

**MANUAL**

**DE**

**PROGRAMAÇÃO**

**DLL para Windows**  
**URANOPOS.DLL**

Revisão 1.00

## Índice

Índice	2
Revisões deste manual	3
Introdução	4
Funções	4
Funções	5
Sequência das Funções	5
Descrição das Funções	6
Le_VersaoDLL	7
Le_Status	8
Le_Status_Gaveta	9
AbrePortaSerial	10
FechaPortaSerial	11
AbreSocket	12
FechaSocket	13
Abre_Gaveta	14
AcionaGuilhotina	15
ImprimeTexto	16
ImprimeTextoLF	17
ImprimeTxFormatado	18
ImprimeTextoInvertido	19
ImprimeTexto180	20
ImprimeAutoTeste	21
ImprimeQRCODE_M1	22
ImprimeQRCODE_M2	23
ImprimePDF417_M2	24
ImprimeDataMatrix_M2	25
ImprimeCode128C	26
ImprimeCode128_C	27
ImprimeCodigoBarras	28
ImprimeCodigoBarras_C	29
ImprimeLogoNV	30
ComandoTx	31
Informações Adicionais sobre a DLL	32
Anexo I	33

## **Revisões deste manual**

Revisão 1.00 – Versão inicial deste manual.

## Introdução

A UranoPos.dll é uma biblioteca dinâmica que deve ser usada para interface entre o aplicativo comercial e a impressora não fiscal que utiliza o protocolo ESC/POS. Esta biblioteca aplica-se para aplicativos que executam sob o sistema operacional Microsoft Windows® 32 bits e 64 bits, possuindo uma versão de DLL para cada sistema operacional.

## Funções

A biblioteca possui um conjunto de funções, sendo que os comandos a serem enviados e seus respectivos parâmetros devem ser montados pelo aplicativo.

Abaixo, são listadas as funções da biblioteca classificadas por situação de uso:

### □ Funções de inicialização e encerramento:

- AbrePortaSerial
- FechaPortaSerial
- AbreSocket
- FechaSocket

### □ Funções de Leitura de status

- Le\_VersaoDLL
- Le\_Status
- Le\_Status\_Gaveta

### □ Funções relativas a transmissão de comando:

- Abre\_Gaveta
- AcionaGuilhotina
- ImprimeTexto
- ImprimeTextoLF
- ImprimeTxFormatado
- ImprimeTextoInvertido
- ImprimeTexto180
- ImprimeAutoTeste
- ImprimeQRCODE\_M1
- ImprimeQRCODE\_M2
- ImprimePDF417\_M2
- ImprimeDataMatrix\_M2

- ImprimeCode128C
- ImprimeCode128\_C
- ImprimeCodigoBarras
- ImprimeCodigoBarras\_C
- ImprimeLogoNV
- ComandoTx

### Sequência das Funções

A sequência normal de uso destas funções é, em geral, a seguinte:

Observação
As funções que estão entre os caracteres '< >' representam uso essencial da função naquele momento. Quando informada entre os caracteres '[ ]', é um indicativo de uso opcional da função.

**1** Abertura de conexão:

< AbrePortaSerial > //Utilizado para comunicação serial.

ou

< AbreSocket > //Utilizado para comunicação ethernet.

**2** Executar um comando:

[Le\_Status]

[Le\_Status\_Gaveta]

[ImprimeLogoNV]

< ImprimeTexto >

[ImprimeCode128C]

< ImprimeTexto >

[ImprimeQRCODE\_M1]

< ImprimeTexto >

**3** Encerramento de Conexão:

< FechaPortaSerial > //Utilizado para comunicação serial.

ou

< FechaSocket > //Utilizado para comunicação ethernet.

## **Descrição das Funções**

As funções serão descritas através dos seguintes atributos:

Nome da função

Descrição

Parâmetro(s): Nome, Tipo e Descrição de cada parâmetro.

Retorno

Declaração

Exemplo

As declarações da função e exemplos serão ilustrados em Embarcadero Delphi XE8<sup>®</sup> compilado para 32 bits. Alguns comandos podem não possuir exemplos por serem similares a outros exemplos já ilustrados em outros comandos.

## ***Le\_VersaoDLL***

Retorna a versão de compilação da DLL.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna uma String onde a será informado a versão atual da DLL que está sendo utilizada. Esta String é composta por 7 caracteres, Ex: 1.0.0.0

### **Declaração:**

```
Function _Le_VersaoDLL(): PAnsiChar; stdcall; external  
'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button1Click(Sender: TObject);  
  
var  
retorno: String;  
  
begin  
  
retorno:= _Le_VersaoDLL();  
  
ShowMessage(retorno);  
  
end;
```

## **Le\_Status**

Retorna o status atual da impressora.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna um valor do tipo inteiro informando o status atual da impressora contendo o seguinte significado:

- 1: Erro de comunicação / Impressora OFFLINE.
- 5: Gaveta Aberta / Tampa Aberta / Sem Papel.
- 9: Tampa aberta / Sem Papel.
- 24: Impressora ONLINE.
- 32: Gaveta Aberta.

