



GIULLIANO GIMENEZ PIVETTA

SIGAP – SISTEMA GERENCIADOR DE APIÁRIOS

Santana do Livramento, dezembro de 2011

GIULLIANO GIMENEZ PIVETTA

SIGAP – SISTEMA GERENCIADOR DE APIÁRIOS

Monografia apresentada ao Curso de Graduação em Informática/Sistemas de Informação, na área de concentração Engenharia de Software, trabalho para conclusão de curso, para obtenção do grau de Bacharel em Informática/Sistemas de Informação

Orientador: Prof. Eduardo Bueno Simões Pires

Sant'ana do Livramento, dezembro de 2011

Dados Internacionais de Catalogação na Publicação (CIP)

P693s Pivetta, Giulliano, Gimenez
 SIGAP – Sistema Gerenciador de Apiários /
 Giulliano Gimenez Pivetta __ Sant’Ana do
 Livramento, 2011.
 p.

 Monografia (Conclusão de Curso) – URCAMP.
 Universidade da Região da Campanha. Faculdade
 de Informática – Sistemas de Informação

 Orientador: Eduardo Bueno Simões Pires

 1. Informática 2. Software - Desenvolvimento
 3. Gerenciamento de Apiário II. Universidade da
 Região da Campanha, Faculdade de Informática
 III. Título

 CDU 681.3.06

Bibliotecária responsável: Rose Mari T. Falcão
CRB 10/1096

GIULLIANO GIMENEZ PIVETTA

SIGAP – SISTEMA GERENCIADOR DE APIÁRIOS

Aprovado em dede.....

BANCA EXAMINADORA

Prof. Ricardo do Espirito Santo Barcellos - URCAMP

Prof. Alfredo Parteli Gomes - URCAMP

Prof. Franco Sampaio - URCAMP

Orientador – Eduardo Bueno Simões Pires - URCAMP

Sant’ana do Livramento, dezembro de 2011

Dedico este trabalho a todos meus familiares que sempre estiveram ao meu lado, e em especial meus pais que me apoiaram desde o início.

AGRADECIMENTOS

Gostaria de agradecer aos meus professores que me passaram seus conhecimentos tornando possível a conclusão deste trabalho, e em especial ao professor Eduardo Bueno Simões Pires pelo apoio em todo o processo de desenvolvimento.

Agradeço aos meus colegas também, por estarem sempre juntos, apoiando uns aos outros, e ao meu avô pelas informações importantes sobre o conteúdo no desenvolvimento do projeto.

"Nós não temos a chance de fazer muitas coisas, e cada uma deve ser realmente excelente. Porque esta é a nossa vida. A vida é breve, e então você morre, sabe? E todos nós escolhemos o que fazer com as nossas vidas. Então é melhor que seja muito bom. É melhor valer a pena."

(Steve Jobs, 1955-2011)

RESUMO

A apicultura é um ramo da zootecnia, onde visa a criação racional de abelhas para produção principalmente de mel e em menor quantidade geleia real, própolis, cera para fins pessoais ou comerciais.

Na apicultura moderna usa-se uma colmeia padrão desenvolvida em pesquisas científicas, e a manutenção de um apiário exige diversas visitas frequentes do apicultor para remoção de matos perto das colmeias, revisão das caixas e colheita, no qual é realizada normalmente duas vezes ao ano.

A apicultura no Brasil vem crescendo e ampliando a sua participação no mercado internacional onde já é o quinto maior exportador mundial, mas mesmo assim muitas vezes o sistema de controle que um apicultor possui sobre seus apiários são precários ou até mesmo não possuem nenhuma forma de controle sobre suas produções, existindo informações imprecisas ou desorganizadas, dificultando e até mesmo diminuindo a produção de mel e outros produtos por falta de cuidado com as informações dos apiários.

A utilização de um software gerenciador ira facilitar o controle e integridade das informações sobre o apiário, uma área no qual existem poucas ferramentas para otimizar a qualidade e quantidade de produção que um apiário pode oferecer.

Palavras Chave:

Apicultura. Software. Apiário. Colmeia.

ABSTRACT

Beekeeping is a branch of animal husbandry, which aims to create rational bees mainly for honey production and fewer royal jelly, wax for personal or commercial. In modern beekeeping uses a honeycomb pattern is developed in scientific research, and maintaining a diverse demands frequent visits apiary of the beekeeper to remove weeds near the hives, boxes and review of the harvest, which is usually held twice a year. Beekeeping in Brazil has been growing and expanding its international market share which is already the fifth largest exporter, yet often the system of control that a beekeeper apiaries on their own are weak or even have no way to control about their productions, there is inaccurate or disorganized, difficult and even decreasing the production of honey and other products through carelessness with the information of the apiaries.

The use of a software manager will facilitate the control and integrity of information about the apiary, an area in which there are few tools to optimize the quality and quantity of production that can offer an apiary.

Keywords:

Beekeeping. Software. Apiary. Beehive.

LISTA DE FIGURAS

Figura 1 – Modelo Entidade - Relacionamento.....	22
Figura 2 – Login de acesso ao sistema.....	27
Figura 3 – Tela principal do sistema.....	28
Figura 4 – Tela de cadastro de Apiários.....	29
Figura 5 – Tela de cadastro de Colmeias.....	30
Figura 6 – Tela de cadastro de apicultores.....	31
Figura 7 – Tela das fichas de controle.....	32
Figura 8 – Tela de controle de equipamentos.....	33

LISTA DE ABREVIATURAS

ER – Entidade-Relacionamento

Integer – Valores Inteiros

Obs. – Observação

RG – Registro Geral

CPF – Cadastro de Pessoas Físicas

LISTA DE TABELAS

Tabela 1 - Apiário.....	24
Tabela 2 - Apicultor.....	24
Tabela 3 - Colheita.....	24
Tabela 4 - Colmeia.....	25
Tabela 5 - Equipamentos.....	25
Tabela 6 - Produção.....	26
Tabela 7 - Revisão.....	26

SUMÁRIO

INTRODUÇÃO.....	14
1. MOTIVAÇÃO.....	15
2. OBJETIVOS.....	16
2.1. OBJETIVO GERAL.....	16
2.2. OBJETIVOS ESPECIFICOS.....	16
2.3. METODOLOGIA	16
3. REVISÃO DE LITERATURA.....	17
3.1. LINGUAGEM DE PROGRAMAÇÃO BORLAND DELPHI.....	17
3.1.1. Histórico e definição.....	17
3.1.2. Principais características do Delphi.....	18
3.1.3 Acessando o banco de dados com Delphi.....	18
3.2. BANCO DE DADOS MYSQL.....	19
3.2.1. Histórico e definição.....	19
3.2.2. As principais características do MySql.....	19
3.2.3. Vantagens para utilização do banco de dados.....	19
4. SIGAP – SISTEMA GERENCIADOR DE APIARIOS.....	21
4.1. PROBLEMA.....	21
4.2. OBJETIVOS.....	21
4.3. CARACTERISTICAS.....	21
4.4. BANCO DE DADOS DO SIGAP.....	22
4.4.1. Diagrama Entidade-Relacionamento.....	22
4.5. DICIONARIO DE DADOS.....	24
4.6. INTERFACE E FUNCIONAMENTO DO SIGAP.....	27
4.6.1. Login de acesso ao sistema.....	27
4.6.2. Tela principal do sistema.....	28
4.6.3. Tela de cadastro de apiários.....	29
4.6.4. Tela de cadastro de colmeias.....	30
4.6.5. Tela de cadastro de apicultores.....	31
4.6.6. Tela das fichas de controle.....	32
4.6.7. Tela de controle de equipamentos.....	33
CONCLUSÃO.....	34

REFERÊNCIAS BIBLIOGRÁFICAS.....	35
ANEXOS.....	36

INTRODUÇÃO

Este trabalho foi realizado para a obtenção do grau de bacharel em sistemas de informação, e uma grande motivação em melhorar a maneira em que os apicultores cuidam de seus apiários, por meio de um software com banco de dados, desenvolvido utilizando a ferramenta Delphi 7 em conjunto com MySql 5.1 para o gerenciamento do banco de dados, onde o apicultor terá as informações mais importantes sobre seus apiários e as respectivas colmeias pertencentes a esse conjunto, para facilitar o manuseio e aumentar a sua produtividade e qualidade. O objetivo do software é aperfeiçoar a maneira como meu avô cuida de suas colmeias, por uma dificuldade encontrada no atual meio de controle (papéis), onde exige uma grande organização para evitar perdas ou extravio de informações. Dessa forma, o software irá armazenar suas informações importantes, reduzindo seus custos com impressões, facilitando o acesso as seus dados de forma rápida e organizada.

1. MOTIVAÇÃO

Os principais motivos para a realização deste trabalho foram, em usar o mesmo com o objetivo de concluir o curso de informática e receber o grau de Bacharel em Informática, e melhorar a forma em que principalmente meu avô (apicultor) gerencia seus apiários, onde a apicultura no brasil vem crescendo rapidamente, e mesmo assim, ainda não existem ferramentas para seu controle, sendo assim, irá melhorar e reduzir gastos com materiais (folhas, fichários, pastas), que são os meios de controles mais utilizados no momento, mas que possui muitas falhas. O software irá melhorar a forma no qual o apicultor gerencia suas colmeias.

2. OBJETIVOS

Buscar melhorar a forma no qual meu avô gerencia seus apiários, facilitando o armazenamento e consulta de informações sobre suas colmeias, colheitas, revisões e equipamentos.

2.1. OBJETIVO GERAL

Este trabalho tem por objetivo facilitar a criação e manutenção de apiários por meio de um software com banco de dados, no qual irá ajudar o apicultor no momento de registrar e pesquisar suas informações sobre determinados apiários e colmeias, aumentando sua produtividade e reduzir o tempo que o apicultor gasta em recolher determinada informação.

2.2. OBJETIVOS ESPECIFICOS

Para a realização deste trabalho foi necessário completar os seguintes objetivos

- a) O funcionamento de um apiário, desde sua criação ate o momento de extração do mel e seus derivados e ser preparado para a venda.
- b) Levantar as necessidades a serem cumpridas pelo software para o apicultor
- c) Estudadas as Ferramentas Delphi7 e MySql.
- d) Modelar o banco de dados
- e) Implementar e testes.

2.3 METODOLOGIA

O desenvolvimento do sistema baseou-se no conhecimento adquirido nas diversas entrevistas realizadas para o entendimento do funcionamento de um apiário, no qual foi necessário compreender o funcionamento de um apiário, como esta relacionado com suas colmeias, a forma no qual são realizadas as revisões e colheitas, os equipamentos utilizados, entre outras informações, e do cumprimento das necessidades que o apicultor sente em relação ao software e por fim do conhecimento das ferramentas de desenvolvimento adquiridos na faculdade.

3. REVISÃO DE LITERATURA

A revisão de literatura trata do desenvolvimento de Software para atender as principais necessidades que os apicultores encontram para gerenciar seus apiários, procurando aperfeiçoar as ações dos apicultores, procurando um melhor controle sobre seu trabalho, adotando os principais fundamentos e conhecimentos que serviram de base para o desenvolvimento e implantação do software.

3.1. LINGUAGEM DE PROGRAMAÇÃO BORLAND DELPHI

3.1.1. Histórico e definição

Segundo informações encontradas no site do Instituto Federal Sul-Rio-Grandense <http://www2.pelotas.ifsul.edu.br/npeil/pagborlanddelphi.htm> escrito por Ricardo Veloso, o Delphi foi lançado em 1995, trata-se de um ambiente de desenvolvimento de aplicações orientadas a objetos, sua linguagem utilizada é o Object Pascal, que pode ser considerada como uma evolução do Pascal padrão, que surgiu no final dos anos 60 e foi utilizado como uma das primeiras linguagens de programação para estudantes de informática, tendo como sua característica principal ser poderoso e ao mesmo tempo relativamente fácil de aprender a usar, sem, no entanto, desprezar a flexibilidade para expansão dos seus recursos.

O Delphi oferece diversos recursos para diferentes tipos de programadores, por ser um completo ambiente de desenvolvimento de aplicações, que se baseia num conceito denominado RAD (Rapid Application Development – Desenvolvimento Rápido de Aplicação). O termo RAD pode nos enganar à primeira vista. Na realidade, essa metodologia não automatiza por completo o desenvolvimento de uma aplicação, mas apenas auxilia em grande parte as diversas fases do projeto, principalmente aquelas relacionadas à criação da interface com o usuário.

O Delphi possui diferentes versões como, Delphi Desktop, para os que trabalham em rede, Delphi Standard, para estudantes, com poucas ferramentas de apoio, Delphi Professional, com a maioria de ferramentas, mas sem suporte a arquitetura cliente/servidor e Delphi Cliente/Server Suíte, versão completa, com todas as ferramentas de apoio para os que trabalham na plataforma cliente/servidor.

O Delphi possui um compilador de código nativo que gera aplicações em 32 bits. Dessa forma, a aplicação criada com ele possui um desempenho muito próximo daquelas

criadas em linguagem C/C++ ou mesmo Assembly. Outro fator que contribui para o desempenho de programas em Delphi diz respeito à maneira como foi escrita sua biblioteca de componentes visuais (VCL). A Borland conseguiu aperfeiçoar de forma surpreendente os algoritmos desses componentes, obtendo um alto desempenho.

