Menú inventario para dar consultas y salidas de madera
- Botón salir cierra el programa



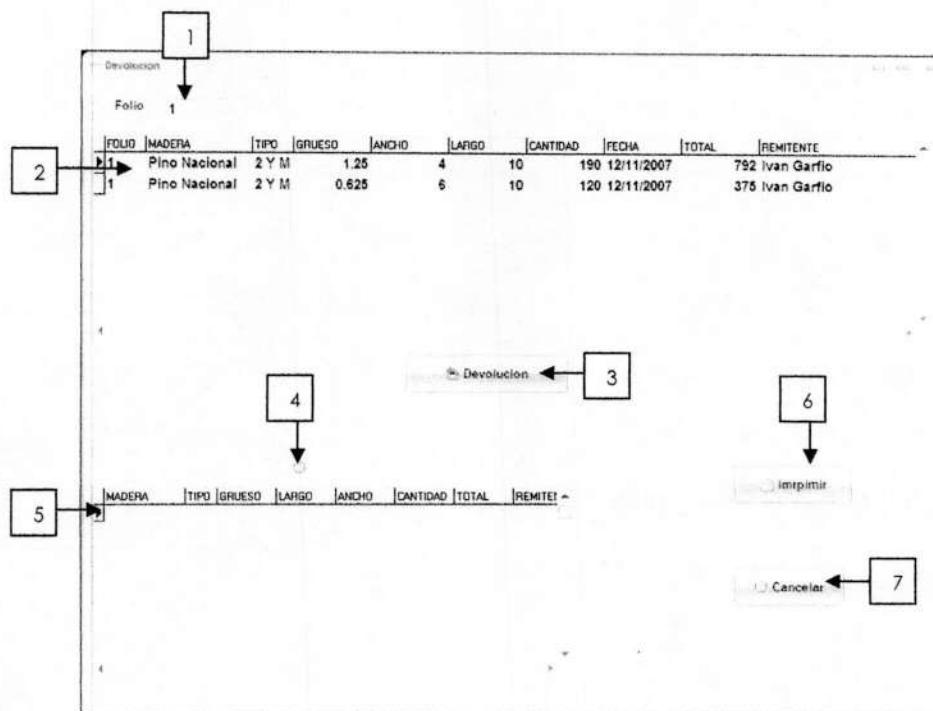
Pantalla de Recepción. Para dar de alta en inventarios en el menú recepción de la pantalla principal



1. Escribir nombre de madera Nacional o extranjera (Radiata)
2. Escribir tipo de madera en una de sus clasificaciones
3. Dar el precio por unidad
4. Ancho de madera en pulgadas
5. Largo de la madera en pies
6. Que cantidad de madera a ingresado
7. Numero de folio de tarja
8. Cubicación de madera que da el sistema automáticamente
9. Dar el grueso de la madera en fracción
10. Hacer click en icono para dar a conocer todos los proveedores
11. Desplegar la barra para asignar el día correcto o fecha
12. Nombre de quien recibió el embarque
13. En esta espacio se escribe el nombre del camión o razón social
14. Espacio para numero de guía
15. Botón cancelar nos retira y se limpia esta pantalla
16. Este botón guarda y agrega a una tarja o registro la información escrita
17. Simple y sencillamente guarda la información para ese folio
18. Con el se imprime la información almacenada en dicho folio.



Pantalla Devoluciones en la cual podrá dar de baja la madera por medio de las tarjas en caso de que la madera este machado o con desperfectos. Se puede acceder a esta pantalla mediante el menú de recepción de la pantalla principal y submenú devoluciones



- Este espacio esta reservado para el numero de folio
- En esta tabla se selecciona el renglón o tarja que desea ser borrado
- Con este botón se limpia la tabla la tarja que se ha seleccionado
- Con este icono podemos contar para borrar la tarja en devolución
- En esta tabla se puede mostrar la tarja devuelta o dada de baja
- Con este botón se imprime las tarjas que se quedan guardadas
- En este botón se cancela la operación, cierra esta pantalla y devuelve a la pantalla principal.



Pantalla de Resumen de Tarjas por fechas y Madera podemos entrar a esta pantalla del submenú tarjas por filtro y nos sirve para mostrar todas las compras o entradas hechas en el lapso de tiempo que creamos conveniente

MADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE
Pino Nacional		1.25	6	16	450	4500	Lizbeth Meraz
Pino Nacional	2 Y M	0.875	8	16	800	7467	Yop
Radiata	Shop3	0.5	8	10	100	333	Ivan Garfio

1. Este cuadro de texto da la opción de escoger las dos maderas que se manejan (Pino nacional, Radiata)
2. Barra donde podemos asignar la fecha de inicio de búsqueda
3. Barra donde podemos asignar la fecha de fin de búsqueda
4. Botón que da inicio a la búsqueda o al filtro de información
5. Nos manda a una ventana auxiliar donde podemos apreciar el formato antes de imprimir
6. Tabla que nos muestra el resultado de la búsqueda que hemos pedido
7. Botón de salir cierra esta pantalla y devuelve a la pantalla principal.



Pantalla de Resumen de Tarjas por fechas y Tipo podemos entrar a esta pantalla del submenú tarjas por filtro y nos sirve para mostrar todas las compras o entradas hechas en el lapso de tiempo que creamos conveniente

TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE
6 Pino Nacional	1.25	6	16	450	4500	Lizbeth Meraz
Pino Nacional	1.25	8	16	300	4000	Yop
Pino Nacional	0.875	8	16	800	7467	Yop
Pino Nacional No 3	1.25	6	16	69	890	Lizbeth Meraz
Radiata	0.5	8	10	100	333	Ivan Garfio
Shop3						

1. Este cuadro de texto da la opción de escoger el tipo de madera que se manejan ya sea nacional o internacionalmente.
2. Barra donde podemos asignar la fecha de inicio de búsqueda
3. Barra donde podemos asignar la fecha de fin de búsqueda
4. Botón que da inicio a la búsqueda o al filtro de información
5. Nos manda a una ventana auxiliar donde podemos apreciar el formato antes de imprimir
6. Tabla que nos muestra el resultado de la búsqueda que hemos pedido
7. Botón de salir cierra esta pantalla y devuelve a la pantalla principal.



Pantalla de Resumen de Tarjas por fechas y Grueso podemos entrar a esta pantalla del submenú tarjas por filtro y nos sirve para mostrar todas las compras o entradas hechas en el lapso de tiempo que creamos conveniente

1	2	3	4	5																																								
Resumen de tarjetas Basa por Reporte																																												
Grueso	1 / 2	De:	1/21/2008	-																																								
A:	1/25/2008	-	Buscar	Imprimir																																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding: 2px;">ADERA</th> <th style="text-align: left; padding: 2px;">TIPO</th> <th style="text-align: left; padding: 2px;">GRUESO</th> <th style="text-align: left; padding: 2px;">ANCHO</th> <th style="text-align: left; padding: 2px;">LARGO</th> <th style="text-align: left; padding: 2px;">CANTIDAD</th> <th style="text-align: left; padding: 2px;">TOTAL</th> <th style="text-align: left; padding: 2px;">REMITENTE</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">6</td> <td style="padding: 2px;">adhera</td> <td style="padding: 2px;">Shop3</td> <td style="padding: 2px;">0.5</td> <td style="padding: 2px;">8</td> <td style="padding: 2px;">10</td> <td style="padding: 2px;">100</td> <td style="padding: 2px;">333 Ivan Garfio</td> </tr> <tr> <td></td> <td style="padding: 2px;">adhera</td> <td style="padding: 2px;">Shop3</td> <td style="padding: 2px;">0.5</td> <td style="padding: 2px;">8</td> <td style="padding: 2px;">10</td> <td style="padding: 2px;">100</td> <td style="padding: 2px;">333 Ivan Garfio</td> </tr> <tr> <td></td> <td style="padding: 2px;">Pino Nacional</td> <td style="padding: 2px;">2 Y M</td> <td style="padding: 2px;">0.875</td> <td style="padding: 2px;">8</td> <td style="padding: 2px;">16</td> <td style="padding: 2px;">800</td> <td style="padding: 2px;">7467 Yop</td> </tr> <tr> <td></td> <td style="padding: 2px;">Pino Nacional</td> <td style="padding: 2px;">1.25</td> <td style="padding: 2px;">6</td> <td style="padding: 2px;">16</td> <td style="padding: 2px;">450</td> <td style="padding: 2px;">4500 Lizbeth Meraz</td> <td style="padding: 2px;"></td> </tr> </tbody> </table>					ADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE	6	adhera	Shop3	0.5	8	10	100	333 Ivan Garfio		adhera	Shop3	0.5	8	10	100	333 Ivan Garfio		Pino Nacional	2 Y M	0.875	8	16	800	7467 Yop		Pino Nacional	1.25	6	16	450	4500 Lizbeth Meraz	
ADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE																																					
6	adhera	Shop3	0.5	8	10	100	333 Ivan Garfio																																					
	adhera	Shop3	0.5	8	10	100	333 Ivan Garfio																																					
	Pino Nacional	2 Y M	0.875	8	16	800	7467 Yop																																					
	Pino Nacional	1.25	6	16	450	4500 Lizbeth Meraz																																						
<input style="width: 100px; height: 30px; border: 1px solid black; background-color: #f0f0f0; border-radius: 5px; font-size: 10px; margin: 0 auto;" type="button" value="Sair"/>																																												

1. Este cuadro de texto da la opción de escoger el grueso de madera
2. Barra donde podemos asignar la fecha de inicio de búsqueda
3. Barra donde podemos asignar la fecha de fin de búsqueda
4. Botón que da inicio a la búsqueda o al filtro de información
5. Nos manda a una ventana auxiliar donde podemos apreciar el formato antes de imprimir
6. Tabla que nos muestra el resultado de la búsqueda que hemos pedido
7. Botón de salir cierra esta pantalla y devuelve a la pantalla principal.



Pantalla de Resumen de Tarjas por fechas y Ancho podemos entrar a esta pantalla del submenú tarjas por filtro y nos sirve para mostrar todas las compras o entradas hechas en el lapso de tiempo que creamos conveniente

MADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE
6 Pino Nacional Estata	Shop3	0.5	1.25	6	16	450	Lizbeth Meraz
Pino Nacional 2 Y M		0.875		8	10	333	Ivan Garfio
					800	7467	Yop

Resumen de tarjas  
Filtros por: Asunto

Ancho: 8 | De: 1/21/2008 | A: 1/25/2008 | Buscar | Imprimir

Sair | 7

1. Este cuadro de texto da la opción de escoger el ancho de madera
2. Barra donde podemos asignar la fecha de inicio de búsqueda
3. Barra donde podemos asignar la fecha de fin de búsqueda
4. Botón que da inicio a la búsqueda o al filtro de información
5. Nos manda a una ventana auxiliar donde podemos apreciar el formato antes de imprimir
6. Tabla que nos muestra el resultado de la búsqueda que hemos pedido
7. Botón de salir cierra esta pantalla y devuelve a la pantalla principal.



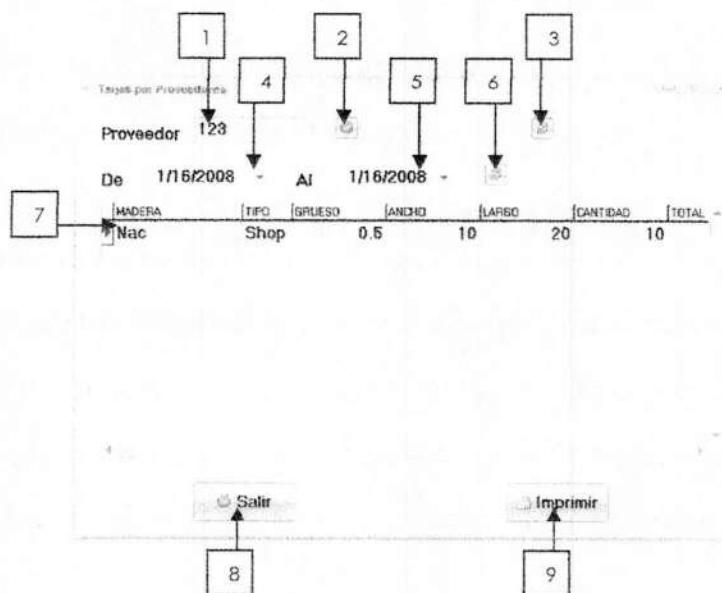
Pantalla de Resumen de Tarjas por fechas y Largo podemos entrar a esta pantalla del submenú tarjas por filtro y nos sirve para mostrar todas las compras o entradas hechas en el lapso de tiempo que creamos conveniente

1	2	3	4	5																								
Resumen de tarjas Filtros por Reporte	De:	A:	Buscar	Imprimir																								
Largo 16	1/21/2008	1/25/2008																										
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>MADERA</th> <th>TIPO</th> <th>GRUESO</th> <th>ANCHO</th> <th>LARGO</th> <th>CANTIDAD</th> <th>TOTAL</th> <th>REMITENTE</th> </tr> </thead> <tbody> <tr> <td>6 Pino Nacional</td> <td>Shop3</td> <td>0.5 1.25</td> <td>8 6</td> <td>10 16</td> <td>100 450</td> <td>333 4500</td> <td>Ivan Garfio Lizbeth Meraz</td> </tr> <tr> <td></td> <td></td> <td>0.875</td> <td>8</td> <td>16</td> <td>800</td> <td>7467</td> <td>Yop</td> </tr> </tbody> </table>					MADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE	6 Pino Nacional	Shop3	0.5 1.25	8 6	10 16	100 450	333 4500	Ivan Garfio Lizbeth Meraz			0.875	8	16	800	7467	Yop
MADERA	TIPO	GRUESO	ANCHO	LARGO	CANTIDAD	TOTAL	REMITENTE																					
6 Pino Nacional	Shop3	0.5 1.25	8 6	10 16	100 450	333 4500	Ivan Garfio Lizbeth Meraz																					
		0.875	8	16	800	7467	Yop																					
<input type="button" value="Salir"/> 7																												

1. Este cuadro de texto da la opción de escoger el largo de madera
2. Barra donde podemos asignar la fecha de inicio de búsqueda
3. Barra donde podemos asignar la fecha de fin de búsqueda
4. Botón que da inicio a la búsqueda o al filtro de información
5. Nos manda a una ventana auxiliar donde podemos apreciar el formato antes de imprimir
6. Tabla que nos muestra el resultado de la búsqueda que hemos pedido
7. Botón de salir cierra esta pantalla y devuelve a la pantalla principal.



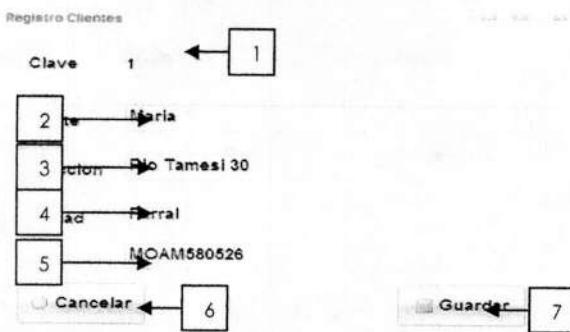
## Pantalla Tarjeta de Proveedores



1. Espacio donde debe escribir la clave de proveedor
2. Botón de empezar a filtrar la información
3. Botón que limpia el espacio de clave de proveedor
4. Organizar fecha de inicio de filtro
5. Poner fecha de fin de filtro de proveedores y su información
6. Actualiza la información que en el momento esta mostrada en la tabla
7. Tabla donde se muestra toda la información del proveedor
8. Botón salir nos devuelve a la pantalla anterior de Resumen de Tarjas
9. Manda a imprimir el formato que se muestra en pantalla.

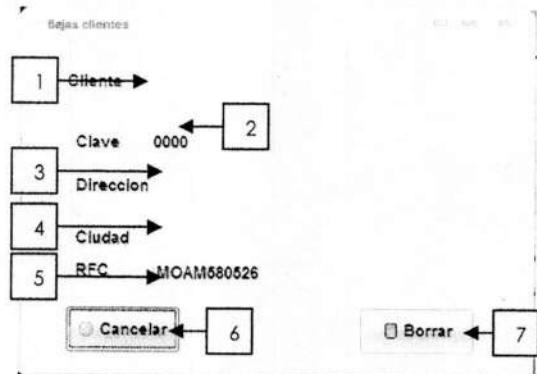


### Pantalla para dar altas a los clientes



1. Es la clave de clientes se genera de forma automatica
2. Espacio para el nombre de cliente
3. Renglon donde debe poner la direccion del cliente
4. Lugar para poder escribir la ciudad del cliente
5. RFC de registro del cliente
6. Botón para cancelar la operación sale sin guardar al nuevo cliente
7. Sirve para guardar los datos del cliente

### Pantalla para dar de baja o eliminan a los Clientes



1. Se escribe el nombre del cliente a dar de baja
2. Se posiciona automáticamente la clave del cliente
3. Espacio donde aparece automáticamente la dirección del cliente
4. En este lugar indica la ciudad de la persona
5. RFC de registro del cliente
6. Botón para cancelar la operación sale sin dar de baja al cliente
7. Borra al cliente de la base de datos.

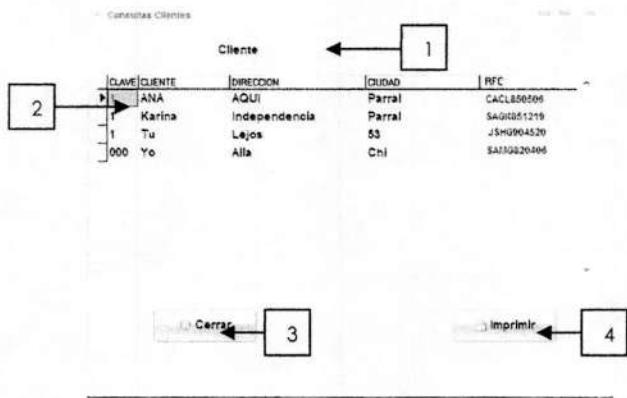


## Pantalla de Cambios en la información de nuestros clientes



1. Se da el nombre del cliente para generar el cambio
2. La clave se genera automáticamente al dar el nombre del cliente
3. La dirección se puede sustituir por otra nueva
4. El espacio de la ciudad puede ser cambiada por otra
5. RFC de registro del cliente
6. Botón para cancelar la operación sale sin dar cambios del cliente
7. Al dar click en este botón podemos actualizar los datos del cliente seleccionado

## Pantalla Consultas



1. Se da el nombre de cliente en este lugar para que nos muestre los datos de esta persona
2. Muestra la información del cliente seleccionado
3. Cierra la ventana que mostró la información
4. Imprime la información mostrada del cliente.

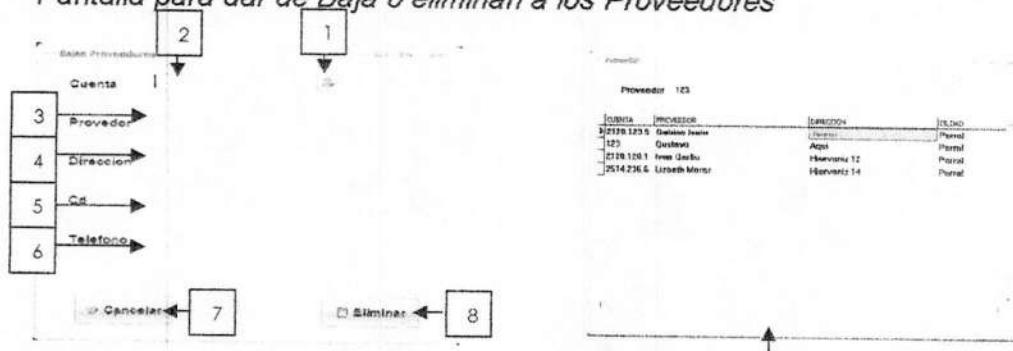


## Pantalla Altas de Proveedores



1. Se asigna cuenta al proveedor manualmente
2. Espacio para poner el nombre del proveedor
3. Escribir la dirección del proveedor
4. Poner la ciudad donde se localiza el proveedor
5. Escribes el numero de telefono
6. Botón para no guardar los datos anteriores
7. Botón de guardar y la informacion.

