



# **Manual de Instalação e Operação**

**LC 2000**

**Versão 7**

OUT/2003

## Sumário

|   |           |
|---|-----------|
| <b>Sobre a LC2000 e este manual</b>                     | <b>3</b>  |
| <b>Introdução</b>                                       | <b>4</b>  |
| <b>Instalação da Leitora</b>                            | <b>4</b>  |
| <b>Programa de Teste da Leitora LC2000</b>              | <b>5</b>  |
| <b>Configuração do LCTest</b>                           | <b>5</b>  |
| <b>Executando a Leitura de Cartões</b>                  | <b>6</b>  |
| <b>Configurando os Cartões para Leitura</b>             | <b>8</b>  |
| <b>Arquivo de Configuração</b>                          | <b>8</b>  |
| <b>Configuração da Leitora e do Cartão</b>              | <b>8</b>  |
| <b>Tipos de Grupos</b>                                  | <b>9</b>  |
| Sem divisão com 12 quadrículas (Binário com 12 dígitos) | 9         |
| Uma divisão de 10 quadrículas (Inscrição de 0 a 9)      | 10        |
| Dígito Verificador                                      | 11        |
| Uma divisão de 10 quadrículas (Numérico de 0 a 9)       | 12        |
| Duas divisões de 5 quadrículas (A B C D E)              | 13        |
| Três divisões de 3 quadrículas (A B C)                  | 13        |
| Quatro divisões de 2 quadrículas (A B)                  | 14        |
| Seis divisões de 1 quadrícula (Binário com 1 dígito)    | 14        |
| <b>Máscara do Cartão</b>                                | <b>15</b> |
| Mascaramento Parcial                                    | 15        |
| Mascaramento Total                                      | 15        |
| <b>Especificações do Cartão</b>                         | <b>16</b> |
| <b>Exemplos de Cartões</b>                              | <b>17</b> |
| <b>Leitura dos Cartões</b>                              | <b>19</b> |
| <b>Arquivo de Configuração</b>                          | <b>19</b> |
| <b>Arquivo de Dados</b>                                 | <b>19</b> |
| <b>Leitura dos Cartões</b>                              | <b>20</b> |
| <b>Exemplo de Arquivo de Dados</b>                      | <b>20</b> |
| <b>Erros de Leitura</b>                                 | <b>22</b> |
| <b>Uso da DLL LC2000.DLL</b>                            | <b>23</b> |
| Função Inicializa a Leitora                             | 23        |
| Função Testa Comunicação                                | 23        |
| Função Recebe Status                                    | 23        |
| Função Envia Configuração                               | 24        |
| Função Lê Cartão  | 24        |
| Função Recebe Cartão                                    | 25        |
| Função Reinicializa Leitora                             | 27        |
| Função Cancela Leitura                                  | 27        |
| Função Envia Máscara                                    | 27        |
| Função Finaliza a Leitora                               | 28        |
| <b>Convenções</b>                                       | <b>29</b> |
| <b>Exemplos para Delphi</b>                             | <b>29</b> |
| Exemplo de Inicialização e Configuração da Leitora      | 30        |
| Exemplo de uso das funções da Leitora                   | 32        |

|  |    |
|--|----|
| Observação sobre interferência eletromagnética | 36 |
| Especificações Técnicas                        | 36 |
| Garantia                                       | 36 |

## Sobre a LC2000 e este manual

Este manual refere-se ao equipamento LC2000 – leitora ótica de marcas – e ao pacote de softwares versão 7. A versão 7 traz algumas melhorias em relação às versões anteriores, das quais destacam-se:

- **Maior velocidade de comunicação com o computador.** Versões anteriores comunicavam-se a uma taxa de 9,6kbps. A versão 7 comunica-se a uma taxa de **56kbps**;
- Capacidade de leitura de cartões maiores. Versões anteriores eram capazes de ler cartões de até 60 linhas. **A versão 7 é capaz de ler cartões de até 90 linhas.**

Por isso, a versão 7 do equipamento não é compatível com as versões anteriores de software. E os softwares do pacote de programas de versão 7 são incompatíveis com leitoras de versões anteriores.

Outras características da LC2000 versão 7 são:

- Compatibilidade com **Windows 95, 98, Me, 2000 e XP**;
- Acompanha programas de teste, de leitura de cartões e de configuração de cartões, com interface gráfica bastante amigável;
- Gravação dos dados lidos é realizada em arquivo texto para posterior processamento. Opcionalmente, o cliente pode desenvolver um software proprietário que poderá se comunicar com a LC2000 com o uso da DLL que acompanha o equipamento;
- Conexão com computador através de porta **serial comum (RS232C) ou USB<sup>1</sup>**;
- Possibilidade de leitura unidirecional ou bidirecional (lê o mesmo cartão 2 vezes – quando puxa e quando devolve – para aumentar a segurança);
- Velocidade de tracionamento do cartão: aprox. 55cm/s;
- Velocidade de leitura: **até 2500 cartões de 38 linhas por hora em modo bidirecional**;
- Tensão de alimentação: 127 ou 220V, 60Hz, selecionado por chave seletora;
- Peso: 3kg;
- Dimensões externas: (229 x 135 x 112) mm;

---

<sup>1</sup> Para que a LC2000 seja conectada a computadores através de porta USB é necessário o uso de um conversor Serial/USB (não incluso). Este conversor pode ser encontrado em lojas de produtos de informática.

## Introdução

A Leitora Ótica de Cartões - LC2000 é um equipamento que permite a leitura de marcas feitas com caneta azul ou preta sobre um cartão de papel, pelo princípio da reflexão luminosa.

Este dispositivo pode ser ligado a qualquer computador do tipo IBM-PC que utilize Windows e disponha de uma interface serial padrão RS-232C ou porta USB disponível. A maioria dos computadores atende a esses requisitos.

Sua aplicação é bastante ampla, podendo ser usada como leitora de cartões de concursos, de provas de escolas, vestibulares, pesquisa de opinião pública, fichas de controle de estoque, cartões de frequência de alunos, comandas para lanchonetes e restaurantes, cartões de ponto, etc.

A leitora pode ainda ser utilizada em qualquer tipo de aplicação que requeira uma coleta de dados marcados pelo usuário. Com o pacote deste dispositivo segue uma biblioteca dinâmica de funções (DLL) que permite que programas específicos feitos pelos clientes possam acessar a leitora.

A figura 1 mostra o painel frontal da LC2000 com os 4 leds de sinalização:

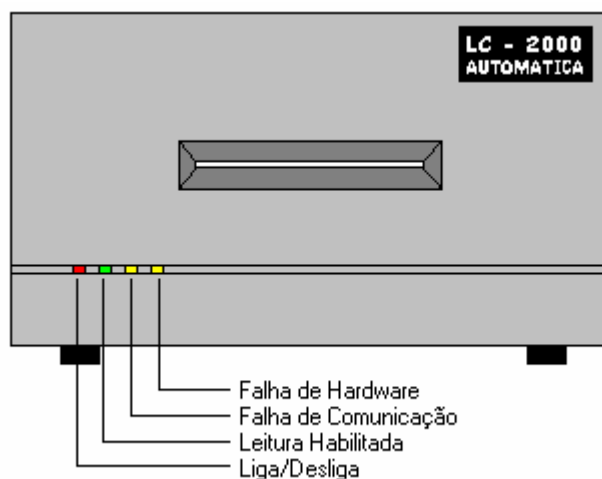


Figura 1

## Instalação da Leitora

- Desligue o computador e conecte o cabo serial da leitora numa das portas seriais RS-232C (porta serial comum) ou USB (através de um adaptador);
- **Verifique se a chave de seleção de voltagem da leitora está coerente com a voltagem local.** Caso contrário, selecione a voltagem correta. Veja a figura 2.
- Conecte o cabo de alimentação atrás da leitora e o outro lado à rede elétrica. Tomada com o fio terra é requerida.
- Ligue o computador e, em seguida, a chave situada atrás da leitora.
- O led vermelho da leitora acende e esta se encontra pronta para o funcionamento.

O pacote de softwares que acompanha o equipamento possui programas para configuração dos cartões, leitura dos cartões e teste da leitora, além das DLLs (de uso opcional para programação). Para instalar todos os programas execute o SETUP existente no disquete número 1 ou no CD de instalação da LC2000 e siga as instruções na tela.

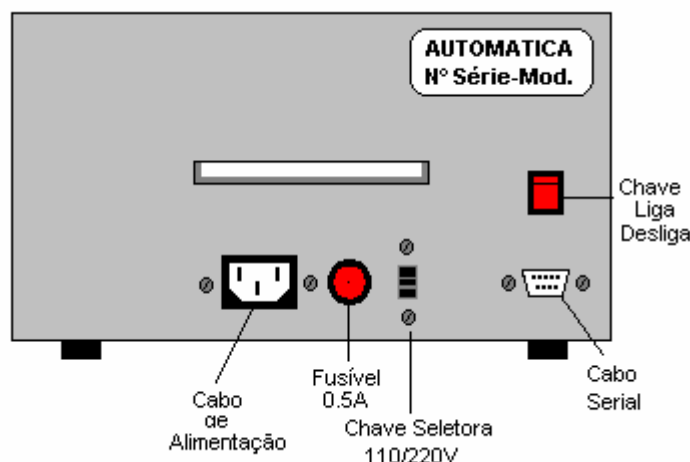


Figura 2

## Programa de Teste da Leitora LC2000

Este programa foi desenvolvido para que o usuário possa testar a Leitora Ótica de Cartões LC2000 e ficar familiarizado com o seu funcionamento.