3.1.2. As principais características do Delphi

- a) Linguagem Orientada a Objetos.
- b) Ambiente gráfico, visual e multimídia.
- c) Arquitetura Baseada em Componentes, assegurando que as aplicações sejam robustas, reutilizáveis e de fácil manutenção.
- d) Fluxo de programação baseado em eventos.
- e) Alto desempenho, em sistemas críticos.
- f) *Tow-Way Tools*, capacidade de alternar entre um form e seu código permite aos desenvolvedores trabalhar tanto na edição de texto como no modo de *design* visual através de total sincronização do código fonte com a representação visual.

3.1.3. Acessando o banco de dados com Delphi

A escolha do Delphi como ferramenta para o desenvolvimento de aplicativos tem aumentado, devido a sua facilidade em interagir com vários tipos de banco de dados. Essa abrangência se deve ao mecanismo BDE (*Borland Database Engine*) que é uma interface de acesso a Banco de Dados que possui suporte para diversos bancos e servidores *SQL* (*Oracle*, *Microsoft SQL Server* e *Interbase*), *drivers* sem a linguagem *SQL* (*Paradox*, *Dbase*) e arquivos de texto. Também aceita os *drivers* do ODBC e suporta bancos de dados do Access.

O BDE permite a navegação em tabelas e *queries SQL*, suporta múltiplos usuários consultando a base de dados ao mesmo tempo, pelo fato de ter um conjunto de funções que compõe uma API (Interface para Programação de Aplicações).

Quando o Delphi precisa acessar um banco de dados, ele usa diretamente o serviço do BDE para facilitar, pois trabalha para leitura de arquivos DB e DBF guardados localmente.

3.2. BANCO DE DADOS MYSQL

3.2.1 Histórico e definição

Segundo o site http://www.oficinadanet.com.br/artigo/2227/mysql_-_o_que_e escrito por francisco leandro, o MySql foi criado na Suécia por David Axmark, Allan Larsson e Michael Widenius, que trabalham juntos desde a década de 80. Hoje seu desenvolvimento emprega mais de 400 profissionais no mundo inteiro e mais de mil contribuem testando o software, integrando-o a outros produtos e escrevendo a respeito dele.

No dia 16 de Janeiro de 2008, a MySQL AB, desenvolvedora do MySQL foi adquirida pela Sun Microsystems, por US\$ 1 bilhão, um preço jamais visto no setor de licenças livres. No dia 20 de Abril de 2009 a Oracle compra a Sun Microsystems e todos os seus produtos, incluindo o MySql.

O sucesso do MySql deve-se em grande medida à fácil integração com o PHP incluído, quase que obrigatoriamente, nos pacotes de hospedagem de sites da Internet oferecidos atualmente. Empresas como Yahoo! Finance, Motorola, NASA, Silicon Graphics e Texas Instruments usam o Mysql em aplicações de missão crítica.

3.2.2. As principais características do MySql

- a) Portabilidade (suporta praticamente qualquer plataforma atual).
- b) Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, Python, Perl, PHP, ASP e Ruby).
- c) Excelente desempenho e estabilidade.
- d) Pouco exigente quanto a recursos de hardware.
- e) Facilidade de uso.
- f) Interfaces gráficas (MySql Toolkit) de fácil utilização cedidos pela MySQL Inc.

3.2.3. Vantagens para utilização do banco de dados

Com a utilização de um sistema gerenciador de banco de dados pode-se notar os seguintes benefícios:

a) Eliminação de Redundâncias – Possibilita a eliminação de dados primitivos de cada sistema. Os dados, que eventualmente são comuns a mais de um sistema, são compartilhados por eles, permitindo o acesso a uma única informação sendo consultada por vários sistemas.

b) Eliminação de Inconsistências – Através do armazenamento da informação em um único local com acesso descentralizado e, sendo compartilhados a vários sistemas, os usuários estarão utilizando uma informação confiável. A inconsistência ocorre quando um mesmo campo tem valores diferentes em sistemas diferentes. Exemplo, o endereço de uma pessoa em um sistema é um, e em outro sistema é diferente. Isso ocorre porque esta pessoa atualizou os dados em um sistema e não o atualizou no outro. Quando o dado é armazenado em um único local e compartilhado pelos sistemas, este problema não ocorre.

c) Compartilhamento de Dados – Permite a utilização simultânea e segura de dados, por mais de uma aplicação ou usuário, independente da operação que esteja sendo realizada. Deve ser observado apenas o processo de atualização concorrente, para não gerar erros de processamento (atualizar simultaneamente o mesmo campo do mesmo registro).

d) Restrições de Segurança – Definem para cada usuário o nível de acesso aos dados a ele concedido (leitura, gravação e leitura ou sem acesso). Este recurso impede que pessoas não autorizadas manipulem ou alterem um determinado campo.

e) Padronização de Dados – Permite que os campos armazenados na base de dados sejam padronizados segundo um determinado formato do armazenamento (padronização de tabela, conteúdo de campos) e ao nome de variáveis seguindo critérios e padrões preestabelecidos pela empresa.

f) Independência de Dados – Representa a forma física de armazenamento dos dados no Banco de Dados e a recuperação das informações pelos programas de aplicação. Esta recuperação deverá ser totalmente independente da maneira com que os dados estão fisicamente armazenados. Quando um programa retira ou inclui dado, o Sistema Gerenciador de Banco de Dados compacta-os para que haja um menor consumo de espaço no disco. Este conhecimento do formato de armazenamento de campo é totalmente transparente para o usuário.

g) Manutenção da Integridade – Consiste em impedir que um determinado código ou chave em uma tabela não tenha correspondência em outra tabela. Exige que o conteúdo dos dados armazenados no Banco de Dados possua valores coerentes ao objetivo do campo.

4. SIGAP – SISTEMA GERENCIADOR DE APIÁRIOS

4.1. PROBLEMA

Por meu avô ser apicultor, constantemente vejo a dificuldade que ele encontra em confirmar se procedimentos de manutenção foram realizados nos apiários ou colmeias, ou até mesmo em conferir as últimas revisões e colheitas, e não ter um controle exato das mesmas, gera muitos transtornos no processo de produção, dificultando o processo de criação até o ponto final de venda dos produtos.

4.2. OBJETIVOS

O objetivo deste trabalho é otimizar o meio utilizado para se gerenciar os apiários, facilitando consultas e cadastros realizados, sanando as dificuldades encontradas para um controle mais aperfeiçoado dos apiários.

4.3. CARACTERÍSTICAS

Dentre as características presentes destacam-se as seguintes:

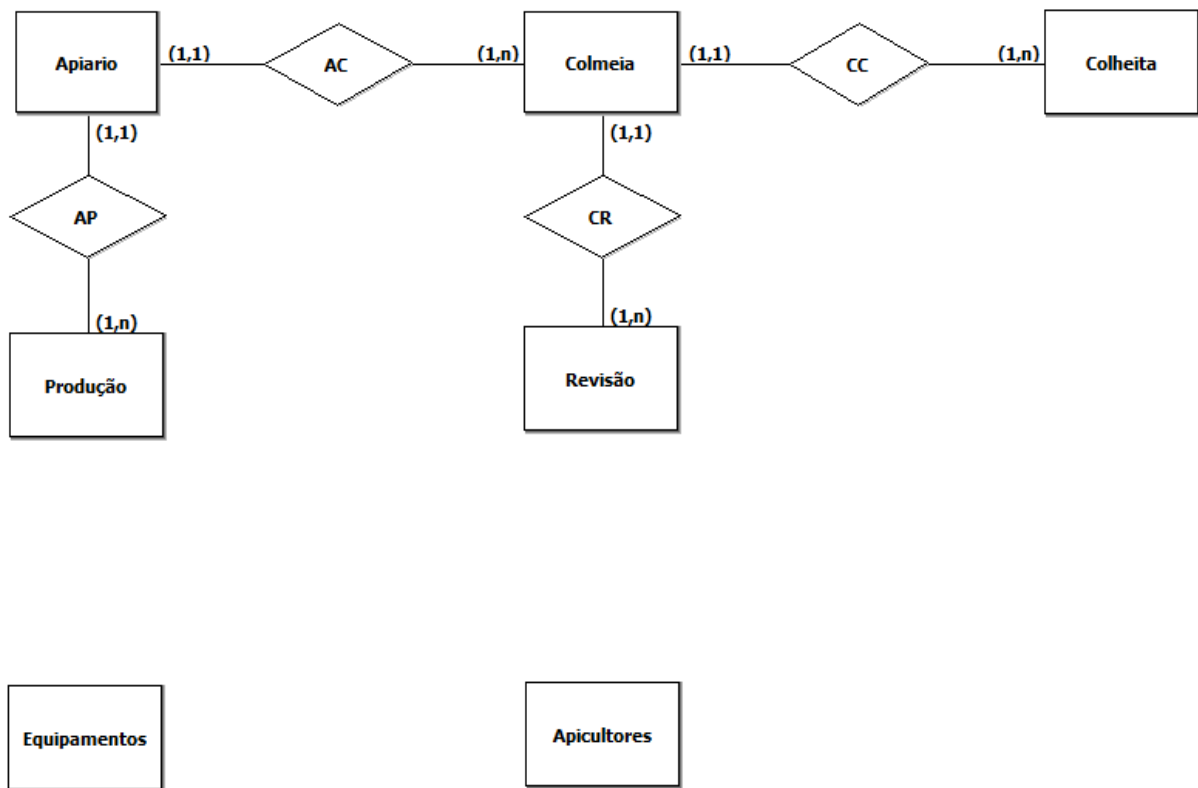
- a) Cadastro de apiários
- b) Cadastro de colmeias por apiários
- c) Cadastro de equipamentos
- d) Controle de visitas (revisões)
- e) Controle de colheitas
- f) Agenda de apicultores
- g) Relatórios

4.4. BANCO DE DADOS DO SIGAP

4.4.1. Diagrama Entidade-Relacionamento

O diagrama entidade-relacionamento possui as seguintes entidades: Apiário, Produção, equipamentos, Colmeia, revisão, Colheita, Apicultores.

- a) **Apiário:** Esta entidade guardará as seguintes informações: Código do apiário, nome, localização, floração.
- b) **Apicultor:** Esta entidade guardará as seguintes informações: Código do apicultor, nome, CPF, RG, N° de registro do apicultor, telefone, celular, localização do(s) apiário(s).
- c) **Colheita:** Esta entidade guardará as seguintes informações: código da colheita, data da colheita, data prevista para próxima colheita, produção, observações.
- d) **Colmeia:** Esta entidade guardará as seguintes informações: Código da colmeia, identificador da colmeia, tipo de ninho, numero de quadros, numero de sobre ninhos, numero de quadros do sobre Ninho, numero de melgueiras, numero de quadros das melgueiras, qualidade do enxame, raça da rainha, idade da rainha, nível defensivo do enxame, adição de pólen, adição de néctar, apiário pertencente.
- e) **Equipamentos:** Esta entidade guardará as seguintes informações: Código do equipamento, nome, descrição, quantidade, categoria.
- f) **Produção:** Esta entidade guardará as seguintes informações: Código da produção, apiário, data, quantidade de mel (peso em Quilogramas).
- g) **Revisão:** Esta entidade guardará as seguintes informações: código da revisão, avaliação, data da revisão, data prevista para próxima revisão, observações.

SIGAP – SISTEMA GERENCIADOR DE APIÁRIOS**Figura 1** – Modelo Entidade-Relacionamento

4.5 DICIONARIO DE DADOS

Tabela 1: Apiário

NOME	TIPO	TAMANHO	DESCRIÇÃO	CP	CE
Cod_apiario	Integer (+)		Código do apiário	*	
Nome	Varchar	45	Nome do apiário		
Local	Varchar	45	Localização do apiário		
Floral	Varchar	45	Floração do apiário		

Tabela 2: Apicultor

NOME	TIPO	TAMANHO	DESCRIÇÃO	CP	CE
N_reg	Integer (+)		Código do apicultor	*	
nome	Varchar	45	Nome do apicultor		
Cpf	Varchar	14	Cpf do apicultor		
Rg	Varchar	15	Rg do Apicultor		
N_talao	Varchar	45	Numero de Registro do Apicultor		
Telefone	Varchar	15	Telefone		
Telefone2	Varchar	15	Celular		
Localizacao	Varchar	45	Localização do(s) Apiario(s)		

Tabela 3: Colheita

NOME	TIPO	TAMANHO	DESCRIÇÃO	CP	CE
Cod_coleta	Integer (+)		Código da coleta	*	
Data_coleta	Date		Data da coleta		
Data_proxcolheita	Date		Data prevista da próxima coleta		
Producao	Varchar	20	Nível de produção da colmeia		
Obs	Varchar	150	Observações		
N_colm	Varchar	45			*

Observações: O campo *N_colm* referencia na Tabela *Colmeia* o campo *N_colm*.

Tabela 4: Colmeia

NOME	TIPO	TAMANHO	DESCRIÇÃO	CP	CE
N_colm	Integer (+)		Código da colmeia	*	
Identificador	Varchar	20	Identificador da colmeia		
Tip_ninho	Varchar	15	Tipo de Ninho da colmeia		
Num_quad	Integer		Numero de Quadros		
N_melg	Integer		Numero de melgueiras		
N_quadmelg	Integer		Numero de quadros da melgueira		
N_sobreninho	Integer		Numero de Sobre Ninhos		
N_quadsobreninho	Integer		Nº de quadros do sobre ninho		
Qual_enxame	Varchar	10	Qualidade do enxame		
Raca_rainha	Varchar	15	Raça da Rainha		
Idade_rainha	Date		Data da Rainha		
Nivel_def	Varchar	10	Nível defensivo do enxame		
Polem	Varchar	5	Adição de Pólen		
Nectar	Varchar	5	Adição de Néctar		
Cod_apiario	Varchar	5	Apiário pertencente		*

Observações: O campo *Cod_apiario* referencia na tabela *Apiario* o campo *Cod_apiario*.