### **Declaração:**

```
Function _Le_Status(): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button2Click(Sender: TObject);
```

```
begin
```

```
  iRetorno:= _Le_Status();
```

```
Case iRetorno of
```

```
  1: ShowMessage('Erro de comunicação / Impressora OFFLINE.');
```

```
  5: ShowMessage('Gaveta Aberta / Tampa Aberta / Sem Papel.');
```

```
  9: ShowMessage('Tampa aberta / Sem Papel');
```

```
 24: ShowMessage('Impressora ONLINE.');
```

```
 32: ShowMessage('Gaveta Aberta.');
```

```
end;
```

```
end;
```

## **Le\_Status\_Gaveta**

Retorna o status atual da gaveta.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna um valor do tipo inteiro informando o status atual da gaveta conectada à impressora contendo o seguinte significado:

- 1: Gaveta Aberta.
- 2: Gaveta Fechada.

### **Declaração:**

```
Function _Le_Status_Gaveta(): Integer; stdcall; external  
'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button3Click(Sender: TObject);  
var  
iRetorno: Integer;  
begin  
  
iRetorno:= _Le_Status_Gaveta();  
  
Case iRetorno of  
  
1: ShowMessage('Gaveta Aberta.');
```

```
2: ShowMessage('Gaveta Fechada.');
```

```
end;
```

```
end;
```

## **AbrePortaSerial**

Abre a porta serial para um canal de comunicação com uma impressora. A velocidade de comunicação utilizada por padrão pela DLL é 115200.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Canal	String AnsiString	Nome da porta serial no formato de uma String. Ex: 'COM1'.
Modelo Imp.	Integer	Modelo de impressora no formato inteiro que é informado na abertura da comunicação serial. <u>Modelos</u> : - 0: ZP250-U, ZK250-E, ZP500-US-QR, ZP200-US-QR, ZP1FIT-S-QR. - 1: ZP300-USE, ZK300-USE, ZP160-W. - 2: ZP250-USE. - 3: ZP80-UBT, ZP220-UBT.

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 0: Insucesso.
- 1: Sucesso.

### **Declaração:**

```
Function _AbrePortaSerial(Canal: AnsiString; ModeloImp: Integer):  
Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button4Click(Sender: TObject);  
var  
iRetorno: Integer;  
begin  
  
iRetorno:= _AbrePortaSerial('COM1',0);  
  
Case iRetorno of  
  
0: ShowMessage('Impossível abrir a porta serial.');
```

```
1: ShowMessage('Abertura de porta realizada com sucesso.');
```

```
end;
```

```
end;
```

## **FechaPortaSerial**

Fecha a porta serial para a comunicação.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 0: Insucesso.
- 1: Sucesso.

### **Declaração:**

```
Function _FechaPortaSerial(): Integer; stdcall; external  
'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button5Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
iRetorno:= _FechaPortaSerial();  
  
Case iRetorno of  
  
0: ShowMessage('Impossível fechar a porta serial.');
```

```
1: ShowMessage('Fechamento de porta realizado com sucesso.');
```

```
end;
```

```
end;
```

## **AbreSocket**

Abre a porta de comunicação via socket com uma impressora.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Endereço IP	String AnsiString	Endereço IP atribuído a impressora no formato de uma String. Ex: '192.168.0.100'.
Modelo Imp.	Integer	Modelo de impressora no formato inteiro que é informado na abertura da comunicação. <u>Modelos</u> - 0: ZK250-E. - 1: ZP300-USE, ZK300-USE, ZP80-UW, ZP160-W. - 2: ZP250-USE.

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 0: Insucesso.
- 1: Sucesso.

### **Declaração:**

```
Function _AbreSocket(End_IP: AnsiString; ModeloImp: Integer):  
Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button6Click(Sender: TObject);  
var  
iRetorno: Integer;  
begin  
  
iRetorno:= _AbreSocket('192.168.0.100',0);  
  
Case iRetorno of  
  
0: ShowMessage('Impossível abrir a comunicação.');
```

```
1: ShowMessage('Abertura realizada com sucesso.');
```

```
end;
```

```
end;
```

## **FechaSocket**

Encerra a comunicação via socket com uma impressora.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 0: Insucesso.
- 1: Sucesso.

