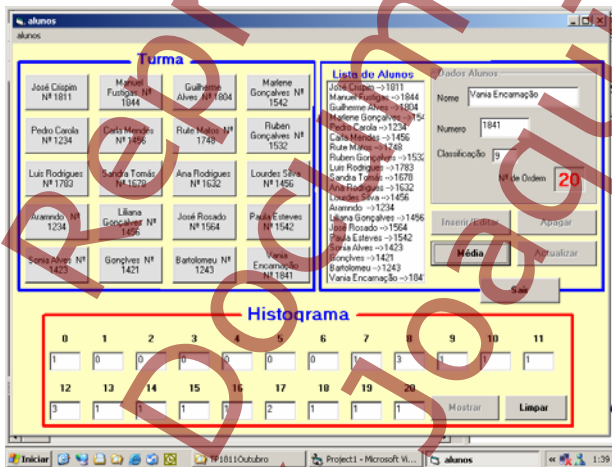


Aplicação em VB 6.0 Gestão de Classificações



Autor:

❖ 1811 - José Joaquim Rosado Crispim

**Instituto Politécnico de Setúbal
Escola Superior de Ciências Empresariais**

Licenciatura Bi-etápica em Gestão de Sistemas de Informação

**Ano Lectivo de 2001/2002
2º Ano 3º Semestre**

Trabalho Final Prático no Âmbito da Disciplina de

Linguagens de Programação I

Docente: Engº Luís Coelho

**Aplicação em VB
Gestão de Classificações**

Autor:

❖ **1811 – José Joaquim Rosado Crispim**

Setúbal, 2002.Outubro.24

Índice

Introdução	2
Manual do Utilizador	3
Manual Técnico	7
Código do Programa	11

Introdução

O Visual Basic é uma linguagem de programação destinada à criação de aplicações em ambiente Windows.

A parte "Visual" refere-se ao método usado para criar a interface gráfica do utilizador (GUI). Ao invés de escrever muitas linhas de código de programação para descrever a aparência e localização de elementos da interface, o programador apenas necessita de arrastar e soltar objectos pré-montados em locais da interface.

A parte "Basic" refere-se à linguagem BASIC (**B**eginners **A**ll-Purpose **S**ymbolic **I**nstruction **C**ode), uma linguagem usada por mais programadores que qualquer outra linguagem na história da computação. O Visual Basic evoluiu da linguagem BASIC original, contendo várias centenas de instruções, funções e palavras-chave, muitas das quais directamente relacionadas com a GUI do Windows.

Quando o Visual Basic para Windows foi lançado em 1991, era uma linguagem de desenvolvimento de propósito geral que oferecia aos programadores uma alternativa para lidar com as complicações do C/C++ e a API do Windows. Naquela altura, a linguagem C/C++ era o único meio de criar uma aplicação Windows boa e eficiente, e por isso o VB sofreu muito no seu início. Infelizmente, o acesso a bases de dados foi uma ideia bem posterior. Na verdade, a versão 1.0 não oferecia nenhum recurso de acesso a base de dados, somente acesso aos rudimentares arquivos ASCII.

O VB evoluiu e passou a ser uma óptima linguagem para acesso a Bases de Dados. Reconhecendo essa tendência, a Microsoft investiu em massa no Visual Basic, aprimorando-o e tornando-o numa poderosa ferramenta para acesso a bases de dados. Utilizando a moderna tecnologia ActiveX e mais uma gama de conjunto de extensões de terceiros, os aplicativos criados em VB que eram restritos aos computadores, estão hoje espalhados por toda a Internet, sendo utilizado por um grande número de utilizadores.

Do exposto se conclui a grande importância desta linguagem de programação e o motivo pelo qual ela foi incluída no Curso de Licenciatura em Gestão de Sistemas de Informação, o qual o autor frequenta.

No projecto que é apresentado, são aplicados os métodos e técnicas de programação em Visual Basic ministrados na Disciplina de Linguagens de Programação I, no âmbito da qual é efectuada a *Aplicação de Gestão de Classificações de Alunos* em conformidade com o enunciado fornecido pelo Docente da Disciplina, Professor Luís Coelho.