### Configuração do LCTeste

A figura 3 mostra a tela de configuração do programa **LCTeste**. Cada função é descrita a seguir:

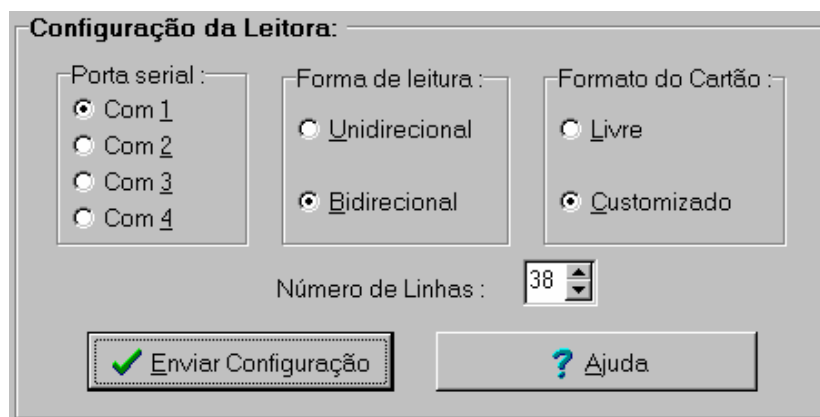


Figura 3

**Porta Serial:** Selecione a porta serial em que leitora foi conectada.

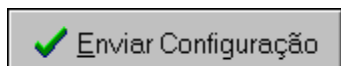
#### Forma de Leitura:

- Selecione o modo **Unidirecional** para que a leitora faça a leitura de cada cartão apenas uma vez. Nesse modo o cartão é colocado pela parte frontal e retirado pela parte traseira.
- Selecione o modo **Bidirecional** para que a leitora execute 2 leituras de cada cartão. Nesse modo o cartão é colocado e retirado pela parte frontal. A leitora executa uma leitura quando puxa o cartão e outra quando o devolve. Se houver alguma diferença entre essas leituras, automaticamente o cartão é tracionado para que mais duas leituras sejam executadas. Se, ainda assim, continuar existindo alguma diferença, o cartão é devolvido após três tentativas e a(s) quadrícula(s) com problema são mostradas em vermelho no monitor.

### Formato do Cartão:

- **Formato Livre:** Neste formato de leitura, a leitora permite que cartões com quantidades diferentes de linhas possam ser lidos. Esta opção é recomendada quando o usuário quer entrar com cartões de tamanhos diferentes e de forma misturada.
- **Formato Customizado:** Neste formato de leitura o número de linhas definido pelo usuário (38 por exemplo) é sempre comparado com o número de linhas detectado durante a leitura. Se forem diferentes, haverá indicação de erro de leitura e a leitora tentará ler o cartão por mais 2 vezes consecutivas.

**Número de Linhas:** Este campo define o número de linhas dos cartões a serem lidos pela leitora. O número máximo de linhas, conseqüentemente o valor máximo que esta variável pode assumir é 60. Este item aparece somente quando a opção pelo formato customizado é habilitada.



Clique no botão **Enviar Configuração** para que a leitora seja configurada de acordo com as informações especificadas.

## Executando a Leitura de Cartões

Após o envio da configuração, o botão **Iniciar Leitura** fica habilitado.



Clique no botão **Iniciar Leitura**. O LED verde (leitura habilitada) se acenderá indicando que o usuário pode colocar o cartão na leitora. Observe na figura 4 a posição correta de inserção do cartão.

**Atenção :** O cartão deve estar com as marcas pintadas voltadas para cima e com a marca de sincronismo (*Strobe*) à direita.

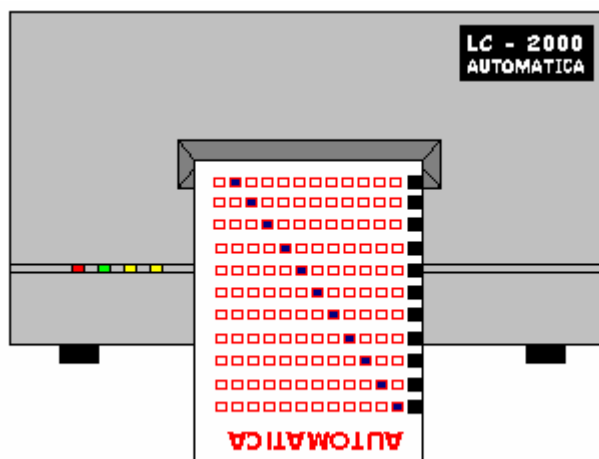


Figura 4

À medida que os cartões são lidos, um contador de cartões é incrementado. Se algum cartão apresentar erro de leitura ou número de linhas incompatível com o programado, o contador de cartões lidos com erro é incrementado.

A figura 5 mostra a imagem gerada pelo programa após a leitura de um cartão com sucesso. Se a leitora estiver em modo bidirecional, caso uma quadrícula esteja mal pintada, havendo divergência da leitura da quadrícula nos dois sentidos, a quadrícula será mostrada em **vermelho** na tela, indicando que é uma marca indefinida.

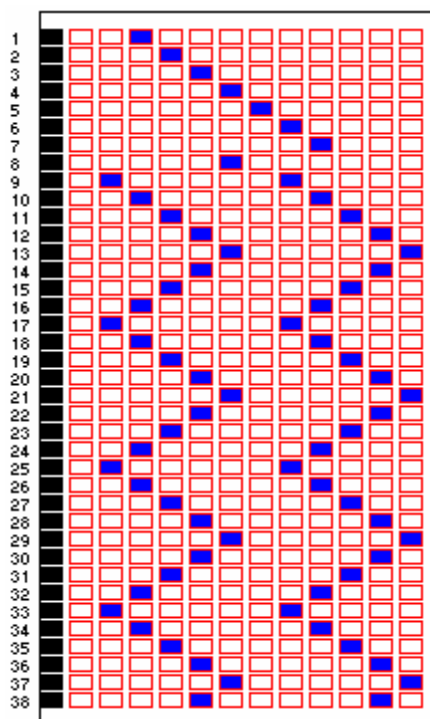


Figura 5



Clique no botão **Parar Leitura** quando desejar finalizar a leitura dos cartões.

**É necessário clicar neste botão antes de sair do programa.**



Para encerrar o programa, fechando a comunicação da porta serial, clique no botão **Sair**. É necessário aguardar alguns segundos para que a porta serial seja corretamente fechada.

## Configurando os Cartões para Leitura

Um dos programas do pacote de softwares que acompanha a leitora foi desenvolvido para que o usuário possa configurar o cartão a ser lido pela Leitora Ótica de Cartões LC2000.

O programa de configuração dos cartões **LCConfig** permite que o usuário configure os mais diversos tipos de cartão para que posteriormente sejam lidos pela leitora, através do programa **LC2000**.

O programa também permite configurar a porta serial onde a leitora está conectada e o modo de leitura (Unidirecional ou Bidirecional) em que a leitora irá ler os cartões.

As informações da configuração do cartão são salvas em arquivos cuja extensão é **.CFG**.

O cartão a ser configurado é formado por grupos de linhas. Cada grupo possui o mesmo conjunto de caracteres. Cada cartão pode ter até 10 tipos de grupos e os grupos podem ser repetidos. Apenas o grupo **Inscrição** deve ser único, por se tratar do grupo identificador do cartão.

Existem 7 grupos disponíveis para o usuário, desde os mais comuns como os grupos **Inscrição** e **Questões ABCDE** até grupos sem divisões, considerados grupos livres, em que o usuário pode associar as quadrículas a qualquer tipo de informação. Nos grupos livres, uma quadrícula não pintada é lida como 0, e uma quadrícula pintada é lida como 1.

## Arquivo de Configuração

A configuração do cartão é guardada em arquivos de configuração, cuja extensão é **.CFG**.

Os arquivos de configuração ficam localizados no sub-diretório **[Diretório de Instalação]\Config**. Por exemplo: **C:\Arquivos de Programas\LC2000\Config**. Estes arquivos são posteriormente usados pelo aplicativo **LC2000**.

A seguir são descritos os botões que formam este campo:



Clique neste botão se desejar iniciar a configuração de um novo cartão. Inicialmente é criado um cartão com uma única linha.



Clique neste botão se desejar abrir um arquivo de configuração já existente para realizar modificações.



Clique neste botão para salvar as modificações feitas no arquivo aberto ou criado. Um nome de arquivo é requerido. Se este arquivo já existir, uma mensagem de arquivo já existente será mostrada, dando a possibilidade de substituí-lo.




Clique neste botão se desejar fechar o programa.

## Configuração da Leitora e do Cartão

Configure os campos conforme a seguir:

- **Porta Serial:** Selecione a porta serial em que a leitora está conectada.



- **Forma de Leitura:** Clique com o *mouse* sobre um dos dois possíveis modos de leitura da leitora: Unidirecional ou Bidirecional. O modo de leitura escolhido será utilizado durante a leitura dos cartões aqui configurados.
- **Formato do Cartão:** Neste campo é programado o **Número de grupos do cartão**. Use  para aumentar ou diminuir o número de grupos do cartão.
- **Número de linhas do cartão:** é o totalizador de linhas de todos os grupos. O número máximo de linhas de cada cartão é 90.

#### Modo Unidirecional:



Neste modo de operação, o cartão entra pela parte frontal da leitora e é expelido por trás, sendo feita uma única leitura das quadrículas. Se houver erro do número de linhas do cartão, o cartão é tracionado para mais duas tentativas de leitura.

#### Modo Bidirecional:

Neste modo de operação, o cartão entra e retorna pela parte frontal da leitora, de modo que há leitura das quadrículas nos dois sentidos de movimento. Este modo de leitura lê pelo menos duas vezes as quadrículas antes de validar os dados. Portanto, este é o modo de operação mais recomendado se for exigida grande segurança na leitura dos dados. Se houver erros de leitura, o cartão é tracionado para mais duas tentativas de leitura.

## Tipos de Grupos

Existem 7 tipos de grupos que podem ser usados para configurar os mais diversos cartões. Cada grupo é representado por uma quantidade de linhas e tem um conjunto de caracteres que o representa.

- **Número do grupo:** identifica a sua posição no cartão. Use  para navegar através dos grupos.
- **Nome do grupo:** identifica a posição do grupo no cartão.
- **Número da linha inicial do grupo:** indica onde o grupo se inicia.
- **Número da linha final do grupo:** indica onde o grupo se encerra. Use  para aumentar ou diminuir o número de linhas do grupo.
- **Habilitado:** Em certas circunstâncias, se o usuário tem um cartão genérico para várias finalidades, ele pode querer que determinados grupos não sejam processados pela leitora. Nesse caso, basta desabilitar os grupos desejados.