Tabela 5: Equipamentos

NOME	TIPO	TAMANHO	DESCRIÇÃO	CP	CE
Cod Equipamento	Integer (+)		Código do equipamento	*	
Nome	Varchar	45	Nome do equipamento		
Categoria	Varchar	45	Categoria do equipamento		
Descrição	Varchar	150	Descrição do equipamento		
Quantidade	Varchar	45	Quantidade		

Tabela 6: Produção

NOME	TIPO	TAMANHO	DESCRIÇÃO	CP	CE
Cod_prod	Integer (+)		Código da produção	*	
Data	Date		Data da produção		
Peso_mel	Double		Quantidade de Mel (peso em Kg)		
Cod_apiario	Integer		Apiário pertencente		*

Observações: O campo *Cod_apiario* referencia na Tabela *Apiario* o campo *Cod_apiario*.

Tabela 7: Revisão

NOME	TIPO	TAMANHO	DESCRIÇÃO	CP	CE
Cod_visita	Integer (+)		Código da visita	*	
avaliacao	Varchar	45	Avaliação da colmeia		
Data_rev	Date		Data da revisão		
Data_proximarevisao	Date		Data estimada para próxima rev.		
Obs	Varchar	100	Observações		
N_colm	Varchar	15	Colmeia		*

Observações: O campo *N_colm* referencia na tabela *Colmeia* o campo *N_colm*.

4.6. INTERFACE E FUNCIONAMENTO DO SIGAP

4.6.1. Login de acesso ao sistema

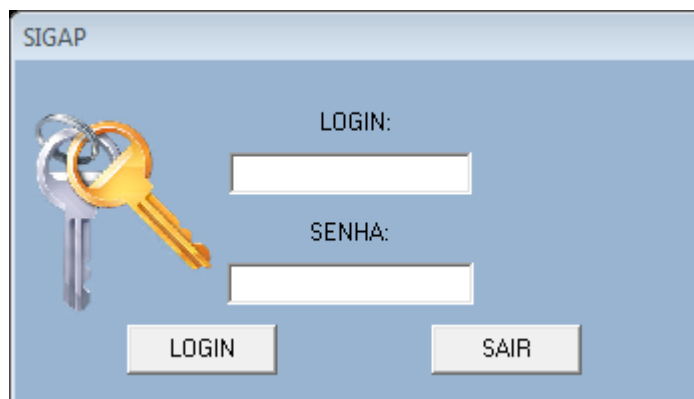


Figura 2 – Login de acesso ao sistema

Ao acessar o *SIGAP* a primeira tela a ser exibida será a de *Login* (**Figura 2** – Login de acesso ao sistema), onde o usuário é identificado. Se for verificado que o usuário ou a senha são inválidos ou não conferem, uma mensagem será exibida e o sistema não será aberto.

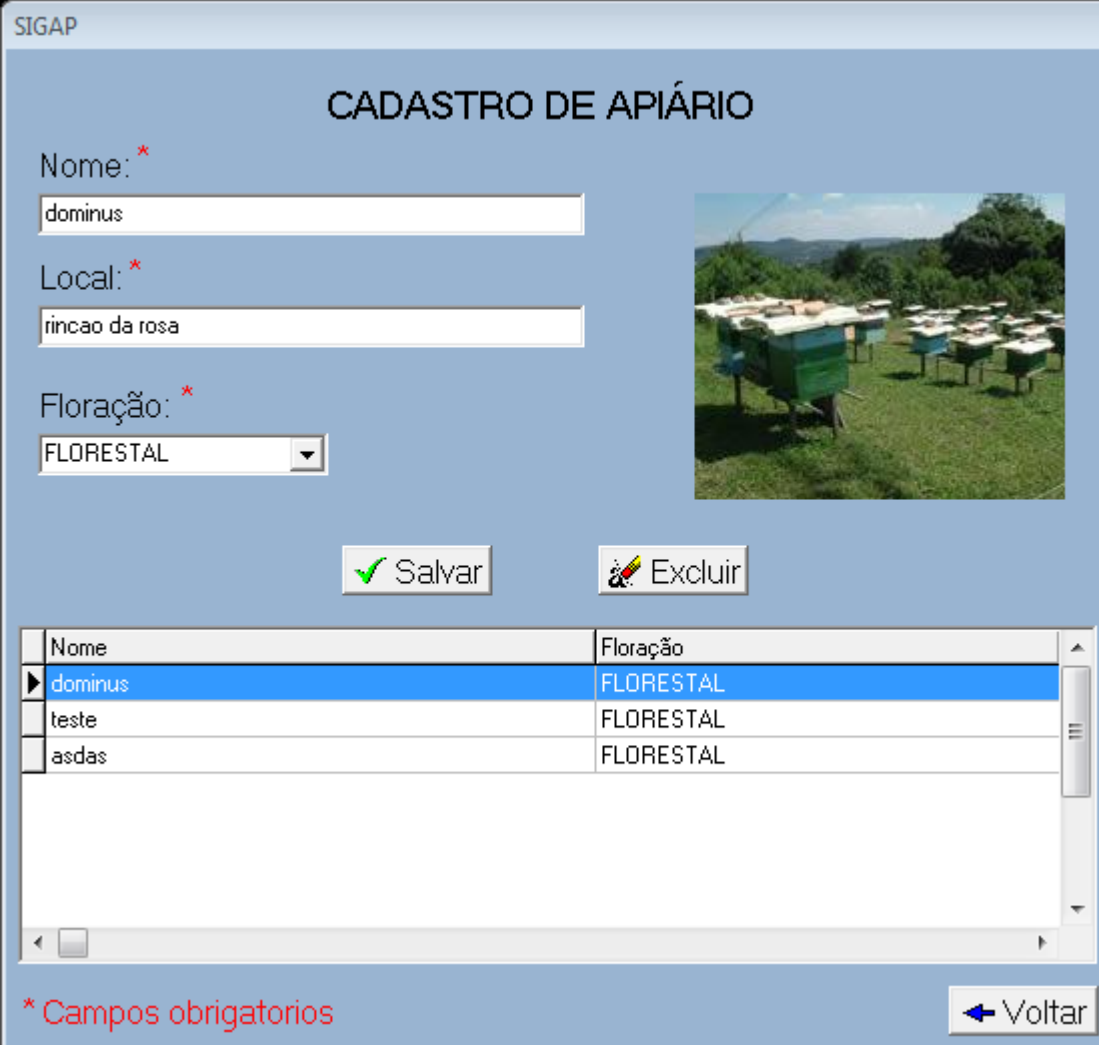
4.6.2. Tela principal do sistema



Figura 3 – Tela principal do sistema

Após a tela de *login* do usuário, será exibida a Tela Principal do Programa (**Figura 3** – tela principal do sistema), onde serão carregadas todas as suas funções, será apresentada esta tela onde aparecem todos os menus do sistema e seus botões de atalho.

4.6.3. Tela de cadastro de Apiários





SIGAP

CADASTRO DE APIÁRIO

Nome: *
dominus

Local: *
rincao da rosa

Floração: *
FLORESTAL

 Salvar  Excluir

Nome	Floração
dominus	FLORESTAL
teste	FLORESTAL
asdas	FLORESTAL

* Campos obrigatorios


 Voltar

Figura 4 – Tela de cadastro de Apiários

Quando o usuário clicar no menu *cadastros* e logo após em *Apiário* da tela principal lhe é apresentado o formulário de cadastro de apiários (**Figura 4** – Tela de cadastro de Apiários), onde são colocadas as principais informações do apiário e será possível fazer consultas rápidas dos mesmos

4.6.4. Tela de cadastro de Colmeias



SIGAP

CADASTRO DE COLMEIA

Tipo de ninho* ID Colmeia.*

Numero de quadros*

Nº de melgueiras: Nº de quadros na melgueira:

Nº de sobreninhos: Nº de quadros do sobreninho:

Raça da rainha:* Data da Rainha: 25/10/11

Qualidade do enxame:

Nível defensivo:

Nectar: Polem:

Apiário Pertencente:*

15:34:39 Terça, 25 de Outubro de 2011

Figura 5 – Tela de cadastro de Colmeias

Quando o usuário clicar no menu *cadastros* e logo após em *Colmeia* da tela principal lhe é apresentado o formulário de cadastro de colmeias (**Figura 5** – Tela de cadastro de Colmeias), onde são colocadas as principais informações da colmeia e a seleção do apiário pertencente cadastrado anteriormente.

4.6.5. Tela de cadastro de apicultores

SIGAP

CADASTRO DE APICULTORES

Nome: *

Cpf: RG:

Nº De Registro do Apicultor:

Telefone: * Celular:

Localização do Apirario:

Salvar Excluir

Nome	Cpf	Rg	Talão
aaaa			
asd			
gauhaashs			
adsdasd			
*			

Voltar

15:37:39 Terça, 25 de Outubro de 2011

Figura 6 – Tela de cadastro de apicultores

Quando o usuário clicar no menu *cadastros* e logo após em *Apicultor* da tela principal lhe é apresentado o formulário de cadastro de apicultores (**Figura 6** – Tela de cadastro de apicultores), onde são colocadas as principais informações dos mesmos, utilizado como uma agenda virtual.

4.6.6. Tela das fichas de Controle

[illegible]

Figura 7 – Tela das fichas de controle

Esta Tela é aberta após selecionar o menu *Fichas de controle* na tela principal do sistema, onde é apresentado o menu de seleção das fichas (**Figura 7** – Tela das Fichas de controle), que funcionam como meio de armazenamento temporário de informações quando não houver a possibilidade de cadastrá-las no software no momento.

4.6.7. Tela de controle de equipamentos

The screenshot shows a web application window titled 'SIGAP' with a sub-header 'CONTROLE DE EQUIPAMENTOS'. The form contains the following fields and controls:

- Categoria:** A dropdown menu with a red asterisk indicating it is a required field.
- Nome:** A text input field with a red asterisk. To its right is a small number '1'.
- Quantidade:** A text input field with a red asterisk, containing the value '1'. To its right are two buttons: a '+' button and a '-' button.
- Descrição:** A large text area with a red asterisk.
- Buttons:** At the bottom, there are four buttons: 'Novo' (with a document icon), 'Salvar' (with a green checkmark icon), 'Excluir' (with a red X icon), and 'Voltar' (with a blue arrow icon).
- Footer:** A button labeled 'Mostrar +' with a red downward arrow icon.

Figura 8 – Tela de controle de equipamentos

Quando o usuário clicar no menu *Equipamentos* na tela principal, lhe é apresentado o formulário de cadastro de equipamentos (**Figura 8** – Tela de controle de equipamentos), onde são colocadas as principais informações dos mesmos, utilizado como um controle de materiais.

CONCLUSÃO

Podemos concluir que desenvolver um software que gerencie todas as funções que um apiário possui foi um desafio enorme, exigindo um estudo mais aprofundado de todas as ferramentas aprendidas até o momento, para extrair o máximo de desempenho e confiabilidade que o software desse ramo da apicultura deve ter, pois muitas informações são essenciais para o trabalho do apicultor.

O sistema foi desenvolvido com uma interface simples e fácil de usar, tanto para o usuário avançado como para o leigo, dessa forma as informações serão encontradas mais rapidamente, agilizando o processo de controle.

O sistema SIGAP foi desenvolvido para utilização do meu avô no qual é apicultor e seus parceiros e para o trabalho de conclusão do curso, mas também como um objetivo próprio, pois será o primeiro de muitos passos que estão por vir, após o término do curso.

Ao término deste trabalho, o software estará pronto para utilização, cumprindo com suas metas, e futuramente com a utilização do software, serão agregadas novas funções, para complementar ainda mais o trabalho dos apicultores.

REFERÊNCIAS BIBLIOGRÁFICAS

CANTÚ, Marco. Dominando o Delphi 7- A Bíblia. Rio de Janeiro: Makron Books, 2003.

FURASTÉ, Pedro Augusto. Normas Técnicas p/ o Trabalho Científico: Elaboração e Formatação. Explicitação das Normas da ABNT. 14. ed. Porto Alegre: Sn, 2005.

HEUSER, Carlos Alberto. Projeto de Banco de Dados. 6. ed. Porto Alegre: BookMan, 2009.