Neste contexto, o presente divide-se em 3 partes. Na primeira, é apresentado um Manual de Utilizador, no qual se descrevem as potencialidades da Aplicação de Gestão de Alunos e a forma de o potencial utilizador lidar com a mesma. A segunda parte constitui um Manual Técnico sendo apresentado e explicado o desenvolvimento de todo o programa incidindo em aspectos relativos ao VB. Por último é apresentada a listagem de todo o programa, devidamente legendada e dividida em blocos.

Manual do Utilizador

A Aplicação de *Gestão de Classificações de Alunos* é um programa desenvolvido em Visual Basic, versão 6.0, que corre em ambiente Windows e destina-se a gerir as classificações de 20 alunos, disponibilizando os seguintes serviços ao utilizador:

- **Botões independentes** para cada aluno, nos quais constam o nome e número do aluno.
- **Consulta de dados** de cada aluno através da simples selecção do botão adstrito ao aluno.
- **Inserção, Edição, Alteração ou Eliminação** dos dados de cada aluno. Neste caso, os dados só se tornarão efectivos após selecção do botão **Actualizar**.
 - Nota: Os Botões *Inserir/Editar, Apagar e Actualizar*, só ficam activos quando é seleccionado um botão de aluno
- Cálculo de **média** aritmética da Turma, com duas casas decimais. O programa efectua o cálculo de acordo com o número de alunos introduzidos. Por exemplo, se só estiverem introduzidos três alunos, a média é efectuada tendo em consideração apenas os três alunos (ou seja, soma as classificações e divide por 3, ignorando os restantes campos).
- **Lista de Alunos**, na qual é mostrada uma listagem dos alunos introduzidos.
- **Histograma** de classificações, no qual são mostradas as quantidades de classificações da Turma
 - Nota: o botão *Limpar* só fica activo após seleccionar o botão *Mostrar*.
- Após seleccionar o botão **Sair**, os dados introduzidos são registados em ficheiro e o programa termina. Os dados mencionados ficam disponíveis quando se activar o programa.
- Menus pendentes de:
 - Ficheiro/Sair, Alunos/Média/Histograma.
- Menu de Contexto de Média e Histograma, acessível através da selecção através do botão direito do rato na Zona de Lista de Alunos e Histograma.
- O programa obriga à introdução de dados. Caso isto não se verifique, quando for seleccionado o botão *Média* o programa mostra uma caixa de diálogo avisando que "*Tem de Introduzir Dados*".
- Está igualmente considerada a hipótese de todos os alunos introduzidos terem classificação zero. Neste caso, quando é seleccionado o botão *media*, o botão mostra uma caixa de diálogo avisando que "*X alunos seleccionados mas todos têm nota zero logo Média = 0*".

Após seleccionar o ficheiro *GestaoAlunos* contido na Pasta "Crispim" do disco C, é apresentada a seguinte interface:

Menus Pendentes

- Ficheiro
- Alunos
- Sair/Guardar
- Media
- Histograma

Botões Individuais para alunos

Botões de:

- Minimizar
- Diminuir/Maximizar
- Fechar

Turma

Nº 0	Nº 0	Nº 0	Nº 0
Nº 0	Nº 0	Nº 0	Nº 0
Nº 0	Nº 0	Nº 0	Nº 0
Nº 0	Nº 0	Nº 0	Nº 0
Nº 0	Nº 0	Nº 0	Nº 0

Lista de Alunos

```

->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0
->0

```

Botões de:

- Inserir/Editar
- Apagar
- Média
- Actualizar
- Sair

Histograma

0	1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20			

Botões de:

- Mostrar Histograma
- Limpar Histograma

Botões de:

- Inserir/Editar Alunos
- Apagar Alunos
- Actualizar Alunos
- Média Turma
- Sair/Gravar Registos