**Atenção: Se um grupo é desabilitado, as informações deste grupo não são lidas pela leitora e estes dados não são salvos no arquivo de dados.**

- **Divisão do grupo:** escolha os grupos que atendam às suas necessidades:
  1. Sem divisão com 12 quadrículas (Binário com 12 dígitos)
  2. Uma divisão de 10 quadrículas (Inscrição de 0 a 9)
  3. Uma divisão de 10 quadrículas (Numérico de 0 a 9)
  4. Duas divisões de 5 quadrículas (A B C D E)
  5. Três divisões de 3 quadrículas (A B C)
  6. Quatro divisões de 2 quadrículas (A B)
  7. Seis divisões de 1 quadrícula (Binário com 1 dígito)

#### Sem divisão com 12 quadrículas (Binário com 12 dígitos)

Este grupo é recomendado quando o usuário deseja associar diversas opções em uma ou mais linhas de forma não-ordenada. O arquivo de dados irá receber um caractere para cada quadrícula deste grupo, sendo o caractere 0 inserido quando a quadrícula não está pintada e o caractere 1 inserido

quando a quadrícula está pintada. Portanto, a cada linha deste grupo, 12 caracteres serão gerados no arquivo de dados.

O **LCConfig** permite que este grupo possa ser configurado mais de uma vez no mesmo cartão.

Devido a ser um grupo de formato livre não há possibilidade de criticá-lo quanto ao seu preenchimento pelo programa **LC2000**.

A figura 6 mostra um exemplo de 5 linhas programadas com este grupo.

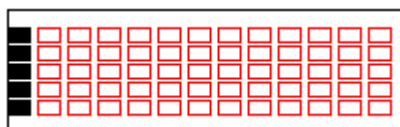


Figura 6

A figura 7 mostra um exemplo de opções de dados que o usuário pode colocar neste grupo. Neste caso, foram usadas 6 linhas para representar dígitos de 0 a 60 e uma linha para anular o campo. Observe que, sendo este formato do tipo livre, nenhuma, várias ou todas as quadrículas de uma linha podem ser pintadas. A primeira e a última quadrículas de cada linha deste grupo podem ser opcionalmente mascaradas.



Figura 7

A figura 8 mostra outro exemplo de aplicação deste grupo, tipicamente usado em comandas de lanchonetes e restaurantes. As quadrículas quarta a nona podem ser opcionalmente mascaradas.



Figura 8

### Uma divisão de 10 quadrículas (Inscrição de 0 a 9)

Este grupo é usado quando o usuário deseja criar um campo de inscrição ou matrícula usando caracteres numéricos de 0 a 9 para cada linha.

Recomenda-se que este grupo seja o primeiro do cartão para facilidade de identificação pelo usuário. Contudo, ele pode ser configurado para ser alocado em qualquer parte do cartão.

**Este grupo deve ser único, por tratar-se do grupo de identificação do cartão.**

**Atenção! Cartões com inscrição preenchida incorretamente, com linha(s) em branco ou com múltipla marcação em uma das linhas geram uma mensagem de erro na tela do programa durante a leitura do cartão. Tais cartões não são inseridos no arquivo de dados.**

A figura 9 mostra um exemplo de 10 linhas programadas com este grupo. **Observe que as duas primeiras quadrículas de cada linha não são utilizadas no grupo de inscrição.**

|    |   |   |   |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|---|---|---|
| 1  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 3  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 4  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 5  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 6  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 7  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 8  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 9  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Figura 9

## Dígito Verificador

O grupo inscrição permite ao usuário o uso de dígito verificador. Este recurso usa o último dígito da inscrição como resultado de um cálculo específico dos dígitos anteriores.

O dígito verificador é muito útil para verificar a consistência da marcação do campo de inscrição, principalmente quando a inscrição é preenchida pelo usuário e não pré-impressa no cartão.

O usuário pode optar por usar ou não o dígito verificador. Existem 4 módulos de cálculo pré-programados. Se optar pelo uso do dígito verificador, o programa de leitura de dados irá efetuar o cálculo durante a leitura do cartão. **Se detectar erro na marcação do dígito verificador, o programa exibirá uma mensagem de erro e os dados do cartão lido não serão gravados no arquivo de dados.**

## Tipos de Dígito Verificador

São fornecidas quatro possibilidades de cálculo de dígito verificador, conforme mostrado a seguir:

### Módulo 10 A - Simples (10-Simp):

Somam-se os dígitos do número;

O resultado da soma é dividido por 10;

A diferença entre 10 e o resto da divisão constitui o dígito verificador.

Exemplo:

Número: 24579  
 $2+4+5+7+9 = 27$   
 $27/10 = 2$  com resto 7

Dígito:  $10 - 7 = 3$

Número de inscrição a ser preenchido no cartão: 245793

**Observação:** Para números cujo dígito verificador não tem representação segundo esse procedimento, a solução prática adotada foi considerar o dígito verificador sendo 0 (zero). Isso ocorre sempre que o resto da divisão é 0 (zero) e a diferença  $10 - 0$  fica sendo representada por dois dígitos, que é impossível.

### Módulo 11 Com Pesos Em Progressão Aritmética (11-A):

Multiplica-se cada dígito do número pelos pesos da seqüência: n, n-1, n-2, n-3, ..., 4, 3, 2; onde **n** é o número total de dígitos, inclusive o dígito verificador.

Somam-se os produtos e divide-se a soma por 11;

A diferença entre 11 e o resto da divisão constitui o dígito verificador.

Exemplo:

Número: 24579  
 $2*6+4*5+5*4+7*3+9*2 = 91$   
 $91/11 = 8$  com resto 3

Dígito:  $11 - 3 = 8$

Novo número: 245798

**Observação:** Para cujo dígito verificador não tem representação segundo esse procedimento, a solução prática adotada foi considerar o dígito verificador sendo 0 (zero). Isso ocorre sempre que o resto da divisão é 0 (zero) e a diferença  $11 - 0$  fica sendo representada por dois dígitos, que é impossível.

#### Módulo 11 Com Pesos Em Progressão Geométrica - Base 2 (11-B):

Multiplica-se cada dígito do número pelos da seqüência:  $2^{\exp(n-1)}$ ,  $2^{\exp(n-2)}$ ,  $2^{\exp(n-3)}$ , ...,  $2^{\exp(3)}$ ,  $2^{\exp(2)}$ ,  $2^{\exp(1)}$ ; onde **n** é o número de dígitos inclusive o dígito verificador.

Somam-se os produtos e divide-se a soma por 11;

A diferença entre 11 e o resto da divisão constitui o dígito verificador.

Exemplo:

Número: 24579  
 $2 \cdot 32 + 4 \cdot 16 + 5 \cdot 8 + 7 \cdot 4 + 9 \cdot 2 = 214$   
 $214 / 11 = 19$  com resto 5

Dígito:  $11 - 5 = 6$

Novo número: 245796

**Observação:** Para aqueles números cujo dígito verificador não tem representação segundo esse procedimento, a solução prática adotada foi considerar o dígito verificador sendo 0 (zero). Isso ocorre sempre que o resto da divisão é 0 (zero) e a diferença  $11 - 0$  fica sendo representada por dois dígitos, que é impossível.

#### Módulo 11 Com Pesos Em Progressão Geométrica - Base 3 (11-C):

Multiplica-se cada dígito do número pelos da seqüência:  $3^{\exp(1)}$ ,  $3^{\exp(2)}$ ,  $3^{\exp(3)}$ , ...,  $3^{\exp(n-3)}$ ,  $3^{\exp(n-2)}$ ,  $3^{\exp(n-1)}$ ; onde **n** é o número de dígitos inclusive o dígito verificador.

Somam-se os produtos e divide-se a soma por 11;

A diferença entre 11 e o resto da divisão constitui o dígito verificador.

Exemplo:

Número: 24579  
 $2 \cdot 3 + 4 \cdot 9 + 5 \cdot 27 + 7 \cdot 81 + 9 \cdot 243 = 2931$   
 $2931 / 11 = 266$  com resto 5

Dígito:  $11 - 5 = 6$

Novo número: 245796

**Observação:** Para aqueles números cujo dígito verificador não tem representação segundo esse procedimento, a solução prática adotada foi considerar o dígito verificador sendo 0 (zero). Isso ocorre sempre que o resto da divisão é 0 (zero) e a diferença  $11 - 0$  fica sendo representada por dois dígitos, que é impossível.

#### Uma divisão de 10 quadrículas (Numérico de 0 a 9)

Este grupo é recomendado quando o usuário deseja associar caracteres numéricos de 0 a 9 a cada linha do grupo. É usado para campos numéricos em geral, como datas, identificação de provas, questões com resposta numérica, etc.

Este grupo pode ser criticado no programa **LC2000**. Isso significa que o programa de leitura avisa ao usuário se alguma linha for lida em branco ou com múltiplas marcações.

O **LCConfig** permite que este grupo possa ser configurado mais de uma vez no mesmo cartão.

No arquivo de dados será escrito o caractere numérico equivalente à marcação efetuada. Para cada linha em branco, o caractere **Z** é adicionado ao arquivo. Para cada linha com múltipla marcação, o caractere **W** é adicionado ao arquivo.

A figura 10 mostra um exemplo de 10 linhas programadas com este grupo.

|    |   |   |   |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|---|---|---|
| 1  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 3  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 4  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 5  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 6  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 7  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 8  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 9  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

Figura 10

### Duas divisões de 5 quadrículas (A B C D E)

Este tipo de grupo é o mais tradicional grupo de questões. Permite 5 opções de escolha, sendo representado por caracteres de **A**, **B**, **C**, **D** e **E**, em duas colunas.

Este grupo pode ser criticado no programa **LC2000**, de modo a avisar ao usuário se alguma questão estiver em branco ou com múltiplas marcações.

O **LCConfig** permite que este grupo possa ser configurado mais de uma vez no mesmo cartão.

No arquivo de dados será escrito o caractere **A**, **B**, **C**, **D** ou **E**, equivalente à marcação efetuada. Para cada questão em branco, o caractere **Z** é adicionado ao arquivo. Para cada questão com múltipla marcação, o caractere **W** é adicionado ao arquivo.