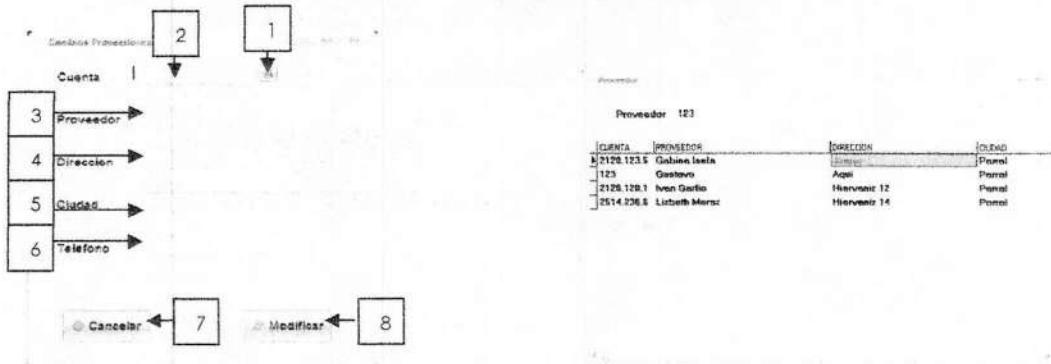
## Pantalla para dar de Baja o eliminar a los Proveedores



1. Dar clic en el icono para mostrar tabla de proveedores
2. El numero de cuenta aparece automáticamente
3. Aparece automático el nombre
4. Se muestra la dirección
5. Muestra la ciudad
6. Nos muestra el telefono
7. Cancela el proceso
8. Elimina al usuario

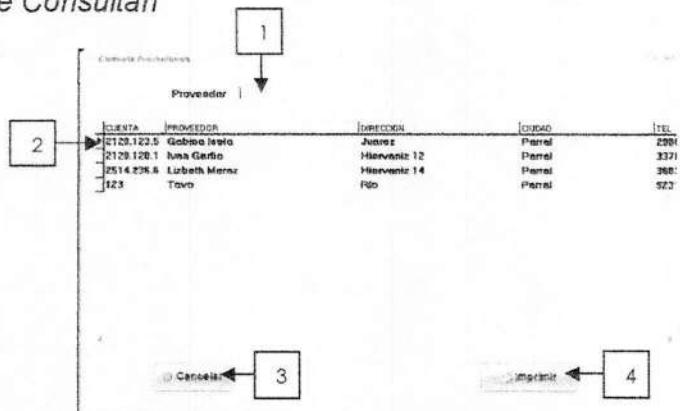


## Pantalla de Cambios en la información de Proveedores



1. Dar clic en el icono para mostrar tabla de proveedores
2. El numero de cuenta aparece automáticamente
3. Se puede cambiar el nombre
4. Aquí cambiamos la dirección
5. Puedes cambiar la ciudad
6. Aquí cambiamos el telefono
7. Cancela el proceso
8. Guarda los cambios creados

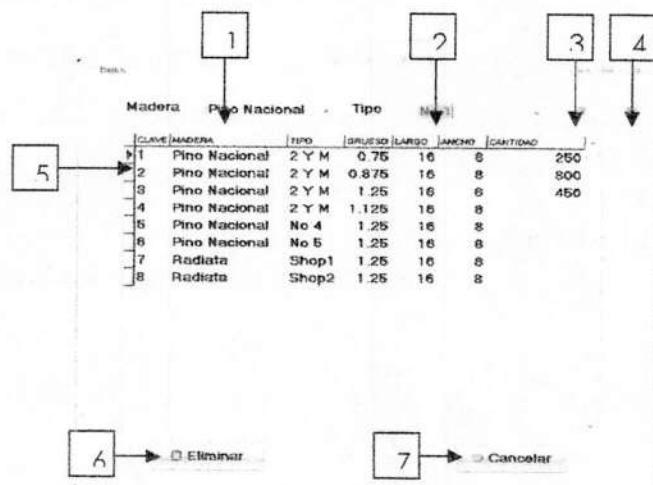
## Pantalla de Consultan



1. Se da el nombre del proveedor en este lugar para que nos muestre la información
2. Muestra la información del proveedor seleccionado
3. Cierra la ventana que mostró la información
4. Imprime la información mostrada del proveedor.



Pantalla Bajas de Madera se accesa desde la pantalla principal en el menú madera y submenú bajas donde podrás dar bajas de inventario de forma total ya sea por madera o por tipo



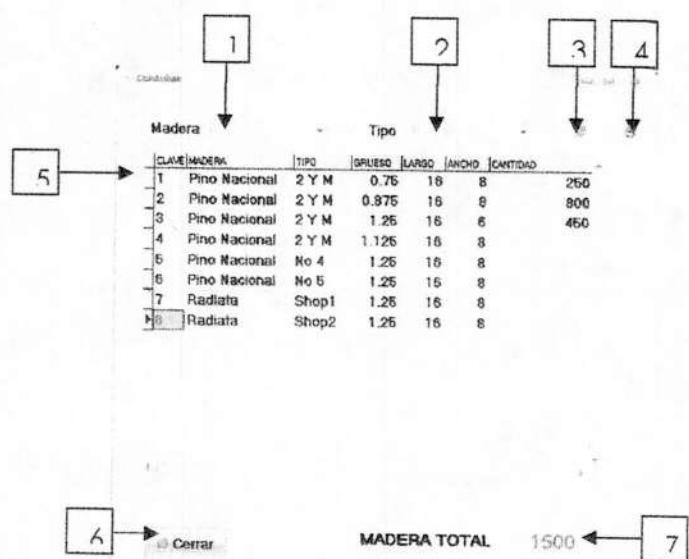
1. Es para escoger la madera a eliminar
2. Lugar reservado para el tipo a escoger
3. Actualiza la tabla de datos
4. Limpia la tabla que muestra la información
5. Tabla de información de madera
6. Una vez seleccionada se presiona este botón para eliminar totalmente
7. Cancela o invalida toda operación

Pantalla de Cambios de Madera aquí podemos cambiar la información de la madera se accesa a través del menú Madera

1. Se le da la clave de la madera que necesitamos cambiar
2. Se guardan los cambios hechos en los recuadros necesarios
3. Se cancela salir sin guardar y nos manda a pantalla principal



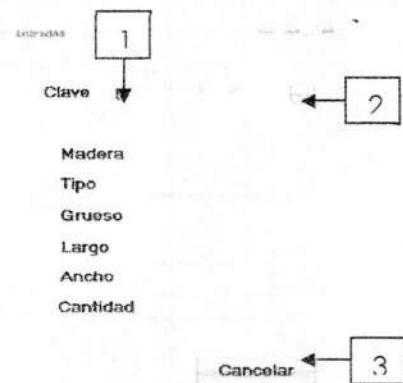
Pantalla Consultas de Madera podemos revisar y visualizar todo lo que hay en madera entramos a través de el menú madera submenú consultas



1. Es para escoger la madera a consultar
2. Lugar reservado para el tipo a escoger
3. Actualiza la tabla de datos
4. Limpia la tabla que muestra la información
5. Tabla de información de madera
6. Cancela o invalida ~~total~~ operación sacándonos de esta pantalla
7. Muestra la existencia de la madera en su total.

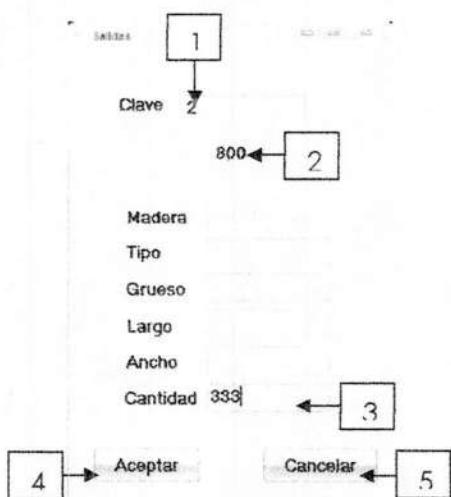


Pantalla de Entrada de Inventarios donde puede ver la entrada de madera según su clave pero no puede modificar sus características. Se entra por la pantalla principal menú Inventarios submenú Entradas



1. Se otorga el numero de clave del producto o combinación que se deseé observar
2. Nos muestra la pantalla de Consultas ver mas adelante
3. Cancelas y sales de esta ventana volviendo a pantalla principal

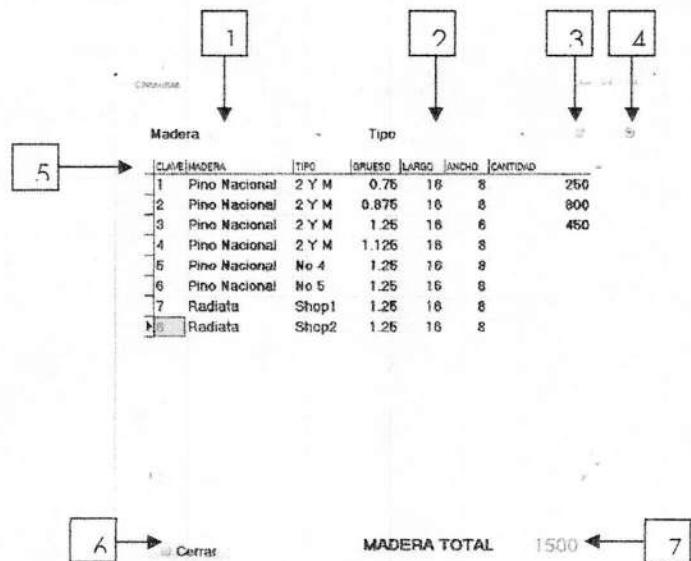
Pantalla de Salida de Inventarios donde puede ver la salida del producto o madera ya terminada que va hacia otro proceso. Se entra por la pantalla principal menú Inventarios submenú Salidas



1. Se le otorga la clave de la combinación a borrar
2. Muestra la cantidad en existencia de dicha madera
3. Se escribe la cantidad a dar de baja debe ser menor a la existencia
4. Se da de baja la cantidad marcada y se limpian los campos
5. Cancela y sale de la operación hacia la pantalla principal



Pantalla de Consultas de Inventario aquí podrá ver el final de la madera cuantos pies quedan y la existencia total. Se llega a esta pantalla en el menú Inventarios y submenú consultas



1. Es para escoger la madera a consultar
2. Lugar reservado para el tipo a escoger
3. Actualiza la tabla de datos
4. Limpia la tabla que muestra la información
5. Tabla de información de madera
6. Cancela o invalida toda operación sacándonos de esta pantalla
7. Muestra la existencia de la madera en su total.

# ANEXO

2



## MANUAL TÉCNICO

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, jpeg, ExtCtrls, xpman;

type
  TForm1 = class(TForm)
    Image1: TImage;
    MainMenu1: TMainMenu;
    Compras1: TMenuItem;
    Recepcion1: TMenuItem;
    Debolucion1: TMenuItem;
    arjas1: TMenuItem;
    Clientes1: TMenuItem;
    Altas1: TMenuItem;
    Bajas1: TMenuItem;
    Cambios1: TMenuItem;
    Busquedas1: TMenuItem;
    Proveedores1: TMenuItem;
    Altas2: TMenuItem;
    Bajas2: TMenuItem;
    Cambios2: TMenuItem;
    Consultas1: TMenuItem;
    Salir1: TMenuItem;
    procedure Salir1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Recepcion1Click(Sender: TObject);
    procedure arjas1Click(Sender: TObject);
    procedure Debolucion1Click(Sender: TObject);
    procedure Altas1Click(Sender: TObject);
    procedure Bajas1Click(Sender: TObject);
    procedure Cambios1Click(Sender: TObject);
    procedure Busquedas1Click(Sender: TObject);
    procedure Altas2Click(Sender: TObject);
    procedure Bajas2Click(Sender: TObject);
    procedure Cambios2Click(Sender: TObject);
    procedure Consultas1Click(Sender: TObject);
    procedure BusquedalClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```



```
var
  Form1: TForm1;

implementation

uses Unit2, Unit3, Unit6, Unit8, Unit12, Unit13, Unit15, Unit17, Unit23,
  Unit19, Unit21, Unit20, Unit22, Unit14, Unit16, Unit4;

{$R *.dfm}

procedure TForm1.Salir1Click(Sender: TObject);
begin
  Form1.Close;
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
  form2.ShowModal;
end;

procedure TForm1.Repcion1Click(Sender: TObject);
begin
  form3.ShowModal;
end;

procedure TForm1.arjas1Click(Sender: TObject);
begin
  form8.ShowModal;
end;

procedure TForm1.Debolucion1Click(Sender: TObject);
begin
  form6.ShowModal;
end;

procedure TForm1.Altas1Click(Sender: TObject);
begin
  form19.ShowModal;
end;

procedure TForm1.Bajas1Click(Sender: TObject);
begin
  form21.showmodal;
end;

procedure TForm1.Cambios1Click(Sender: TObject);
begin
  form20.showmodal;
end;
```



```
procedure TForm1.Busquedas1Click(Sender: TObject);
begin
form22.showmodal;
end;

procedure TForm1.Altas2Click(Sender: TObject);
begin
form12.ShowModal;
end;

procedure TForm1.Bajas2Click(Sender: TObject);
begin
form13.Showmodal;
end;

procedure TForm1.Cambios2Click(Sender: TObject);
begin
form15.ShowModal;
end;

procedure TForm1.Consultas1Click(Sender: TObject);
begin
form17.ShowModal;
end;

procedure TForm1.BusquedalClick(Sender: TObject);
begin
form14.ShowModal;
end;

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, ComCtrls,xpman;

type
  TForm2 = class(TForm)
    Image1: TImage;
    Timer1: TTimer;
    ProgressBar1: TProgressBar;
    Timer2: TTimer;
    procedure Timer1Timer(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
```



```
public
  { Public declarations }
end;

var
  Form2: TForm2;

implementation

{$R *.dfm}

procedure TForm2.Timer1Timer(Sender: TObject);
begin
  form2.Close;
end;

procedure TForm2.Timer2Timer(Sender: TObject);
begin
  progressbar1.position:= progressbar1.position + 15;
end;

end.

unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Buttons, ComCtrls, DB, DBTables,xpman;

type
  TForm3 = class(TForm)
    Bevel1: TBevel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    LbTot: TLabel;
    LbFolio: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label5: TLabel;
```



```
LbTt: TLabel;
Label6: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
SpeedButton1: TSpeedButton;
Label19: TLabel;
EdAncho: TEdit;
EdLargo: TEdit;
EdCanti: TEdit;
EdRemit: TEdit;
EdConsig: TEdit;
EdDir: TEdit;
EdChof: TEdit;
EdCam: TEdit;
EdGuia: TEdit;
EdRem: TEdit;
DTPFecha: TDateTimePicker;
BtCancel: TBitBtn;
BtGuar: TBitBtn;
BtGuyAg: TBitBtn;
BtImp: TBitBtn;
EdRecib: TEdit;
PTot: TPanel;
GroupBox1: TGroupBox;
Label1: TLabel;
EdDos: TEdit;
Eduno: TEdit;
Edit1: TEdit;
EdClase: TEdit;
EdTipo: TEdit;
EdCuenta: TEdit;
EdPre: TEdit;
Timer1: TTimer;
TbMad: TTable;
TbInv: TTable;
TbFol: TTable;
TbProv: TTable;
DsProv: TDataSource;
TbMadMADERA: TStringField;
TbMadTIPO: TStringField;
TbMadGRUESO: TFloatField;
TbMadANCHO: TFloatField;
TbMadLARGO: TFloatField;
TbMadCANTIDAD: TFloatField;
TbMadTOTAL: TFloatField;
TbMadREMITENTE: TStringField;
TbMadCONSIG: TStringField;
TbMadDIRECCION: TStringField;
TbMadCHOFER: TStringField;
```



```
TbMadCAMION: TStringField;
TbMadFECHA: TDateField;
TbMadGUIA: TStringField;
TbMadREMFTAL: TStringField;
TbMadRICIBIO: TStringField;
TbMadFOLIO: TStringField;
TbMadTOTALTARJA: TFloatField;
TbMadPRECIO: TFloatField;
TbInvMADERA: TStringField;
TbInvTIPO: TStringField;
TbInvGRUESO: TFloatField;
TbInvANCHO: TFloatField;
TbInvLARGO: TFloatField;
TbInvCANTIDAD: TFloatField;
TbInvFOLIO: TStringField;
TbFolFOLIO: TFloatField;
TbProvCUENTA: TStringField;
TbProvPROVEEDOR: TStringField;
TbProvDIRECCION: TStringField;
TbProvCIUDAD: TStringField;
TbProvTEL: TStringField;
TbTem: TTable;
TbTemMADERA: TStringField;
TbTemTIPO: TStringField;
TbTemGRUESO: TFloatField;
TbTemLARGO: TFloatField;
TbTemANCHO: TFloatField;
TbTemCANTIDAD: TFloatField;
TbTemTOTAL: TFloatField;
TbTemREMITENTE: TStringField;
TbTemCONSIG: TStringField;
TbTemDIRECCION: TStringField;
TbTemCHOFER: TStringField;
TbTemCAMION: TStringField;
TbTemFECHA: TDateField;
TbTemGUIA: TStringField;
TbTemREMFTAL: TStringField;
TbTemRECIBIO: TStringField;
TbTemFOLIO: TStringField;
TbTemTOTALTARJA: TFloatField;
TbTemPRECIO: TFloatField;
procedure EdClaseKeyPress(Sender: TObject; var Key: Char);
procedure EdTipoKeyPress(Sender: TObject; var Key: Char);
procedure EdunoKeyPress(Sender: TObject; var Key: Char);
procedure EdDosKeyPress(Sender: TObject; var Key: Char);
procedure EdPreKeyPress(Sender: TObject; var Key: Char);
procedure EdAnchoKeyPress(Sender: TObject; var Key: Char);
procedure EdLargoKeyPress(Sender: TObject; var Key: Char);
procedure EdCantiKeyPress(Sender: TObject; var Key: Char);
procedure EdCantiExit(Sender: TObject);
```



```

procedure EdRemitKeyPress(Sender: TObject; var Key: Char);
procedure EdConsigKeyPress(Sender: TObject; var Key: Char);
procedure EdDirKeyPress(Sender: TObject; var Key: Char);
procedure EdChofKeyPress(Sender: TObject; var Key: Char);
procedure EdCamKeyPress(Sender: TObject; var Key: Char);
procedure EdGuiaKeyPress(Sender: TObject; var Key: Char);
procedure EdRemKeyPress(Sender: TObject; var Key: Char);
procedure EdRecibKeyPress(Sender: TObject; var Key: Char);
procedure BtCancelClick(Sender: TObject);
procedure BtGuyAgClick(Sender: TObject);
procedure BtGuarClick(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure EdCuentaKeyPress(Sender: TObject; var Key: Char);
procedure FormActivate(Sender: TObject);
procedure EdDosExit(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure BtImpClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form3: TForm3;

implementation

uses Unit4, Unit5;

{$R *.dfm}

procedure TForm3.EdClaseKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8])) then
      key:=#0;
  end;

procedure TForm3.EdTipoKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);