Quantidades de Classificações

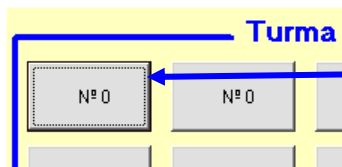
Menu de Contexto

- Seleccionando, com o botão direito do rato, as Zonas *Lista de Alunos* e *Histograma*

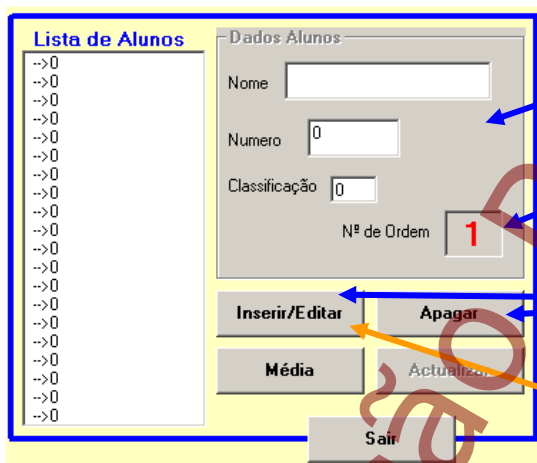
© José Joaquim Rosado Crispim - Outubro de 2002 - ESCE/IPS - TP Final de LPI

Inserir Dados:

Para inserir dados, proceda da seguinte forma:

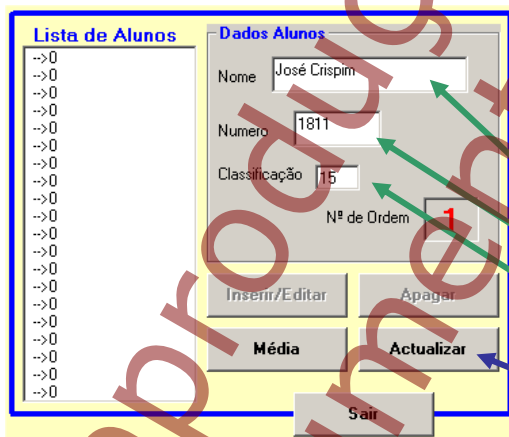


1 – Com o botão esquerdo do rato, seleccione um dos botões da zona **Turma**



- Automaticamente fica activa a Caixa onde vai inserir os dados dos alunos
- É mostrado o Número do botão Seleccionado (a contar da esquerda para a direita e de cima para baixo)
- Os botões *Inserir/Editar* e *Apagar* ficam activos

2 – Com o botão Esquerdo do Rato, seleccione o botão **Inserir/Editar**

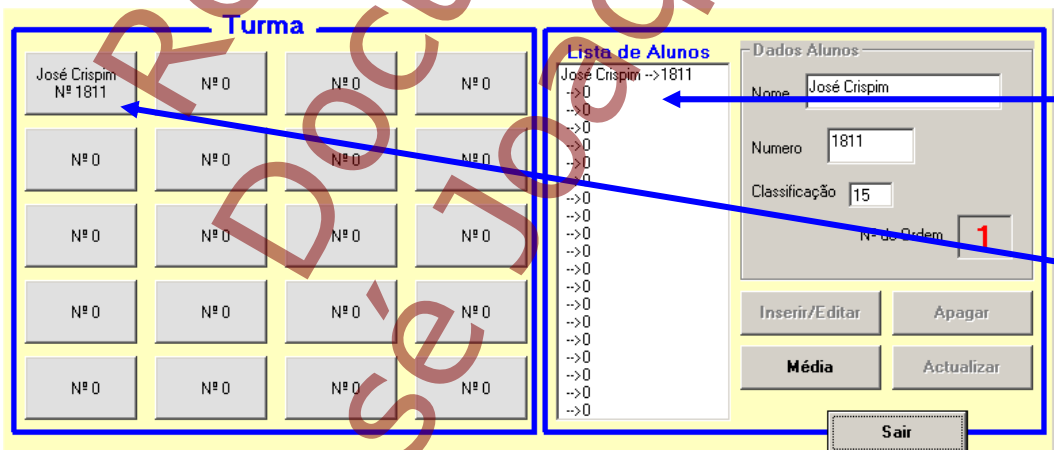


Os Botões *Inserir/Editar* e *Apagar* ficam inactivos

3 - Introduza os dados do Aluno, seleccionando com o botão esquerdo do rato na caixa respectiva:

- Em **Nome** - Primeiro e último nome
- Em **Número** - O número do aluno
- Em **Classificação** - Classificação do aluno

4 – Com o botão esquerdo do rato, seleccione o botão **Actualizar**



Na lista de alunos ficaram registados o Nome e o Número do Aluno

No botão seleccionado ficaram registados o Nome e o Número do Aluno

Editar/Alterar Dados:

Para alterar dados, proceda da mesma forma explicada anteriormente, seleccionando o botão do aluno pretendido, e depois o botão **Inserir/Editar**.

Seleccione a(s) caixa(s) do(s) dados(s) que pretende alterar e seleccione o botão **Actualizar**.

A **Lista de Alunos** e Botão do Aluno, serão actualizados após a acção anterior.

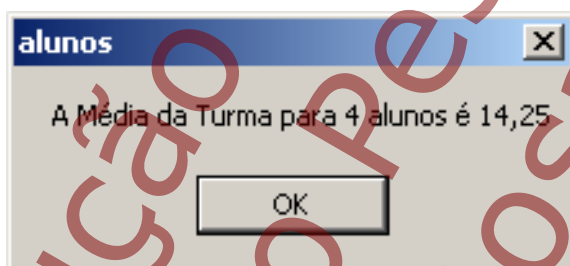
Apagar Dados:

Para apagar dados, o procedimento é idêntico. Seleccione o aluno a Apagar e seleccione o botão **Apagar** e seguidamente o botão **Actualizar**

Consulta da Média da Turma:

Após introduzir mais alguns dados de alunos, seleccione o botão **Média**

Aparecerá uma caixa de diálogo, com a média dos alunos introduzidos



Consulta do Histograma:

Com o botão esquerdo do rato, seleccione o botão **Mostrar** da zona Histograma.



No histograma acima, verificamos que existe 1 aluno com classificação 12, 1 com 13, 1 com 15 e 1 com 16. Para limpar o histograma seleccione o botão **Limpar**

Nota: Pode efectuar as consultas referidas através do menu **Alunos** ou através do menu de contexto seleccionando as Zonas **Lista de Alunos** ou **Histograma**, com o **botão direito** do rato.

Para sair do programa seleccione o botão **Sair** e o programa termina, mantendo a integridade dos dados actualizados, os quais serão apresentados quando iniciar uma nova sessão

Manual Técnico

Aplicação:

A aplicação foi desenvolvida em Visual Basic, Versão 6.0, a qual é uma linguagem orientada para **objectos**, onde estes possuem **propriedades**, executam **métodos** e respondem a **eventos**.

A presente aplicação (projecto) tem o nome de alunos, tendo sido gerado um ficheiro executável com o nome GestaoAlunos.

Objectos e Classes:

Como objecto base temos uma **forma (form alunos)**, na qual estão inseridos os restantes objectos de 4 classes diferentes, cuja descrição é a seguinte:

- ✓ Classe ListBox:
 - List1
- ✓ Classe CommandButton:
 - Command1 (0 a 20) (Alunos), Command2(Inserir), Command3(Apagar), Command4 (Mostrar), Command5 (Média), Command6 (Limpar), Command7 (Actualizar), command8 (Sair).
- ✓ Classe Frame:
 - Frame1
- ✓ Classe Label:
 - Label1 a Label30
- ✓ Classe mnu:
 - mnuAlunos, mnuFicheiro, mnuHistograma, mnuMedia, mnuSairGardar,
- ✓ Classe Line:
 - Line1 a Line15
- ✓ Classe TextBox:
 - Text1 a Text4 (0 a 20)

Como já foi referido, os objectos têm propriedades que correspondem à sua **característica** e que por sua vez assumem determinados valores a que damos o nome de **Atributos**.