A figura 11 mostra um exemplo de 20 questões programadas em 10 linhas com este grupo.

|    |   |   |   |   |   |    |   |   |   |   |   |
|----|---|---|---|---|---|----|---|---|---|---|---|
| 1  | A | B | C | D | E | 11 | A | B | C | D | E |
| 2  | A | B | C | D | E | 12 | A | B | C | D | E |
| 3  | A | B | C | D | E | 13 | A | B | C | D | E |
| 4  | A | B | C | D | E | 14 | A | B | C | D | E |
| 5  | A | B | C | D | E | 15 | A | B | C | D | E |
| 6  | A | B | C | D | E | 16 | A | B | C | D | E |
| 7  | A | B | C | D | E | 17 | A | B | C | D | E |
| 8  | A | B | C | D | E | 18 | A | B | C | D | E |
| 9  | A | B | C | D | E | 19 | A | B | C | D | E |
| 10 | A | B | C | D | E | 20 | A | B | C | D | E |

Figura 11

### Três divisões de 3 quadrículas (A B C)

Este tipo permite 3 opções de escolha, sendo representado por caracteres de **A**, **B** e **C** em três colunas.

Este grupo pode ser criticado no programa **LC2000**, de modo a avisar ao usuário se alguma questão estiver em branco ou com múltiplas marcações.

O **LCConfig** permite que este grupo possa ser configurado mais de uma vez no mesmo cartão.

No arquivo de dados será escrito o caractere **A**, **B**, ou **C**, equivalente à marcação efetuada. Para cada questão em branco, o caractere **Z** é adicionado ao arquivo. Para cada questão com múltipla marcação, o caractere **W** é adicionado ao arquivo.

A figura 12 mostra um exemplo de 30 questões programadas em 10 linhas com este grupo.

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

Figura 12

#### Quatro divisões de 2 quadrículas (A B)

Este tipo permite 2 opções de escolha, sendo representado por caracteres de **A** e **B** em quatro colunas. Pode ser usado como questão do tipo **Falso** e **Verdadeiro**, bastando apenas que sejam impressos os caracteres **F** e **V** como opção para **A** e **B**. Outra opção seria o uso para escolha do Sexo (**M** e **F**). O arquivo de dados, independente dos rótulos impressos no cartão, sempre armazenará os caracteres **A** ou **B**. Para cada questão em branco, o caractere **Z** é adicionado ao arquivo. Para cada questão com múltipla marcação, o caractere **W** é adicionado ao arquivo.

Este grupo pode ser criticado no programa **LC2000**, de modo a avisar ao usuário se alguma questão estiver em branco ou com múltiplas marcações.

O **LCConfig** permite que este grupo possa ser configurado mais de uma vez no mesmo cartão.

A figura 13 mostra um exemplo de 40 questões programadas em 10 linhas com este grupo.

|    |   |   |   |    |   |   |   |    |   |   |   |    |   |   |   |
|----|---|---|---|----|---|---|---|----|---|---|---|----|---|---|---|
| 1  | A | B | C | 11 | A | B | C | 21 | A | B | C | 31 | A | B | C |
| 2  | A | B | C | 12 | A | B | C | 22 | A | B | C | 32 | A | B | C |
| 3  | A | B | C | 13 | A | B | C | 23 | A | B | C | 33 | A | B | C |
| 4  | A | B | C | 14 | A | B | C | 24 | A | B | C | 34 | A | B | C |
| 5  | A | B | C | 15 | A | B | C | 25 | A | B | C | 35 | A | B | C |
| 6  | A | B | C | 16 | A | B | C | 26 | A | B | C | 36 | A | B | C |
| 7  | A | B | C | 17 | A | B | C | 27 | A | B | C | 37 | A | B | C |
| 8  | A | B | C | 18 | A | B | C | 28 | A | B | C | 38 | A | B | C |
| 9  | A | B | C | 19 | A | B | C | 29 | A | B | C | 39 | A | B | C |
| 10 | A | B | C | 20 | A | B | C | 30 | A | B | C | 40 | A | B | C |

Figura 13

#### Seis divisões de 1 quadrícula (Binário com 1 dígito)

Este grupo é recomendado quando o usuário deseja associar dois estados a uma determinada questão. Para cada questão deste grupo, o caractere **0** é gravado no arquivo de dados quando a quadrícula não está marcada, e o caractere **1** é gravado quando a quadrícula está marcada.

Este grupo pode ser usado para controle de frequência de alunos, por exemplo.

O **LCConfig** permite que este grupo possa ser configurado mais de uma vez no mesmo cartão.

Não há possibilidade de criticar este campo quanto ao seu preenchimento.

A figura 14 mostra um exemplo de 10 linhas programadas com este grupo.

|    |  |  |  |    |  |  |  |    |  |  |  |    |  |  |  |    |  |  |  |    |  |  |  |
|----|--|--|--|----|--|--|--|----|--|--|--|----|--|--|--|----|--|--|--|----|--|--|--|
| 1  |  |  |  | 11 |  |  |  | 21 |  |  |  | 31 |  |  |  | 41 |  |  |  | 51 |  |  |  |
| 2  |  |  |  | 12 |  |  |  | 22 |  |  |  | 32 |  |  |  | 42 |  |  |  | 52 |  |  |  |
| 3  |  |  |  | 13 |  |  |  | 23 |  |  |  | 33 |  |  |  | 43 |  |  |  | 53 |  |  |  |
| 4  |  |  |  | 14 |  |  |  | 24 |  |  |  | 34 |  |  |  | 44 |  |  |  | 54 |  |  |  |
| 5  |  |  |  | 15 |  |  |  | 25 |  |  |  | 35 |  |  |  | 45 |  |  |  | 55 |  |  |  |
| 6  |  |  |  | 16 |  |  |  | 26 |  |  |  | 36 |  |  |  | 46 |  |  |  | 56 |  |  |  |
| 7  |  |  |  | 17 |  |  |  | 27 |  |  |  | 37 |  |  |  | 47 |  |  |  | 57 |  |  |  |
| 8  |  |  |  | 18 |  |  |  | 28 |  |  |  | 38 |  |  |  | 48 |  |  |  | 58 |  |  |  |
| 9  |  |  |  | 19 |  |  |  | 29 |  |  |  | 39 |  |  |  | 49 |  |  |  | 59 |  |  |  |
| 10 |  |  |  | 20 |  |  |  | 30 |  |  |  | 40 |  |  |  | 50 |  |  |  | 60 |  |  |  |

Figura 14

## Máscara do Cartão

O objetivo de se mascarar uma determinada quadrícula ou questão é desabilitá-la para que o seu conteúdo seja ignorado pela leitora. Quando uma quadrícula é mascarada no programa, independente de estar ou não marcada, ela não será lida pelo programa.

Pode-se mascarar alguns campos quando o usuário não quer se sejam lidas determinadas quadrículas. Marcações indefinidas podem gerar erro de leitura. Por isso é interessante mascarar campos que não serão utilizados.



**Definir Manualmente os Campos Mascarados**

Clique no botão para habilitar o mascaramento e selecione (com o clique do mouse) as quadrículas a serem mascaradas. A cor da quadrícula selecionada muda para verde. Para desmarcar, clique novamente sobre a quadrícula. Sua cor volta a ser branca.

Os grupos têm mascaramento automático das quadrículas que não são usadas. Mascaramento adicional deve ser feito de forma manual. Existem dois tipos de mascaramento, conforme descrito a seguir:

### Mascaramento Parcial

Quando apenas algumas quadrículas de algum campo (divisão de um grupo) são mascaradas diz-se que o mascaramento é parcial. Neste caso, as quadrículas restantes são válidas e seu conteúdo marcado é acrescido ao arquivo de dados, desde que o grupo esteja habilitado.

A figura 15 mostra um grupo numérico com uma linha em que o usuário optou por não haver opção 0 e de 5 a 9.



Figura 15

### Mascaramento Total

Quando todas as quadrículas de um campo são mascaradas diz-se que o mascaramento é total. Esta opção é muito útil quando existe um campo em que muitas questões não serão lidas. **Neste caso, o conteúdo marcado nestes campos não é acrescido ao arquivo de dados.**

A figura 16 mostra um grupo ABCDE com 20 questões em que o usuário mascarou as questões de 15 a 20. Portanto no arquivo de dados somente constarão as respostas relativas às questões 1 a 14.

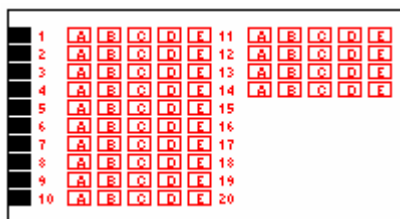


Figura 16

Um outro exemplo de mascaramento total acontece na figura 17. O usuário decidiu mascarar as duas primeiras linhas um grupo numérico de 7 linhas por considerar desnecessário o preenchimento total do grupo nesta situação. A primeira e a segunda linhas não são acrescidas ao arquivo de dados.



Figura 17

Também pode-se mascarar questões no meio de um determinado grupo. Isso é útil, por exemplo, para invalidar questões anuladas. Tenha sempre em mente que estas questões mascaradas totalmente **não** são acrescidas ao arquivo de dados. A figura 18 mostra este caso em um grupo do tipo ABC com 30 questões. A questão 20 foi anulada. No arquivo constarão as questões 1 a 19 e 21 a 30, em sequência e nesta ordem.

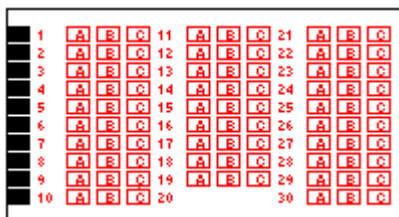
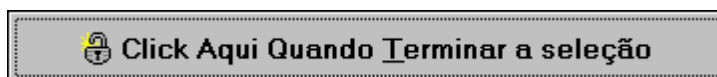


Figura 18



Clique no botão para que as modificações feitas na máscara do cartão sejam efetivadas.

## Especificações do Cartão

O cartão deve ser confeccionado com os campos pré-impressos, desenhados de acordo com a aplicação. Veja um exemplo na figura 19. As seguintes especificações devem ser seguidas ao se confeccionar os cartões.