```



```
end;

procedure TForm3.EdunoKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm3.EdDosKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm3.EdPreKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8,'.'])) then
  key:=#0;
end;

procedure TForm3.EdAnchoKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm3.EdLargoKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm3.EdCantiKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm3.EdCantiExit(Sender: TObject);
var larg,anch,grueso,uno,dos,cubico,cant,cubicar,piezaje:extended;
x,y:integer;
begin
Larg:=StrToInt(EdLargo.Text);
Anch:=StrToInt(EdAncho.Text);
cant:=StrToInt(edCanti.Text);
uno:=strtoint(eduno.Text);
dos:=strtoint(eddos.Text);
Grueso:=uno/dos;
cubicar:=larg*anch;
cubico:=(cubicar/12)*grueso;
```



```
piezaje:=(cubico*cant);
x:=round(piezaje);
ptot.Caption:=inttostr(x);
if lbtt.Caption='0000' then
  lbtt.Caption:= ptot.Caption
else
begin
  y:=(stroint(lbtt.Caption))+x;
  lbtt.Caption:=inttostr(y);
end;

end;

procedure TForm3.EdRemitKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9',''])) then
      key:=#0;
  end;

procedure TForm3.EdConsigKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9',''])) then
      key:=#0;
  end;

procedure TForm3.EdDirKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9',''])) then
      key:=#0;
  end;

procedure TForm3.EdChofKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
```



```

or (SelLength = Length(Text)) then
if Key in ['a'..'z'] then
  Key:= UpCase(Key);
if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
  key:=#0;
end;

procedure TForm3.EdCamKeyPress(Sender: TObject; var Key: Char);
begin
  Key:= UpCase(Key);
if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
  key:=#0;
end;

procedure TForm3.EdGuiaKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9'])) then
      key:=#0;
    end;

procedure TForm3.EdRemKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
    end;

procedure TForm3.EdRecibKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
    end;

procedure TForm3.BtCancelClick(Sender: TObject);
var ctd, jh:integer;
begin

```



```
ctd:= tbitem.RecordCount;
if ctd=0 then
begin
tbitem.Close;
end
else
begin
for jh:=1 to ctd do
begin
tbitem.Delete;
tbitem.Next;
end;
end;
form3.Close;
edtipo.Text:="";
eduno.Text:="";
eddos.Text:="";
edclase.Clear;
edlargo.Text:="";
edcanti.Text:="";
ptot.Caption:='0000';
edremit.Text:="";
edconsig.Text:="";
eddir.Text:="";
edchof.Text:="";
edguia.Text:="";
edrem.Text:="";
edcam.Clear;
edancho.Clear;
edrecib.Clear;
edpre.Clear;
lbtt.Caption:='0000';
edcuenta.Clear;
end;

procedure TForm3.BtGuyAgClick(Sender: TObject);
var grues:extended;
begin
grues:=(strtofloat(eduno.Text))/(strtofloat(eduno.Text));

tbmad.Append;
tbmadMadera.AsString:=edclase.Text;
tbmadtipo.AsString:=edtipo.Text;
tbmadgrueso.AsFloat:=strtofloat(edit1.Text);
tbmadancho.AsFloat:=strtofloat(edancho.Text);
tbmadprecio.AsFloat:=strtofloat(edpre.Text);
tbmadLargo.AsFloat:=strtofloat(edlargo.Text);
tbmadCantidad.AsInteger:=strtoint(edcanti.Text);
tbmadtotal.AsInteger:=strtoint(ptot.Caption);
```



```
tbmadremitente.AsString:=edremit.Text;
tbmadconsig.AsString:=edconsig.Text;
tbmaddiraccion.AsString:=eddir.Text;
tbmadchofer.AsString:=edchof.Text;
tbmadcamion.AsString:=edcam.Text;
tbmadfecha.AsDateTime:=dtpfecha.DateTime;
tbmadguia.AsString:=edguia.Text;
tbmadremftal.AsString:=edrem.Text;
tbmadricibio.AsString:=edrecib.Text;
tbmadfolio.AsString:=lbfolio.Caption;
tbmad.Post;

tbTem.Append;
tbTemMadera.AsString:=edclase.Text;
tbTemtipo.AsString:=editipo.Text;
tbTemgrueso.AsFloat:=strtofloat(edit1.Text);
tbTemancho.AsFloat:=strtofloat(edancho.Text);
tbTemprecio.AsFloat:=strtofloat(edpre.Text);
tbTemLargo.AsFloat:=strtofloat(edlargo.Text);
tbTemCantidad.AsInteger:=strtoint(edcanti.Text);
tbTemtotal.AsInteger:=strtoint(ptot.Caption);

tbTemremitente.AsString:=edremit.Text;
tbTemconsig.AsString:=edconsig.Text;
tbTemdireccion.AsString:=eddir.Text;
tbTemchofer.AsString:=edchof.Text;
tbTemcamion.AsString:=edcam.Text;
tbTemfecha.AsDateTime:=dtpfecha.DateTime;
tbTemguia.AsString:=edguia.Text;
tbTemremftal.AsString:=edrem.Text;
tbTemrecibio.AsString:=edrecib.Text;
tbTemfolio.AsString:=lbfolio.Caption;
tbTem.Post;

tbinv.Append;
tbInvMadera.AsString:=edclase.Text;
tbInvtipo.AsString:=editipo.Text;
tbInvgrueso.AsFloat:=grues;
tbInvancho.AsFloat:=strtofloat(edancho.Text);
tbInvLargo.AsFloat:=strtofloat(edlargo.Text);
tbInvCantidad.AsInteger:=strtoint(edcanti.Text);
tbInvFolio.AsString:=lbfolio.Caption;
tbinv.Post;

edremit.Enabled:=false;
edconsig.Enabled:=false;
eddir.Enabled:=false;
edchof.Enabled:=false;
edcam.Enabled:=false;
dtpfecha.Enabled:=false;
```



```
edguia.Enabled:=false;
edrem.Enabled:=false;
edrecib.Enabled:=false;
edcuenta.Enabled:=false;
edancho.Text:="";
edlargo.Text:="";
edcanti.Text:="";
edclase.Clear;
edtipo.Clear;
eduno.Clear;
eddos.Clear;
edpre.Clear;
edit1.Clear;
edclase.SetFocus;
ptot.Caption:='0000';
end;

procedure TForm3.BtGuarClick(Sender: TObject);
var grues:extended;
folio:integer;

begin
folio:=strtoint(lbfolio.Caption);
grues:=strtofloat(edit1.Text);
tbmad.Append;
tbmadMadera.AsString:=edclase.Text;
tbmadtipo.AsString:=edtipo.Text;
tbmadgrueso.AsFloat:=grues;
tbmadancho.AsFloat:=strtofloat(edancho.Text);
tbmadLargo.AsFloat:=strtofloat(edlargo.Text);
tbmadCantidad.AsInteger:=strtoint(edcanti.Text);
tbmadremitente.AsString:=edremit.Text;
tbmadconsig.AsString:=edconsig.Text;
tbmaddiraccion.AsString:=eddir.Text;
tbmadchofer.AsString:=edchof.Text;
tbmadcamion.AsString:=edcam.Text;
tbmadfecha.AsDateTime:=dtpfecha.DateTime;
tbmadguia.AsString:=edguia.Text;
tbmadremftal.AsString:=edrem.Text;
tbmadricibio.AsString:=edrecib.Text;
tbmadprecio.AsFloat:=strtofloat(edpre.Text);
tbmadfolio.AsString:=lbfolio.Caption;
tbmadtotaltarja.AsInteger:=strtoint(ptot.Caption);
tbmadprecio.AsFloat:=strtofloat(edpre.Text);
if (lbtt.Caption)='0000' then
begin
tbmadtotal.AsFloat:=strtofloat(ptot.Caption);
tbmadtotaltarja.AsFloat:=strtofloat(ptot.Caption);
end
else
```



```

begin
tbmadtotal.AsFloat:=strtofloat(ptot.Caption);
tbmadtotaltarja.AsInteger:=strtoint(lbtt.Caption);
end;
tbmad.Post;

tbTem.Append;
tbTemMadera.AsString:=edclase.Text;
tbTemtipo.AsString:=editipo.Text;
tbTemgrueso.AsFloat:=strtofloat(edit1.Text);
tbTemancho.AsFloat:=strtofloat(edancho.Text);
tbTemprecio.AsFloat:=strtofloat(edpre.Text);
tbTemLargo.AsFloat:=strtofloat(edlargo.Text);
tbTemCantidad.AsInteger:=strtoint(edcanti.Text);
tbTemtotal.AsFloat:=strtofloat(ptot.Caption);

tbTemremitente.AsString:=edremit.Text;
tbTemconsig.AsString:=edconsig.Text;
tbTemdirección.AsString:=eddir.Text;
tbTemchofer.AsString:=edchof.Text;
tbTemcamion.AsString:=edcam.Text;
tbTemfecha.AsDateTime:=dtpfecha.DateTime;
tbTemguia.AsString:=edguia.Text;
tbTemremftal.AsString:=edrem.Text;
tbTemrecibio.AsString:=edrecib.Text;
tbTemfolio.AsString:=lbfolio.Caption;
if (lbtt.Caption)='0000' then
begin
tbTemtotal.AsInteger:=strtoint(ptot.Caption);
tbTemtotaltarja.AsInteger:=strtoint(ptot.Caption);
end
else
begin
tbTemtotal.AsInteger:=strtoint(ptot.Caption);
tbTemtotaltarja.AsInteger:=strtoint(lbtt.Caption);
end;
tbTem.Post;

tbinv.Append;
tbInvMadera.AsString:=edclase.Text;
tbInvtipo.AsString:=editipo.Text;
tbInvgrueso.AsFloat:=grues;
tbInvancho.AsFloat:=strtofloat(edancho.Text);
tbInvLargo.AsFloat:=strtofloat(edlargo.Text);
tbInvCantidad.AsInteger:=strtoint(edcanti.Text);
tbInvFolio.AsString:=lbfolio.Caption;
tbinv.Post;
editipo.Text:="";
eduno.Text:="";
eddos.Text:="";

```



```
edancho.Text:="";
edlargo.Text:="";
edcanti.Text:="";
ptot.Caption:='0000';
edremit.Text:="";
edconsig.Text:="";
eddir.Text:="";
edchof.Text:="";
edcam.text:="";
edguia.Text:="";
edrem.Text:="";
edrecib.Text:="";
edit1.Text:="";
edcuenta.Clear;
edclase.Text:="";
edpre.Clear;
lbtt.Caption:='0000';
folio:=folio+1;
tbfol.Edit;
tbfolfolio.AsInteger:=folio;
tbfol.Post;
lbfolio.Caption:=inttostr(tbfolfolio.AsInteger);
edremit.Enabled:=true;
edconsig.Enabled:=true;
eddir.Enabled:=true;
edchof.Enabled:=true;
edcam.Enabled:=true;
dtpfecha.Enabled:=true;
edguia.Enabled:=true;
edrem.Enabled:=true;
edrecib.Enabled:=true;
end;

procedure TForm3.SpeedButton1Click(Sender: TObject);
begin
form4.ShowModal;
end;

procedure TForm3.EdCuentaKeyPress(Sender: TObject; var Key: Char);
begin
if (not( key in['0'..'9','#8,'.'])) then
  key:=#0;
end;

procedure TForm3.FormActivate(Sender: TObject);
begin
tbmad.Open;
TbInv.Open;
tbfol.Open;
TbTem.Open;
```



```

dtpfecha.Date:= now;
lbfolio.Caption:=tbfolio.AsString;
edclase.SetFocus;
end;

procedure TForm3.EdDosExit(Sender: TObject);
var grues:extended;
begin
grues:=(strtoint(eduno.Text))/(strtoint(eddos.Text));
edit1.Text:=floattostr(grues);
end;

procedure TForm3.Timer1Timer(Sender: TObject);
begin
if (trim(Eduno.Text)<>"")and(trim(eddos.Text)<>)and(trim(edancho.Text)<>)and
(trim(edlargo.Text)<>)and(trim(Edcanti.Text)<>)and (trim(edremit.Text)<>) and
(trim(edrecib.Text)<>) and (trim(edpre.Text)<>)and (trim(edclase.Text)<>)and
(trim(edtipo.Text)<>)then
begin
Btguyag.Enabled:=true;
btguar.Enabled:=true;
end
ELSE
begin
btguyag.Enabled:= false;
btguar.Enabled:=false;
end;
end;

procedure TForm3.BtImpClick(Sender: TObject);
begin
form5.QuickRep1.Preview;
end;

end.

unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids,xpman;

type
  TForm4 = class(TForm)
    Label1: TLabel;
    DBGrid1: TDBGrid;
    EdProv: TEdit;

```



```

procedure EdProvChange(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure DBGrid1CellClick(Column: TColumn);
procedure EdProvKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form4: TForm4;
implementation

uses Unit3;
{$R *.dfm}

procedure TForm4.EdProvChange(Sender: TObject);
begin
  Form3.tbProv.SetKey;
  Form3.TBProv['Provedor]:=trim(EdProv.Text);
  Form3.tbProv.GotoNearest;
end;

procedure TForm4.FormActivate(Sender: TObject);
begin
  Form3.tbprov.open;
  Form3.tbprov.IndexName:='Proveedor';
  Form3.tbprov.First;
  edProv.SetFocus;
end;

procedure TForm4.DBGrid1CellClick(Column: TColumn);
begin
  Form3.EdCuenta.Text:=Form3.TbProvCuentaAsString;
  Form3.EdRemit.Text:=Form3.TbProvProveedorAsString;
  Form3.EdDir.Text:=Form3.TbProvDireccionAsString;
  form4.Close;
end;

procedure TForm4.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
end;

```



```
end.  
unit Units5;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, jpeg, QRCtrls, QuickRpt, ExtCtrls, xpman;  
  
type  
  TForm5 = class(TForm)  
    QuickRep1: TQuickRep;  
    TitleBand1: TQRBand;  
    QRShape12: TQRShape;  
    QRLLabel1: TQRLLabel;  
    QRLLabel2: TQRLLabel;  
    QRLLabel3: TQRLLabel;  
    QRLLabel4: TQRLLabel;  
    QRLLabel5: TQRLLabel;  
    QRLLabel6: TQRLLabel;  
    QRLLabel7: TQRLLabel;  
    QRLLabel8: TQRLLabel;  
    QRLLabel9: TQRLLabel;  
    QRLLabel10: TQRLLabel;  
    QRLLabel11: TQRLLabel;  
    QRLLabel12: TQRLLabel;  
    QRLLabel13: TQRLLabel;  
    QRLLabel14: TQRLLabel;  
    QRLLabel15: TQRLLabel;  
    QRDBText1: TQRDBText;  
    QRDBText2: TQRDBText;  
    QRDBText3: TQRDBText;  
    QRDBText4: TQRDBText;  
    QRDBText5: TQRDBText;  
    QRDBText6: TQRDBText;  
    QRDBText7: TQRDBText;  
    QRDBText8: TQRDBText;  
    QRDBText9: TQRDBText;  
    QRDBText10: TQRDBText;  
    QRDBText11: TQRDBText;  
    QRDBText12: TQRDBText;  
    QRLLabel16: TQRLLabel;  
    QRLLabel17: TQRLLabel;  
    QRLLabel18: TQRLLabel;  
    QRLLabel19: TQRLLabel;  
    QRLLabel20: TQRLLabel;  
    QRLLabel21: TQRLLabel;  
    QRShape1: TQRShape;  
    QRShape2: TQRShape;  
    QRShape3: TQRShape;  
    QRShape4: TQRShape;  
    QRShape5: TQRShape;
```



```
QRShape6: TQRShape;
QRShape7: TQRShape;
QRShape8: TQRShape;
QRShape9: TQRShape;
QRShape10: TQRShape;
QRShape11: TQRShape;
QRDBText17: TQRDBText;
QRLabel11: TQRLLabel;
QRShape13: TQRShape;
QRImage1: TQRImage;
DetailBand2: TQRBand;
QRDBText13: TQRDBText;
QRDBText14: TQRDBText;
QRDBText15: TQRDBText;
QRDBText19: TQRDBText;
QRDBText16: TQRDBText;
QRSubDetail1: TQRSubDetail;
SummaryBand1: TQRBand;
QRLabel12: TQRLLabel;
QRShape14: TQRShape;
QRDBText18: TQRDBText;
QRLabel23: TQRLLabel;
QRExpr1: TQRExpr;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form5: TForm5;

implementation

uses Unit4, Unit3;

{$R *.dfm}

end.

unit Unit6;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, StdCtrls, Grids, DBGrids, Buttons, xpman;

type
```



```
TForm6 = class(TForm)
  LbFolio: TLabel;
  BtGuyAg: TBitBtn;
  BtImp: TBitBtn;
  BtCancel: TBitBtn;
  DBGrid1: TDBGrid;
  edfolio: TEdit;
  TbMad: TTable;
  TbMadMADERA: TStringField;
  TbMadTIPO: TStringField;
  TbMadGRUESO: TFloatField;
  TbMadANCHO: TFloatField;
  TbMadLARGO: TFloatField;
  TbMadCANTIDAD: TFloatField;
  TbMadTOTAL: TFloatField;
  TbMadREMITENTE: TStringField;
  TbMadCONSIG: TStringField;
  TbMadDIRECCION: TStringField;
  TbMadCHOFER: TStringField;
  TbMadCAMION: TStringField;
  TbMadFECHA: TDateField;
  TbMadGUIA: TStringField;
  TbMadREMFTAL: TStringField;
  TbMadRICIBIO: TStringField;
  TbMadFOLIO: TStringField;
  TbMadTOTALTARJA: TFloatField;
  TbMadPRECIO: TFloatField;
  DataSource1: TDataSource;
  TbTem: TTable;
  TbTemMADERA: TStringField;
  TbTemTIPO: TStringField;
  TbTemGRUESO: TFloatField;
  TbTemLARGO: TFloatField;
  TbTemANCHO: TFloatField;
  TbTemCANTIDAD: TFloatField;
  TbTemTOTAL: TFloatField;
  TbTemREMITENTE: TStringField;
  TbTemCONSIG: TStringField;
  TbTemDIRECCION: TStringField;
  TbTemCHOFER: TStringField;
  TbTemCAMION: TStringField;
  TbTemFECHA: TDateField;
  TbTemGUIA: TStringField;
  TbTemREMFTAL: TStringField;
  TbTemRECIBIO: TStringField;
  TbTemFOLIO: TStringField;
  TbTemTOTALTARJA: TFloatField;
  TbTempRECIO: TFloatField;
  TbDev: TTable;
  TbDevMADERA: TStringField;
```



```

TbDevTIPO: TStringField;
TbDevGRUESO: TFloatField;
TbDevLARGO: TFloatField;
TbDevANCHO: TFloatField;
TbDevCANTIDAD: TFloatField;
TbDevTOTAL: TFloatField;
TbDevREMITENTE: TStringField;
TbDevCONSIG: TStringField;
TbDevDIRECCION: TStringField;
TbDevCHOFER: TStringField;
TbDevCAMION: TStringField;
TbDevFECHA: TDateField;
TbDevGUIA: TStringField;
TbDevREMFTAL: TStringField;
TbDevRECIBIO: TStringField;
TbDevFOLIO: TStringField;
TbDevTOTALTARJA: TFloatField;
TbDevPRECIO: TFloatField;
DBGrid2: TDBGrid;
DataSource2: TDataSource;
SpeedButton1: TSpeedButton;
procedure edfolioKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure FormActivate(Sender: TObject);
procedure DBGrid1DblClick(Sender: TObject);
procedure BtGuyAgClick(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure BtImpClick(Sender: TObject);
procedure BtCancelClick(Sender: TObject);
procedure edfolioKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form6: TForm6;
implementation
uses Unit7;

{$R *.dfm}

procedure TForm6.edfolioKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
if key=vk_return then
begin
  tbMad.filter:=(Folio ='+' QuotedStr(EdFolio.Text) '+');
  tbMad.Filtered:=true;
end;