NOBEL, Maria Da Glória, Noções Básicas Sobre a Criação de Abelhas. 1º ed. São Paulo 1997

WIESE, Helmuth. Apicultura - novos tempos, 2º Ed. Agrolivros Rio de Janeiro 2005

www2.pelotas.ifsul.edu.br/npeil/pagborlanddelphi.htm Acessado dia 03/11/11 as 22:35

<http://www.oficinadanet.com.br/artigo/2227/mysql> Acessado dia 04/11/11 às 21:20

ANEXOS

```

unit Unit9;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, Grids, DBGrids, Mask, DBCtrls, Menus, jpeg,
  ExtCtrls;
type
  Tcadastro_apiario = class(TForm)
    BitBtn1: TBitBtn;
    DBGrid1: TDBGrid;
    Label1: TLabel;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    Label3: TLabel;
    DBEdit2: TDBEdit;
    Label4: TLabel;
    DBEdit3: TDBEdit;
    Label5: TLabel;
    DBText1: TDBText;
    DBComboBox1: TDBComboBox;
    Image1: TImage;
    Label2: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

```

var
    cadastro_apiario: Tcadastro_apiario;
implementation

uses Unit1, Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8;
{$R *.dfm}
procedure Tcadastro_apiario.BitBtn1Click(Sender: TObject);
Begin
    cadastro_apiario.CLOSE;
end;
procedure Tcadastro_apiario.BitBtn2Click(Sender: TObject);
begin
    DATAMODULE3.apiario.Append;
end;
procedure Tcadastro_apiario.BitBtn4Click(Sender: TObject);
begin
    if (dbedit2.text = "") or (dbedit3.text = "") or (dbcombobox1.Text = "") then
    begin
        messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
        dbedit2.SetFocus;
    end
    else
    begin
        DATAMODULE3.apiario.Post;
        messagedlg('Apiario cadastrado com sucesso!', mtinformation, [mbok], 0);
        DATAMODULE3.apiario.append;
    end;
end;
procedure Tcadastro_apiario.BitBtn3Click(Sender: TObject);
begin
    DATAMODULE3.apiario.DELETE;
end;
end.
unit Unit4;

```

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DBCtrls, ExtCtrls, Grids, DBGrids, StdCtrls, Buttons, Mask, jpeg,
Menus, ComCtrls;

type

```
Tcadastro_apicultor = class(TForm)
  Label1: TLabel;
  Label3: TLabel;
  DBEdit2: TDBEdit;
  Label4: TLabel;
  DBEdit3: TDBEdit;
  Label5: TLabel;
  DBEdit4: TDBEdit;
  Label6: TLabel;
  DBEdit5: TDBEdit;
  Label7: TLabel;
  DBEdit6: TDBEdit;
  Label8: TLabel;
  DBEdit7: TDBEdit;
  Label9: TLabel;
  DBEdit8: TDBEdit;
  BitBtn1: TBitBtn;
  DBGrid1: TDBGrid;
  DBText1: TDBText;
  BitBtn2: TBitBtn;
  BitBtn3: TBitBtn;
  StatusBar1: TStatusBar;
  Timer1: TTimer;
  Label10: TLabel;
  Label11: TLabel;
  labelerro: TLabel;
  Button1: TButton;
  procedure BitBtn1Click(Sender: TObject);
```



```

    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
    procedure DBEdit3Exit(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    cadastro_apicultor: Tcadastro_apicultor;
implementation
uses Unit3, Unit1, Unit2, Unit5, Unit6, Unit7;
{$R *.dfm}
procedure Tcadastro_apicultor.BitBtn1Click(Sender: TObject);
begin
    INICIO.Visible:=TRUE;
    cadastro_apicultor.Close;
end;
procedure Tcadastro_apicultor.BitBtn3Click(Sender: TObject);
begin
    DataModule3.apicultor.Delete;
end;
procedure Tcadastro_apicultor.BitBtn2Click(Sender: TObject);
begin
    if (dbedit2.text = "") or (dbedit6.text='( ) - ') then
    begin
        messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
        dbedit2.SetFocus;
    end
    else
    begin

```

```

DataModule3.apicultor.POST;
messagedlg('Apicultor cadastrado com sucesso!', mtinformation, [mbok], 0);
DATAMODULE3.apicultor.Append;
end
end;

procedure Tcadastro_apicultor.BitBtn4Click(Sender: TObject);
begin
DATAMODULE3.apicultor.Append;
end;
procedure Tcadastro_apicultor.Timer1Timer(Sender: TObject);
var
vw_ano,vw_mes,vw_dia, vw_semana: word;
vs_diasem : String;
vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of

        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
        12:vs_mes:= 'Dezembro';
    end;
end;
vw_semana:= dayofweek(date);

```

```

case vw_semana of
1: vs_diasem := 'Domingo';
2: vs_diasem := 'Segunda';
3: vs_diasem := 'Terça';
4: vs_diasem := 'Quarta';
5: vs_diasem := 'Quinta';
6: vs_diasem := 'Sexta';
7: vs_diasem := 'Sábado';
end;

StatusBar1.Panels[1].Text:=' '+ vs_diasem + ', ' +inttostr(vw_dia) +' de '+ vs_mes + ' de '
+inttostr(vw_ano);
end;

procedure Tcadastro_apicultor.BitBtn5Click(Sender: TObject);
var
dig1,dig2,i,cal: integer;
digitos: array [1..11] of integer;
controle: boolean;
begin
controle := false;
For i := 1 to length(dbedit3.text) - 3 do
Begin
If i < 4 then
digitos[i] := StrToInt(dbedit3.text [i])
Else
If (i >= 4) and (i < 7) then
digitos[i] := StrToInt(dbedit3.text [i+1])
Else
If (i >= 7) and (i < 10) then
digitos[i] := StrToInt(dbedit3.text [i+2])
Else
If (i >= 10) then
digitos[i] := StrToInt(dbedit3.text [i+3]);
End;
// Verifica se os digitos são iguais

```

```

For i := 1 to 10 do
If digitos[i] <> digitos[i+1] then
Begin
controle := true;
break;
End;
If controle then
Begin
controle := false;
// Primeiro Dígito
cal := (10 * digitos[1]) + (9 * digitos[2]) + (8 * digitos[3]) + (7 * digitos[4]) + (6 * digitos[5]) +
(5 * digitos[6]) + (4 * digitos[7]) + (3 * digitos[8]) + (2 * digitos[9]);
dig1 := 11 - (cal mod 11);
if dig1 >= 10 then
dig1 := 0;
// Segundo Dígito
cal := (11 * digitos[1]) + (10 * digitos[2]) + (9 * digitos[3]) + (8 * digitos[4]) + (7 * digitos[5]) +
(6 * digitos[6]) + (5 * digitos[7]) + (4 * digitos[8]) + (3 * digitos[9]) + (2 * dig1);
dig2 := 11 - (cal mod 11);
If dig2 >= 10 then
dig2 := 0;
If ( dbedit3.text[13] = IntToStr(dig1) ) and (dbedit3.text [14] = IntToStr(dig2) ) then
Begin
LabelErro.Caption:='CPF Válido';
LabelErro.Color:=ClGreen;
LabelErro.Font.Color:=ClWhite;
End
Else
Begin
LabelErro.Caption:='CPF Inválido';
LabelErro.Color:=ClRed;
LabelErro.Font.Color:=ClWhite;
End;
End

```

```

Else
Begin
LabelErro.Caption:='CPF Inválido';
LabelErro.Color:=ClRed;
LabelErro.Font.Color:=ClWhite;
End;
End;

procedure Tcadastro_apicultor.DBEdit3Exit(Sender: TObject);
Var
dig1,dig2,i,cal : integer;
digitos : array [1..11] of integer;
controle : boolean;
Begin
controle := false;
For i := 1 to length(dbedit3.text) - 3 do
Begin
If i < 4 then
digitos[i] := StrToInt(dbedit3.text [i])
Else
If (i >= 4) and (i < 7) then
digitos[i] := StrToInt(dbedit3.text [i+1])
Else
If (i >= 7) and (i < 10) then
digitos[i] := StrToInt(dbedit3.text [i+2])
Else
If (i >= 10) then
digitos[i] := StrToInt(dbedit3.text [i+3]);
End;
// Verifica se os digitos são iguais
For i := 1 to 10 do
If digitos[i] <> digitos[i+1] then
Begin
controle := true;
break;

```

```

End;
If controle then
Begin
controle := false;
// Primeiro Dígito
cal := (10 * digitos[1]) + (9 * digitos[2]) + (8 * digitos[3]) + (7 * digitos[4]) + (6 * digitos[5]) +
(5 * digitos[6]) + (4 * digitos[7]) + (3 * digitos[8]) + (2 * digitos[9]);
dig1 := 11 - (cal mod 11);
if dig1 >= 10 then
dig1 := 0;
// Segundo Dígito
cal := (11 * digitos[1]) + (10 * digitos[2]) + (9 * digitos[3]) + (8 * digitos[4]) + (7 * digitos[5]) +
(6 * digitos[6]) + (5 * digitos[7]) + (4 * digitos[8]) + (3 * digitos[9]) + (2 * dig1);
dig2 := 11 - (cal mod 11);
If dig2 >= 10 then
dig2 := 0;
If ( dbedit3.text[13] = IntToStr(dig1) ) and (dbedit3.text [14] = IntToStr(dig2) ) then
Begin
LabelErro.Caption:='CPF Válido';
LabelErro.Color:=ClGreen;
LabelErro.Font.Color:=ClWhite;
End
Else
Begin
LabelErro.Caption:='CPF Inválido';
LabelErro.Color:=ClRed;
LabelErro.Font.Color:=ClWhite;
End;
End
Else
Begin
LabelErro.Caption:='CPF Inválido';
LabelErro.Color:=ClRed;
LabelErro.Font.Color:=ClWhite;

```

End;

End;

end.

unit Unit2;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, Grids, DBGrids, Mask, DBCtrls, ExtCtrls,
Menus, ComCtrls, jpeg;

type

Tcadastro_colmeia = class(TForm)

Label1: TLabel;

BitBtn1: TBitBtn;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

DBComboBox1: TDBComboBox;

DBComboBox2: TDBComboBox;

DBComboBox3: TDBComboBox;

DBComboBox4: TDBComboBox;

DBComboBox5: TDBComboBox;

DBComboBox6: TDBComboBox;

DBText1: TDBText;

BitBtn3: TBitBtn;

```

BitBtn4: TBitBtn;
Label2: TLabel;
DBLookupComboBox1: TDBLookupComboBox;
DBComboBox7: TDBComboBox;
DBComboBox8: TDBComboBox;
DBComboBox9: TDBComboBox;
DBComboBox10: TDBComboBox;
DBComboBox11: TDBComboBox;
DBEdit1: TDBEdit;
StatusBar1: TStatusBar;
Timer1: TTimer;
Image1: TImage;
Label15: TLabel;
DBEdit2: TDBEdit;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Label21: TLabel;
Label22: TLabel;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Colmia1Click(Sender: TObject);
procedure Apicultor1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
    { Private declarations }
public
    { Public declarations }
end;
var

```



```

    cadastro_colmeia: Tcadastro_colmeia;
implementation
uses Unit1, Unit3, Unit4, Unit5, Unit6, Unit7;
{$R *.dfm}
procedure Tcadastro_colmeia.BitBtn1Click(Sender: TObject);
begin
    cadastro_colmeia.Close;
end;
procedure Tcadastro_colmeia.BitBtn2Click(Sender: TObject);
begin
    DATAMODULE3.colmeia.Append;
end;
procedure Tcadastro_colmeia.BitBtn3Click(Sender: TObject);
begin
    if (dblookupcombobox1.keyvalue=null) or (dbcombobox3.text='') or (dbedit2.text='') or
    (dbcombobox7.text='') or (dbcombobox6.text='') then
    begin
        messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
        DBCombobox1.SetFocus;
    end
    else
    begin
        datamodule3.apiario.Edit;
        datamodule3.apiarion_colmeia.Value:=datamodule3.apiarion_colmeia.Value+1;
        datamodule3.apiario.Post;
        datamodule3.colmeiaCod_apiario.Text:=datamodule3.apiarioCod_apiario.Text;
        DATAMODULE3.colmeia.Post;
        dbedit1.Text:= datetostr(now);
        messagedlg('Colmeia cadastrada com sucesso!', mtinformation, [mbok], 0);
        DATAMODULE3.colmeia.Append;
    end;
end;
procedure Tcadastro_colmeia.BitBtn4Click(Sender: TObject);
begin

```

```

DATAMODULE3.colmeia.ClearFields;

end;

procedure Tcadastro_colmeia.Timer1Timer(Sender: TObject);
var
vw_ano,vw_mes,vw_dia, vw_semana: word;
vs_diasem : String;
vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of
        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
        12:vs_mes:= 'Dezembro';
    end;
    vw_semana:= dayofweek(date);
    case vw_semana of

        1: vs_diasem := 'Domingo';
        2: vs_diasem := 'Segunda';
        3: vs_diasem := 'Terça';
        4: vs_diasem := 'Quarta';
        5: vs_diasem := 'Quinta';
        6: vs_diasem := 'Sexta';
        7: vs_diasem := 'Sábado';
    end;
end;

```

```

end;
StatusBar1.Panels[1].Text:=' '+ vs_diasem + ', ' +inttostr(vw_dia) +' de '+ vs_mes + ' de '
+inttostr(vw_ano);
end;
procedure Tcadastro_colmeia.Colmia1Click(Sender: TObject);
begin
datamodule3.colmeia.append;
INICIO.VISIBLE:=FALSE;
cadastro_colmeia.ShowModal;
end;
procedure Tcadastro_colmeia.Apicultor1Click(Sender: TObject);
begin
datamodule3.colmeia.append;
INICIO.VISIBLE:=FALSE;
cadastro_apicultor.ShowModal;
end;
procedure Tcadastro_colmeia.FormShow(Sender: TObject);
begin
dbedit1.Text:= datetostr(now);
end;
procedure Tcadastro_colmeia.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
datamodule3.colmeia.Cancel;
end;
end.
unit Unit11;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Grids, DBGrids, StdCtrls, Buttons, DBCtrls, Mask, ComCtrls,
    ExtCtrls;
type
    Tcadastro_usuario = class(TForm)

```