- Gramatura do papel = 90 g/m<sup>2</sup> ou superior
- Número de marcas por linha = 12
- Largura do *Strobe (Clock)* **d<sub>1</sub>** = 3,17mm (1/8")
- Distância entre a primeira marca e a margem esquerda do cartão **d<sub>2</sub>** = 4,50mm
- Distância entre as marcas **d<sub>3</sub>** = 6,35mm (1/4")
- Distância mínima entre as linhas **d<sub>4</sub>** = 2,54mm (1/10")
- Largura da marca **d<sub>5</sub>** = 4,76mm (3/16")
- Altura da marca **d<sub>6</sub>** = 2,54mm (1/10")
- Distância mínima entre o topo do cartão e a primeira linha **d<sub>7</sub>** = 45 mm
- Distância mínima entre o fundo do cartão e a última linha **d<sub>8</sub>** = 10 mm
- Largura do cartão **d<sub>9</sub>** = 82,55 mm (3 1/4")
- Número máximo de linhas = 90
- Número máximo de grupos = 10
- Comprimento do cartão: variável, dependendo da aplicação
- Cor dos *Strobes (Clocks)* = Preta
- Toda impressão adicional deverá ser feita nas cores:



Laranja Pérsia Super Cor N- 060345 ou  
Laranja Clássico Super Cor N- 060344

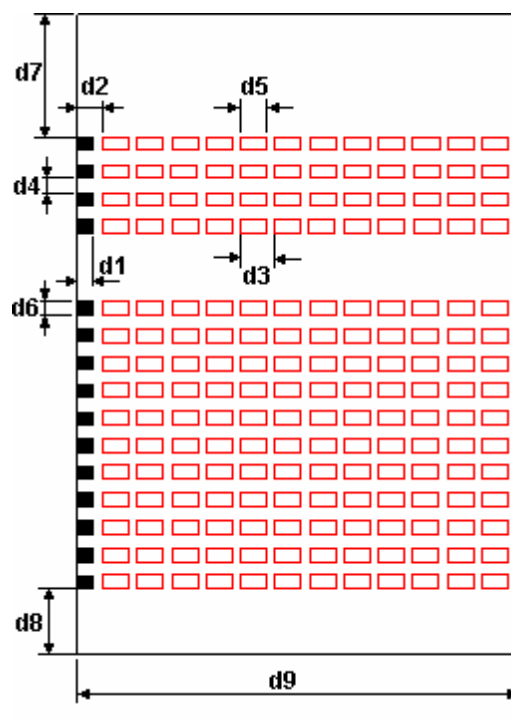


Figura 19

**Atenção:** A confecção do cartão requer uma gráfica de precisão. As medidas devem ser rigorosamente respeitadas para que não comprometam a leitura dos dados pela leitora.

As quadriculas também podem ser feitas com bordas arredondadas. Se preferir fazer quadriculas redondas, posicione cada círculo de modo que seus centros coincidam com os centros de cada quadricula. O diâmetro de cada círculo deve ser igual a **d6**.

Em qualquer caso, deve-se instruir as pessoas que forem marcar os cartões a seguirem as seguintes recomendações:

- Utilizar caneta esferográfica azul ou preta para efetuar as marcações;
- Não dobrar ou amassar o cartão;
- Marcar completamente a(s) quadricula(s) desejada(s), sem ultrapassar seus limites;
- Tomar muito cuidado para não riscar a coluna destinada aos strobes;
- Não utilizar corretivo tipo *Liquid Paper*.

## Exemplos de Cartões

A figura 20 mostra um exemplo de cartão configurado com 3 grupos: Inscrição (tipo uma divisão de 10 quadriculas - Inscrição de 0 a 9), Prova (tipo uma divisão de 10 quadriculas - Numérico de 0 a 9) e Questões (tipo duas divisões de 5 quadriculas ABCDE). Observe que os dígitos 0 e de 5 a 9 do grupo PROVA estão mascarados por não existirem tais provas. O grupo Questões apresenta 60 questões com as opções ABCDE.

A figura 21 apresenta um outro exemplo de cartão com características semelhantes às do primeiro, mas com o grupo de questões configurado de maneira um pouco diferente. Nesse caso, cada uma das 30 questões tem as opções ABCD e V ou F. Nesse caso, o grupo de Questões pode ser configurado como sendo do tipo sem divisão com 12 quadriculas (Binário com 12 dígitos).

**AUTOMATICA**  
TECNOLOGIA  
TEL. (27) 3325-6778

INSCRIÇÃO

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

PROVA 1 2 3 4

|    |   |   |   |   |   |    |   |   |   |   |   |
|----|---|---|---|---|---|----|---|---|---|---|---|
| 1  | A | B | C | D | E | 31 | A | B | C | D | E |
| 2  | A | B | C | D | E | 32 | A | B | C | D | E |
| 3  | A | B | C | D | E | 33 | A | B | C | D | E |
| 4  | A | B | C | D | E | 34 | A | B | C | D | E |
| 5  | A | B | C | D | E | 35 | A | B | C | D | E |
| 6  | A | B | C | D | E | 36 | A | B | C | D | E |
| 7  | A | B | C | D | E | 37 | A | B | C | D | E |
| 8  | A | B | C | D | E | 38 | A | B | C | D | E |
| 9  | A | B | C | D | E | 39 | A | B | C | D | E |
| 10 | A | B | C | D | E | 40 | A | B | C | D | E |
| 11 | A | B | C | D | E | 41 | A | B | C | D | E |
| 12 | A | B | C | D | E | 42 | A | B | C | D | E |
| 13 | A | B | C | D | E | 43 | A | B | C | D | E |
| 14 | A | B | C | D | E | 44 | A | B | C | D | E |
| 15 | A | B | C | D | E | 45 | A | B | C | D | E |
| 16 | A | B | C | D | E | 46 | A | B | C | D | E |
| 17 | A | B | C | D | E | 47 | A | B | C | D | E |
| 18 | A | B | C | D | E | 48 | A | B | C | D | E |
| 19 | A | B | C | D | E | 49 | A | B | C | D | E |
| 20 | A | B | C | D | E | 50 | A | B | C | D | E |
| 21 | A | B | C | D | E | 51 | A | B | C | D | E |
| 22 | A | B | C | D | E | 52 | A | B | C | D | E |
| 23 | A | B | C | D | E | 53 | A | B | C | D | E |
| 24 | A | B | C | D | E | 54 | A | B | C | D | E |
| 25 | A | B | C | D | E | 55 | A | B | C | D | E |
| 26 | A | B | C | D | E | 56 | A | B | C | D | E |
| 27 | A | B | C | D | E | 57 | A | B | C | D | E |
| 28 | A | B | C | D | E | 58 | A | B | C | D | E |
| 29 | A | B | C | D | E | 59 | A | B | C | D | E |
| 30 | A | B | C | D | E | 60 | A | B | C | D | E |

ASSINATURA

Figura 20

**AUTOMATICA**  
TECNOLOGIA  
TEL. (27) 3325-6778

INSCRIÇÃO

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

PROVA 1 2 3 4

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

ASSINATURA

Figura 21

## Leitura dos Cartões

O programa de Leitura dos Cartões foi desenvolvido para que os dados marcados nos cartões sejam lidos e gravados num arquivo texto. Os cartões a serem lidos devem ser configurados e marcados de acordo com instruções anteriormente descritas.

O usuário poderá, então, usar os dados gravados no arquivo da forma que melhor lhe convier. Pode, por exemplo, copiar e colar os dados numa planilha eletrônica para posterior processamento. Ou pode desenvolver um software que leia o arquivo de dados e transfira-os diretamente para seu banco de dados.

O programa de Leitura dos Cartões **LC2000** permite que o usuário leia os mais diversos tipos de cartão que previamente foram configurados no programa **LCConfig**.

O usuário deve carregar o arquivo de configuração compatível com os cartões a serem lidos. A extensão destes arquivos é **.CFG**.

Os dados da leitura dos cartões são salvos num arquivo texto cuja extensão é **.DAT**.

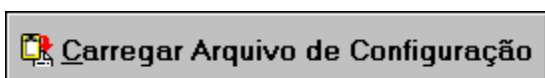
Se o usuário optar por crítica aos campos, toda vez que houver alternativas em branco ou com múltipla marcação, o programa irá parar e pedir confirmação do usuário para salvar ou não os dados do cartão no arquivo de dados.

Caso exista um campo do tipo inscrição com linha(s) em branco ou com múltipla marcação, os dados do cartão não são armazenados no arquivo de dados. Se o usuário optar por usar dígito verificador e o cálculo executado pelo programa de leitura para verificação deste dígito não conferir com o dígito lido, os dados do cartão não são armazenados no arquivo de dados.

Se o programa detectar um cartão com número de linhas diferente do configurado, seus dados não são transferidos para o arquivo de dados e uma mensagem é mostrada na tela.

## Arquivo de Configuração

A configuração do cartão é guardada em arquivos de configuração, cuja extensão é **.CFG**. Estes arquivos são previamente criados pelo aplicativo **LCConfig**.



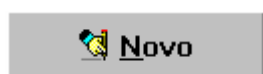
Clique neste botão se desejar carregar um arquivo de configuração de um determinado tipo cartão.

**Atenção: O arquivo de configuração deve ser compatível com o cartão a ser lido!**

## Arquivo de Dados

O arquivo de dados tem extensão **.DAT** e é onde são armazenados os dados dos cartões lidos.

Os arquivos de dados ficam alocados no diretório **[Diretório de Instalação]\Dados**, cujo *default* é **C:\Arquivos de Programas\LC2000\Dados**. Existem dois botões associados a este arquivo, conforme a seguir:



Clique neste botão se desejar iniciar um novo arquivo de dados para leitura de um novo grupo de cartões. Digite um nome para o novo arquivo.



Clique neste botão se desejar continuar a leitura dos cartões em um arquivo já existente. Se o cartão já contiver cartões lidos anteriormente, o número da inscrição do último cartão e o número de cartões lidos contidos neste arquivo são mostrados na tela.

**Atenção: Esteja certo de abrir um arquivo que contenha dados de cartões que tenham a mesma configuração carregada previamente. Se o arquivo de dados contiver dados de cartões incompatíveis com a configuração carregada, o número de inscrição mostrado na tela poderá estar incorreto.**

## Leitura dos Cartões



 **Iniciar Leitura** Clique neste botão para iniciar a leitura dos cartões. Quando este botão é clicado, inicia-se a comunicação entre a leitora e o computador.