Propriedades:

Quando a aplicação arranca, as propriedades **Enabled** e **Visible** da **Frame1** têm o atributo **false** e a propriedade **Enabled** dos **Command2** (Botão Inserir), **Command3** (Botão Apagar), **Command6** (Botão Limpar) e **Command7** (Botão Actualizar) tem o atributo **False**.

Após o utilizador seleccionar o Botão de Inserir (Command2) a aplicação liberta os controlos referidos, para que o utilizador tenha acesso à Frame para introdução dos dados.

Métodos:

Métodos são funções disponíveis no Visual Basic que levam o objecto a executar uma determinada acção. Assim, no objecto **List1** são utilizados os métodos **AddItem** de forma a serem adicionados os nomes dos alunos à lista e o método **Clear** de forma a que a Lista seja Limpa.

Eventos:

Eventos são acontecimentos que são activados por uma determinada acção do utilizador. No presente caso, o evento utilizado é o **click**, quando o utilizador selecciona com o botão do rato um determinado botão da interface (Objecto) este desenrola uma determinada acção.

Módulos:

O código de programação no Visual Basic está organizado em módulos, que podem assumir três tipos diferentes:

- Form Modules
- Standard Modules
- Class Modules

Estes guardam dados introduzidos através de declarações e o código através de procedimentos. Nesta aplicação foi utilizado apenas o Form Module, pois trata-se de uma aplicação de pequena dimensão.

Dados:

No Visual Basic existem diversos tipos de dados, os quais podem ou não ser declarados. Os que foram utilizados na presente aplicação são os seguintes:

- Integer – variável numérica que assume valores entre -32.768 a 32.767
- String – variável de carácter que vai aproximadamente até 2 milhões de caracteres.
- Array – são listas de dados, podendo ser vectores ou multi-dimensionais. Por defeito a base é zero, mas pode ser alterado para base 1. Outra característica é podermos definir o limite inferior da matriz.
- Definidas pelo Utilizador – Taluno.

Procedimentos:

Procedimentos são blocos de código, que facilitam a estruturação de referido, e dividem-se em três grupos:

- **Sub procedures** – dividem-se em **event** procedures e **general** procedures. No primeiro caso estão ligados a um determinado evento e no segundo embora não estando ligados a qualquer evento podem ser acedidos por estes ou por outros em qualquer altura.
- **Function procedures**
- **Property procedures.**

Os procedimentos podem ser **Públicos** e serem acedidos por outros módulos de programação, ou **privados** e serem acedidos apenas dentro do respectivo módulo.

A presente aplicação é constituída por 24 procedimentos, cujo detalhe é apresentado seguidamente:

- ✓ **Option Explicit** – na primeira linha de código parece este procedimentos, o qual vai obrigar a declarar todas as variáveis e a primeira variável declarada em **general** da form é uma array designada por **Talunos**. Esta é declarada como **private** por ser de um tipo definido pelo programador e está inserida num módulo form e apenas pode ser acedida por este módulo.
- ✓ **Private Sub Form_Load** – general procedure accionada pelo programador que procede à abertura do ficheiro de dados através da rotina **carregadados** (que carregará os dados existentes ou não), e define a forma como a interface virá a ser apresentada, nomeadamente com o histograma limpo (limpahist), com a frame invisível e dissabled (Frame1.Visible = False, Frame1.Enabled = False), e os botões de Inserir, Apagar, Actualizar e limpar (do Histograma) inactivos (Command2.Enabled = False, Command3.Enabled = False, Command6.Enabled = False, Command7.Enabled = False).
- ✓ **Sub carregadados** – abre o ficheiro **TURMAxyz.INI** e carrega os dados existentes no referido
- ✓ **Sub GUARDA** – procede à operação inversa da interior, ou seja, guarda os dados existentes na aplicação e fecha a aplicação. Esta rotina será executada quando o utilizador seleccionar o botão Guardar.
- ✓ **Sub InserirRegisto** – Activa a frame onde vão ser introduzidos os dados pelo utilizador (Frame1.Enabled = True), desactiva os botões Inserir e Apagar (Command2.Enabled = False, Command3.Enabled = False) e activa o botão Actualizar (Command7.Enabled = True). Esta rotina será activada quando o utilizador seleccionar um dos botões de alunos.
- ✓ **Sub ApagarRegisto** – Quando o utilizador seleccionar o botão apagar, esta rotina colocará o campo Nome da Form em branco (Text1.Text = "") e os campos Número e Classificação com Zero (Text2.Text = "0", Text3.Text = "0"). Igualmente inactiva os Botões de Inserir e Apagar (Command2.Enabled = False, Command3.Enabled = False) e activa o botão de Actualizar (Command7.Enabled = True).