```



```
end;
procedure TForm6.FormActivate(Sender: TObject);
begin
tbMad.open;
tbMad.IndexName:='Folio';
tbMad.First;
end;
procedure TForm6.DBGrid1DblClick(Sender: TObject);
begin
tbtem.Open;
tbtem.Append;
tbTemMaderaAsString:=tbmadMaderaAsString;
tbTemtipoAsString:=tbmadTipoAsString;
tbTemgruesoAsFloat:=tbmadgruesoAsFloat;
tbTemanchoAsFloat:=tbmadanchoAsFloat;
tbTemprecioAsFloat:=tbmadprecioAsFloat;
tbTemLargoAsFloat:=tbmadlargoAsFloat;
tbTemCantidadAsInteger:=tbmadcantidadAsInteger;
tbTemtotalAsFloat:=tbmadtotalAsInteger;
tbTemremitenteAsString:=tbmadremitenteAsString;
tbTemconsigAsString:=tbmadconsigAsString;
tbTemdireccionAsString:=tbmaddiraccionAsString;
tbTemchoferAsString:=tbmadchoferAsString;
tbTemcamionAsString:=tbmadcamionAsString;
tbTemfechaAsDateTime:=tbmadfechaAsDateTime;
tbTemguiaAsString:=tbmadguiaAsString;
tbTemremftalAsString:=tbmadremftalAsString;
tbTemrecibioAsString:=tbmadricibioAsString;
tbTemfolioAsString:=tbmadfolioAsString;
tbTemtotaltarjaAsInteger:=tbmadtotaltarjaAsInteger;
tbtem.Post;
end;
procedure TForm6.BtGuyAgClick(Sender: TObject);
var i, conta:integer;
begin
tbdev.Open;
conta:= tbtem.RecordCount;
for i:=1 to conta do
begin
tbdev.Append;
tbdevMaderaAsString:=tbmadMaderaAsString;
tbdevtipoAsString:=tbmadTipoAsString;
tbdevgruesoAsFloat:=tbmadgruesoAsFloat;
tbdevanchoAsFloat:=tbmadanchoAsFloat;
tbdevprecioAsFloat:=tbmadprecioAsFloat;
tbdevLargoAsFloat:=tbmadlargoAsFloat;
tbdevCantidadAsInteger:=tbmadcantidadAsInteger;
tbdevtotalAsFloat:=tbmadtotalAsInteger;
tbdevremitenteAsString:=tbmadremitenteAsString;
tbdevconsigAsString:=tbmadconsigAsString;
```



```
tbdevdireccion.AsString:=tbtaddireccion.AsString;
tbdevchofer.AsString:=tbtmadchofer.AsString;
tbdevcamion.AsString:=tbtmadcamion.AsString;
tbdevfecha.AsDateTime:=tbtmadfecha.AsDateTime;
tbdevguia.AsString:=tbtmadguia.AsString;
tbdevremtal.AsString:=tbtmadremtal.AsString;
tbdevrecibio.AsString:=tbtmadrecibio.AsString;
tbdevfolio.AsString:=tbtmadfolio.AsString;
tbdevtotaltarja.AsInteger:=tbtmadtotaltarja.AsInteger;
tbdev.Post;
tbdev.Next;
tbtmad.Delete;
tbtmad.Next;
end;
showmessage('procesos terminado');
tbdev.Refresh;
tbtmad.Refresh;
end;
procedure TForm6.SpeedButton1Click(Sender: TObject);
begin
tbtitem.Delete;
end;
procedure TForm6.BtImpClick(Sender: TObject);
begin
form7.QuickRep1.Preview;
end;
procedure TForm6.BtCancelClick(Sender: TObject);
var ctd, jh:integer;
begin
ctd:= tbtitem.RecordCount;
if ctd=0 then
begin
tbtitem.Close;
end
else
begin
for jh:=1 to ctd do
begin
tbtitem.Delete;
tbtitem.Next;
end;
end;
form6.Close;
tbtmad.Filtered:=false;
end;
procedure TForm6.edfolioKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in[#8,'0'..'9'])) then
  key:=#0;
end;
```



```
end.  
unit Unit7;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, QCRCtrls, QuickRpt, jpeg, ExtCtrls, xpman;  
  
type  
  TForm7 = class(TForm)  
    QuickRep1: TQuickRep;  
    TitleBand1: TQRBand;  
    QRShape12: TQRShape;  
    QRLLabel1: TQRLLabel;  
    QRLLabel2: TQRLLabel;  
    QRLLabel3: TQRLLabel;  
    QRLLabel4: TQRLLabel;  
    QRLLabel5: TQRLLabel;  
    QRLLabel6: TQRLLabel;  
    QRLLabel7: TQRLLabel;  
    QRLLabel8: TQRLLabel;  
    QRLLabel9: TQRLLabel;  
    QRLLabel10: TQRLLabel;  
    QRLLabel13: TQRLLabel;  
    QRLLabel14: TQRLLabel;  
    QRLLabel15: TQRLLabel;  
    QRDBText1: TQRDBText;  
    QRDBText2: TQRDBText;  
    QRDBText3: TQRDBText;  
    QRDBText4: TQRDBText;  
    QRDBText5: TQRDBText;  
    QRDBText6: TQRDBText;  
    QRDBText7: TQRDBText;  
    QRDBText8: TQRDBText;  
    QRDBText9: TQRDBText;  
    QRDBText10: TQRDBText;  
    QRDBText11: TQRDBText;  
    QRDBText12: TQRDBText;  
    QRLLabel16: TQRLLabel;  
    QRLLabel17: TQRLLabel;  
    QRLLabel18: TQRLLabel;  
    QRLLabel19: TQRLLabel;  
    QRLLabel20: TQRLLabel;  
    QRLLabel21: TQRLLabel;  
    QRShape1: TQRShape;  
    QRShape2: TQRShape;  
    QRShape3: TQRShape;  
    QRShape4: TQRShape;  
    QRShape5: TQRShape;
```



```
QRShape6: TQRShape;
QRShape7: TQRShape;
QRShape8: TQRShape;
QRShape9: TQRShape;
QRShape10: TQRShape;
QRShape11: TQRShape;
QRDBText17: TQRDBText;
QRLabel11: TQRLLabel;
QRShape13: TQRShape;
QRIImage1: TQRIImage;
DetailBand2: TQRBand;
QRDBText13: TQRDBText;
QRDBText14: TQRDBText;
QRDBText15: TQRDBText;
QRDBText19: TQRDBText;
QRDBText16: TQRDBText;
QRSubDetail1: TQRSubDetail;
SummaryBand1: TQRBand;
QRLabel12: TQRLLabel;
QRShape14: TQRShape;
QRDBText18: TQRDBText;
QRLabel23: TQRLLabel;
QRExpr1: TQRExpr;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form7: TForm7;

implementation

uses Unit6;

{$R *.dfm}
end.

unit Unit8;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, Menus, StdCtrls, ComCtrls, ExtCtrls, Buttons,
  Grids, DBGrids, xpman;

type
```



```
TForm8 = class(TForm)
  DBGrid1: TDBGrid;
  BtSalir: TBitBtn;
  PFechas: TPanel;
  Label1: TLabel;
  Label2: TLabel;
  DTPFecha1: TDateTimePicker;
  DtpFecha2: TDateTimePicker;
  BitBtn1: TBitBtn;
  PMad: TPanel;
  CbMad: TComboBox;
  BitBtn6: TBitBtn;
  PTipo: TPanel;
  Label4: TLabel;
  EdTipo: TEdit;
  BitBtn3: TBitBtn;
  PGrueso: TPanel;
  Label6: TLabel;
  Label7: TLabel;
  EdGrueso: TEdit;
  BitBtn2: TBitBtn;
  EdGrueso2: TEdit;
  Pancho: TPanel;
  Label5: TLabel;
  EdAncho: TEdit;
  BitBtn5: TBitBtn;
  Plargo: TPanel;
  Label3: TLabel;
  EdLargo: TEdit;
  BitBtn4: TBitBtn;
  MainMenu1: TMainMenu;
  Filtrospor1: TMenuItem;
  Fechas1: TMenuItem;
  Maderas1: TMenuItem;
  ipos1: TMenuItem;
  Grueso1: TMenuItem;
  Ancho1: TMenuItem;
  Largo1: TMenuItem;
  TbMad: TTable;
  DataSource1: TDataSource;
  TbMadMADERA: TStringField;
  TbMadTIPO: TStringField;
  TbMadGRUESO: TFloatField;
  TbMadANCHO: TFloatField;
  TbMadARGO: TFloatField;
  TbMadCANTIDAD: TFloatField;
  TbMadTOTAL: TFloatField;
  TbMadREMITENTE: TStringField;
  TbMadCONSIG: TStringField;
  TbMadDIRECCION: TStringField;
```



```
TbMadCHOFER: TStringField;
TbMadCAMION: TStringField;
TbMadFECHA: TDateField;
TbMadGUIA: TStringField;
TbMadREMFTAL: TStringField;
TbMadRICIBIO: TStringField;
TbMadFOLIO: TStringField;
TbMadTOTALTARJA: TFloatField;
TbMadPRECIO: TFloatField;
Reportel: TMenuItem;
Proveedores1: TMenuItem;
BitBtn7: TBitBtn;
BitBtn8: TBitBtn;
BitBtn9: TBitBtn;
BitBtn10: TBitBtn;
BitBtn11: TBitBtn;
BitBtn12: TBitBtn;
procedure Fechas1Click(Sender: TObject);
procedure Maderas1Click(Sender: TObject);
procedure ipos1Click(Sender: TObject);
procedure Grueso1Click(Sender: TObject);
procedure Ancho1Click(Sender: TObject);
procedure Largo1Click(Sender: TObject);
procedure EdGruesoKeyPress(Sender: TObject; var Key: Char);
procedure EdGrueso2KeyPress(Sender: TObject; var Key: Char);
procedure EdAnchoKeyPress(Sender: TObject; var Key: Char);
procedure EdLargoKeyPress(Sender: TObject; var Key: Char);
procedure EdTipoKeyPress(Sender: TObject; var Key: Char);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure BtSalirClick(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure BitBtn11Click(Sender: TObject);
procedure BitBtn12Click(Sender: TObject);
procedure Proveedores1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var



```
Form8: TForm8;  
  
implementation  
  
uses Unit9, Unit10;  
  
{$R *.dfm}  
  
procedure TForm8.Fechas1Click(Sender: TObject);  
begin  
  PFechas.Visible:=True;  
  tbMad.open;  
  tbMad.IndexName:='Fecha';  
  tbMad.First;  
  DTPFecha1.SetFocus;  
  Pancho.Visible:=False;  
  Plargo.Visible:=False;  
  PMad.Visible:=False;  
  PTipo.Visible:=False;  
  PGrueso.Visible:=False;  
end;  
  
procedure TForm8.Maderas1Click(Sender: TObject);  
begin  
  PMad.Visible:=True;  
  tbMad.open;  
  tbMad.IndexName:='Madera';  
  tbMad.First;  
  CbMad.SetFocus;  
  Pancho.Visible:=False;  
  Plargo.Visible:=False;  
  PFechas.Visible:=False;  
  PTipo.Visible:=False;  
  PGrueso.Visible:=False;  
end;  
  
procedure TForm8.ipos1Click(Sender: TObject);  
begin  
  Ptipo.Visible:=True;  
  tbMad.open;  
  tbMad.IndexName:='Tipo';  
  tbMad.First;  
  EdTipo.SetFocus;  
  Pancho.Visible:=False;  
  Plargo.Visible:=False;  
  PFechas.Visible:=False;  
  PMad.Visible:=False;  
  PGrueso.Visible:=False;  
end;
```



```
procedure TForm8.Grueso1Click(Sender: TObject);
begin
  PGrueso.Visible:=True;
  tbMad.open;
  tbMad.IndexName:='Grueso';
  tbMad.First;
  EdGrueso.SetFocus;
  Pancho.Visible:=False;
  Plargo.Visible:=False;
  PFechas.Visible:=False;
  PMad.Visible:=False;
  PTipo.Visible:=False;
end;

procedure TForm8.Ancho1Click(Sender: TObject);
begin
  Pancho.Visible:=True;
  tbMad.Open;
  tbMad.IndexName:='Ancho';
  tbMad.First;
  EdAncho.SetFocus;
  PTipo.Visible:=False;
  Plargo.Visible:=False;
  PFechas.Visible:=False;
  PMad.Visible:=False;
  PGrueso.Visible:=False;
end;

procedure TForm8.Largo1Click(Sender: TObject);
begin
  PLargo.Visible:=True;
  tbMad.open;
  tbMad.IndexName:='Largo';
  tbMad.First;
  EdLargo.SetFocus;
  Pancho.Visible:=False;
  PTipo.Visible:=False;
  PFechas.Visible:=False;
  PMad.Visible:=False;
  PGrueso.Visible:=False;
end;

procedure TForm8.EdGruesoKeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9','#8'])) then
    key:=#0;
end;

procedure TForm8.EdGrueso2KeyPress(Sender: TObject; var Key: Char);
begin
```



```
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm8.EdAnchoKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm8.EdLargoKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm8.EdTipoKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
  end;

procedure TForm8.BitBtn1Click(Sender: TObject);
var fech1, fech2:tdatetime;
begin
  fech1:= dtpfech1.DateTime;
  fech2:= dtpfech2.DateTime;
  tbMad.filter:='((Fecha >='+QuotedStr(DateToStr(Fech1))+') and (Fecha <=' +
  QuotedStr(DateToStr(Fech2))+'))';
  tbMad.Filtered:=true;

end;

procedure TForm8.BitBtn2Click(Sender: TObject);
var uno, dos:integer;
  grueso: extended;
begin
  uno:= Strtoint(edgrueso.Text);
  dos:= Strtoint(edgrueso2.Text);
  grueso:= (uno)/(dos);
  tbMad.filter:='(Grueso =' + QuotedStr(floattosrt(grueso))+')';
  tbMad.Filtered:=true;

end;

procedure TForm8.BitBtn5Click(Sender: TObject);
begin
```



```
tbMad.filter:='(Ancho =' + QuotedStr(EdAncho.Text) + ')';
tbMad.Filtered:=true;
end;

procedure TForm8.BitBtn4Click(Sender: TObject);
begin
tbMad.filter:='(Largo =' + QuotedStr(EdLargo.Text) + ')';
tbMad.Filtered:=true;
end;

procedure TForm8.BitBtn6Click(Sender: TObject);
begin
tbMad.filter:='(Madera =' + QuotedStr(cbmad.Text) + ')';
tbMad.Filtered:=true;
end;

procedure TForm8.BitBtn3Click(Sender: TObject);
begin
tbMad.filter:='( Tipo =' + QuotedStr(EdTipo.Text) + ')';
tbMad.Filtered:=true;
end;

procedure TForm8.FormActivate(Sender: TObject);
begin
tbMad.open;
Pmad.Visible:=false;
Pancho.Visible:=False;
Plargo.Visible:=False;
PFechas.Visible:=False;
PTipo.Visible:=False;
PGrueso.Visible:=False;
end;

procedure TForm8.BtSalirClick(Sender: TObject);
begin
form8.Close;
tbmad.Filtered:=false;
end;

procedure TForm8.BitBtn7Click(Sender: TObject);
begin
form9.QrFecha.Preview;
end;

procedure TForm8.BitBtn8Click(Sender: TObject);
begin
form9.QrMad.Preview;
end;

procedure TForm8.BitBtn9Click(Sender: TObject);
```



```
begin
form9.QrTipo.Preview;
end;

procedure TForm8.BitBtn10Click(Sender: TObject);
begin
form9.QrGrues.Preview;
end;

procedure TForm8.BitBtn11Click(Sender: TObject);
begin
form9.QrAncho.Preview;
end;

procedure TForm8.BitBtn12Click(Sender: TObject);
begin
form9.QrLargo.Preview;
end;

procedure TForm8.Proveedores1Click(Sender: TObject);
begin
form10.ShowModal;
end;

end.

unit Unit9;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, QuickRpt, QRCtrls, ExtCtrls, xpman;

type
  TForm9 = class(TForm)
    QrFecha: TQuickRep;
    DetailBand1: TQRBand;
    QRBand1: TQRBand;
    QRBand2: TQRBand;
    QRLabel1: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRLabel10: TQRLabel;
```



```
QRLLabel11: TQRLLabel;  
QRLLabel12: TQRLLabel;  
QRLLabel14: TQRLLabel;  
QRLLabel15: TQRLLabel;  
QRLLabel16: TQRLLabel;  
QRDBText1: TQRDBText;  
QRDBText2: TQRDBText;  
QRDBText3: TQRDBText;  
QRDBText4: TQRDBText;  
QRDBText5: TQRDBText;  
QRDBText6: TQRDBText;  
QRDBText7: TQRDBText;  
QRDBText8: TQRDBText;  
QRDBText9: TQRDBText;  
QrTipo: TQuickRep;  
TitleBand2: TQRBand;  
QRLLabel17: TQRLLabel;  
QRLLabel26: TQRLLabel;  
QRLLabel27: TQRLLabel;  
QRLLabel28: TQRLLabel;  
QRLLabel29: TQRLLabel;  
QRLLabel30: TQRLLabel;  
QRLLabel31: TQRLLabel;  
QRLLabel32: TQRLLabel;  
QRLLabel33: TQRLLabel;  
QRLLabel34: TQRLLabel;  
DetailBand3: TQRBand;  
QRDBText10: TQRDBText;  
QRDBText19: TQRDBText;  
QRDBText20: TQRDBText;  
QRDBText21: TQRDBText;  
QRDBText22: TQRDBText;  
QRDBText23: TQRDBText;  
QRDBText24: TQRDBText;  
QRDBText25: TQRDBText;  
QRSubDetail2: TQRSubDetail;  
QrMad: TQuickRep;  
TitleBand1: TQRBand;  
QRLLabel2: TQRLLabel;  
QRLLabel13: TQRLLabel;  
QRLLabel18: TQRLLabel;  
QRLLabel19: TQRLLabel;  
QRLLabel20: TQRLLabel;  
QRLLabel21: TQRLLabel;  
QRLLabel22: TQRLLabel;  
QRLLabel23: TQRLLabel;  
QRLLabel24: TQRLLabel;  
QRLLabel25: TQRLLabel;  
DetailBand2: TQRBand;  
QRDBText11: TQRDBText;
```