```

Label1: TLabel;
Label2: TLabel;
DBEdit1: TDBEdit;
Label3: TLabel;
DBEdit2: TDBEdit;
Label4: TLabel;
DBEdit3: TDBEdit;
DBText1: TDBText;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
BitBtn5: TBitBtn;
Timer1: TTimer;
StatusBar1: TStatusBar;
Label7: TLabel;
Label6: TLabel;
Label8: TLabel;
DBGrid1: TDBGrid;
BitBtn3: TBitBtn;
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    cadastro_usuario: Tcadastro_usuario;
implementation
uses Unit1, Unit10, Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8, Unit9,
    Unit13;

```

```

{$R *.dfm}

procedure Tcadastro_usuario.BitBtn3Click(Sender: TObject);
begin
  DATAMODULE3.USUARIO.delete;
end;

procedure Tcadastro_usuario.BitBtn1Click(Sender: TObject);
begin
  if (dbedit1.text='') or (dbedit2.text='') or (dbedit3.text='') then
  begin
    messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
    DBedit1.SetFocus;
  end
  else
  begin
    DATAMODULE3.USUARIO.post;
    DATAMODULE3.USUARIO.Append;
    messagedlg('Usuario cadastrado com sucesso!', mtinformation, [mbok], 0);
  end
end;

procedure Tcadastro_usuario.BitBtn4Click(Sender: TObject);
begin
  cadastro_usuario.Visible:=false;
  gerenciamento_usuario.showmodal;
end;

procedure Tcadastro_usuario.BitBtn2Click(Sender: TObject);
begin
  DATAMODULE3.USUARIO.ClearFields;
end;

procedure Tcadastro_usuario.BitBtn5Click(Sender: TObject);
begin
  INICIO.Visible:=TRUE;
  cadastro_usuario.Close;
end;

procedure Tcadastro_usuario.Timer1Timer(Sender: TObject);

```

```

var
vw_ano,vw_mes,vw_dia, vw_semana: word;
vs_diasem : String;
vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of
        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
        12:vs_mes:= 'Dezembro';
    end;
    vw_semana:= dayofweek(date);
    case vw_semana of
        1: vs_diasem := 'Domingo';
        2: vs_diasem := 'Segunda';
        3: vs_diasem := 'Terça';
        4: vs_diasem := 'Quarta';
        5: vs_diasem := 'Quinta';
        6: vs_diasem := 'Sexta';
        7: vs_diasem := 'Sábado';
    end;
    StatusBar1.Panels[1].Text:=' ' + vs_diasem + ', ' +inttostr(vw_dia) +' de ' + vs_mes + ' de '
    +inttostr(vw_ano);
end;

```

```

end.
unit Unit5;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, Buttons, Mask, DBCtrls, ExtCtrls, DB,
  DBTables, ComCtrls, Menus;
type
  Tcontrole_apiarario = class(TForm)
    Label1: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    DBLookupComboBox1: TDBLookupComboBox;
    Label3: TLabel;
    DBGrid1: TDBGrid;
    Timer1: TTimer;
    StatusBar1: TStatusBar;
    BitBtn5: TBitBtn;
    procedure Button1Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure DBLookupComboBox1Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var

```

```

    controle_apiario: Tcontrole_apiario;
implementation
uses Unit1, Unit2, Unit3, Unit4, Unit6, Unit7, Unit9, Unit16;
{$R *.dfm}
procedure Tcontrole_apiario.Button1Click(Sender: TObject);
begin
    controle_apiario.Close;
end;

procedure Tcontrole_apiario.BitBtn1Click(Sender: TObject);
begin
    datamodule3.colmeia.Append;
    cadastro_colmeia.SHOWMODAL;
end;

procedure Tcontrole_apiario.BitBtn2Click(Sender: TObject);
begin
    DataModule3.colmeia.Delete;
end;

procedure Tcontrole_apiario.BitBtn3Click(Sender: TObject);
begin
    INICIO.Visible:=TRUE;
    Controle_apiario.Close;
end;

procedure Tcontrole_apiario.DBLookupComboBox1Click(Sender: TObject);
var
    cod_apiario:string;
begin
    cod_apiario:=datamodule3.apiarioCod_apiario.Text;
    datamodule3.colmeia.Close;
    datamodule3.colmeia.CommandText:=('select      *      from      colmeia      where
cod_apiario="'+cod_apiario+'" ');
    datamodule3.colmeia.open;
end;

procedure Tcontrole_apiario.Timer1Timer(Sender: TObject);

```



```

var
vw_ano,vw_mes,vw_dia, vw_semana: word;
vs_diasem : String;
vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of
        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
        12:vs_mes:= 'Dezembro';
    end;
    vw_semana:= dayofweek(date);
    case vw_semana of
        1: vs_diasem := 'Domingo';
        2: vs_diasem := 'Segunda';
        3: vs_diasem := 'Terça';
        4: vs_diasem := 'Quarta';
        5: vs_diasem := 'Quinta';
        6: vs_diasem := 'Sexta';
        7: vs_diasem := 'Sábado';
    end;
    StatusBar1.Panels[1].Text:=' ' + vs_diasem + ', ' +inttostr(vw_dia) +' de ' + vs_mes + ' de '
    +inttostr(vw_ano);
end;

```

```

procedure Tcontrole_apiario.BitBtn4Click(Sender: TObject);
begin
    datamodule3.apiario.Append;
    cadastro_apiario.showmodal;
end;

procedure Tcontrole_apiario.BitBtn5Click(Sender: TObject);
begin
    datamodule3.producao.Append;
    controle_producao.showmodal;
end;
end.

unit Unit15;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, ExtCtrls, ComCtrls, StdCtrls, Buttons, Grids, DBGrids, DB,
    ADODB, DBCtrls;

type
    Tcontrole_colheitas = class(TForm)
        Label1: TLabel;
        DBGrid1: TDBGrid;
        BitBtn1: TBitBtn;
        BitBtn2: TBitBtn;
        Label2: TLabel;
        Label3: TLabel;
        StatusBar1: TStatusBar;
        Timer1: TTimer;
        DBLookupComboBox1: TDBLookupComboBox;
        DBLookupComboBox2: TDBLookupComboBox;
        ADOQuery1: TADOQuery;
        ADOQuery2: TADOQuery;
        DataSource1: TDataSource;
        DataSource2: TDataSource;
        procedure BitBtn2Click(Sender: TObject);
    end;

```

```

procedure Timer1Timer(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure DBLookupComboBox1CloseUp(Sender: TObject);
procedure DBLookupComboBox2CloseUp(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  controle_colheitas: Tcontrole_colheitas;
implementation
uses Unit1, Unit12, Unit10, Unit11, Unit13, Unit14, Unit2, Unit3, Unit4,
  Unit5, Unit6, Unit7, Unit8, Unit9;
{$R *.dfm}
procedure Tcontrole_colheitas.BitBtn2Click(Sender: TObject);
begin
  inicio.Visible:=true;
  controle_colheitas.Close;
end;
procedure Tcontrole_colheitas.Timer1Timer(Sender: TObject);
var
  vw_ano,vw_mes,vw_dia, vw_semana: word;
  vs_diasem : String;
  vs_mes : String;
begin
  StatusBar1.Panels[0].Text:= '  ' + timetostr(time);
  decodedate(now(),vw_ano,vw_mes,vw_dia);
  case vw_mes of
    1: vs_mes:= 'Janeiro';
    2: vs_mes:= 'Fevereiro';
    3: vs_mes:= 'Março';
    4: vs_mes:= 'Abril';
    5: vs_mes:= 'Maio';

```

```

        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
        12:vs_mes:= 'Dezembro';
    end;
    vw_semana:= dayofweek(date);
    case vw_semana of
        1: vs_diasem := 'Domingo';
        2: vs_diasem := 'Segunda';
        3: vs_diasem := 'Terça';
        4: vs_diasem := 'Quarta';
        5: vs_diasem := 'Quinta';
        6: vs_diasem := 'Sexta';
        7: vs_diasem := 'Sábado';
    end;
    StatusBar1.Panels[1].Text:=' '+ vs_diasem + ', ' +inttostr(vw_dia) + ' de '+ vs_mes + ' de '
    +inttostr(vw_ano);
    end;
    procedure Tcontrole_colheitas.BitBtn1Click(Sender: TObject);
    begin
        if (dblookupcombobox1.Keyvalue=null) or (dblookupcombobox2.KeyValue=null) then
            begin
                messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
                dblookupcombobox1.SetFocus;
            end
        else
            INSERIR_COLETA.SHOWMODAL;
        end;
    procedure Tcontrole_colheitas.DBLookupComboBox1CloseUp(Sender: TObject);
    var
        codapiario: integer;

```

```

begin
codapiario:= Datamodule3.apiario['cod_apiario'];
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('SELECT * FROM apiario a, colmeia c WHERE ');
ADOQuery1.SQL.add('a.cod_apiario      =      c.cod_apiario      and      c.cod_apiario      =
''+inttostr(codapiario)+'');
ADOQuery1.Open;
end;

procedure Tcontrole_colheitas.DBLookupComboBox2CloseUp(Sender: TObject);
var
codcolmeia: integer;
begin
if not ADOQuery1.Eof then
begin
codcolmeia:= ADOQuery1['n_colm'];
ADOQuery2.Close;
ADOQuery2.SQL.Clear;
ADOQuery2.SQL.Add('SELECT R.N_COLM, R.DATA_coleta, r.producao, R.OBS
FROM colmeia c, coleta r WHERE ');
ADOQuery2.SQL.add('c.N_colm = r.N_colm and r.N_colm = ''+inttostr(codcolmeia)+' ');
ADOQuery2.Open;
end
else
showmessage('Apiario nao possui colmeias');
end;
end.

unit Unit6;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, Grids, DBGrids, DBCtrls, Mask, DB, ADODB,
ExtCtrls;

type

```

```

Tcontrole_equip = class(TForm)
  Label1: TLabel;
  BitBtn1: TBitBtn;
  Label3: TLabel;
  DBEdit2: TDBEdit;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  DBText1: TDBText;
  DBMemo1: TDBMemo;
  DBText2: TDBText;
  DBComboBox1: TDBComboBox;
  BitBtn2: TBitBtn;
  BitBtn3: TBitBtn;
  BitBtn4: TBitBtn;
  DBGrid1: TDBGrid;
  BitBtn5: TBitBtn;
  ADOQuery1: TADOQuery;
  DataSource1: TDataSource;
  ComboBox1: TComboBox;
  BitBtn6: TBitBtn;
  Button1: TButton;
  Button2: TButton;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  DBText3: TDBText;
  procedure BitBtn1Click(Sender: TObject);
  procedure BitBtn3Click(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);
  procedure BitBtn4Click(Sender: TObject);
  procedure ComboBox1CloseUp(Sender: TObject);
  procedure BitBtn5Click(Sender: TObject);

```

```

procedure BitBtn6Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure DBGrid1DbClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  controle_equip: Tcontrole_equip;
implementation
uses Unit5, Unit1, Unit2, Unit3, Unit4, Unit7, Unit8, Unit9;
{$R *.dfm}
procedure Tcontrole_equip.BitBtn1Click(Sender: TObject);
begin
  INICIO.Visible:=TRUE;
  controle_equip.Close;
end;
procedure Tcontrole_equip.BitBtn3Click(Sender: TObject);
begin
  datamodule3.equipamento.Filtered:= false;
  DATAMODULE3.equipamento.APPEND;
  Datamodule3.equipamento['quantidade']:= 1;
  DBCombobox1.ItemIndex:= 0;
end;
procedure Tcontrole_equip.BitBtn2Click(Sender: TObject);
begin
  if (DBEdit2.Text = "") or (DBMemo1.Text = "") or (DBCombobox1.ItemIndex = 0) then
  begin
    messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
    DBCombobox1.SetFocus;
  end
end

```

```

else
begin
DATAMODULE3.equipamento.POST;
messagedlg('Equipamento cadastrado com sucesso!', mtinformation, [mbok], 0);
end;
end;
procedure Tcontrole_equip.BitBtn4Click(Sender: TObject);
begin
DATAMODULE3.equipamento.DELETE;
end;
procedure Tcontrole_equip.ComboBox1CloseUp(Sender: TObject);
begin
ADOQUERY1.CLOSE;
adoquery1.SQL.clear;
adoquery1.SQL.Add('select nome as Nome, categoria as Categoria, quantidade as
Quantidade, descricao as Descrição from equipamento where categoria =
''' + combobox1.items[combobox1.itemindex] + ''');
adoquery1.Open;
end;
procedure Tcontrole_equip.BitBtn5Click(Sender: TObject);
begin
controle_equip.clientheight:=600;
bitbtn5.Visible:= false;
end;
procedure Tcontrole_equip.BitBtn6Click(Sender: TObject);
begin
controle_equip.clientheight:=365;
bitbtn5.Visible:= true;
end;
procedure Tcontrole_equip.Button1Click(Sender: TObject);
begin
Datamodule3.equipamento.Edit;
Datamodule3.equipamento['quantidade']:= Datamodule3.equipamento['quantidade'] + 1;
Datamodule3.equipamento.Post;

```