Passe os cartões na Leitora, observando que as quadrículas ficam com a face para cima, e as marcas de sincronismo (*Strobes*) ficam do lado direito, de maneira que o cartão entra de ponta a cabeça. A figura 21 mostra a posição de inserção do cartão na leitora.

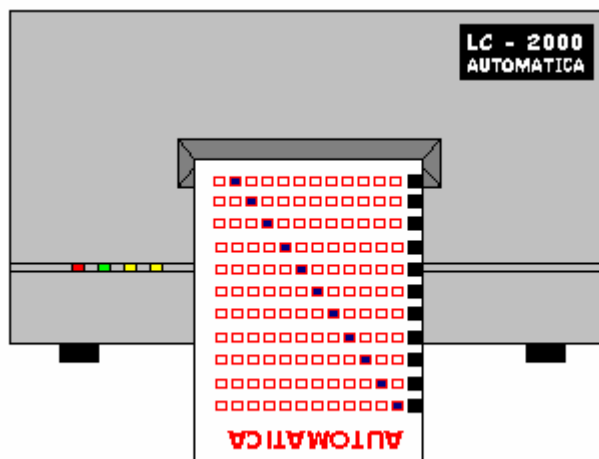
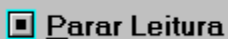



Figura 22

Na medida em que os cartões são lidos, uma reprodução do cartão é mostrada na tela para simples conferência. Se não houver nenhum erro, os dados dos cartões são armazenados no arquivo de dados.

Se o usuário optar por fazer crítica aos campos, todos os grupos que têm este recurso serão criticados. Ou seja, se houver campos em branco ou com múltipla marcação, o programa é interrompido e uma tela de advertência é mostrada. O usuário pode optar por aceitar ou não o cartão.



 **Parar Leitura** Clique neste botão quando desejar parar a leitura dos cartões para encerrar o programa ou mudar de arquivo de configuração e/ou de dados.

## Exemplo de Arquivo de Dados

A seguir é mostrado o arquivo texto gerado pela leitura do cartão da figura 23. O cartão foi lido apenas uma vez. O arquivo gerado fica da seguinte forma:

```
01234562ABCDED CBABCDED CBABCDED CBABCDED ABCDED CBABCDED CBABCDED CBABCDED <EOL> <E
OF>
```

Cada registro de cartão é finalizado com um caracter de fim de linha (*end of line*) <EOL>. O arquivo é finalizado com o caracter de fim de arquivo (*end of file*) <EOF>. Esses caracteres não são mostrados quando o arquivo de dados é aberto no bloco de notas.

Figura 23

O arquivo texto gerado pela leitura dos cartões deve ser então processado pelo usuário para comparação com os gabaritos e armazenamento dos dados finais nos bancos de dados.

## Erros de Leitura

Os erros de leitura de um cartão podem ocorrer em função de:

- Marcações mal feitas;
- Cartão marcado com tinta muito reflexiva;
- Cartão confeccionado com cores fora da especificação;
- Cartão confeccionado fora do padrão das medidas;
- Cartão amassado;
- Cartão molhado;
- Número de linhas não corresponde ao configurado;
- Campo de Inscrição preenchido incorretamente;

Para evitar erros de leitura é muito importante que os cartões sejam confeccionados com as dimensões descritas neste manual. Se as quadrículas estiverem deslocadas ou com dimensões diferentes, podem ocorrer leituras incorretas. Se os *strokes* estiverem muito claros ou muito pequenos, podem ocorrer erros devido a número de linhas incompatível com a configuração. Se a largura do cartão estiver maior que o especificado, o cartão pode agarrar e travar o motor da máquina. Se estiver menor, o cartão pode ser tracionado na diagonal, fazendo com que as quadrículas se desloquem de sua posição correta e gerando leituras incorretas.

Para marcação dos cartões deve ser utilizada caneta esferográfica azul ou preta. A quadrícula deve ser completamente preenchida, sem que sua delimitação seja ultrapassada. Nenhum risco ou marcação deve ser feito na coluna destinada aos *strokes* para que a leitora não interprete incorretamente como uma nova linha.

**Atenção:** Não coloque na leitora cartões que tenham sido corrigidos com corretivo líquido. A tinta pode se acumular nos sensores gerando erros de leitura.

## Uso da DLL LC2000.DLL

A LC2000.DLL foi originalmente construída em *Delphi*. Esta biblioteca dinâmica de funções permite que o usuário possa fazer aplicativos para controlar diretamente a Leitora de Cartões LC2000 usando as várias linguagens de programação visual desenvolvidas para *Windows*.

**OBS.: Programas feitos utilizando DLLs das versões 5.0, 5.1 e 6.0 NÃO podem usar diretamente a DLL versão 7.0. Além da substituição do arquivo .DLL, devem ser feitas modificações no código para adequar o tamanho da matriz de dados enviados e recebidos da leitora.**

Abaixo, são descritas as funções que compõem a DLL:

- 1 Inicializa
- 2 Testa\_Comunicacao
- 3 Recebe\_Status
- 4 Envia\_Configuracao
- 5 Le\_Cartao
- 6 Recebe\_Cartao
- 7 Reseta\_Leitora
- 8 Cancela\_Leitura
- 9 Envia\_Mascara
- 10 Finaliza

### Função Inicializa a Leitora

Esta função inicializa a comunicação entre a leitora e o computador.  
Declaração:

**Function Inicializa (Porta : byte): boolean; stdcall;**

Comentários:

Se não houver nenhum problema de erro de comunicação (*checksum*) e/ou falha de comunicação (*time-out*), a função Inicializa retorna *True*.

**Porta:** Esta variável pode assumir qualquer valor entre 1 e 9, desde que existam tais portas seriais configuradas no computador. A comunicação entre a leitora e o computador é feita em 56000 bps, 8 bits de dados, 1 stop bit e sem paridade.

### Função Testa Comunicação

Esta função testa a comunicação entre a leitora e o computador.  
Declaração:

**Function Testa\_Comunicacao (var Status: boolean): boolean; stdcall;**

Comentários:

A função retorna *True* se não houver falha ou erro de comunicação. Um comando é enviado para a leitora e ela retorna o *status* de funcionamento.

**Status:** Esta variável retorna *True* se a leitora estiver OK. Caso contrário retorna *False*.

### Função Recebe Status

Esta função recebe o *status* do último cartão lido.  
Declaração:

**Function Recebe\_Status (var Status: boolean): boolean; stdcall;**

Comentários:

A função retorna *True* se não houver falha ou erro de comunicação.

**Status:** Esta variável retorna *True* se a leitora tiver lido o último cartão sem nenhum erro de leitura e/ou de número de linhas (*strokes*). Caso contrário retorna *False*.

## Função Envia Configuração

Esta função envia a configuração do cartão para a leitora.  
Declaração:

**Function Envia\_Configuracao (NumStrobe, TipoLeitura: byte): boolean; stdcall;**

Comentários:

A função retorna *True* se não houver falha ou erro de comunicação.

**NumStrobe:** Esta variável define o número de linhas dos cartões a serem lidos pela leitora. O número máximo de linhas, conseqüentemente o valor máximo que esta variável pode assumir é 90. Neste caso o cartão apresenta um formato customizado. Se o usuário quiser que a leitora leia cartões com tamanhos diferentes de linhas, ele deve definir NumStrobe=0. Vale salientar que se optar por formato livre dos cartões, o teste de verificação do tamanho do cartão não é feito.

**TipoLeitura:** Esta variável permite que a leitora leia os cartões de duas maneiras: Unidirecional e Bidirecional. Para optar por unidirecional, TipoLeitura deve conter o valor ordinal do caracter 'U' ( ord('U') = 85) e para bidirecional deve conter o valor ordinal do caracter 'B' ( ord('B')=66).

### Formato Livre:

A leitora se encontra neste modo se o número de linhas do cartão for definido como zero (NumStrobe = 0). Esta opção é recomendada quando o usuário quer entrar com cartões de tamanhos diferentes e de forma misturada. Neste caso, o teste de verificação do tamanho do cartão não é habilitado.

### Formato Customizado:

Este modo de leitura garante que a leitora só vai ler cartões com um número de linhas definido pelo usuário (NumStrobe = 38, por exemplo). Neste caso, se for inserido algum cartão com um número menor ou maior de linhas do que o programado pela configuração, uma mensagem de erro será mostrada na tela.

### Modo Unidirecional:

Neste modo de operação, o cartão entra na leitora pela parte frontal e é expelido por trás, sendo feita uma única leitura das quadrículas. Se houver erros de leitura, o cartão é reposicionado para mais duas tentativas de leitura. Se, após as três tentativas, continuar havendo erro, o *Status* de leitura da leitora sinalizará *False*.

### Modo Bidirecional:

Neste modo de operação, o cartão entra e retorna pela frente, de modo que há leitura das quadrículas nos dois sentidos de movimento. Este modo de leitura lê pelo menos duas vezes as quadrículas antes de validar os dados, portanto este modo de operação é o mais recomendado. Se houver erros de leitura, o cartão é reposicionado para mais duas tentativas de leitura. Se, após as três tentativas, continuar havendo erro, o *Status* de leitura da leitora sinalizará *False*.

## Função Lê Cartão

Esta função posiciona a leitora para aguardar a leitura de um cartão. O *led* verde do painel da leitora acende sinalizando a espera do cartão.  
Declaração:

**Function Le\_Cartao (var Status: boolean): boolean ; stdcall;**



Comentários:

A função retorna *True* se não houver falha ou erro de comunicação. A função não retorna enquanto não ler um cartão. Para cancelar esta função e obrigar o seu retorno imediato, use a função *Cancela\_Leitura*.

**Status:** Esta variável retorna *True* quando o cartão for lido. Se a função *Cancela\_Leitura* for chamada antes da leitura do cartão, esta variável retorna *False*.

## Função Recebe Cartão

Declaração:

```
type TMat = array [1..1081] of char;  
Function Recebe_Cartao (var Matriz :TMat) : boolean ; stdcall;
```

Comentários:

Esta função recebe os dados do último cartão lido pela leitora e armazena-os em um vetor. Como o tamanho máximo do cartão é de 90 linhas, o vetor é dimensionado para 1081 posições (90 \* 12) mais o caractere *NULL* do fim do arranjo. Cada linha do cartão contém 12 quadrículas.