```
QRDBText12: TQRDBText;
QRDBText13: TQRDBText;
QRDBText14: TQRDBText;
QRDBText15: TQRDBText;
QRDBText16: TQRDBText;
QRDBText17: TQRDBText;
QRDBText18: TQRDBText;
QRSUBDetail1: TQRSUBDetail;
QrGrues: TQuickRep;
TitleBand3: TQRBand;
DetailBand4: TQRBand;
QRSUBDetail3: TQRSUBDetail;
QRLLabel35: TQLLabel;
QRLLabel36: TQLLabel;
QRLLabel37: TQLLabel;
QRLLabel38: TQLLabel;
QRLLabel39: TQLLabel;
QRLLabel40: TQLLabel;
QRLLabel41: TQLLabel;
QRLLabel42: TQLLabel;
QRLLabel43: TQLLabel;
QRLLabel44: TQLLabel;
QRLLabel45: TQLLabel;
QRLLabel46: TQLLabel;
QRDBText26: TQRDBText;
QRDBText27: TQRDBText;
QRDBText28: TQRDBText;
QRDBText29: TQRDBText;
QRDBText30: TQRDBText;
QRDBText31: TQRDBText;
QRDBText32: TQRDBText;
QRDBText33: TQRDBText;
QrAncho: TQuickRep;
TitleBand4: TQRBand;
DetailBand5: TQRBand;
QRSUBDetail4: TQRSUBDetail;
QRLLabel47: TQLLabel;
QRLLabel48: TQLLabel;
QRLLabel49: TQLLabel;
QRLLabel50: TQLLabel;
QRLLabel51: TQLLabel;
QRLLabel52: TQLLabel;
QRLLabel53: TQLLabel;
QRLLabel54: TQLLabel;
QRLLabel55: TQLLabel;
QRLLabel56: TQLLabel;
QRDBText34: TQRDBText;
QRDBText35: TQRDBText;
QRDBText36: TQRDBText;
QRDBText37: TQRDBText;
```



```
QRDBText38: TQRDBText;
QRDBText39: TQRDBText;
QRDBText40: TQRDBText;
QRDBText41: TQRDBText;
QrLargo: TQuickRep;
DetailBand6: TQRBand;
TitleBand5: TQRBand;
QRSubDetail5: TQRSubDetail;
QRLLabel57: TQRLLabel;
QRLLabel58: TQRLLabel;
QRLLabel67: TQRLLabel;
QRLLabel68: TQRLLabel;
QRLLabel69: TQRLLabel;
QRLLabel70: TQRLLabel;
QRLLabel71: TQRLLabel;
QRLLabel72: TQRLLabel;
QRLLabel73: TQRLLabel;
QRLLabel74: TQRLLabel;
QRDBText42: TQRDBText;
QRDBText43: TQRDBText;
QRDBText44: TQRDBText;
QRDBText45: TQRDBText;
QRDBText46: TQRDBText;
QRDBText47: TQRDBText;
QRDBText48: TQRDBText;
QRDBText49: TQRDBText;
procedure QrFechaBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
procedure QrMadBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
procedure QrTipoBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
procedure QrGruesBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
procedure QrAnchoBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
procedure QrLargoBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form9: TForm9;
implementation
uses Unit8;

{$R *.dfm}
```



```
procedure TForm9.QrFechaBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
  qrlabel14.Caption:=datetostr(form8.DTPFechal.DateTime);
  qrlabel16.Caption:=datetostr(form8.DTPFech2.DateTime);
end;

procedure TForm9.QrMadBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
  qrlabel13.Caption:=form8.CbMad.Text;
end;

procedure TForm9.QrTipoBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
  qrlabel26.Caption:=form8.EdTipo.Text;
end;

procedure TForm9.QrGruesBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
  qrlabel36.Caption:=form8.EdGrueso.Text;
  qrlabel37.Caption:=form8.EdGrueso2.Text;
end;

procedure TForm9.QrAnchoBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
  qrlabel48.Caption:=form8.EdAncho.Text;
end;

procedure TForm9.QrLargoBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
  qrlabel58.Caption:=form8.EdLargo.Text;
end;
```



```
unit Unit10;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, Buttons, StdCtrls, Grids, DBGrids, ComCtrls, xpman;

type
  TForm10 = class(TForm)
    Label1: TLabel;
    EdProv: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    DtFechal: TDateTimePicker;
    DtFecha2: TDateTimePicker;
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    TbMad: TTable;
    DsMad: TDataSource;
    TbMadMADERA: TStringField;
    TbMadTIPO: TStringField;
    TbMadGRUESO: TFloatField;
    TbMadANCHO: TFloatField;
    TbMadLARGO: TFloatField;
    TbMadCANTIDAD: TFloatField;
    TbMadTOTAL: TFloatField;
    TbMadREMITENTE: TStringField;
    TbMadCONSIG: TStringField;
    TbMadDIRECCION: TStringField;
    TbMadCHOFER: TStringField;
    TbMadCAMION: TStringField;
    TbMadFECHA: TDateField;
    TbMadGUIA: TStringField;
    TbMadREMFTAL: TStringField;
    TbMadRICIBIO: TStringField;
    TbMadFOLIO: TStringField;
    TbMadTOTALTARJA: TFloatField;
    TbMadPRECIO: TFloatField;
    SpeedButton3: TSpeedButton;
    procedure FormActivate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
```



```
procedure EdProvKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form10: TForm10;

implementation

uses Unit11;

{$R *.dfm}

procedure TForm10.FormActivate(Sender: TObject);
begin
  dtfechal.DateTime:=now;
  dtfecha2.DateTime:=now;
  tbmad.Open;
  tbMad.IndexName:='Remitente';
  tbMad.First;
  edprov.SetFocus;
end;

procedure TForm10.BitBtn1Click(Sender: TObject);
begin
  edprov.Clear;
  tbmad.Filtered:=false;
  form10.Close;
end;

procedure TForm10.SpeedButton2Click(Sender: TObject);
begin
  tbmad.Filtered:=false;
  edprov.Clear;
  edprov.SetFocus;
end;

procedure TForm10.SpeedButton1Click(Sender: TObject);
begin
  tbMad.filter:=(remitente =''+QuotedStr(Edprov.Text)+')';
  tbMad.Filtered:=true;
end;

procedure TForm10.BitBtn2Click(Sender: TObject);
begin
  form11.QrResProv.Preview;
end;
```



```

procedure TForm10.SpeedButton3Click(Sender: TObject);
var fechal, fecha2:tdatetime;
begin
  fechal:= dtfecha1.DateTime;
  fecha2:= dtfecha2.DateTime;
  tbMad.filter:='((Fecha >=' + QuotedStr(DateToStr(Fechal)) + ') and (Fecha <=' +
  QuotedStr(DateToStr(Fecha2)) + ')';
  tbMad.Filtered:=true;
end;

procedure TForm10.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
    or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;
end.

unit Unit11;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, QRCtrls, QuickRpt, ExtCtrls, xpman;

type
  TForm11 = class(TForm)
    QrResProv: TQuickRep;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    QRSubDetail1: TQRSubDetail;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRLabel10: TQRLabel;
    QRLabel11: TQRLabel;
  end;

```



```
QRLLabel12: TQRLLabel;
QRLLabel13: TQRLLabel;
QRLLabel14: TQRLLabel;
QRLLabel15: TQRLLabel;
QRLLabel16: TQRLLabel;
QRLLabel17: TQRLLabel;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
QRDBText5: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRDBText8: TQRDBText;
QRExpr1: TQRExpr;
QRLLabel18: TQRLLabel;
QRDBText9: TQRDBText;
SummaryBand1: TQRBand;
QRLLabel19: TQRLLabel;
QRLLabel20: TQRLLabel;
QRExpr2: TQRExpr;
QRExpr3: TQRExpr;
procedure QrResProvBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form11: TForm11;

implementation

uses Unit10;

{$R *.dfm}

procedure TForm11.QrResProvBeforePrint(Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
  qrlabel3.Caption:=form10.EdProv.Text;
  qrlabel16.Caption:=datetostr(form10.DtFecha1.DateTime);
  qrlabel17.Caption:=datetostr(form10.DtFecha2.DateTime);
end;

end.
```



```
unit Unit12;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, ExtCtrls, StdCtrls, Buttons, xpman;

type
  TForm12 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    EdCuenta: TEdit;
    EdProv: TEdit;
    EdDir: TEdit;
    EdCd: TEdit;
    EdTel: TEdit;
    BtCancel: TBitBtn;
    BtGuar: TBitBtn;
    Timer1: TTimer;
    TbProv: TTable;
    TbProvCUENTA: TStringField;
    TbProvPROVEEDOR: TStringField;
    TbProvDIRECCION: TStringField;
    TbProvCIUDAD: TStringField;
    TbProvTEL: TStringField;
    procedure EdCuentaKeyPress(Sender: TObject; var Key: Char);
    procedure EdProvKeyPress(Sender: TObject; var Key: Char);
    procedure EdCuentaExit(Sender: TObject);
    procedure EdDirKeyPress(Sender: TObject; var Key: Char);
    procedure EdCdKeyPress(Sender: TObject; var Key: Char);
    procedure EdTelKeyPress(Sender: TObject; var Key: Char);
    procedure Timer1Timer(Sender: TObject);
    procedure BtCancelClick(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure BtGuarClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form12: TForm12;
```

implementation



```
{$R *.dfm}

procedure TForm12.EdCuentaKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8,'.'])) then
  key:=#0;
end;

procedure TForm12.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
With Sender as TEdit do
  if (Text = "")or (Text[SelStart] = ' ')
    or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;

procedure TForm12.EdCuentaExit(Sender: TObject);
begin
TbProv.Open;
if tbProv.FindKey([EdCuenta.Text]) and
  not(Trim(EdCuenta.Text)!="")then
begin
  ShowMessage('Cuenta ya registrada!!');
  EdCuenta.Text:="";
  EdCuenta.SetFocus;
end
else
begin
  EdProv.SetFocus;
end;
end;

procedure TForm12.EdDirKeyPress(Sender: TObject; var Key: Char);
begin
With Sender as TEdit do
  if (Text = "")or (Text[SelStart] = ' ')
    or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;

procedure TForm12.EdCdKeyPress(Sender: TObject; var Key: Char);
begin
With Sender as TEdit do
  if (Text = "")or (Text[SelStart] = ' ')
```



```

or (SelLength = Length(Text)) then
if Key in ['a'..'z'] then
  Key:= UpCase(Key);
if (not(key in['a'..'z','A'..'Z','#8,'0'..'9','.'])) then
  key:=#0;
end;

procedure TForm12.EdTelKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9','#8])) then
  key:=#0;
end;

procedure TForm12.Timer1Timer(Sender: TObject);
begin
if (trim(EdCuenta.Text)<>"")and(trim(edprov.Text)<>"")and(trim(eddir.Text)<>")
  and (trim(edcd.Text)<>)and(trim(Edtel.Text)<>)then
begin
BtGuar.Enabled:=true;
end
ELSE
begin
btGuar.Enabled:= false ;
end;
end;

procedure TForm12.BtCancelClick(Sender: TObject);
begin
form12.Close;
edcuenta.Clear;
edprov.Clear;
edcd.Clear;
edtel.Clear;
end;

procedure TForm12.FormActivate(Sender: TObject);
begin
BtGuar.Enabled:=False;
EdCuenta.Text:="";
EdProv.Text:="";
Eddir.Text:="";
Edcd.Text:="";
Edtel.Text:="";
EdCuenta.SetFocus;
end;

procedure TForm12.BtGuarClick(Sender: TObject);
begin
tbProv.Append;
tbProvCuentaAsString:= EdCuenta.Text;

```



```

tbProvProveedor.AsString:=EdProv.Text;
tbProvDireccion.AsString:=Eddir.Text;
tbProvCiudad.AsString:=Edcd.Text;
tbProvTel.AsString:=Edtel.Text;

tbProv.Post;
EdCuenta.Text:="";
EdProv.Text:="";
EdDir.Text:="";
EdCd.Text:="";
EdTel.Text:="";
EdProv.SetFocus;
end;

end.

unit Unit13;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, StdCtrls, Buttons, ExtCtrls,xpman;

type
  TForm13 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Bevel1: TBevel;
    BtBus: TSpeedButton;
    EdCuenta: TEdit;
    EdProv: TEdit;
    EdDir: TEdit;
    EdCd: TEdit;
    EdTel: TEdit;
    BtCancel: TBitBtn;
    BtElim: TBitBtn;
    TbProv: TTable;
    TbProvCUENTA: TStringField;
    TbProvPROVEEDOR: TStringField;
    TbProvDIRECCION: TStringField;
    TbProvCIUDAD: TStringField;
    TbProvTEL: TStringField;
    DsProv: TDataSource;
    procedure EdCuentaKeyPress(Sender: TObject; var Key: Char);
    procedure EdProvKeyPress(Sender: TObject; var Key: Char);
  end;

```



```

procedure EdDirKeyPress(Sender: TObject; var Key: Char);
procedure EdCdKeyPress(Sender: TObject; var Key: Char);
procedure EdTelKeyPress(Sender: TObject; var Key: Char);
procedure BtElimClick(Sender: TObject);
procedure BtBusClick(Sender: TObject);
procedure EdCuentaKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure BtCancelClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form13: TForm13;

implementation

uses Unit14;

{$R *.dfm}

procedure TForm13.EdCuentaKeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9','#8,'.'])) then
    key:=#0;
end;

procedure TForm13.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
end;

procedure TForm13.EdDirKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
end;

```



```
procedure TForm13.EdCdKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
      if (not(key in['a'..'z','A'..'Z',' ',#8,'0'..'9',''])) then
        key:=#0;
      end;
end;

procedure TForm13.EdTelKeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9',#8])) then
    key:=#0;
  end;

procedure TForm13.BtElimClick(Sender: TObject);
begin
  tbProv.Delete;
  btelim.Enabled:=false;
  EdCuenta.Text:="";
  EdProv.Text:="";
  EdDir.Text:="";
  EdCd.Text:="";
  EdTel.Text:="";
  EdCuenta.SetFocus;
end;

procedure TForm13.BtBusClick(Sender: TObject);
begin
  form14.ShowModal;
end;

procedure TForm13.EdCuentaKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  tbProv.Open;
  if key=vk_return then
    begin
      if tbProv.FindKey([EdCuenta.Text]) and
        not(Trim(EdCuenta.Text)!="")then
        begin
          EdProv.Text:= TbprovProveedor.asstring;
          EdDir.Text:= TbProvDireccion.AsString;
          EdCd.Text:= TbprovCiudadAsString;
          EdTel.Text:=TbProvTelAsString;
          btElim.Enabled:=True;
          BtElim.setFocus;
        end;
    end;
end;
```



```
end
else
begin
  ShowMessage('Cuenta no encontrada!!');
  EdCuenta.Text:="";
  EdCuenta.SetFocus;
end;
end;
end;

procedure TForm13.BtCancelClick(Sender: TObject);
begin
form13.Close;
edcuenta.Clear;
edprov.Clear;
eddir.Clear;
edcd.Clear;
edtel.Clear;
end;
end.

unit Unit14;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, StdCtrls,xpman;

type
TForm14 = class(TForm)
  Label1: TLabel;
  EdProv: TEdit;
  DBGrid1: TDBGrid;
  procedure EdProvChange(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure DBGrid1CellClick(Column: TColumn);
  procedure EdProvKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
Form14: TForm14;

implementation
```



```
uses Unit13;

{$R *.dfm}

procedure TForm14.EdProvChange(Sender: TObject);
begin
  form13.tbProv.SetKey;
  form13.TBProv['proveedor]:=trim(form13.EdCuenta.Text);
  form13.tbProv.GotoNearest;
end;

procedure TForm14.FormActivate(Sender: TObject);
begin
  form13.tbprov.open;
  form13.tbprov.IndexName:='Proveedor';
  form13.tbprov.First;
  form13.edcuenta.SetFocus;
end;

procedure TForm14.DBGrid1CellClick(Column: TColumn);
begin
  form13.EdCuenta.Text:=form13.TbProvCuentaAsString;
  form14.Close;
end;

procedure TForm14.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;
end.

unit Unit15;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, ExtCtrls, StdCtrls, Buttons, xpman;

type
  TForm15 = class(TForm)
    Label1: TLabel;
```



```

Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Bevel1: TBevel;
BtBusc: TSpeedButton;
EdCuenta: TEdit;
EdProv: TEdit;
EdDir: TEdit;
EdCd: TEdit;
EdTel: TEdit;
BtCancel: TBitBtn;
BtModif: TBitBtn;
Timer1: TTimer;
TbProv: TTable;
TbProvCUENTA: TStringField;
TbProvPROVEEDOR: TStringField;
TbProvDIRECCION: TStringField;
TbProvCIUDAD: TStringField;
TbProvTEL: TStringField;
DsProv: TDataSource;
procedure EdCuentaKeyPress(Sender: TObject; var Key: Char);
procedure EdProvKeyPress(Sender: TObject; var Key: Char);
procedure EdDirKeyPress(Sender: TObject; var Key: Char);
procedure EdCdKeyPress(Sender: TObject; var Key: Char);
procedure EdTelKeyPress(Sender: TObject; var Key: Char);
procedure BtBuscClick(Sender: TObject);
procedure BtModifClick(Sender: TObject);
procedure BtCancelClick(Sender: TObject);
procedure EdCuentaKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form15: TForm15;

implementation

uses Unit16;

{$R *.dfm}

procedure TForm15.EdCuentaKeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9','#8,'.'])) then
    key:=#0;

```



end;

```
procedure TForm15.EdProvKeyPress(Sender: TObject; var Key: Char);  
begin
```

```
  With Sender as TEdit do
```

```
    if (Text = "")or (Text[SelStart] = ' ')  
      or (SelLength = Length(Text)) then
```

```
    if Key in ['a'..'z'] then
```

```
      Key:= UpCase(Key);
```

```
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','!'])) then
```

```
      key:=#0;
```

```
  end;
```

```
procedure TForm15.EdDirKeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
  With Sender as TEdit do
```

```
    if (Text = "")or (Text[SelStart] = ' ')  
      or (SelLength = Length(Text)) then
```

```
    if Key in ['a'..'z'] then
```

```
      Key:= UpCase(Key);
```

```
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','!'])) then
```

```
      key:=#0;
```

```
  end;
```

```
procedure TForm15.EdCdKeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
  With Sender as TEdit do
```

```
    if (Text = "")or (Text[SelStart] = ' ')  
      or (SelLength = Length(Text)) then
```

```
    if Key in ['a'..'z'] then
```

```
      Key:= UpCase(Key);
```

```
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','!'])) then
```

```
      key:=#0;
```

```
  end;
```

```
procedure TForm15.EdTelKeyPress(Sender: TObject; var Key: Char);
```

```
begin
```

```
  With Sender as TEdit do
```

```
    if (Text = "")or (Text[SelStart] = ' ')  
      or (SelLength = Length(Text)) then
```

```
    if Key in ['a'..'z'] then
```

```
      Key:= UpCase(Key);
```

```
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','!'])) then
```

```
      key:=#0;
```

```
  end;
```

```
procedure TForm15.BtBuscClick(Sender: TObject);
```

```
begin
```

```
  form16.ShowModal;
```

```
end;
```



```
procedure TForm15.BtModifClick(Sender: TObject);
begin
TbProv.edit;
  TbprovProveedor.asstring :=EdProv.Text;
  TbProvdireccion.AsString :=Eddir.Text;
  Tbprovciudad.AsString:= Edcd.Text;
  TbProvtel.AsString :=Edtel.Text;
  tbProv.Post;
  btmodif.Enabled:=FALSE;