- ✓ **Sub ActMatriz** – Quando for seleccionado o botão actualizar, esta rotina procederá à desactivação dos Botões Inserir/Editar, Apagar e Actualizar (Command2.Enabled = False, Command3.Enabled = False, Command7.Enabled = False) e à desactivação dos campos da frame, embora mantendo-a visível (Frame1.Enabled = False).
- ✓ **Sub actlista** – Rotina que actualiza os campos da matriz Lista de Alunos.
- ✓ **Sub Mostrabot** – Rotina que coloca o Nome e Número do Aluno no Botão respectivo.
- ✓ **Sub medias** – Rotina que calcula a média da turma, quando é activado o botão média.
- ✓ **Sub histograma** – rotina que mostra os dados no Histograma quando é accionado o botão Mostrar.
- ✓ **Sub limpahist** – Rotina que limpa o Histograma quando é seleccionado o botão Limpar mostrando uma caixa de texto de confirmação.
- ✓ **Private Sub Command1** – general procedure que define as acções a tomar quando for seleccionado um dos botões de Aluno. Mostra o conteúdo das caixas de texto da frame, e activa os botões Inserir e Apagar.
- ✓ **Private Sub Command2_Click** – executa a rotina InserirRegisto, quando é seleccionado o botão Inserir/Editar.
- ✓ **Private Sub Command3_Click** - executa a rotina ApagarRegisto, quando é seleccionado o botão Apagar.
- ✓ **Private Sub Command7_Click** - executa a rotina ActMatriz, quando é seleccionado o botão Actualizar.
- ✓ **Private Sub Command5_Click** - executa a rotina medias, quando é seleccionado o botão Media.
- ✓ **Private Sub Command8_Click** - executa a rotina GUARDA, quando é seleccionado o botão Sair.
- ✓ **Private Sub Command4_Click** - executa a rotina histograma, quando é seleccionado o botão Mostrar (Histograma).
- ✓ **Private Sub Command4_Click** - executa a rotina limpahist, quando é seleccionado o botão Limpar (Histograma), mostrando uma caixa de diálogo de confirmação.
- ✓ **Private Sub mnuFicheiro_Click** – menu pendente ficheiro.
- ✓ **Private Sub mnuhistograma_Click** – Menu Pendente Histograma. Executa a rotina histograma
- ✓ **Private Sub mnumedia_Click** – Menu Pendente Média. Executa a rotina medias
- ✓ **Private Sub mnuSairGardar_Click** – Menu pendente Sair – executa a rotina GUARDA e fecha o programa (End).
- ✓ **Private Sub Form_MouseUp** – mostra o menu de contexto quando o utilizador selecciona o clique direito do rato (If Button = 2) numa das zonas da interface.

```
'
```

```
'Obrigatoriedade de declarar as variaveis
```

```
'
```

```
Option Explicit
```

```
'
```

```
'Variavel defenida pelo utilizador
```

```
'
```

```
Private Type Taluno
```

```
    Nome As String
```

```
    numero As Integer
```

```
    nota As Integer
```

```
End Type
```

```
'
```

```
' Não pode ser uma variável Global porque é definida pelo
```

```
' utilizador e não se encontra num Modulo Standard, apenas
```

```
' é visível neste modulo
```

```
'
```

```
Dim alunos(19) As Taluno
```

```
'
```

```
' Variável numérica Global é visível em todas as funções e
```

```
' procedimentos existentes em todos os módulos de uma aplicação
```

```
'
```

```
Public index, botao As Integer
```

```
'
```

```
'Procedimento de arranque
```

```
'
```

```
Private Sub Form_Load()
```

```
    index = 0
```

```
    carregadados
```

```
    limpahist
```

```
    Mostrabot (index)
```

```
    actlista (index)
```

```
    Frame1.Visible = False
```

```
    Frame1.Enabled = False
```

```
    Command2.Enabled = False
```

```
    Command3.Enabled = False
```

```
    Command6.Enabled = False
```

```
    Command7.Enabled = False
```

```
End Sub
```

```
'  
' Início do Programa - Abre e lê sequencialmente ficheiro turmaxyz.ini  
'
```

```
Sub carregados()
```

```
'  
' variável de carácter (-1)- declarada dentro do procedimento  
' só é reconhecida no próprio procedimento)
```

```
' (Dim ou Private)
```

```
Dim dados As String
```

```
Open "C:\Crispim\TURMAxyz.INI" For Input As #1
```

```
While Not EOF(1)
```

```
Input #1, alunos(index).Nome, alunos(index).numero, alunos(index).nota
```

```
index = index + 1
```

```
Wend
```

```
Close #1
```

```
index = 0
```

```
End Sub
```

```
'  
' Fim do Programa - Guarda alterações no ficheiro turmaxyz.INI  
'
```

```
Sub GUARDA()
```

```
Open "C:\Crispim\TURMAxyz.INI" For Output As #1
```

```
For index = 0 To 19
```

```
Print #1, alunos(index).Nome + "," + CStr(alunos(index).numero) + "," +  
CStr(alunos(index).nota)
```

```
Next index
```

```
Close #1
```

```
index = 0
```

```
End Sub
```