**Matriz:** Este vetor armazena um caractere para cada quadrícula das linhas do cartão. Se a quadrícula estiver marcada, o caractere '1' (valor ordinal 49) é armazenado na respectiva posição, caso contrário o caractere '0' (valor ordinal 48) é armazenado. O caractere *NULL* é colocado logo após a última quadrícula do cartão, como delimitador do fim do vetor.

Para o cartão da figura 24, veja o vetor gerado a seguir:

**AT AUTOMATICA**  
TECNOLOGIA  
TEL. (27) 3325-6778  
www.automatica.com.br

INSCRIÇÃO

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

PROVA 1 3 4

|    |   |   |   |   |   |    |   |   |   |   |   |
|----|---|---|---|---|---|----|---|---|---|---|---|
| 1  | A | B | C | D | E | 31 | A | B | C | D | E |
| 2  | A | B | C | D | E | 32 | A | B | C | D | E |
| 3  | A | B | C | D | E | 33 | A | B | C | D | E |
| 4  | A | B | C | D | E | 34 | A | B | C | D | E |
| 5  | A | B | C | D | E | 35 | A | B | C | D | E |
| 6  | A | B | C | D | E | 36 | A | B | C | D | E |
| 7  | A | B | C | D | E | 37 | A | B | C | D | E |
| 8  | A | B | C | D | E | 38 | A | B | C | D | E |
| 9  | A | B | C | D | E | 39 | A | B | C | D | E |
| 10 | A | B | C | D | E | 40 | A | B | C | D | E |
| 11 | A | B | C | D | E | 41 | A | B | C | D | E |
| 12 | A | B | C | D | E | 42 | A | B | C | D | E |
| 13 | A | B | C | D | E | 43 | A | B | C | D | E |
| 14 | A | B | C | D | E | 44 | A | B | C | D | E |
| 15 | A | B | C | D | E | 45 | A | B | C | D | E |
| 16 | A | B | C | D | E | 46 | A | B | C | D | E |
| 17 | A | B | C | D | E | 47 | A | B | C | D | E |
| 18 | A | B | C | D | E | 48 | A | B | C | D | E |
| 19 | A | B | C | D | E | 49 | A | B | C | D | E |
| 20 | A | B | C | D | E | 50 | A | B | C | D | E |
| 21 | A | B | C | D | E | 51 | A | B | C | D | E |
| 22 | A | B | C | D | E | 52 | A | B | C | D | E |
| 23 | A | B | C | D | E | 53 | A | B | C | D | E |
| 24 | A | B | C | D | E | 54 | A | B | C | D | E |
| 25 | A | B | C | D | E | 55 | A | B | C | D | E |
| 26 | A | B | C | D | E | 56 | A | B | C | D | E |
| 27 | A | B | C | D | E | 57 | A | B | C | D | E |
| 28 | A | B | C | D | E | 58 | A | B | C | D | E |
| 29 | A | B | C | D | E | 59 | A | B | C | D | E |
| 30 | A | B | C | D | E | 60 | A | B | C | D | E |

ASSINATURA

Figura 24

O cartão da figura 24 é composto de 38 linhas em 3 grupos, e o vetor retornará com o seguinte valor:

Matriz = {

|     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '0' | '0' |
| '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' |
| '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' |
| '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' |
| '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' | '0' |
| '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' |
| '0' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' |
| '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' | '0' |
| '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' |
| '0' | '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' | '0' |
| '0' | '0' | '0' | '1' | '0' | '0' | '0' | '0' | '0' | '1' | '0' | '0' |

```
'0','0','1','0','0','0','0','0','1','0','0','0',
'0','1','0','0','0','0','0','1','0','0','0','0',
'0','0','1','0','0','0','0','0','1','0','0','0',
'0','0','0','1','0','0','0','0','0','1','0','0',
'0','0','0','0','1','0','0','0','0','0','1','0',
'0','0','0','0','0','1','0','0','0','0','0','1',
'0','0','0','0','1','0','0','0','0','0','1','0',
'0','0','0','1','0','0','0','0','0','1','0','0',
'0','0','1','0','0','0','0','0','1','0','0','0',
'0','0','1','0','0','0','0','0','1','0','0','0',
'0','1','0','0','0','0','0','1','0','0','0','0',
'0','1','0','0','0','0','0','1','0','0','0','0',
'0','0','1','0','0','0','0','1','0','0','0','0',
'0','0','0','1','0','0','0','0','1','0','0','0',
'0','0','0','0','1','0','0','0','0','1','0','0',
'0','0','0','0','0','1','0','0','0','0','1','0',
'0','0','0','0','1','0','0','0','0','0','1','1',
'0','0','0','0','1','0','0','0','0','0','1','0',
#0, ... , #0 }
```

**Observação:** todas as posições do vetor que não são lidas pelo cartão, neste exemplo de 457 a 1081 estarão ocupadas com o caractere *NULL* (#0).

## Função Reinicializa Leitora

Esta função reinicializa a comunicação entre a leitora e o computador.  
Declaração:

```
Function Reseta_Leitora: boolean; stdcall;
```

Comentários:

A função retorna *True* se não houver falha ou erro de comunicação.

## Função Cancela Leitura

Esta função cancela a função **Le\_Cartao**. O *led* verde de sinalização da leitora apaga impedindo a leitura do cartão.

Declaração:

```
Function Cancela_Leitura : boolean; stdcall;
```

Comentários:

A função retorna *True* se não houver falha ou erro de comunicação.

## Função Envia Máscara

Esta função envia o vetor para mascaramento das quadrículas da leitora.

Declaração:

```
type TMat = array [1..1081] of char;  
Function Envia_Mascara (var Mascara: TMat): boolean ; stdcall;
```

Comentários:

O objetivo do mascaramento de algumas quadrículas do cartão é permitir que elas possam ser usadas para escrita com caneta ou impressos como, por exemplo, o campo de inscrição (Figura 23). Em alguns casos os números são impressos na lateral e o usuário tem que pintar as quadrículas correspondentes.

O conteúdo de toda quadrícula mascarada é ignorado, ou seja, mesmo se houver marcação definida a leitora considera a quadrícula em branco.

**Mascara:** Neste vetor deve-se armazenar um caractere para cada quadrícula das linhas do cartão. Se a quadrícula for **mascarada**, o caractere '0' (valor ordinal 48) deve ser armazenado na

Para o cartão da figura 23, foram mascaradas as quadrículas conforme o vetor abaixo:

**Observação:** todas as posições do vetor que não existam no cartão, neste exemplo, de 457 a 1080, devem ser preenchidas com o caractere '1'. Isso é feito para que não haja mascaramento de linhas extras caso um cartão com maior número de linhas seja lido. A posição 1081 do vetor deve ser preenchida com o caractere *NULL* (#0), delimitando o seu fim.

**Function Finaliza : boolean ; stdcall;**

Comentários:

A função retorna *True* se não houver falha ou erro de comunicação.

**Tipo *byte*:**

O tipo *byte* varia de 0 a 255 não tem sinal e é formado por 8 *bits* (*Unsigned 8-bit*).

**Tipo *char*:**

O tipo *char* genérico corresponde ao *AnsiChar*. Seu valor ocupa um *byte-character*, ordenado de acordo com o conjunto de caracteres *ANSI* estendido.

**Tipo *boolean*:**

O tipo *boolean* ocupa um *byte*. Ele é considerado *False* quando o seu valor ordinal é zero e é considerado *True* quando seu valor ordinal é não-zero.

## Convenções

A convenção se refere à passagem de parâmetros. Parâmetros são transferidos para funções e procedimentos via registradores da CPU e/ou através da pilha (*stack*), dependendo da convenção de chamada das funções e procedimentos.

Esta biblioteca de funções foi compilada para 5 convenções de chamada, conforme descrito abaixo. Use a que melhor se adequar para *link* com a sua linguagem de programação:

| Convenção       | DLL Compilada |
|-----------------|---------------|
| <i>stdcall</i>  | LC2000.dll    |
| <i>register</i> | LC2000r.dll   |
| <i>pascal</i>   | LC2000p.dll   |
| <i>cdecl</i>    | LC2000c.dll   |
| <i>safecall</i> | LC2000s.dll   |

As convenções ***register*** e ***pascal*** passam parâmetros da esquerda para a direita, isto é, o parâmetro mais à esquerda é avaliado e passado primeiramente e o parâmetro mais à direita é avaliado e passado por último.

As convenções ***cdecl***, ***stdcall*** e ***safecall*** passam parâmetros da direita para a esquerda. Para todas as convenções exceto ***cdecl***, o procedimento ou função remove os parâmetros da pilha após retorno.

Com a convenção ***register***, usa-se até 3 registradores da CPU para passar parâmetros, enquanto as outras convenções sempre passam todos os parâmetros pela pilha.

Uma interface em que a intenção é ser usada entre módulos, especialmente módulos escritos em linguagens de programação diferentes, todos os métodos devem ser declarados usando ***stdcall***. A convenção ***safecall*** deve ser usada para implementar métodos de interfaces duais.