  EdCuenta.Text:="";
  EdProv.text:="";
  Eddir.Text:="";
  Edcd.Text:="";
  Edtel.Text:="";
  EdCuenta.SetFocus;
end;

procedure TForm15.BtCancelClick(Sender: TObject);
begin
form15.Close;
EdCuenta.Text:="";
  EdProv.text:="";
  Eddir.Text:="";
  Edcd.Text:="";
  Edtel.Text:="";
end;

procedure TForm15.EdCuentaKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
tbProv.Open;
if key=vk_return then
begin
  if tbProv.FindKey([EdCuenta.Text]) and
    not(Trim(EdCuenta.Text)!="")then
  begin
    EdProv.Text:= TbprovProveedor.asstring;
    Eddir.Text:= TbProvdireccion.AsString;
    Edcd.Text:= Tbprovciudad.AsString;
    Edtel.Text:=TbProvtel.AsString;
    EdProv.Enabled:=true;
    Eddir.Enabled:=true;
    Edcd.Enabled:=true;
    Edtel.Enabled:=true;
    btmodif.Enabled:=true;
    EdProv.setFocus;
  end
  else

```



```
begin
  ShowMessage('Cuenta no encontrada!!');
  EdCuenta.Text:="";
  EdCuenta.SetFocus;
end ;
end;
end;

end.

unit Unit16;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, StdCtrls,xpman;

type
  TForm16 = class(TForm)
    Label1: TLabel;
    EdProv: TEdit;
    DBGrid1: TDBGrid;
    procedure EdProvChange(Sender: TObject);
    procedure EdProvKeyPress(Sender: TObject; var Key: Char);
    procedure FormActivate(Sender: TObject);
    procedure DBGrid1CellClick(Column: TColumn);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form16: TForm16;

implementation

uses Unit15;

{$R *.dfm}

procedure TForm16.EdProvChange(Sender: TObject);
begin
  form15.tbProv.SetKey;
  form15.TBProv['proveedor]:=trim(form15.EdCuenta.Text);
  form15.tbProv.GotoNearest;
end;
```



```

procedure TForm16.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;

procedure TForm16.FormActivate(Sender: TObject);
begin
  form15.tbprov.open;
  form15.tbprov.IndexName:='Proveedor';
  form15.tbprov.First;
  form15.edcuenta.SetFocus;
end;

procedure TForm16.DBGrid1CellClick(Column: TColumn);
begin
  form15.EdCuenta.Text:=form15.TbProvCuenta.AsString;
  form16.Close;
end;

unit Unit17;
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Grids, DBGrids, DB, DBTables,xpman;

type
  TForm17 = class(TForm)
    Label1: TLabel;
    EdProv: TEdit;
    DBGrid1: TDBGrid;
    BtCancel: TBitBtn;
    BitBtn1: TBitBtn;
    TbProv: TTable;
    TbProvCUENTA: TStringField;
    TbProvPROVEEDOR: TStringField;
    TbProvDIRECCION: TStringField;
    TbProvCIUDAD: TStringField;
    TbProvTEL: TStringField;
    dsProv: TDataSource;
    procedure BtCancelClick(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  end;

```



```
procedure EdProvChange(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure EdProvKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form17: TForm17;
implementation
uses Unit18;
{$R *.dfm}

procedure TForm17.BtCancelClick(Sender: TObject);
begin
  form17.Close;
end;

procedure TForm17.FormActivate(Sender: TObject);
begin
  tbprov.open;
  tbprov.IndexName:='Proveedor';
  tbprov.First;
  edprov.SetFocus;
end;

procedure TForm17.EdProvChange(Sender: TObject);
begin
  tbProv.SetKey;
  TBProv['proveedor]:=trim(Edprov.Text);
  tbProv.GotoNearest;
end;

procedure TForm17.BitBtn1Click(Sender: TObject);
begin
  form18.QuickRep1.Preview;
end;

procedure TForm17.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z',' ',#8,'0'..'9'])) then
      key:=#0;
end;
```



```
end.  
unit Unit18;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, ExtCtrls, QuickRpt, QRCtrls, DB, DBTables, xpman;  
  
type  
  TForm18 = class(TForm)  
    QuickRep1: TQuickRep;  
    DetailBand1: TQRBand;  
    TitleBand1: TQRBand;  
    QRSubDetail1: TQRSubDetail;  
    QRLabel1: TQRLabel;  
    QRLabel2: TQRLabel;  
    QRLabel3: TQRLabel;  
    QRLabel4: TQRLabel;  
    QRLabel5: TQRLabel;  
    QRLabel6: TQRLabel;  
    QRDBText1: TQRDBText;  
    QRDBText2: TQRDBText;  
    QRDBText3: TQRDBText;  
    QRDBText4: TQRDBText;  
    QRDBText5: TQRDBText;  
    Table1: TTable;  
    Table1CUENTA: TStringField;  
    Table1PROVEEDOR: TStringField;  
    Table1DIRECCION: TStringField;  
    Table1CIUDAD: TStringField;  
    Table1TEL: TStringField;  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form18: TForm18;  
  
implementation  
  
uses Unit17;  
  
{$R *.dfm}  
  
end.
```



```

unit Unit19;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, ExtCtrls, StdCtrls, Buttons, xpman;

type
  TForm19 = class(TForm)
    Label1: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label2: TLabel;
    EdNomb: TEdit;
    EdDir: TEdit;
    EdCd: TEdit;
    BtCancel: TBitBtn;
    BtGuardar: TBitBtn;
    Timer1: TTimer;
    TbClient: TTable;
    TbClientCLAVE: TStringField;
    TbClientCLIENTE: TStringField;
    TbClientDIRECCION: TStringField;
    TbClientCIUDAD: TStringField;
    procedure EdNombKeyPress(Sender: TObject; var Key: Char);
    procedure EdDirKeyPress(Sender: TObject; var Key: Char);
    procedure EdCdKeyPress(Sender: TObject; var Key: Char);
    procedure FormActivate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure BtCancelClick(Sender: TObject);
    procedure BtGuardarClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form19: TForm19;
  clave:integer;

implementation

{$R *.dfm}

procedure TForm19.EdNombKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do

```



```

if (Text = "")or (Text[SelStart] = ' ')
  or (SelLength = Length(Text)) then
if Key in ['a'..'z'] then
  Key:= UpCase(Key);
if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9'])) then
  key:=#0;
end;

procedure TForm19.EdDirKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9'])) then
      key:=#0;
    end;

procedure TForm19.EdCdKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
    if Key in ['a'..'z'] then
      Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9'])) then
      key:=#0;
    end;

procedure TForm19.FormActivate(Sender: TObject);
begin
  tbClient.Open;
  clave:=strToInt(tbclientclave.AsString);
  label2.Caption:=inttostr(clave+1);
  EdNomb.Text:="";
  EdDir.Text:="";
  EdCd.Text:="";
end;

procedure TForm19.Timer1Timer(Sender: TObject);
begin
  if (trim(Ednomb.Text)<>"")and(trim(edDir.Text)<>"")and(trim(edCd.Text)<>"")then
    begin
      BtGuardar.Enabled:=true;
    end
  ELSE
    begin
      btGuardar.Enabled:= false ;
    end;
end;

```



```
end;

procedure TForm19.BtCancelarClick(Sender: TObject);
begin
form19.Close;
ednomb.Clear;
edcd.Clear;
eddir.Clear;
end;

procedure TForm19.BtGuardarClick(Sender: TObject);
begin
tbClient.Append;
tbClientClaveAsString:=label2.Caption;
tbClientClienteAsString:=EdNomb.Text;
tbClientDireccionAsString:=EdDir.Text;
tbClientCiudadAsString:=EdCd.Text;

tbClient.Post;
EdNomb.Text:="";
EdDir.Text:="";
EdCd.Text:="";
EdNomb.SetFocus;
clave:=strToInt(tbclientclaveAsString);
label2.Caption:=IntToStr(clave+1);
end;

end.

unit Unit20;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, ExtCtrls, StdCtrls, Buttons, xpman;

type
TForm20 = class(TForm)
  Bevel1: TBevel;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  EdNomb: TEdit;
  EdDir: TEdit;
  EdCd: TEdit;
  BtCancelar: TBitBtn;
```



```

BtCambiar: TBitBtn;
Timer1: TTimer;
TbClient: TTable;
TbClientCLAVE: TStringField;
TbClientCLIENTE: TStringField;
TbClientDIRECCION: TStringField;
TbClientCIUDAD: TStringField;
procedure EdNombKeyPress(Sender: TObject; var Key: Char);
procedure EdDirKeyPress(Sender: TObject; var Key: Char);
procedure EdCdKeyPress(Sender: TObject; var Key: Char);
procedure BtCancelClick(Sender: TObject);
procedure BtCambiarClick(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure EdNombKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form20: TForm20;

implementation

{$R *.dfm}

procedure TForm20.EdNombKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','#8,'0'..'9','.'])) then
      key:=#0;
  end;

procedure TForm20.EdDirKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','#8])) then
      key:=#0;
  end;

```



```
procedure TForm20.EdCdKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','#8])) then
      key:=#0;
  end;

procedure TForm20.BtCancelClick(Sender: TObject);
begin
  form20.Close;
  ednomb.Clear;
  edcd.Clear;
  eddir.Clear;
end;

procedure TForm20.BtCambiarClick(Sender: TObject);
begin
  TbClient.edit;
  TbClientCliente.asstring :=EdNomb.Text;
  TbClientClave.AsString :=Label2.Caption;
  TbClientDireccion.AsString:= Eddir.Text;
  TbClientCiudad.AsString :=EdCd.Text;
  tbClient.Post;
  btCambiar.Enabled:=FALSE;

  EdNomb.Text:="";
  Label2.Caption:="";
  EDDir.Text:="";
  EdCd.Text:="";
  EdNomb.SetFocus;
end;

procedure TForm20.Timer1Timer(Sender: TObject);
begin
  if (trim(Ednomb.Text)<>"")and(trim(edDir.Text)<>"")and(trim(edCd.Text)<>"")then
  begin
    BtCambiar.Enabled:=true;
  end
  ELSE
  begin
    btCambiar.Enabled:= false ;
  end;
end;

procedure TForm20.EdNombKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
```



```

begin
  tbClient.Open;
  if key= vk_return then
  begin
    if tbClient.FindKey([Ednomb.Text]) and
      not(Trim(Ednomb.Text)!="")then
    begin
      Label2.Caption:= TbClientClave.AsString;
      EdDir.Text:= TbClientDireccion.AsString;
      EdCd.Text:= TbClientCiudad.AsString;
      EdNomb.setFocus;
      EdDir.Enabled:=True;
      EdCd.Enabled:=True;
      btcambiar.Enabled:=true;
    end
    else
    begin
      ShowMessage('Cliente no encontrado!!');
      Ednomb.Text="";
      Ednomb.SetFocus;
    end ;
  end;
end;
end;

end.

```

unit Unit21;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, StdCtrls, Buttons, ExtCtrls, xpman;

type

```

TForm21 = class(TForm)
  Bevel1: TBevel;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  EdNomb: TEdit;
  EdDir: TEdit;
  EdCd: TEdit;
  BtCancelar: TBitBtn;
  BtBorrar: TBitBtn;
  TbClient: TTable;
  TbClientCLAVE: TStringField;

```



```

TbClientCLIENTE: TStringField;
TbClientDIRECCION: TStringField;
TbClientCIUDAD: TStringField;
procedure EdNombKeyPress(Sender: TObject; var Key: Char);
procedure BtCancelClick(Sender: TObject);
procedure BtBorrarClick(Sender: TObject);
procedure EdNombKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure EdDirKeyPress(Sender: TObject; var Key: Char);
procedure EdCdKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form21: TForm21;

implementation

{$R *.dfm}

procedure TForm21.EdNombKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;

procedure TForm21.BtCancelClick(Sender: TObject);
begin
  form21.Close;
  ednomb.Clear;
  edcd.Clear;
  eddir.Clear;
end;

procedure TForm21.BtBorrarClick(Sender: TObject);
begin
  tbClient.Delete;
  btBorrar.Enabled:=false;
  EdNomb.Text:="";
  EdDir.Text:="";
  EdCd.Text:="";
  EdNomb.SetFocus;

```



end;

```
procedure TForm21.EdNombKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  tbClient.Open;
  if key=vk_return then
  begin
    if tbClient.FindKey([Ednomb.Text]) and
      not(Trim(Ednomb.Text)="")then
      begin
        Label2.Caption:= TbClientClave.AsString;
        EdDir.Text:= TbClientDireccion.AsString;
        EdCd.Text:= TbClientCiudad.AsString;
        EdNomb.setfocus;
        btBorrar.Enabled:=true;
      end
    else
      begin
        ShowMessage('Cliente no encontrado!!');
        Ednomb.Text:="";
        Ednomb.SetFocus;
      end ;
  end;
end;
```

```
procedure TForm21.EdDirKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
end;
```

```
procedure TForm21.EdCdKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8])) then
      key:=#0;
end;
```

end.



```

unit Unit22;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, StdCtrls, Buttons, Grids, DBGrids, xpman;
type
  TForm22 = class(TForm)
    Label1: TLabel;
    DBGrid1: TDBGrid;
    Edit1: TEdit;
    BtCerrar: TBitBtn;
    BtImp: TBitBtn;
    TbClient: TTable;
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Edit1Change(Sender: TObject);
    procedure BtCerrarClick(Sender: TObject);
    procedure BtImpClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form22: TForm22;
implementation
uses Unit23;
{$R *.dfm}

procedure TForm22.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;

procedure TForm22.Edit1Change(Sender: TObject);
begin
  tbclient.SetKey;
  TBclient['cliente']:=trim(Edit1.Text);
  tbclient.GotoNearest;
end;
procedure TForm22.BtCerrarClick(Sender: TObject);
begin
  form22.Close;
  edit1.Clear;

```



```
end;
procedure TForm22.BtImpClick(Sender: TObject);
begin
form23.QuickRep1.Preview;
end;

end.

unit Unit23;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, QuickRpt, QRCtrls, DB, DBTables,xpman;

type
  TForm23 = class(TForm)
    QuickRep1: TQuickRep;
    DetailBand1: TQRBand;
    TitleBand1: TQRBand;
    QRSubDetail1: TQRSubDetail;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    Table1: TTable;
    Table1CLAVE: TStringField;
    Table1CLIENTE: TStringField;
    Table1DIRECCION: TStringField;
    Table1CIUDAD: TStringField;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form23: TForm23;

implementation
uses Unit22;
{$R *.dfm}
end.
```



```

unit Unit24;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, DB, Grids, DBGrids, DBTables;

type
  TForm24 = class(TForm)
    TbProv: TTable;
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Label1: TLabel;
    EdProv: TEdit;
    TbProvCUENTA: TStringField;
    TbProvPROVEEDOR: TStringField;
    TbProvDIRECCION: TStringField;
    TbProvCIUDAD: TStringField;
    TbProvTEL: TStringField;
    BitBtn1: TBitBtn;
    procedure EdProvKeyPress(Sender: TObject; var Key: Char);
    procedure EdProvChange(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form24: TForm24;
implementation
{$R *.dfm}
procedure TForm24.EdProvKeyPress(Sender: TObject; var Key: Char);
begin
  With Sender as TEdit do
    if (Text = "")or (Text[SelStart] = ' ')
      or (SelLength = Length(Text)) then
      if Key in ['a'..'z'] then
        Key:= UpCase(Key);
    if (not(key in['a'..'z','A'..'Z','','#8,'0'..'9','.'])) then
      key:=#0;
  end;

procedure TForm24.EdProvChange(Sender: TObject);
begin
  tbProv.SetKey;
  TBProv['proveedor]:=trim(Edprov.Text);
  tbProv.GotoNearest;
end;

```



```
procedure TForm24.BitBtn1Click(Sender: TObject);
begin
form24.Close;
end;
end.

unit Unit25;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, ExtCtrls, DB, DBTables;

type
TForm25 = class(TForm)
TbPino: TTable;
TbPinoCLAVE: TStringField;
TbPinoMADERA: TStringField;
TbPinoTIPO: TStringField;
TbPinoGRUESO: TFloatField;
TbPinoLARGO: TFloatField;
TbPinoANCHO: TFloatField;
TbPinoCANTIDAD: TFloatField;
Ppin: TPanel;
PRad: TPanel;
GroupBox1: TGroupBox;
RadioButton1: TRadioButton;
RadioButton2: TRadioButton;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
CbTipp: TComboBox;
CbAncp: TComboBox;
CbLarp: TComboBox;
Label7: TLabel;
CbTipr: TComboBox;
CbAncr: TComboBox;
CbLarr: TComboBox;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
BtPino: TBitBtn;
BitBtn2: TBitBtn;
BtRad: TBitBtn;
TbConta: TTable;
```



```
TbContaCONTADOR: TSmallintField;
Timer1: TTimer;
Timer2: TTimer;
EdDos: TEdit;
Label11: TLabel;
Eduno: TEdit;
Edit1: TEdit;
Edit2: TEdit;
Label12: TLabel;
Edit3: TEdit;
procedure FormActivate(Sender: TObject);
procedure RadioButton1Click(Sender: TObject);
procedure RadioButton2Click(Sender: TObject);
procedure BtPinoClick(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure BtRadClick(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure EdunoKeyPress(Sender: TObject; var Key: Char);
procedure EdDosKeyPress(Sender: TObject; var Key: Char);
procedure EdDosExit(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form25: TForm25;

implementation

{$R *.dfm}

procedure TForm25.FormActivate(Sender: TObject);
begin
  tbconta.Open;
  ppin.Visible:=false;
  prad.Visible:=false;
  label2.Caption:=inttostr(tbcontacontador.AsInteger);
  btpino.Visible:=false;
  btrad.Visible:=false;
  btpino.Enabled:=false;
  btrad.Enabled:=false;
end;

procedure TForm25.RadioButton1Click(Sender: TObject);
begin
  ppin.Visible:=true;
  prad.Visible:=false;
```



```
btpino.Visible:=true;  
end;  
  
procedure TForm25.RadioButton2Click(Sender: TObject);  
begin  
prad.Visible:=true;  
ppin.Visible:=false;  
btrad.Visible:=true;  
end;  
  
procedure TForm25.BtPinoClick(Sender: TObject);  
var conta:integer;  
begin  
tbpino.Open;  
tbpino.Append;  
tbpinoclaveAsString:=label2.Caption;  
tbpinomaderaAsString:='Pino Nacional';  
tbpinotipoAsString:=cbtipp.Text;  
tbpinogrueso.AsFloat:=strtofloat(edit1.Text);  
tbpinolargoAsString:=cblarp.Text;  
tbpinoanchoAsString:=cbancp.Text;  
tbpino.Post;  
conta:=strToInt(label2.Caption);  
conta:=conta+1;  
tbconta.Open;  
tbconta.Edit;  
tbcontacontador.AsInteger:=conta;  
tbconta.Post;  
label2.Caption:=tbcontacontadorAsString;  
ppin.Visible:=false;  
btpino.Enabled:=false;  
btpino.Visible:=false;  
cbtipp.Text:='';  
edit1.Clear;  
edit2.Clear;  
edit3.Clear;  
cbancp.Text:='';  
cblarp.Text:='';  
end;  
  
procedure TForm25.Timer1Timer(Sender: TObject);  
begin  
if (trim(cbtipp.Text)<>'')and(trim(edit1.Text)<>'')and(trim(cblarp.Text)<>'')and  
(trim(cbancp.Text)<>'')then  
begin  
Btpino.Enabled:=true;  
end  
ELSE  
begin
```