'Rotina para Inserir Registo

```
Sub InserirRegisto()  
    Frame1.Enabled = True  
    Command2.Enabled = False  
    Command3.Enabled = False  
    Command7.Enabled = True  
End Sub
```

'Rotina para Apagar Registo

```
Sub ApagarRegisto()  
    Text1.Text = ""  
    Text2.Text = "0"  
    Text3.Text = "0"  
    Command2.Enabled = False  
    Command3.Enabled = False  
    Command7.Enabled = True  
End Sub
```

'Rotina para Actualizar matriz

```
Sub ActMatriz()  
    Frame1.Enabled = False  
    Command2.Enabled = False  
    Command3.Enabled = False  
    Command7.Enabled = False  
    alunos(botao).Nome = Text1.Text  
    alunos(botao).numero = Text2.Text  
    alunos(botao).nota = Text3.Text  
    Mostrabot (index)  
    actlista (index)  
End Sub
```

```
' Rotina para Limpar, Encher e Limpar a Lista
```

```
Sub actlista(index As Integer)
```

```
' Limpa a lista
```

```
List1.Clear
```

```
' Enche a lista
```

```
For index = 0 To 19
```

```
' Actualiza a lista
```

```
List1.AddItem alunos(index).Nome + " -->" + CStr(alunos(index).numero)
```

```
Next
```

```
index = 0
```

```
End Sub
```

```
' Rotina para colocar o nome do aluno nos botões
```

```
Sub Mostrabot(index As Integer)
```

```
For index = 0 To 19
```

```
' coloca nome dos produtos nos botões
```

```
Command1(index).Caption = alunos(index).Nome + " N° " + CStr(alunos(index).numero)
```

```
Next
```

```
index = 0
```

```
End Sub
```

```
'  
' Rotina para efectuar a medias dos alunos existentes  
' na matriz Alunos
```

```
'  
' variáveis numéricas (2)  
'
```

```
Sub medias()
```

```
Dim Media, contador, validos As Single
```

```
Media = 0: contador = 0
```

```
For contador = 0 To 19
```

```
    If Len(alunos(contador).Nome) > 0 Then
```

```
        Media = Media + alunos(contador).nota
```

```
        validos = validos + 1
```

```
    End If
```

```
Next contador
```

```
If validos > 0 Then
```

```
    If Media = 0 Then
```

```
        MsgBox CStr(validos) + " alunos seleccionados mas todos têm nota zero  
logo Média = 0"
```

```
    Else
```

```
        MsgBox "A Média da Turma para " + CStr(validos) + " alunos é " +  
CStr(Round(Media / validos, 2))
```

```
    End If
```

```
    Else
```

```
        MsgBox "Tem de Introduzir Dados"
```

```
    End If
```

```
End Sub
```

```
'  
' Rotina para mostrar Histograma  
'
```

```
Sub histograma()  
'
```

```
' Variáveis numéricas (2)  
'
```

```
Dim contador As Single  
'
```

```
' Vector apenas visível neste procedimento  
'
```

```
Dim mathist(20) As Integer
```

```
contador = 0  
'