```

end;
procedure Tcontrole_equip.Button2Click(Sender: TObject);
begin
if datamodule3.equipamento['quantidade'] = 0 then
begin
    messagedlg('Quantidade não pode ser menor que 0', mterror, [mbok], 0);
end
else
begin
    Datamodule3.equipamento.Edit;
    Datamodule3.equipamento['quantidade']:= Datamodule3.equipamento['quantidade'] - 1;
    Datamodule3.equipamento.Post;
end;
end;
procedure Tcontrole_equip.DBGrid1DbClick(Sender: TObject);
begin
datamodule3.equipamento.Filter:= 'cod_equipamento='+
inttostr(ADOQuery1['cod_equipamento']);
datamodule3.equipamento.Filtered:= true;
end;
procedure Tcontrole_equip.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
datamodule3.equipamento.Filtered:= false;
end;
end.
unit Unit16;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Buttons, Grids, DBGrids, DBCtrls, DB, ADODB;
type
    Tcontrole_producao = class(TForm)
        BitBtn1: TBitBtn;

```

```

DBGrid1: TDBGrid;
BitBtn2: TBitBtn;
DBLookupComboBox1: TDBLookupComboBox;
Label1: TLabel;
ADOQuery1: TADOQuery;
DataSource1: TDataSource;
DBText1: TDBText;
ADOQuery2: TADOQuery;
DataSource2: TDataSource;
Label2: TLabel;
BitBtn3: TBitBtn;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure DBLookupComboBox1CloseUp(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  controle_producao: Tcontrole_producao;
implementation
uses Unit1, Unit10, Unit11, Unit12, Unit13, Unit14, Unit15, Unit2, Unit3,
  Unit4, Unit5, Unit6, Unit7, Unit8, Unit9, Unit17;
{$R *.dfm}
procedure Tcontrole_producao.BitBtn1Click(Sender: TObject);
begin
  INICIO.Visible:=TRUE;
  controle_producao.Close;
end;
procedure Tcontrole_producao.BitBtn2Click(Sender: TObject);
begin
  inserir_producao.showmodal;

```

```

end;

procedure Tcontrole_producao.DBLookupComboBox1CloseUp(Sender: TObject);
begin
if dblookupcombobox1.keyvalue=null then
exit
else
begin
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('SELECT data as Data,peso_mel as Mel_Kg FROM producao
WHERE cod_apiario = '+inttostr(dblookupcombobox1.KeyValue)+'');
ADOQuery1.Open;
ADOQuery1.Last;
{
ADOQuery2.Close;
ADOQuery2.SQL.Clear;
ADOQuery2.SQL.Add('SELECT avg(peso_mel) FROM producao WHERE cod_apiario =
'+inttostr(dblookupcombobox1.KeyValue)+'          AND          data          =
'+datetostr(ADOQuery1['data'])+' ');
ADOQuery2.Open;
}
end ;
end;

procedure Tcontrole_producao.BitBtn3Click(Sender: TObject);
begin
adoquery1.Delete;
end;
end.

unit Unit7;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, DBCtrls, Grids, DBGrids, ComCtrls, ExtCtrls,
DB, ADODB;

```

```

type
  Tcontrole_revisao = class(TForm)
    Label2: TLabel;
    BitBtn2: TBitBtn;
    Label3: TLabel;
    Timer1: TTimer;
    StatusBar1: TStatusBar;
    BitBtn1: TBitBtn;
    Label1: TLabel;
    DBGrid1: TDBGrid;
    Label4: TLabel;
    DBLookupComboBox1: TDBLookupComboBox;
    ADOQuery1: TADOQuery;
    DataSource1: TDataSource;
    DBLookupComboBox2: TDBLookupComboBox;
    ADOQuery2: TADOQuery;
    DataSource2: TDataSource;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure DBLookupComboBox1CloseUp(Sender: TObject);
    procedure DBLookupComboBox2CloseUp(Sender: TObject);
  private
    { Private declarations }
  public
  end;
var
  controle_revisao: Tcontrole_revisao;
implementation
uses Unit6, Unit1, Unit2, Unit3, Unit4, Unit5, Unit8, Unit12, Unit10,
  Unit11, Unit9;
{$R *.dfm}
procedure Tcontrole_revisao.BitBtn1Click(Sender: TObject);
begin

```

```

inicio.Visible:=true;
controle_revisao.Close;
end;
procedure Tcontrole_revisao.BitBtn2Click(Sender: TObject);
begin
if (dblookupcombobox1.Keyvalue=null) or (dblookupcombobox2.KeyValue=null) then
begin
messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
dblookupcombobox1.SetFocus;
end
else
controle_visita.SHOWMODAL;
end;
procedure Tcontrole_revisao.Timer1Timer(Sender: TObject);
var
vw_ano,vw_mes,vw_dia, vw_semana: word;
vs_diasem : String;
vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of
        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';

        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
    end
end;

```

```

12:vs_mes:= 'Dezembro';
end;
vw_semana:= dayofweek(date);
case vw_semana of
1: vs_diasem := 'Domingo';
2: vs_diasem := 'Segunda';
3: vs_diasem := 'Terça';
4: vs_diasem := 'Quarta';
5: vs_diasem := 'Quinta';
6: vs_diasem := 'Sexta';
7: vs_diasem := 'Sábado';
end;
StatusBar1.Panels[1].Text:= ' ' + vs_diasem + ', ' +inttostr(vw_dia) +' de ' + vs_mes + ' de '
+inttostr(vw_ano);
end;
procedure Tcontrole_revisao.DBLookupComboBox1CloseUp(Sender: TObject);
var
codapiario: integer;
begin
codapiario:= Datamodule3.apiario['cod_apiario'];
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('SELECT * FROM apiario a, colmeia c WHERE ');
ADOQuery1.SQL.add('a.cod_apiario      =      c.cod_apiario      and      c.cod_apiario      =
''+inttostr(codapiario)+'''');
ADOQuery1.Open;
end;

procedure Tcontrole_revisao.DBLookupComboBox2CloseUp(Sender: TObject);
VAR
codcolmeia: integer;
begin
if not ADOQuery1.Eof then
begin

```

```

codcolmeia:= ADOQuery1['n_colm'];
ADOQuery2.Close;
ADOQuery2.SQL.Clear;
ADOQuery2.SQL.Add('SELECT                R.AVALIACAO,          R.DATA_REV,
R.DATA_PROXIMAREVISAO, R.OBS FROM colmeia c, revisao r WHERE ');
ADOQuery2.SQL.add('c.N_colm = r.N_colm and r.N_colm = '"+inttostr(codcolmeia)+'" ');
ADOQuery2.Open;
end
else
  showmessage('Apiario nao possui colmeias');
end;
end.
Unit Unit8;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, DBCtrls, Buttons, ComCtrls, ExtCtrls;
type
  Tcontrole_visita = class(TForm)
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    lbcolm: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Label1: TLabel;
    Timer1: TTimer;
    StatusBar1: TStatusBar;
    Label6: TLabel;
    eddata: TEdit;
    cbstadocolm: TComboBox;
    cbproxrevisao: TComboBox;
    MMobs: TMemo;
    procedure BitBtn1Click(Sender: TObject);
  end;
end.

```

```

    procedure BitBtn2Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormShow(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    controle_visita: Tcontrole_visita;
implementation
uses Unit3, Unit1, Unit2, Unit4, Unit5, Unit6, Unit7, Unit10, Unit11,
    Unit12, Unit13, Unit14, Unit15, Unit9;
{$R *.dfm}
procedure Tcontrole_visita.BitBtn1Click(Sender: TObject);
begin
    controle_visita.CLOSE;
end;
procedure Tcontrole_visita.BitBtn2Click(Sender: TObject);
begin
    //NOMETABELA['CAMPO']:=EDIT.TEXT
    DATAMODULE3.revisao.INSERT;
    datamodule3.revisao['avaliacao']:=cbestadocolm.Text;
    datamodule3.revisao['data_proximarevisao']:= now + strtoint(cbproxrevisao.Text);
    datamodule3.revisao['data_rev']:=eddata.text;
    datamodule3.revisao['obs']:=MMobs.text;
    datamodule3.revisao['n_colm']:= strtoint(lbcolm.caption);
    DATAMODULE3.revisao.Post;
    controle_visita.Close;
    Controle_revisao.ADOQuery2.close;
    Controle_revisao.ADOQuery2.open;
end;
procedure Tcontrole_visita.Timer1Timer(Sender: TObject);
var

```



```

vw_ano,vw_mes,vw_dia, vw_semana: word;
vs_diasem : String;
vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of
        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
        12:vs_mes:= 'Dezembro';
    end;
    vw_semana:= dayofweek(date);
    case vw_semana of
        1: vs_diasem := 'Domingo';
        2: vs_diasem := 'Segunda';
        3: vs_diasem := 'Terça';
        4: vs_diasem := 'Quarta';
        5: vs_diasem := 'Quinta';
        6: vs_diasem := 'Sexta';
        7: vs_diasem := 'Sábado';
    end;
    StatusBar1.Panels[1].Text:=' ' + vs_diasem + ', ' +inttostr(vw_dia) +' de ' + vs_mes + ' de '
    +inttostr(vw_ano);
end;
procedure Tcontrole_visita.FormShow(Sender: TObject);

```

```

begin
lbcolm.Caption:= CONTROLE_REVISAO.ADOQuery1['N_COLM'];
eddata.Text:= datetostr(now);
end;
end.
unit Unit19;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, RpDefine, RpRave, ExtCtrls, jpeg;
type
  Tfichas = class(TForm)
    BitBtn1: TBitBtn;
    RvProject1: TRvProject;
    RvProject2: TRvProject;
    RvProject3: TRvProject;
    RvProject4: TRvProject;
    imgcolheita: TImage;
    RadioGroup1: TRadioGroup;
    RadioButton1: TRadioButton;
    RadioButton2: TRadioButton;
    RadioButton3: TRadioButton;
    RadioButton4: TRadioButton;
    BitBtn7: TBitBtn;
    imgrevisao: TImage;
    imgcolmeia: TImage;
    imgapiario: TImage;
    Label1: TLabel;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn7Click(Sender: TObject);
    procedure RadioButton4Click(Sender: TObject);
    procedure RadioButton3Click(Sender: TObject);
  end;

```

```

    procedure RadioButton1Click(Sender: TObject);
    procedure RadioButton2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    fichas: Tfichas;
implementation
uses Unit1, Unit10, Unit11, Unit12, Unit13, Unit14, Unit15, Unit16, Unit17,
    Unit18, Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8, Unit9;
{$R *.dfm}
procedure Tfichas.BitBtn1Click(Sender: TObject);
begin
    fichas.Close;
end;
procedure Tfichas.BitBtn2Click(Sender: TObject);
begin
    rvproject1.Execute;
end;
procedure Tfichas.BitBtn3Click(Sender: TObject);
begin
    rvproject2.Execute;
end;
procedure Tfichas.BitBtn7Click(Sender: TObject);
begin
    if radiobutton4.Checked=true then
        rvproject3.Execute;
    if radiobutton3.Checked=true then
        rvproject4.Execute;
    if radiobutton1.Checked=true then
        rvproject1.Execute;
    if radiobutton2.Checked=true then

```

```
rvproject2.Execute;
end;
procedure Tfichas.RadioButton4Click(Sender: TObject);
begin
if radiobutton4.Checked=true then
begin
imgapiario.Visible:=true;
imgcolmeia.Visible:=false;
imgrevisao.Visible:=false;
imgcolheita.Visible:=false;
end
end;
procedure Tfichas.RadioButton3Click(Sender: TObject);
begin
if radiobutton3.Checked=true then
begin
imgapiario.Visible:=false;
imgcolmeia.Visible:=true;
imgrevisao.Visible:=false;
imgcolheita.Visible:=false;
end
end;
procedure Tfichas.RadioButton1Click(Sender: TObject);
begin
if radiobutton1.Checked=true then
begin
imgapiario.Visible:=false;
imgcolmeia.Visible:=false;
imgrevisao.Visible:=true;
imgcolheita.Visible:=false;
end
end;
procedure Tfichas.RadioButton2Click(Sender: TObject);
begin
```

```

if radiobutton2.Checked=true then
begin
imgapiario.Visible:=false;
imgcolmeia.Visible:=false;
imgrevisao.Visible:=false;
imgcolheita.Visible:=true;
end
end;
end.
unit Unit1;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Buttons, Menus, ComCtrls, ExtCtrls, jpeg, AppEvnts,
    XPMAN;
type
    TINICIO = class(TForm)
        BitBtn6: TBitBtn;
        StatusBar1: TStatusBar;
        Image1: TImage;
        Timer1: TTimer;
        ApplicationEvents1: TApplicationEvents;
        MainMenu1: TMainMenu;
        Sair1: TMenuItem;
        CadastrodeApiario1: TMenuItem;
        Apiario1: TMenuItem;
        Colmeia1: TMenuItem;
        Apicultor1: TMenuItem;
        Controles1: TMenuItem;
        Revises1: TMenuItem;
        Colheitas1: TMenuItem;
        ApiariosColmeias1: TMenuItem;
        Usuario1: TMenuItem;
        FichasdeControle1: TMenuItem;
    end;

```