```
btpino.Enabled:= false;
end;
end;

procedure TForm25.Timer2Timer(Sender: TObject);
begin
if (trim(cbtipr.Text)<>"")and(trim(edit1.Text)<>"")and(trim(cblarr.Text)<>"")and
(trim(cbancr.Text)<>"")then
begin
Btrad.Enabled:=true;
end
ELSE
begin
btrad.Enabled:= false;
end;
end;

procedure TForm25.BtRadClick(Sender: TObject);
var conta:integer;
begin
tbpino.Open;
tbpino.Append;
tbpinoclaveAsString:=label2.Caption;
tbpinomaderaAsString:='Radiata';
tbpinotipoAsString:=cbtipr.Text;
tbpinogrueso.AsFloat:=strtofloat(edit1.Text);
tbpinolargoAsString:=cblarr.Text;
tbpinoanchoAsString:=cbancr.Text;
tbpino.Post;
conta:=strtoint(label2.Caption);
conta:=conta+1;
tbconta.Open;
tbconta.Edit;
tbcontacontador.AsInteger:=conta;
tbconta.Post;
label2.Caption:=tbcontacontadorAsString;
prad.Visible:=false;
btrad.Enabled:=false;
btrad.Visible:=false;
cbtipr.Text:="";
edit1.Text:="";
eduno.Clear;
eddos.Clear;
cbancr.Text:="";
cblarr.Text:="";

end;

procedure TForm25.BitBtn2Click(Sender: TObject);
begin
```



```
form25.Close;
end;

procedure TForm25.EdunoKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm25.EdDosKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9',#8])) then
  key:=#0;
end;

procedure TForm25.EdDosExit(Sender: TObject);
var grues:extended;
begin
grues:=(strtoint(eduno.Text))/(strtoint(eddos.Text));
edit1.Text:=floattostr(grues);
end;

end.

unit Unit26;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons, DB, DBTables;

type
  TForm26 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    BtPino: TBitBtn;
    Panel1: TPanel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    CbMad: TComboBox;
    Cbtipp: TComboBox;
    CbAnc: TComboBox;
    CbLarg: TComboBox;
    CbTipr: TComboBox;
  end;
```



```

TbPino: TTable;
TbPinoCLAVE: TStringField;
TbPinoMADERA: TStringField;
TbPinoTIPO: TStringField;
TbPinoGRUESO: TFloatField;
TbPinoLARGO: TFloatField;
TbPinoANCHO: TFloatField;
TbPinoCANTIDAD: TFloatField;
BtCan: TBitBtn;
SpeedButton1: TSpeedButton;
Edit2: TEdit;
Eduno: TEdit;
Label11: TLabel;
EdDos: TEdit;
procedure Btcanclick(Sender: TObject);
procedure CbMadExit(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure BtPinoClick(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Edit1KeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure SpeedButton1Click(Sender: TObject);
procedure EdDosExit(Sender: TObject);
procedure EdunoKeyPress(Sender: TObject; var Key: Char);
procedure EdDosKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form26: TForm26;

implementation

uses Unit28;

{$R *.dfm}

procedure TForm26.CbMadExit(Sender: TObject);
begin
if cbmad.Text='Pino Nacional' then
begin
  cbtipp.Visible:=true;
  cbtipr.Visible:=false;
  btpino.Visible:=true;
end
else
begin

```



```
cbtipr.Visible:=true;
cbtipp.Visible:=false;
end;
end;

procedure TForm26.FormActivate(Sender: TObject);
begin
panel1.Visible:=false;
btptime.Visible:=false;
cbtipp.Visible:=false;
cbtipr.Visible:=false;
edit2.Visible:=false;

end;

procedure TForm26.BtPinoClick(Sender: TObject);
begin
tbptime.Edit;
tbpinomaderaAsString:=cbmad.Text;
if cbtipp.Visible=true then
  tbpinotipoAsString:=cbtipp.Text
else
  tbpinotipoAsString:=cbtipr.Text;
tbpinogrueso.AsFloat:=strtofloat(edit2.Text);
tbpinolargoAsString:=cblarg.Text;
tbpinoanchoAsString:=cbanc.Text;
tbptime.Post;
panel1.Visible:=false;
btptime.Enabled:=false;
btptime.Visible:=false;
cbtipr.Text:="";
edit2.Text:="";
edit1.Clear;
edit2.Visible:=false;
eduno.Clear;
eddos.Clear;
cbanc.Text:="";
cblarg.Text:="";
cbmad.Text:="";
cbtipp.Text:="";
cbtipr.Visible:=false;
cbtipp.Visible:=false;
end;

procedure TForm26.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9','#8'])) then
  key:=#0;
end;
```



```
procedure TForm26.Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  tbpino.Open;
  if key=vk_return then
    begin
      if tbpino.FindKey([Edit1.Text]) and
        not(Trim(Edit1.Text)='')then
        begin
          panel1.Visible:=true;
          cbmad.Text:= Tbpinomadera.asstring;
          if cbmad.Text='Pino Nacional'then
            begin
              cbtipp.Visible:=true;
              edit2.Visible:=true;
              cbtipp.Text:=tbpinotipo.AsString;
            end
          else
            begin
              cbtipr.Visible:=true;
              cbtipr.Text:=tbpinotipo.AsString;
            end;
          edit2.Text:= floattostr( Tbpinogrueso.AsFloat);
          cblarg.Text:=Tbpinolargo.AsString;
          cbanc.Text:=tbpinoancho.AsString;
          btpino.Visible:=true;
          btpino.Enabled:=true;
        end
      else
        begin
          ShowMessage('Clave no existe!!');
          Edit1.Text:="";
          Edit1.SetFocus;
        end;
    end;
end;
procedure TForm26.BtCanClick(Sender: TObject);
begin
  form26.Close;
  panel1.Visible:=false;
  btpino.Enabled:=false;
  btpino.Visible:=false;
  cbtipr.Text:="";
  edit1.Text:="";
  eduno.Clear;
  eddos.Clear;
  cbanc.Text:="";
  cblarg.Text:="";
  cbmad.Text:="";
  cbtipp.Text:="";
```



```
cbtipr.Visible:=false;
cbtipp.Visible:=false;
edit2.Clear;
edit2.Visible:=false;
end;

procedure TForm26.SpeedButton1Click(Sender: TObject);
begin
form28.ShowModal;
end;

procedure TForm26.EdDosExit(Sender: TObject);
var grues:extended;
begin
grues:=(strtoint(eduno.Text))/(strtoint(eddos.Text));
edit2.Text:=floattostr(grues);
end;

procedure TForm26.EdunoKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9','#8'])) then
  key:=#0;
end;

procedure TForm26.EdDosKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9','#8'])) then
  key:=#0;
end;
end.

unit Unit27;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, StdCtrls, Buttons, DBTables, Grids, DBGrids;

type
  TForm27 = class(TForm)
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    TbPino: TTable;
    Label1: TLabel;
    cbmad: TComboBox;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
```



```
TbPinoCLAVE: TStringField;
TbPinoMADERA: TStringField;
TbPinoTIPO: TStringField;
TbPinoGRUESO: TFloatField;
TbPinoLARGO: TFloatField;
TbPinoANCHO: TFloatField;
TbPinoCANTIDAD: TFloatField;
Label4: TLabel;
cbtipo: TComboBox;
SpeedButton1: TSpeedButton;
SpeedButton2: TSpeedButton;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form27: TForm27;

implementation

{$R *.dfm}

procedure TForm27.BitBtn1Click(Sender: TObject);
begin
if tbpinocantidad.AsInteger=0 then
  tbpino.Delete
else
begin
  showmessage('No se puede eliminar');
  Cbmad.SetFocus;
end;
end;

procedure TForm27.BitBtn2Click(Sender: TObject);
begin
form27.Close;
end;

procedure TForm27.SpeedButton1Click(Sender: TObject);
begin
tbpino.Filter:='( madera=' + quotedstr(cbmad.Text)+')and ( tipo=' +
quotedstr(cbtipo.Text)+')';
tbpino.Filtered:=true;
end;
```



```
procedure TForm27.SpeedButton2Click(Sender: TObject);
begin
  tbpino.Filtered:=false;
  cbmad.Clear;
  cbtipo.Clear;
end;

end.

unit Unit28;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, DB, DBTables, Grids, DBGrids;

type
  TForm28 = class(TForm)
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    TbPino: TTable;
    Label1: TLabel;
    cbmad: TComboBox;
    TbPinoCLAVE: TStringField;
    TbPinoMADERA: TStringField;
    TbPinoTIPO: TStringField;
    TbPinoGRUESO: TFloatField;
    TbPinoLARGO: TFloatField;
    TbPinoANCHO: TFloatField;
    TbPinoCANTIDAD: TFloatField;
    BitBtn1: TBitBtn;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    cbtipo: TComboBox;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
```



```
Form28: TForm28;
implementation
{$R *.dfm}

procedure TForm28.BitBtn1Click(Sender: TObject);
begin
form28.Close;
label3.Caption:='0000';
tbpino.Filtered:=false;
end;

procedure TForm28.FormActivate(Sender: TObject);
var suma,rango, i:integer;
begin
tbpino.Refresh;
tbpino.Filtered:=false;
suma:=0;
rango:=tbpino.recordcount;
for i:=1 to rango do
begin
suma:=tbpinocantidad.AsInteger+suma;
tbpino.Next;
end;
label3.Caption:=inttostr(suma);
end;

procedure TForm28.SpeedButton1Click(Sender: TObject);
begin
tbpino.Filter:='( madera=' + quotedstr(cbmad.Text)+')and ( tipo=' +
quotedstr(cbtipo.Text)+ ')';
tbpino.Filtered:=true;
end;

procedure TForm28.SpeedButton2Click(Sender: TObject);
begin
tbpino.Filtered:=false;
cbmad.Clear;
cbtipo.Clear;
end;
end.

unit Unit29;
interface
uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```



Dialogs, StdCtrls, Buttons, ExtCtrls, DB, DBTables;

```

type
  TForm29 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    SpeedButton1: TSpeedButton;
    Panel1: TPanel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    EdMad: TEdit;
    EdTip: TEdit;
    EdGru: TEdit;
    EdLarg: TEdit;
    EdCant: TEdit;
    EdAnc: TEdit;
    Timer1: TTimer;
    BtAccept: TBitBtn;
    BitBtn2: TBitBtn;
    TbPino: TTable;
    TbPinoCLAVE: TStringField;
    TbPinoMADERA: TStringField;
    TbPinoTIPO: TStringField;
    TbPinoGRUESO: TFloatField;
    TbPinoLARGO: TFloatField;
    TbPinoANCHO: TFloatField;
    TbPinoCANTIDAD: TFloatField;
    procedure SpeedButton1Click(Sender: TObject);
    procedure Edit1KeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure EdCantKeyPress(Sender: TObject; var Key: Char);
    procedure BtAcceptClick(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form29: TForm29;
implementation

```



```
uses Unit28;
{$R *.dfm}
procedure TForm29.SpeedButton1Click(Sender: TObject);
begin
form28.ShowModal;
end;

procedure TForm29.Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
tbpino.Open;
if key=vk_return then
begin
  if tbpino.FindKey([Edit1.Text]) and
    not(Trim(Edit1.Text)="")then
  begin
    panel1.Visible:=true;
    edmad.Text:= Tbpinomadera.AsString;
    editip.Text:=tbpinotipo.AsString;
    edgru.Text:= Tbpinogrueso.AsString;
    edlarg.Text:=Tbpinolargo.AsString;
    edanc.Text:=tbpinoancho.AsString;
    btAcept.Enabled:=true;
  end
  else
  begin
    ShowMessage('Clave no existe!!');
    Edit1.Text:="";
    Edit1.SetFocus;
  end;
end;
end;

procedure TForm29.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9','#8])) then
  key:=#0;
end;

procedure TForm29.EdCantKeyPress(Sender: TObject; var Key: Char);
begin
if (not(key in['0'..'9','#8])) then
  key:=#0;
end;

procedure TForm29.BtAcceptClick(Sender: TObject);
var cant:integer;
begin
tbpino.Open;
tbpino.Edit;
```



```
cant:=tbpinocantidad.AsInteger;
tbpinocantidad.AsInteger:=(strToInt(edcant.Text))+ cant;
tbpino.Post;
edcant.Clear;
edmad.Clear;
editip.Clear;
edgru.Clear;
edlarg.Clear;
panel1.Visible:=false;
edit1.Clear;
btacept.Enabled:=false;
edit1.SetFocus;
end;

procedure TForm29.BitBtn2Click(Sender: TObject);
begin
form29.Close;
edcant.Clear;
edmad.Clear;
editip.Clear;
edgru.Clear;
edlarg.Clear;
panel1.Visible:=false;
edit1.Clear;
btacept.Enabled:=false;
end;

procedure TForm29.Timer1Timer(Sender: TObject);
begin
if(trim(edcant.Text)<>"")then
begin
Btacept.Enabled:=true;
end
ELSE
begin
btacept.Enabled:= false;
end;
end;

procedure TForm29.FormActivate(Sender: TObject);
begin
edcant.Clear;
edmad.Clear;
editip.Clear;
edgru.Clear;
edlarg.Clear;
panel1.Visible:=false;
edit1.Clear;
btacept.Enabled:=false;
end;
```



```
end.  
unit Unit30;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, DB, DBTables, StdCtrls, Buttons, ExtCtrls;  
  
type  
  TForm30 = class(TForm)  
    Label1: TLabel;  
    Edit1: TEdit;  
    SpeedButton1: TSpeedButton;  
    Panel1: TPanel;  
    Label2: TLabel;  
    Label3: TLabel;  
    Label4: TLabel;  
    Label5: TLabel;  
    Label6: TLabel;  
    Label7: TLabel;  
    EdMad: TEdit;  
    EdTip: TEdit;  
    EdGru: TEdit;  
    EdLarg: TEdit;  
    EdCant: TEdit;  
    EdAnc: TEdit;  
    Timer1: TTimer;  
    BtAccept: TBitBtn;  
    BitBtn2: TBitBtn;  
    TbPino: TTable;  
    TbPinoCLAVE: TStringField;  
    TbPinoMADERA: TStringField;  
    TbPinoTIPO: TStringField;  
    TbPinoGRUESO: TFloatField;  
    TbPinoLARGO: TFloatField;  
    TbPinoANCHO: TFloatField;  
    TbPinoCANTIDAD: TFloatField;  
    Label8: TLabel;  
    procedure SpeedButton1Click(Sender: TObject);  
    procedure Edit1KeyDown(Sender: TObject; var Key: Word;  
      Shift: TShiftState);  
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);  
    procedure BtAcceptClick(Sender: TObject);  
    procedure BitBtn2Click(Sender: TObject);  
    procedure EdCantKeyPress(Sender: TObject; var Key: Char);  
    procedure EdCantExit(Sender: TObject);  
    procedure FormActivate(Sender: TObject);  
  private  
    { Private declarations }
```



```
public
  { Public declarations }
end;

var
  Form30: TForm30;

implementation

uses Unit28;

{$R *.dfm}

procedure TForm30.SpeedButton1Click(Sender: TObject);
begin
  form28.ShowModal;
end;

procedure TForm30.Edit1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  tbpino.Open;
  if key=vk_return then
    begin
      if tbpino.FindKey([Edit1.Text]) and
        not(Trim(Edit1.Text)='')then
        begin
          panel1.Visible:=true;
          edmad.Text:= Tbpinomadera.AsString;
          editip.Text:=tbpinotipo.AsString;
          edgru.Text:= Tbpinogrueso.AsString;
          edlarg.Text:=Tbpinolargo.AsString;
          edanc.Text:=tbpinoancho.AsString;
          label8.Caption:=tbpinocantidad.AsString;
          btAcept.Enabled:=true;
        end
      else
        begin
          ShowMessage('Clave no existe!!');
          Edit1.Text:='';
          Edit1.SetFocus;
        end;
    end;
end;
end;

procedure TForm30.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9','#8'])) then
    key:=#0;
end;
```



```
procedure TForm30.BtAcceptClick(Sender: TObject);
var can:integer;
begin
  tbpino.Open;
  can:=tbpinocantidad.AsInteger;
  can:=can-(strToInt(edcant.Text));
  tbpino.Edit;
  tbpinocantidad.AsInteger:=can;
  tbpino.Post;
  edcant.Clear;
  edmad.Clear;
  edgru.Clear;
  edlarg.Clear;
  edanc.Clear;
  edtip.Clear;
  panel1.Visible:=false;
  edit1.SetFocus;
  btaccept.Enabled:=false;
  label8.Caption:='00';
end;

procedure TForm30.BitBtn2Click(Sender: TObject);
begin
  form30.Close;
  edcant.Clear;
  edmad.Clear;
  edtip.Clear;
  edgru.Clear;
  edlarg.Clear;
  panel1.Visible:=false;
  edit1.Clear;
  btaccept.Enabled:=false;
end;

procedure TForm30.EdCantKeyPress(Sender: TObject; var Key: Char);
begin
  if (not(key in['0'..'9',#8])) then
    key:=#0;
end;

procedure TForm30.EdCantExit(Sender: TObject);
var can:integer;
begin
  can:=strToInt(label8.Caption);
  if can<(strToInt(edcant.Text)) then
  begin
    showmessage('Excede el numero de existencias');
    edcant.Clear;
    edcant.SetFocus;
  end;
end;
```



```
end
else
begin
btaccept.SetFocus;
btaccept.Enabled:=true;
end;
end;

procedure TForm30.FormActivate(Sender: TObject);
begin
btaccept.Enabled:=false;
end;
end.
```

# ANEXO

3



# TABLA DE DICCCIONARIO DE DATOS

## ESTRUCTURA DE LA TABLA DE CLIENTE.

NOMBRE DEL CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO
Clave	Identifica de forma única al cliente	Carácter	10
Cliente	Nombre personal del cliente	Carácter	50
Direccion	Domicilio del cliente.	Carácter	50
Ciudad	Ubicación del cliente	Carácter	15

**NOMBRE** = Clave.

**DESCRIPCIÓN** = Identificador de forma única.

**LONGITUD** = (10)

ALFABETO

**FORMATO DE ENTRADA** X (30)

NUMERICO

**FORMATO DE SALIDA** X (30)

ALFANUMERICO

**NOMBRE** = Cliente

**DESCRIPCIÓN** = Nombre personal del cliente.

**LONGITUD** = (50)

ALFABETO

**FORMATO DE ENTRADA** X (50)

NUMERICO

**FORMATO DE SALIDA** X (50)

ALFANUMERICO

**NOMBRE** = Direccion.

**DESCRIPCIÓN** = Domicilio del cliente.

**LONGITUD** = (50)

ALFABETO

**FORMATO DE ENTRADA** X (50)

NUMERICO

**FORMATO DE SALIDA** X (50)

ALFANUMERICO



**NOMBRE** = Ciudad

**DESCRIPCIÓN** = Ubicación del cliente.

**LONGITUD** = (15)

ALFABETO

**FORMATO DE ENTRADA** X (15)

NUMERICO

**FORMATO DE SALIDA** X (15)

ALFANUMERICO

### ESTRUCTURA DE LA TABLA DEVOLUCION.

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO
Madera	Identifica de la madera	Carácter	14
Tipo	Tipo de madera a registrar.	Numerico	5
Grueso	Grueso de madera a registrar.	Numerico	5
Ancho	Ancho de madera a registrar.	Numerico	5
Cantidad	Cuenta madera a registrar.	Numerico	5
Total	Piesaje de madera a registrar	Numerico	5
Remitente	Proveedor quien envio.	Carácter	45
Consig	A nombre a quien queda la madera	Caracter	35
Direccion	Domicilio del proveedor	Carácter	45
Chofer	Nombre de persona que llevo el transporte	Carácter	30
Camion	Tipo de camion o razon social	Carácter	10
Fecha	El dia actualizado	Caracter	6
Guia	Numero de referencia de la transportacion	Numerico	10
Remftal	Quien envio finalmente la madera	Carácter	20
Recibio	Quien recibio en patio	Caracter	35
Folio	Numero de tarja que se maneja	Numerico	5
Totaltarja	Total de recibido	Numerico	9
Precio	La cantidad a pagar	Numerico	5