```

```
' Enche matriz com nota igual por índice  
'
```

```
For contador = 0 To 19
```

```
    If Len(alunos(contador).Nome) > 0 Then
```

```
        mathist(alunos(contador).nota) = mathist(alunos(contador).nota) + 1
```

```
    End If
```

```
Next contador  
'
```

```
' mostra histograma  
'
```

```
For contador = 0 To 20
```

```
    Text4(contador).Text = CStr(mathist(contador))
```

```
Next contador
```

```
Command6.Enabled = True
```

```
Command4.Enabled = False
```

```
End Sub
```

```
' Rotina para limpar histograma
```

```
Sub limpahist()
```

```
Dim contador As Integer
```

```
contador = 0
```

```
For contador = 0 To 20
```

```
Text4(contador).Text = " "
```

```
Next contador
```

```
End Sub
```

```
' Botões Alunos
```

```
Private Sub Command1_Click(index As Integer)
```

```
Text1.Text = alunos(index).Nome
```

```
Text2.Text = alunos(index).numero
```

```
Text3.Text = alunos(index).nota
```

```
botao = index
```

```
Label26.Caption = CStr(botao + 1)
```

```
Frame1.Visible = True
```

```
Command2.Enabled = True
```

```
Command3.Enabled = True
```

```
End Sub
```

```
' Botão inserir registo
```

```
Private Sub Command2_Click()
```

```
InserirRegisto
```

```
End Sub
```

```
' Botão apagar registo
```

```
Private Sub Command3_Click()
```

```
ApagarRegisto
```

```
End Sub
```

' Botão Actualizar

Private Sub Command7_Click()

 ActMatriz

End Sub

' Botão Média

Private Sub Command5_Click()

 medias

End Sub

' Botão Sair do Programa

Private Sub Command8_Click()

 GUARDA

End

End Sub

' Botão Mostrar Histograma

Private Sub Command4_Click()

 histograma

End Sub

```
'  
'Botão Limpar do Histograma  
'
```

```
Private Sub Command6_Click()
```

```
    Dim resposta As Integer
```

```
    resposta = MsgBox("Tem a certeza ? ", vbYesNo, "--- Histograma ---")
```

```
    If resposta = vbYes Then
```

```
' Procedimento para Limpar o histograma  
'
```

```
    limpahist
```

```
    Command6.Enabled = False
```

```
    Command4.Enabled = True
```

```
    End If
```

```
End Sub
```

```
'Menus Pendentes  
'
```

```
Private Sub mnuFicheiro_Click()
```

```
End Sub
```

```
Private Sub mnuhistograma_Click()
```

```
histograma
```

```
End Sub
```

```
Private Sub mnumedia_Click()
```

```
medias
```

```
End Sub
```

```
Private Sub mnuSairGardar_Click()
```

```
    GUARDA
```

```
    End
```

```
End Sub
```

```
'  
' menu de contexto  
'
```

```
Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
    If Button = 2 Then
```

```
        PopupMenu mnualunos
```

```
    End If
```

```
End Sub
```