```

Relatorios1: TMenuItem;
Equipamentos1: TMenuItem;
Produo1: TMenuItem;
Image2: TImage;
XPManifest1: TXPManifest;
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure BitBtn9Click(Sender: TObject);
procedure Colmia1Click(Sender: TObject);
procedure Apicultor1Click(Sender: TObject);
procedure Apiario1Click(Sender: TObject);
    procedure Produo1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure Sair1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure BitBtn10Click(Sender: TObject);
procedure ApplicationEvents1Exception(Sender: TObject; E: Exception);
procedure BitBtn11Click(Sender: TObject);
procedure Colmeia1Click(Sender: TObject);
procedure Equipamentos1Click(Sender: TObject);
procedure Revises1Click(Sender: TObject);
procedure Colheitas1Click(Sender: TObject);
procedure ApiariosColmeias1Click(Sender: TObject);
procedure Usuario1Click(Sender: TObject);
procedure Relatorios1Click(Sender: TObject);
procedure FichasdeControle1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }

```

```

end;
var
    INICIO: TINICIO;
implementation
uses Unit2, Unit4, Unit5, Unit6, Unit7, Unit3, Unit9, Unit8, Unit10,
    Unit11, Unit14, Unit15, Unit12, Unit13, Unit16, Unit18, Unit19;
{$R *.dfm}
procedure TINICIO.BitBtn6Click(Sender: TObject);
begin
    APPLICATION.TERMINATE;
end;
procedure TINICIO.BitBtn1Click(Sender: TObject);
begin
    datamodule3.colmeia.append;
    INICIO.VISIBLE:=FALSE;
    cadastro_colmeia.ShowModal;
end;
procedure TINICIO.BitBtn3Click(Sender: TObject);
begin
    INICIO.VISIBLE:=FALSE;
    datamodule3.apicultor.append;
    cadastro_apicultor.SHOWMODAL;
end;
procedure TINICIO.BitBtn2Click(Sender: TObject);
begin
    INICIO.VISIBLE:=FALSE;
    Controle_apiario.SHOWMODAL;
end;
procedure TINICIO.BitBtn8Click(Sender: TObject);
begin
    controle_equip.ShowModal;
end;
procedure TINICIO.BitBtn7Click(Sender: TObject);
begin

```

```

INICIO.VISIBLE:=FALSE;
controle_revisao.SHOWMODAL;
end;
procedure TINICIO.BitBtn9Click(Sender: TObject);
begin
INICIO.Visible:=FALSE;
datamodule3.apiario.append;
cadastro_apiario.SHOWMODAL;
end;
procedure TINICIO.Colmia1Click(Sender: TObject);
begin
datamodule3.colmeia.append;
INICIO.VISIBLE:=FALSE;
cadastro_colmeia.ShowModal;
end;
procedure TINICIO.Apicultor1Click(Sender: TObject);
begin
datamodule3.apicultor.append;
cadastro_apicultor.SHOWMODAL;
end;
procedure TINICIO.Apiario1Click(Sender: TObject);
begin
datamodule3.apiario.append;
cadastro_apiario.SHOWMODAL;
end;
procedure TINICIO.Produo1Click(Sender: TObject);
begin

controle_producao.SHOWMODAL;
end;
procedure TINICIO.Timer1Timer(Sender: TObject);
var
vw_ano,vw_mes,vw_dia, vw_semana: word;
vs_diasem : String;

```



```

vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of
        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
        10:vs_mes:= 'Outubro';
        11:vs_mes:= 'Novembro';
        12:vs_mes:= 'Dezembro';
    end;
    vw_semana:= dayofweek(date);
    case vw_semana of
        1: vs_diasem := 'Domingo';
        2: vs_diasem := 'Segunda';
        3: vs_diasem := 'Terça';
        4: vs_diasem := 'Quarta';
        5: vs_diasem := 'Quinta';
        6: vs_diasem := 'Sexta';
        7: vs_diasem := 'Sábado';
    end;
    StatusBar1.Panels[1].Text:=' ' + vs_diasem + ', ' +inttostr(vw_dia) +' de ' + vs_mes + ' de '
    +inttostr(vw_ano);
end;
procedure TINICIO.Sair1Click(Sender: TObject);
begin
    APPLICATION.TERMINATE;

```

```

end;

procedure TINICIO.FormCreate(Sender: TObject);
begin
login:=tlogin.Create(application);
login.showmodal;
end;

procedure TINICIO.BitBtn10Click(Sender: TObject);
begin
login_cad_usuario.showmodal;
end;

procedure TINICIO.ApplicationEvents1Exception(Sender: TObject;
E: Exception);
var
mensagem: string;
Pos1, Pos2: integer;
begin
if Pos(UpperCase('is not a valid date'), UpperCase(E.Message)) <> 0 then
mensagem:='Data inválida, proceda a correção.'
else if Pos(UpperCase('must have a value'), UpperCase(E.Message)) <> 0 then
begin
Pos1:=Pos(' ', E.Message);
mensagem:=E.Message;
Delete(mensagem, Pos1, 1);
Pos2:=Pos(' ', mensagem);
mensagem:=copy(E.Message, Pos1 + 1, Pos2 - Pos1);
mensagem := 'É obrigatório o preenchimento do campo '+ mensagem + ' ';
end
else if Pos(UpperCase('key violation'), UpperCase(E.Message)) <> 0 then
mensagem := 'Houve violação de Chave. Registro já incluído.'
else if Pos(UpperCase('is not a valid time'), UpperCase(E.Message)) <> 0 then
mensagem := 'Hora inválida, proceda a correção.'
else if Pos(UpperCase('is not a valid float'), UpperCase(E.Message)) <> 0 then
begin
Pos1 :=Pos(' ', E.Message);

```

```

mensagem := E.Message;
Delete(mensagem, Pos1, 1);
Pos2 := Pos("", mensagem);
mensagem := copy(E.Message, Pos1 + 1, Pos2 - Pos1);
mensagem := 'O valor ' + mensagem + ' não é válido.';
end
else if Pos(UpperCase('field value required'), UpperCase(E.Message)) <> 0 then
begin
Pos1 := Pos('column ', E.Message) + 7;
Pos2 := Pos(',', E.Message);
mensagem := copy(E.Message, Pos1, Pos2 - Pos1);
mensagem := 'Campo ' + mensagem + ' deve ser preenchido.';
end
else if Pos(UpperCase('ATTEMPT TO STORE DUPLICATE VALUE'),
UpperCase(E.Message)) <> 0
then
mensagem := 'Não é permitido valor duplicado. '
else if Pos(UpperCase('FOREIGN KEY'), UpperCase(E.Message)) <> 0 then
mensagem := 'Operação não permitida, registro vinculado em outra tabela está faltando.'
else if Pos('VIOLATION OF PRIMARY OR UNIQUE KEY CONSTRAINT',
UpperCase(E.Message)) <> 0
then
mensagem := 'Registro
Duplicado' + #13#10 + Copy(UpperCase(E.Message), Pos('VIOLATION OF PRIMARY OR
UNIQUE KEY CONSTRAINT', UpperCase(E.Message)) + 47, 100)
else if (Pos(UpperCase('MUST APPLY UPDATES BEFORE REFRESHING DATA'),
UpperCase(E.Message)) <> 0) then
mensagem := 'É necessário aplicara as alterações antes de atualizar os dados.'
else if (Pos(UpperCase('INVALID INPUT VALUE'), UpperCase(E.Message)) <> 0) then
mensagem := 'Valor digitado não é valido conforme a máscara.'
else
mensagem := 'Ocorreu o seguinte erro: ' + #13 + UpperCase(E.Message);
MessageDlg(mensagem, mtError, [mbOk], 0);
ABORT;

```

```

end;

procedure TINICIO.BitBtn11Click(Sender: TObject);
begin
    CONTROLE_colheitas.SHOWMODAL;
end;

procedure TINICIO.Colmeia1Click(Sender: TObject);
begin
    datamodule3.colmeia.append;
    cadastro_colmeia.ShowModal;
end;

procedure TINICIO.Equipamentos1Click(Sender: TObject);
begin
    datamodule3.equipamento.append;
    Datamodule3.equipamento['quantidade']:= 1;
    controle Equip.DBCombobox1.ItemIndex:= 0;
    controle Equip.ShowModal;
end;

procedure TINICIO.Revises1Click(Sender: TObject);
begin
    controle_revisao.SHOWMODAL;
end;

procedure TINICIO.Colheitas1Click(Sender: TObject);
begin
    CONTROLE_colheitas.SHOWMODAL;
end;

procedure TINICIO.ApiariosColmeias1Click(Sender: TObject);
begin
    Controle_apiario.SHOWMODAL;
end;

procedure TINICIO.Usuario1Click(Sender: TObject);
begin
    login_cad_usuario.showmodal;
end;

```

```

procedure TINICIO.Relatorios1Click(Sender: TObject);
begin
relatorio.showmodal;
end;
procedure TINICIO.FichasdeControle1Click(Sender: TObject);
begin
fichas.showmodal;
end;
end.
unit Unit12;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls, ComCtrls, DBCtrls, Mask;
type
  Tinserrir_coleta = class(TForm)
    BitBtn1: TBitBtn;
    StatusBar1: TStatusBar;
    Timer1: TTimer;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    BitBtn2: TBitBtn;
    PRODEDIT: TComboBox;
    DCEDIT: TEdit;
    MOBS: TMemo;
    PCEDIT: TEdit;
    Label6: TLabel;
    LCOLM: TLabel;
  procedure BitBtn1Click(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure BitBtn2Click(Sender: TObject);

```

```

    procedure FormShow(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    inserir_coleta: Tinsrerir_coleta;
implementation
uses Unit3, Unit15, Unit1, Unit10, Unit11, Unit13, Unit14, Unit2, Unit4,
    Unit5, Unit6, Unit7, Unit8, Unit9;
{$R *.dfm}
procedure Tinsrerir_coleta.BitBtn1Click(Sender: TObject);
begin
    INSERIR_COLETA.Close;
end;
procedure Tinsrerir_coleta.Timer1Timer(Sender: TObject);
var
    vw_ano,vw_mes,vw_dia, vw_semana: word;
    vs_diasem : String;
    vs_mes : String;
begin
    StatusBar1.Panels[0].Text:= ' ' + timetostr(time);
    decodedate(now(),vw_ano,vw_mes,vw_dia);
    case vw_mes of
        1: vs_mes:= 'Janeiro';
        2: vs_mes:= 'Fevereiro';
        3: vs_mes:= 'Março';
        4: vs_mes:= 'Abril';
        5: vs_mes:= 'Maio';
        6: vs_mes:= 'Junho';
        7: vs_mes:= 'Julho';
        8: vs_mes:= 'Agosto';
        9: vs_mes:= 'Setembro';
    end;
end;

```

```

10:vs_mes:= 'Outubro';
11:vs_mes:= 'Novembro';
12:vs_mes:= 'Dezembro';
end;
vw_semana:= dayofweek(date);
case vw_semana of
1: vs_diasem := 'Domingo';
2: vs_diasem := 'Segunda';
3: vs_diasem := 'Terça';
4: vs_diasem := 'Quarta';
5: vs_diasem := 'Quinta';
6: vs_diasem := 'Sexta';
7: vs_diasem := 'Sábado';
end;
StatusBar1.Panels[1].Text:=' '+ vs_diasem + ', ' +inttostr(vw_dia) +' de '+ vs_mes + ' de '
+inttostr(vw_ano);
end;
procedure Tinserir_coleta.BitBtn2Click(Sender: TObject);
begin
if (dcedit.Text = '') or (prodedit.ItemIndex = 0) then
begin
messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
prodedit.SetFocus;
end
else
begin
DATAMODULE3.COLETA.INSERT;
datamodule3.COLETA['PRODUCAO']:=PRODEDIT.Text;
datamodule3.COLETA['data_proxcolheita']:= pcedit.text;
datamodule3.COLETA['data_COLETA']:=strtodate(DCEDIT.text);
datamodule3.COLETA['obs']:=Mobs.text;
datamodule3.COLETA['n_colm']:= strtoint(lcolm.caption);

DATAMODULE3.COLETA.Post;

```

```

INSERIR_COLETA.Close;
Controle_COLHEITAS.ADOQuery2.close;
Controle_COLHEITAS.ADOQuery2.open;
end
end;

procedure Tinsrer_coleta.FormShow(Sender: TObject);
begin
prodedit.Itemindex:=0;
mobs.Clear;
lcolm.Caption:= CONTROLE_COLHEITAS.ADOQuery1['N_COLM'];
DCEDIT.Text:= datetostr(now);
pcedit.Text:=datetostr(strtodate(dccedit.Text)+ 180);
end;
end.

unit Unit17;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, DBCtrls, Mask;
type
  Tinsrer_producao = class(TForm)
    BitBtn1: TBitBtn;
    Label2: TLabel;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit4: TDBEdit;
    DBText1: TDBText;
    BitBtn2: TBitBtn;
    DBLookupComboBox1: TDBLookupComboBox;
    Label9: TLabel;
    Label11: TLabel;
    Label5: TLabel;
    Label6: TLabel;

```