**NOMBRE** = Madera

**DESCRIPCIÓN** = Identifica de madera

**LONGITUD** = (14)

ALFABETO

**FORMATO DE ENTRADA** X (14)

NUMERICO

**FORMATO DE SALIDA** X (14)

ALFANUMERICO

**NOMBRE** = Tipo

**DESCRIPCIÓN** = Tipo de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Grueso

**DESCRIPCIÓN** = Grueso de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Ancho

**DESCRIPCIÓN** = Ancho de madera a registrar.

**LONGITUD** = (50)

ALFABETO

**FORMATO DE ENTRADA** X (50)

NUMERICO

**FORMATO DE SALIDA** X (50)

ALFANUMERICO



**NOMBRE** = Cantidad

**DESCRIPCIÓN** = Cuanto de madera se va a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Total

**DESCRIPCIÓN** = Piesaje de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Remitente.

**DESCRIPCIÓN** = Proveedor quien envio

**LONGITUD** = (45)

ALFABETO

**FORMATO DE ENTRADA** X (45)

NUMERICO

**FORMATO DE SALIDA** X (45)

ALFANUMERICO

**NOMBRE** = Consig.

**DESCRIPCIÓN** = A nombre a quien queda la madera

**LONGITUD** = (35)

ALFABETO

**FORMATO DE ENTRADA** X (35)

NUMERICO

**FORMATO DE SALIDA** X (35)

ALFANUMERICO



**NOMBRE** = Direccion

**DESCRIPCIÓN** = Domicilio del proveedor

**LONGITUD** = (45)

ALFABETO

**FORMATO DE ENTRADA** X (45)

NUMERICO

**FORMATO DE SALIDA** X (45)

ALFANUMERICO

**NOMBRE** = Chofer

**DESCRIPCIÓN** = Nombre de persona que llevo el transporte

ALFABETO

**LONGITUD** = (30)

NUMERICO

**FORMATO DE ENTRADA** X (30)

ALFANUMERICO

**FORMATO DE SALIDA** X (30)

**NOMBRE** = Camion

**DESCRIPCIÓN** = Tipo de camion o razon social

ALFABETO

**LONGITUD** = (10)

NUMERICO

**FORMATO DE ENTRADA** X (10)

ALFANUMERICO

**FORMATO DE SALIDA** X (10)

**NOMBRE** = Fecha

**DESCRIPCIÓN** = El dia actualizado

**LONGITUD** = (6)

ALFABETO

**FORMATO DE ENTRADA** X (6)

NUMERICO

**FORMATO DE SALIDA** X (6)

ALFANUMERICO



**NOMBRE** = Guia

**DESCRIPCIÓN** = Numero de referencia de la transportacion

**LONGITUD** = (10)

ALFABETO

**FORMATO DE ENTRADA** X (10)

NUMERICO

**FORMATO DE SALIDA** X (10)

ALFANUMERICO

**NOMBRE** = Remftal

**DESCRIPCIÓN** = Quien envio finalmente la madera

**LONGITUD** = (20)

ALFABETO

**FORMATO DE ENTRADA** X (20)

NUMERICO

**FORMATO DE SALIDA** X (20)

ALFANUMERICO

**NOMBRE** = Recibio

**DESCRIPCIÓN** = Quien recibio en patio

**LONGITUD** = (35)

ALFABETO

**FORMATO DE ENTRADA** X (35)

NUMERICO

**FORMATO DE SALIDA** X (35)

ALFANUMERICO

**NOMBRE** = Folio

**DESCRIPCIÓN** = Numero de tarja que se maneja

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO



**NOMBRE** = Totaltarja

**DESCRIPCIÓN** = Total de recibido

**LONGITUD** = (9)

ALFABETO

**FORMATO DE ENTRADA** X (9)

NUMERICO

**FORMATO DE SALIDA** X (9)

ALFANUMERICO

**NOMBRE** = Precio

**DESCRIPCIÓN** = La cantidad a pagar

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO



## ESTRUCTURA DE LA TABLA TEMPORAL DEVOLUCION.

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO
Madera	Identifica de la madera	Carácter	14
Tipo	Tipo de madera a registrar.	Numerico	5
Grueso	Grueso de madera a registrar.	Numerico	5
Ancho	Ancho de madera a registrar.	Numerico	5
Cantidad	Cuenta madera a registrar.	Numerico	5
Total	Piesaje de madera a registrar	Numerico	5
Remitente	Proveedor quien envio.	Carácter	45
Consig	A nombre a quien queda la madera	Caracter	35
Direccion	Domicilio del proveedor	Carácter	45
Chofer	Nombre de persona que llevo el transporte	Carácter	30
Camion	Tipo de camion o razon social	Carácter	10
Fecha	El dia actualizado	Caracter	6
Guia	Numero de referencia de la transportacion	Numerico	10
Remftal	Quien envio finalmente la madera	Carácter	20
Recibio	Quien recibio en patio	Caracter	35
Folio	Numero de tarja que se maneja	Numerico	5
Totaltarja .	Total de recibido	Numerico	9
Precio	La cantidad a pagar	Numerico	5



**NOMBRE** = Madera

**DESCRIPCIÓN** = Identifica de madera

**LONGITUD** = (14)

ALFABETO

**FORMATO DE ENTRADA** X (14)

NUMERICO

**FORMATO DE SALIDA** X (14)

ALFANUMERICO

**NOMBRE** = Tipo

**DESCRIPCIÓN** = Tipo de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Grueso

**DESCRIPCIÓN** = Grueso de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Ancho

**DESCRIPCIÓN** = Ancho de madera a registrar.

**LONGITUD** = (50)

ALFABETO

**FORMATO DE ENTRADA** X (50)

NUMERICO

**FORMATO DE SALIDA** X (50)

ALFANUMERICO



**NOMBRE** = Cantidad

**DESCRIPCIÓN** = Cuanto de madera se va a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICo

**NOMBRE** = Total

**DESCRIPCIÓN** = Piesaje de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICo

**NOMBRE** = Remitente.

**DESCRIPCIÓN** = Proveedor quien envio

**LONGITUD** = (45)

ALFABETO

**FORMATO DE ENTRADA** X (45)

NUMERICO

**FORMATO DE SALIDA** X (45)

ALFANUMERICo

**NOMBRE** = Consig.

**DESCRIPCIÓN** = A nombre a quien queda la madera

**LONGITUD** = (35)

ALFABETO

**FORMATO DE ENTRADA** X (35)

NUMERICO

**FORMATO DE SALIDA** X (35)

ALFANUMERICo



**NOMBRE** = Direccion

**DESCRIPCIÓN** = Domicilio del proveedor

**LONGITUD** = (45)

**FORMATO DE ENTRADA** X (45)

**FORMATO DE SALIDA** X (45)

ALFABETO

NUMERICO

ALFANUMERICO

**NOMBRE** = Chofer

**DESCRIPCIÓN** = Nombre de persona que llevo el transporte

ALFABETO

**LONGITUD** = (30)

NUMERICO

**FORMATO DE ENTRADA** X (30)

ALFANUMERICO

**FORMATO DE SALIDA** X (30)

**NOMBRE** = Camion

**DESCRIPCIÓN** = Tipo de camion o razon social

ALFABETO

**LONGITUD** = (10)

NUMERICO

**FORMATO DE ENTRADA** X (10)

ALFANUMERICO

**FORMATO DE SALIDA** X (10)

**NOMBRE** = Fecha

**DESCRIPCIÓN** = El dia actualizado

**LONGITUD** = (6)

ALFABETO

**FORMATO DE ENTRADA** X (6)

NUMERICO

**FORMATO DE SALIDA** X (6)

ALFANUMERICO



**NOMBRE** = Guia

**DESCRIPCIÓN** = Numero de referencia de la transportacion

**LONGITUD** = (10)

ALFABETO

**FORMATO DE ENTRADA** X (10)

NUMERICO

**FORMATO DE SALIDA** X (10)

ALFANUMERICO

**NOMBRE** = Remftal

**DESCRIPCIÓN** = Quien envio finalmente la madera

**LONGITUD** = (20)

ALFABETO

**FORMATO DE ENTRADA** X (20)

NUMERICO

**FORMATO DE SALIDA** X (20)

ALFANUMERICO

**NOMBRE** = Recibio

**DESCRIPCIÓN** = Quien recibio en patio

**LONGITUD** = (35)

ALFABETO

**FORMATO DE ENTRADA** X (35)

NUMERICO

**FORMATO DE SALIDA** X (35)

ALFANUMERICO

**NOMBRE** = Folio

**DESCRIPCIÓN** = Numero de tarja que se maneja

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO



**NOMBRE** = Totaltarja

**DESCRIPCIÓN** = Total de recibido

**LONGITUD** = (9)

ALFABETO

**FORMATO DE ENTRADA** X (9)

NUMERICO

**FORMATO DE SALIDA** X (9)

ALFANUMERICO

**NOMBRE** = Precio

**DESCRIPCIÓN** = La cantidad a pagar

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

### ESTRUCTURA DE LA TABLA DE PROVEEDORES.

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO
Cuenta	Identifica de forma única al proveedor.	Numerico	11
Proveedor	Nombre del proveedor	Carácter	45
Direccion	Domicilio del proveedor	Carácter	45
Ciudad	Ciudad del proveedor	Carácter	25
Tel	Numero de telefono del proveedor	Numerico	10

**NOMBRE** = Cuenta

**DESCRIPCIÓN** Identifica de forma única al proveedor.

**LONGITUD** = (11)

ALFABETO

**FORMATO DE ENTRADA** X (11)

NUMERICO

**FORMATO DE SALIDA** X (11)

ALFANUMERICO



**NOMBRE** = Proveedor

**DESCRIPCIÓN** = Nombre del proveedor

**LONGITUD** = (45)

**FORMATO DE ENTRADA** X (45)

**FORMATO DE SALIDA** X (45)

- |                                     |                     |
|-------------------------------------|---------------------|
| <input checked="" type="checkbox"/> | <b>ALFABETO</b>     |
| <input type="checkbox"/>            | <b>NUMERICO</b>     |
| <input type="checkbox"/>            | <b>ALFANUMERICo</b> |

**NOMBRE** = Direccion

**DESCRIPCIÓN** = Domicilio del proveedor.

**LONGITUD** = (45)

**FORMATO DE ENTRADA** X (45)

**FORMATO DE SALIDA** X (45)

- |                                     |                     |
|-------------------------------------|---------------------|
| <input type="checkbox"/>            | <b>ALFABETO</b>     |
| <input type="checkbox"/>            | <b>NUMERICO</b>     |
| <input checked="" type="checkbox"/> | <b>ALFANUMERICo</b> |

**NOMBRE** = Ciudad.

**DESCRIPCIÓN** = Ciudad del proveedor

**LONGITUD** = (25)

**FORMATO DE ENTRADA** X (25)

**FORMATO DE SALIDA** X (25)

- |                                     |                     |
|-------------------------------------|---------------------|
| <input checked="" type="checkbox"/> | <b>ALFABETO</b>     |
| <input type="checkbox"/>            | <b>NUMERICO</b>     |
| <input type="checkbox"/>            | <b>ALFANUMERICo</b> |

**NOMBRE** = Tel.

**DESCRIPCIÓN** = Numero de telefono del proveedor

**LONGITUD** = (10)

**FORMATO DE ENTRADA** X (10)

**FORMATO DE SALIDA** X (10)

- |                                     |                     |
|-------------------------------------|---------------------|
| <input type="checkbox"/>            | <b>ALFABETO</b>     |
| <input checked="" type="checkbox"/> | <b>NUMERICO</b>     |
| <input type="checkbox"/>            | <b>ALFANUMERICo</b> |



### ESTRUCTURA DE LA TABLA DE TIPO DE CONSULTA.

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO
Madera	Identifica de la madera	Carácter	14
Tipo	Tipo de madera a registrar.	Numerico	5
Grueso	Grueso de madera a registrar.	Numerico	5
Ancho	Ancho de madera a registrar.	Numerico	5
Cantidad	Cuenta madera a registrar.	Numerico	5
Total	Piesaje de madera a registrar	Numerico	5
Remitente	Proveedor quien envio.	Carácter	45
Consig	A nombre a quien queda la madera	Caracter	35
Direccion	Domicilio del proveedor	Carácter	45
Chofer	Nombre de persona que llevo el transporte	Carácter	30
Camion	Tipo de camion o razon social	Carácter	10
Fecha	El dia actualizado	Caracter	6
Guia	Numero de referencia de la transportacion	Numerico	10
Remftal	Quien envio finalmente la madera	Carácter	20
Recibio	Quien recibio en patio	Caracter	35
Folio	Numero de tarja que se maneja	Numerico	5
Totaltarja	Total de recibido	Numerico	9
Precio	La cantidad a pagar	Numerico	5



**NOMBRE** = Madera

**DESCRIPCIÓN** = Identifica de madera

**LONGITUD** = (14)

ALFABETO

**FORMATO DE ENTRADA** X (14)

NUMERICO

**FORMATO DE SALIDA** X (14)

ALFANUMERICO

**NOMBRE** = Tipo

**DESCRIPCIÓN** = Tipo de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Grueso

**DESCRIPCIÓN** = Grueso de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Ancho

**DESCRIPCIÓN** = Ancho de madera a registrar.

**LONGITUD** = (50)

ALFABETO

**FORMATO DE ENTRADA** X (50)

NUMERICO

**FORMATO DE SALIDA** X (50)

ALFANUMERICO



**NOMBRE** = Cantidad

**DESCRIPCIÓN** = Cuanto de madera se va a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Total

**DESCRIPCIÓN** = Piesaje de madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Remitente.

**DESCRIPCIÓN** = Proveedor quien envio

**LONGITUD** = (45)

ALFABETO

**FORMATO DE ENTRADA** X (45)

NUMERICO

**FORMATO DE SALIDA** X (45)

ALFANUMERICO

**NOMBRE** = Consig.

**DESCRIPCIÓN** = A nombre a quien queda la madera

**LONGITUD** = (35)

ALFABETO

**FORMATO DE ENTRADA** X (35)

NUMERICO

**FORMATO DE SALIDA** X (35)

ALFANUMERICO



**NOMBRE** = Direccion

**DESCRIPCIÓN** = Domicilio del proveedor

**LONGITUD** = (45)

ALFABETO

**FORMATO DE ENTRADA** X (45)

NUMERICO

**FORMATO DE SALIDA** X (45)

ALFANUMERICo

**NOMBRE** = Chofer

**DESCRIPCIÓN** = Nombre de persona que maneja el camion

ALFABETO

**LONGITUD** = (30)

NUMERICO

**FORMATO DE ENTRADA** X (30)

ALFANUMERICo

**FORMATO DE SALIDA** X (30)

**NOMBRE** = Camion

**DESCRIPCIÓN** = Tipo de camion o razon social

ALFABETO

**LONGITUD** = (10)

NUMERICO

**FORMATO DE ENTRADA** X (10)

ALFANUMERICo

**FORMATO DE SALIDA** X (10)

**NOMBRE** = Fecha

**DESCRIPCIÓN** = El dia actualizado

**LONGITUD** = (6)

ALFABETO

**FORMATO DE ENTRADA** X (6)

NUMERICO

**FORMATO DE SALIDA** X (6)

ALFANUMERICo



**NOMBRE** = Guia

**DESCRIPCIÓN** = Numero de referencia de la transportacion

**LONGITUD** = (10)

ALFABETO

**FORMATO DE ENTRADA** X (10)

NUMERICO

**FORMATO DE SALIDA** X (10)

ALFANUMERICO

**NOMBRE** = Remftal

**DESCRIPCIÓN** = Quien envio finalmente la madera

ALFABETO

**LONGITUD** = (20)

NUMERICO

**FORMATO DE ENTRADA** X (20)

ALFANUMERICO

**FORMATO DE SALIDA** X (20)

**NOMBRE** = Recibio

**DESCRIPCIÓN** = Quien recibio en patio

ALFABETO

**LONGITUD** = (35)

NUMERICO

**FORMATO DE ENTRADA** X (35)

ALFANUMERICO

**FORMATO DE SALIDA** X (35)

**NOMBRE** = Folio

**DESCRIPCIÓN** = Numero de tarja que se maneja

ALFABETO

**LONGITUD** = (5)

NUMERICO

**FORMATO DE ENTRADA** X (5)

ALFANUMERICO

**FORMATO DE SALIDA** X (5)



**NOMBRE** = Totaltarja

**DESCRIPCIÓN** = Total de recibido

**LONGITUD** = (9)

ALFABETO

**FORMATO DE ENTRADA** X (9)

NUMERICO

**FORMATO DE SALIDA** X (9)

ALFANUMERICO

**NOMBRE** = Precio

**DESCRIPCIÓN** = La cantidad a pagar

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

### ESTRUCTURA DE LA TABLA PINO.

CAMPO	DESCRIPCIÓN	TIPO	TAMAÑO
Clave	Identifica de forma única de la combinacion	Numerico	2
Madera	La madera que existe nacional o radiata	Carácter	14
Tipo	Tipo de madera que exsite	Carácter	5
Grueso	El grueso de la madera a registrar	Numerico	5
Largo	El largo de la madera a registrar	Numerico	5
Ancho	El ancho de la madera a registrar	Numerico	5
Cantidad	La cantidad de madera a registrar	Numerico	6

**NOMBRE** = Clave

**DESCRIPCIÓN** = Identifica de forma única de la combinacion

**LONGITUD** = (2)

ALFABETO

**FORMATO DE ENTRADA** X (2)

NUMERICO

**FORMATO DE SALIDA** X (2)

ALFANUMERICO



**NOMBRE** = Madera

**DESCRIPCIÓN** = La madera que existe nacional o radiata

**LONGITUD** = (14)

ALFABETO

**FORMATO DE ENTRADA** X (14)

NUMERICO

**FORMATO DE SALIDA** X (14)

ALFANUMERICO

**NOMBRE** = Tipo

**DESCRIPCIÓN** = Tipo de madera que existe ue esta en uso.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Grueso

**DESCRIPCIÓN** = El grueso de la madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

X

**NOMBRE** = Largo

**DESCRIPCIÓN** = El largo de la madera a registrar.

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO



**NOMBRE** = Ancho.

**DESCRIPCIÓN** = El ancho de la madera a registrar

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO

**NOMBRE** = Cantidad.

**DESCRIPCIÓN** = La cantidad de madera a registrar

**LONGITUD** = (6)

ALFABETO

**FORMATO DE ENTRADA** X (6)

NUMERICO

**FORMATO DE SALIDA** X (6)

ALFANUMERICO

### ESTRUCTURA DE LA TABLA FOLIO.

Folio	Identifica de forma única a las tarjas	Numerico	5
-------	----------------------------------------	----------	---

**NOMBRE** = Folio

**DESCRIPCIÓN** = Identifica de forma única a las tarjas

**LONGITUD** = (5)

ALFABETO

**FORMATO DE ENTRADA** X (5)

NUMERICO

**FORMATO DE SALIDA** X (5)

ALFANUMERICO



### ESTRUCTURA DE LA TABLA RADIANA.

Contador	Lleva el control de los folio por numeracion	Numerico	3
----------	----------------------------------------------	----------	---

**NOMBRE** = Contador

**DESCRIPCIÓN** = Lleva el control de los folio por numeracion

**LONGITUD** = (3)

ALFABETO

**FORMATO DE ENTRADA** X (3)

NUMERICO

**FORMATO DE SALIDA** X (3)

ALFANUMERICO