Abaixo uma tabela sumariza as convenções:

| Convenção       | Ordem              | Limpeza    | Registradores |
|-----------------|--------------------|------------|---------------|
| <i>register</i> | esquerda - direita | função     | sim           |
| <i>pascal</i>   | esquerda - direita | função     | não           |
| <i>cdecl</i>    | direita - esquerda | quem chama | não           |
| <i>stdcall</i>  | direita - esquerda | função     | não           |
| <i>safecall</i> | direita - esquerda | função     | não           |

## Exemplos para *Delphi*

## Exemplo de Inicialização e Configuração da Leitora

A seguir é descrito um exemplo em *Delphi* para inicialização e configuração da Leitora:

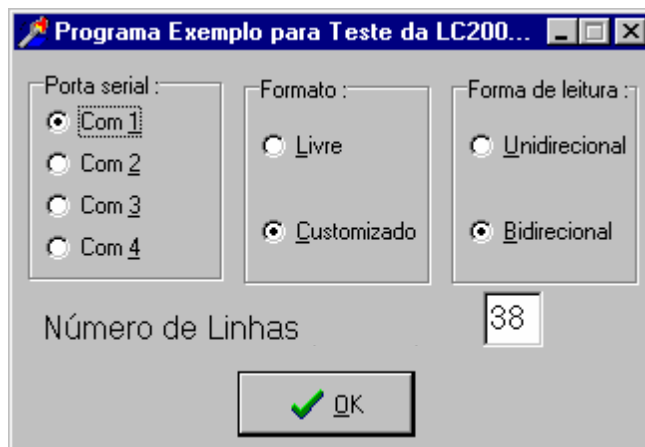


Figura 25

Ao clicar o botão OK, a seguinte rotina de configuração sugerida é executada:

---

```

var
  PrimeiraVez : Boolean; { Variavel global }

type TMat = array [1..1081] of char;

{ funções exportadas da Lc2000.dll }

Function Inicializa (Porta : byte) : boolean; stdcall;
  external 'Lc2000.dll' ;

Function Envia_Configuracao (NumStrobe, TipoLeitura : byte) : boolean; stdcall;
  external 'Lc2000.dll' ;

Function Testa_Comunicacao (var Status : boolean) : boolean; stdcall;
  external 'Lc2000.dll' ;

Function Recebe_Status (var Status : boolean) : boolean; stdcall;
  external 'Lc2000.dll' ;

Function Recebe_Cartao (var Matriz : TMat): boolean; stdcall;
  external 'Lc2000.dll' ;

Function Le_Cartao (var Leu : boolean) : boolean; stdcall;
  external 'Lc2000.dll' ;

Function Reseta_Leitora : boolean; stdcall;
  external 'Lc2000.dll' ;
  
```

Function Finaliza : boolean; stdcall;

external 'Lc2000.dll' ;

Function Cancela\_Leitura : boolean; stdcall;

external 'Lc2000.dll' ;

Function Envia\_Mascara (var Mascara : TMat): boolean; stdcall;

external 'Lc2000.dll' ;

procedure TFrmSetup.BitBtn1Click(Sender: TObject);

var

Status : boolean;

NumStrobe: byte;

TipoLeitura : byte;

Porta : byte;

begin

Case RdgPorta.ItemIndex of

0 : Porta := 1;

1 : Porta := 2;

2 : Porta := 3;

3 : Porta := 4;

end;

If RdgLeitura.ItemIndex = 0

Then

TipoLeitura := ord('U') { leitura Unidirecional foi escolhida }

Else

TipoLeitura := ord('B'); { leitura Bidirecional foi escolhida }

If RdgFormato.ItemIndex = 0

Then

NumStrobe := 0; { Número de linhas livre }

Else

NumStrobe := StrToInt(Edit1.Text); { Número de Strobes = Número de linhas }

If PrimeiraVez then {Confirma se é a primeira vez}

begin

If Inicializa (Porta) then {Se primeira vez apenas inicializa leitora }

```
begin
    Testa_Comunicacao ( Status ); {Testa comunicação}
    If Status Then
        begin
            Envia_Configuracao ( NumStrobe , TipoLeitura );
        end;
    end;
end
else
begin
    If Finaliza then {Caso contrário, finaliza leitora, pois a porta serial pode ser outra }
    begin
        If Inicializa (Porta) then { e depois reinicializa }
        begin
            Testa_Comunicacao ( Status ); {Testa comunicação}
            If Status then
                begin
                    Envia_Configuracao ( NumStrobe , TipoLeitura );
                end;
            end;
        end;
    end;
    PrimeiraVez := False;
end;
```

## Exemplo de uso das funções da Leitora

A seguir é descrito um exemplo em *Delphi* para uso das funções da Leitora:



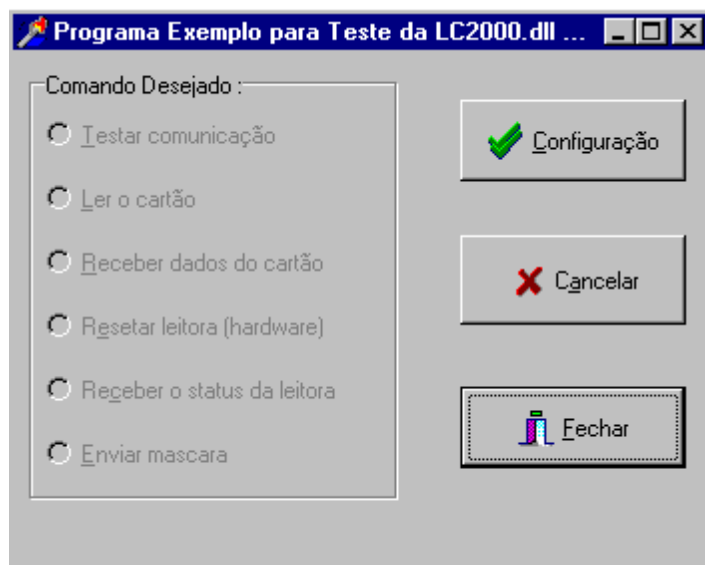


Figura 26

Ao clicar sobre o *RadioGroup*, a seguinte rotina de funções sugerida é executada:

---

```

procedure TFrmLeitora.RdgComandoClick(Sender: TObject);
var
  Matriz, Mascara : TMat;
  I, Y : integer;
  Status, Leu : boolean;
begin
  case RdgComando.ItemIndex of
    0 : begin {Testar comunicação}
      Testa_Comunicacao (Status);
      If Status
      Then
        ShowMessage('A leitora está comunicando corretamente.') ;
      Else
        ShowMessage('A leitora não está comunicando corretamente.') ;
      end;
    1 : begin {Ler o cartão}
      Le_Cartao(Leu);    { programa fica esperando passar o cartão na leitora }
      If Leu
      Then
        ShowMessage('O cartão foi lido corretamente.') ;
      Else
        ShowMessage('A leitura foi abortada, a função Cancela_Leitura foi executada.');
```

```
If Recebe_Cartao (Matriz)
Then
  ShowMessage('O cartão foi recebido corretamente.') ;
  {Mostrar aqui o cartão}
Else
  ShowMessage('O cartão não foi recebido corretamente.') ;
end;
3 : begin  {Resetar leitora}
  Reseta_Leitora;
end;
4 : begin  {Receber status}
  Recebe_Status (Status);
  if Status
  Then
    ShowMessage('A leitora está OK.') ;
  Else
    ShowMessage('A leitora está com problema.') ;
  end;
5 : begin  {Envia Mascara }
  For I := 1 To 1081 do
  begin
    Mascara[I] := '0';
    { estamos mandando uma máscara para anular todas as quadrículas }
  end;
  if Envia_Mascara (Mascara)
  Then
    ShowMessage('A mascara foi enviada.') ;
  Else
    ShowMessage('Falha ao enviar mascara.') ;
  end;
end;
end;
```

Ao clicar o botão Cancela, a seguinte rotina de configuração sugerida é executada:

---

```
procedure TFrmLeitora.BtnCancelarClick(Sender: TObject);
begin
  Cancela_Leitura ;
end;
```

Ao clicar o botão Fechar, a seguinte rotina de configuração sugerida é executada:

---

```
procedure TFrmLeitora.BtnCloseClick(Sender: TObject);  
begin  
    Finaliza; { Avisa a leitora que o programa fechou }  
    Close;    {sai do programa}  
end;
```

## Observação sobre interferência eletromagnética

Como qualquer equipamento eletrônico, a LC2000 pode apresentar funcionamento inadequado devido a interferência eletromagnética. Para evitar problemas com a LC2000 devido a interferência eletromagnética, evite o uso de rádios comunicadores (tipo walkie-talkie) ou de dispositivos de transmissão “wireless” próximos à máquina.

## Especificações Técnicas

- Peso aprox.: 3,0 kg;
- Alimentação : 110 ou 220 V, 60 Hz (seleção através de chave seletora);
- Dimensões externas : 229 mm x 135 mm x 112 mm;
- Processo de Leitura : Ótico Reflexivo;
- Tamanho das Marcas : 2,54 mm x 4,76 mm (1/10” x 3/16”);
- Velocidade de Leitura : 2500 cartões de 38 linhas por hora em modo bidirecional (testada em computador com clock de 700 MHz);
- Comunicação : Serial RS232C (pode ser ligada à porta USB através de adaptador não incluso);
- Acabamento : pintura epóxi cor bege.

## Garantia

1. A garantia da Leitora Ótica de Cartões LC2000 estende-se por 180 dias (6 meses) contados a partir da data de emissão da nota fiscal.
2. A garantia e a assistência técnica da leitora é dada pela Automatica Tecnologia S.A. em sua sede, sendo que as despesas de frete para envio e retorno são de responsabilidade do cliente.
3. Em caso de mau funcionamento da leitora, o cliente deve contactar imediatamente nosso Departamento de Suporte Técnico por telefone no horário de 8h às 12h e de 14h às 18h, de segunda a sexta-feira, ou por e-mail.
4. Após contato com o técnico, caso o problema não possa ser solucionado por telefone, a leitora deverá ser enviada devidamente embalada, com os cabos, disquetes, e um conjunto (10 unidades) dos cartões que estão sendo utilizados na leitura.
5. A troca imediata da leitora defeituosa só será feita se o defeito for notificado até 15 dias após a data da emissão da nota fiscal e, caso não haja leitora em estoque, o cliente deverá aguardar a reposição. Após este prazo, o equipamento será reparado.
6. A garantia pressupõe que a leitora tenha sido instalada e operada rigorosamente de acordo com as instruções do manual do equipamento.
7. Excluem-se da garantia os danos causados por transporte, uso indevido fora das especificações técnicas e conexão da leitora à rede de alimentação elétrica de 220V com a chave de seleção de voltagem ajustada para 110V.
8. A garantia cessará imediatamente se for constatado que a leitora foi aberta sem autorização.
9. O equipamento deverá ser acompanhado de nota fiscal de remessa para conserto. Não nos responsabilizamos por equipamentos enviados sem nota fiscal e nem por avarias de transporte.

Endereço:

**AUTOMATICA TECNOLOGIA S.A.**

**Departamento Técnico**

**Rua Comissário Octávio de Queiroz, 40, Jardim da Penha – Vitória, ES**

**Tel: (27) 3325 6778**

**Fax: (27) 3225 2234**

**Web: [www.automatica.com.br](http://www.automatica.com.br)**

**e-mail: [suporte@automatica.com.br](mailto:suporte@automatica.com.br)**