```

procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  inserir_producao: Tinsrerir_producao;
implementation
uses Unit3, Unit1, Unit10, Unit11, Unit12, Unit13, Unit14, Unit15, Unit16,
  Unit2, Unit4, Unit5, Unit6, Unit7, Unit8, Unit9;
{$R *.dfm}
procedure Tinsrerir_producao.BitBtn1Click(Sender: TObject);
begin
  inserir_producao.Close;
end;
procedure Tinsrerir_producao.BitBtn2Click(Sender: TObject);
begin
  if (dbedit4.Text = "") or (dbedit3.text="") then
  begin
    messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
    dbedit4.setfocus;
  end
  else
  begin
    datamodule3.producao['cod_apiario']:= DbLookupCombobox1.KeyValue;
    datamodule3.producao.Post;
    controle_producao.ADOQuery1.Close;
    controle_producao.ADOQuery1.SQL.Clear;
    controle_producao.ADOQuery1.SQL.Add('SELECT * FROM producao WHERE cod_apiario
    = '"+inttostr(dblockupcombobox1.KeyValue)+"");
  end
end;

```

```

controle_producao.ADOQuery1.Open;
messagedlg('Equipamento cadastrado com sucesso!', mtinformation, [mbok], 0);
DATAMODULE3.producao.Append;
end
end;

procedure Tinsirir_producao.FormShow(Sender: TObject);
begin
    datamodule3.producao.Insert;
    DBLookupComboBox1.KeyValue:= controle_producao.DblookupCombobox1.KeyValue;
    dbedit3.Text:=datetostr(now);
    dbedit4.SetFocus;
end;

procedure Tinsirir_producao.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
    Datamodule3.producao.Cancel;
end;

end.

unit Unit10;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, DB, DBTables, Grids, DBGrids, ADODB, jpeg, ExtCtrls;

type

    Tlogin = class(TForm)
        Edit1: TEdit;
        Edit2: TEdit;
        Button1: TButton;
        Button2: TButton;
        ADOQuery1: TADOQuery;
        DataSource1: TDataSource;
        Label1: TLabel;
        Label2: TLabel;
        Image1: TImage;
    end;

```

```

    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    login: Tlogin;
implementation
uses Unit1, Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8, Unit9;
{$R *.dfm}
procedure Tlogin.Button2Click(Sender: TObject);
begin
    application.Terminate;
end;
procedure Tlogin.Button1Click(Sender: TObject);
begin
    adoquery1.Close;
    adoquery1.SQL.Clear;
    adoquery1.SQL.Add('select * from usuario where login="'+edit1.Text+'" and
senha="'+edit2.text+'"');
    adoquery1.Open;
    adoquery1.Active:=true;

    if adoquery1.Eof then
        showmessage('Usuário invalido')
    else
        login.close;
    end;
procedure Tlogin.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    if key= #13 then

```

```

button1click(sender);
end;
end.
unit Unit14;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Buttons, DB, ADODB, jpeg, ExtCtrls;

type
    Tlogin_cad_usuario = class(TForm)
        DataSource1: TDataSource;
        ADOQuery1: TADOQuery;
        BitBtn1: TBitBtn;
        BitBtn2: TBitBtn;
        Edit1: TEdit;
        Edit2: TEdit;
        Label1: TLabel;
        Label2: TLabel;
        Image1: TImage;
        procedure BitBtn1Click(Sender: TObject);
        procedure BitBtn2Click(Sender: TObject);
        procedure Edit2KeyPress(Sender: TObject; var Key: Char);
        procedure FormShow(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    login_cad_usuario: Tlogin_cad_usuario;

implementation

```

```
uses Unit1, Unit10, Unit11, Unit12, Unit13, Unit2, Unit3, Unit4, Unit5,
    Unit6, Unit7, Unit8, Unit9;
```

```
{ $R *.dfm }
```

```
procedure Tlogin_cad_usuario.BitBtn1Click(Sender: TObject);
begin
    adoquery1.Close;
    adoquery1.SQL.Clear;
    adoquery1.SQL.Add('select * from usuario where login="'+edit1.Text+'" and
        senha="'+edit2.text+'"');
    adoquery1.Open;
    adoquery1.Active:=true;
```

```
if adoquery1.Eof then
begin
    showmessage('Usuário invalido');
    edit1.clear;
    edit2.clear;
end
else
begin
    cadastro_usuario.Visible:=true;
    datamodule3.usuario.append;
    login_cad_usuario.close;
end;
end;
```

```
procedure Tlogin_cad_usuario.BitBtn2Click(Sender: TObject);
begin
    login_cad_usuario.close;
end;

procedure Tlogin_cad_usuario.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
```

```

if key= #13 then
bitbtn1click(sender);
end;
procedure Tlogin_cad_usuario.FormShow(Sender: TObject);
begin
edit1.Clear;
edit2.Clear;
end;
end.
unit Unit18;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, Buttons, RpRave, RpDefine, RpCon, RpConDS, DB, ADODB,
    ExtCtrls, Grids, DBGrids, DBCtrls;
type
    Trelatorio = class(TForm)
        BitBtn1: TBitBtn;
        RvDataSetConnection1: TRvDataSetConnection;
        RvProject1: TRvProject;
        RadioGroup1: TRadioGroup;
        RadioButton1: TRadioButton;
        RadioButton2: TRadioButton;
        RadioButton3: TRadioButton;
        RadioButton4: TRadioButton;
        RadioButton5: TRadioButton;
        ADOQuery1: TADOQuery;
        DataSource1: TDataSource;
        BitBtn2: TBitBtn;
        DBLookupComboBox1: TDBLookupComboBox;
        DBLookupComboBox2: TDBLookupComboBox;
        DBLookupComboBox3: TDBLookupComboBox;
        DBLookupComboBox4: TDBLookupComboBox;
        DBLookupComboBox5: TDBLookupComboBox;

```

```

DBLookupComboBox6: TDBLookupComboBox;
ADOQuery2: TADOQuery;
DataSource2: TDataSource;
RvProject2: TRvProject;
RvProject3: TRvProject;
RvProject4: TRvProject;
RvProject5: TRvProject;
Label1: TLabel;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure DBLookupComboBox2CloseUp(Sender: TObject);
procedure DBLookupComboBox4CloseUp(Sender: TObject);
procedure RadioButton2Click(Sender: TObject);
procedure RadioButton3Click(Sender: TObject);
procedure RadioButton4Click(Sender: TObject);
procedure RadioButton5Click(Sender: TObject);
procedure DBLookupComboBox3CloseUp(Sender: TObject);
procedure DBLookupComboBox5CloseUp(Sender: TObject);
procedure DBLookupComboBox6CloseUp(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  relatorio: Trelatorio;

implementation

uses Unit1, Unit10, Unit11, Unit12, Unit13, Unit14, Unit15, Unit16, Unit17,
  Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8, Unit9;

{$R *.dfm}

```

```
procedure Trelatorio.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
relatorio.close;
```

```
end;
```

```
procedure Trelatorio.BitBtn2Click(Sender: TObject);
```

```
begin
```

```
if radiobutton1.checked = true then
```

```
begin
```

```
adoquery1.Close;
```

```
adoquery1.SQL.Clear;
```

```
adoquery1.SQL.add('select * from Apiario');
```

```
adoquery1.Open;
```

```
RvProject1.Execute;
```

```
end;
```

```
if radiobutton2.checked = true then
```

```
begin
```

```
if (dblookupcombobox1.KeyValue=null)then
```

```
begin
```

```
messedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
```

```
DBlookupcombobox1.SetFocus;
```

```
end
```

```
else
```

```
begin
```

```
adoquery1.Close;
```

```
adoquery1.SQL.Clear;
```

```
adoquery1.SQL.add('select * from colmeia where
```

```
cod_apiario='"+inttostr(dblookupcombobox1.keyvalue)+" '');
```

```
adoquery1.Open;
```

```
RvProject2.Execute;
```

```
end;
```

```
end;
```



```

if radiobutton3.checked = true then
begin
    if (dblookupcombobox2.KeyValue=null)then
    begin
        messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
        DBlookupcombobox2.SetFocus;
    end
else
    begin
        adoquery1.Close;
        adoquery1.SQL.Clear;
        adoquery1.SQL.add('select * from producao p, apiario a where p.cod_apiario=
'+inttostr(dblookupcombobox2.keyvalue)+' and a.cod_apiario = p.cod_apiario ');
        adoquery1.Open;
        RvProject3.Execute;
    end;
end;
if radiobutton4.checked = true then
begin
    if (dblookupcombobox3.KeyValue=null) or (dblookupcombobox4.KeyValue=null) then
    begin
        messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
        DBlookupcombobox3.SetFocus;
    end
else
    begin
        adoquery1.Close;
        adoquery1.SQL.Clear;
        adoquery1.SQL.add('SELECT c.identificador, R.* FROM colmeia c, coleta r WHERE
c.N_colm = r.N_colm and r.N_colm ='+inttostr(dblookupcombobox4.keyvalue)+' ');
        adoquery1.Open;
        RvProject4.Execute;
    end;
end;

```

```

if radiobutton5.checked = true then
begin
    if (dblookupcombobox5.KeyValue=null) or (dblookupcombobox6.KeyValue=null) then
    begin
        messagedlg('Complete os campos obrigatórios!', mterror, [mbok], 0);
        DBlookupcombobox5.SetFocus;
    end
    else
    begin
        adoquery1.Close;
        adoquery1.SQL.Clear;
        adoquery1.SQL.add('SELECT      c.identificador,      R.N_COLM,      R.AVALIACAO,
R.DATA_REV, R.DATA_PROXIMAREVISAO, R.OBS FROM colmeia c, revisao r
WHERE c.N_colm = r.N_colm and r.N_colm ='+inttostr(dblookupcombobox6.keyvalue)+'
');
        adoquery1.Open;
        RvProject5.Execute;
    end;
end;
end;

procedure Trelatorio.DBLookupComboBox2CloseUp(Sender: TObject);
var
codapiario:integer;
begin
ADOQuery2.Close;
ADOQuery2.SQL.Clear;
ADOQuery2.SQL.Add('SELECT * FROM apiario a, colmeia c WHERE ');
ADOQuery2.SQL.add('a.cod_apiario      =      c.cod_apiario      and      c.cod_apiario      =
'+inttostr(codapiario)+'');
ADOQuery2.Open;
end;

procedure Trelatorio.DBLookupComboBox4CloseUp(Sender: TObject);
var
codcolmeia:integer;

```

```

begin
if not ADOQuery1.Eof then
begin
//a codcolmeia:= ADOQuery1['n_colm'];
ADOQuery2.Close;
ADOQuery2.SQL.Clear;
ADOQuery2.SQL.Add('SELECT  R.N_COLM,  R.DATA_coleta,  r.producao,  R.OBS
FROM colmeia c, coleta r WHERE ');
ADOQuery2.SQL.add('c.N_colm = r.N_colm and r.N_colm = '"+inttostr(codcolmeia)+'" ');
ADOQuery2.Open;
end;
end;
procedure Trelatorio.RadioButton2Click(Sender: TObject);
begin
if radiobutton2.Checked=true then
begin
dblookupcombobox1.Enabled:=true;
dblookupcombobox2.Enabled:=false;
dblookupcombobox3.Enabled:=false;
dblookupcombobox4.Enabled:=false;
dblookupcombobox5.Enabled:=false;
dblookupcombobox6.Enabled:=false;
end;
end;
procedure Trelatorio.RadioButton3Click(Sender: TObject);
begin
if radiobutton3.Checked=true then
begin
dblookupcombobox1.Enabled:=false;
dblookupcombobox2.Enabled:=true;
dblookupcombobox3.Enabled:=false;
dblookupcombobox4.Enabled:=false;
dblookupcombobox5.Enabled:=false;
dblookupcombobox6.Enabled:=false;

```

```
end;
end;
procedure Trelatorio.RadioButton4Click(Sender: TObject);
begin
if radiobutton4.Checked=true then
begin
dblookupcombobox1.Enabled:=false;
dblookupcombobox2.Enabled:=false;
dblookupcombobox3.Enabled:=true;
dblookupcombobox4.Enabled:=false;
dblookupcombobox5.Enabled:=false;
dblookupcombobox6.Enabled:=false;
end;
end;
procedure Trelatorio.RadioButton5Click(Sender: TObject);
begin
if radiobutton5.Checked=true then
begin
dblookupcombobox1.Enabled:=false;
dblookupcombobox2.Enabled:=false;
dblookupcombobox3.Enabled:=false;
dblookupcombobox4.Enabled:=false;
dblookupcombobox5.Enabled:=true;
dblookupcombobox6.Enabled:=false;
end;
end;
procedure Trelatorio.DBLookupComboBox3CloseUp(Sender: TObject);
begin
begin
dblookupcombobox4.Enabled:=true;
end;
end;
procedure Trelatorio.DBLookupComboBox5CloseUp(Sender: TObject);
begin
```

```

dblookupcombobox6.Enabled:=true;
end;
procedure Trelatorio.DBLookupComboBox6CloseUp(Sender: TObject);
VAR
codcolmeia: integer;
begin
if not ADOQuery1.Eof then
begin
//codcolmeia:= DBLookupComboBox6.KeyValue;
ADOQuery1.Close;
ADOQuery1.SQL.Clear;
ADOQuery1.SQL.Add('SELECT    c.identificador,    R.N_COLM,    R.AVALIACAO,
R.DATA_REV, R.DATA_PROXIMAREVISAO, R.OBS FROM colmeia c, revisao r
WHERE ');
ADOQuery1.SQL.add('c.N_colm    =    r.N_colm    and    r.N_colm    =
'+inttostr(DbLookupComboBox6.KeyValue)+' ');
ADOQuery1.Open;
end
end ;
end.

```