### **Declaração:**

```
Function _FechaSocket(): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button7Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
iRetorno:= _FechaSocket();  
  
Case iRetorno of  
  
0: ShowMessage('Impossível fechar o socket.');
```

```
1: ShowMessage('Fechamento de socket realizado com sucesso.');
```

```
end;
```

```
end;
```

## **Abre\_Gaveta**

Aciona o comando de abertura na porta da gaveta de moedas da impressora.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _Abre_Gaveta(): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button8Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
iRetorno:= _Abre_Gaveta();  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## **AcionaGuilhotina**

Aciona a guilhotina (Cutter) da impressora.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _AcionaGuilhotina(): Integer; stdcall; external  
'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button9Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
iRetorno:= _AcionaGuilhotina();  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ***ImprimeTexto***

Imprime os dados que estão no buffer de impressão e avança uma linha.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.
Alinhamento Posição	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Fonte	Integer	Tamanho da fonte a ser utilizado na impressão. 0: Fonte A (12x24 - 48 Col) 1: Fonte B (9x17- 64 Col)
Espaço Linha	Integer	Espaço de impressão entre linhas. 0~6
Negrito	Integer	Texto será impresso em modo negrito. 0: Modo desativado 1: Modo Ativado

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeTexto(sTexto: AnsiString; iPosicao: Integer;  
iFonte: Integer; iEspacoLinha: Integer; iNegrito: Integer):  
Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button10Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
        //String, Posicao, Fonte, EspacLinha, Negrito  
iRetorno:= _ImprimeTexto('Texto impresso',1,1,2,1);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ***ImprimeTextoLF***

Imprime os dados que estão no buffer de impressão e avança n° linhas.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.
Alinhamento Posição	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Fonte	Integer	Tamanho da fonte a ser utilizado na impressão. 0: Fonte A (12x24 - 48 Col) 1: Fonte B (9x17- 64 Col)
Espaço Linha	Integer	Espaço de impressão entre linhas. 0~6
Negrito	Integer	Texto será impresso em modo negrito. 0: Modo desativado 1: Modo Ativado
Número Linhas	Integer	Número de linhas que serão avançadas. 0~255

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeTextoLF(sTexto: AnsiString; iPosicao: Integer;  
iFonte: Integer; iEspacoLinha: Integer; iNegrito: Integer;  
iNLinhas: Integer): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button11Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //String, Posicao, Fonte, EspacLinha, Negrito, Nlinhas  
    iRetorno:= _ImprimeTextoLF('Texto impresso',1,1,2,1,5);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ***ImprimeTxFormatado***

Imprime os dados que estão no buffer de impressão no formato selecionado e avança uma linha.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.
Alinhamento Posição	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Fonte	Integer	Tamanho da fonte a ser utilizado na impressão. 0: Fonte A (12x24 - 48 Col) 1: Fonte B (9x17- 64 Col)
Espaço Linha	Integer	Espaço de impressão entre linhas. 0~6
Formato	Integer	Formato em que o texto será impresso. 1~6 *

*\*Consulte Anexo I para mais informações.*

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeTxFormatado(sTexto: AnsiString; iPosicao: Integer;  
iFonte: Integer; iEspacoLinha: Integer; iFormato: Integer):  
Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button12Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //String, Posicao, Fonte, EspacLinha, NFormato  
iRetorno:= _ImprimeTxFormatado('Texto impresso',1,1,2,5);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

### ***ImprimeTextoInvertido*** (*Inverte a cor preto por branco na impressão*)

Imprime os dados que estão no buffer de impressão e avança uma linha.

#### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.
Alinhamento Posição	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Fonte	Integer	Tamanho da fonte a ser utilizado na impressão. 0: Fonte A (12x24 - 48 Col) 1: Fonte B (9x17- 64 Col)
Espaço Linha	Integer	Espaço de impressão entre linhas. 0~6

#### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

#### **Declaração:**

```
Function _ImprimeTextoInvertido(sTexto: AnsiString; iPosicao:  
Integer; iFonte: Integer; iEspacoLinha): Integer; stdcall; external  
'UranoPos.dll';
```

#### **Exemplo:**

```
procedure TForm1.Button13Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
                //String, Posicao, Fonte, EspacLinha  
iRetorno:= _ImprimeTextoInvertido('Texto impresso',1,1,2);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

### ***ImprimeTexto180*** (*Imprime o texto de cabeça para baixo*)

Imprime os dados que estão no buffer de impressão e avança uma linha.

#### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.
Alinhamento Posição	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Fonte	Integer	Tamanho da fonte a ser utilizado na impressão. 0: Fonte A (12x24 - 48 Col) 1: Fonte B (9x17- 64 Col)
Espaço Linha	Integer	Espaço de impressão entre linhas. 0~6
Negrito	Integer	Texto será impresso em modo negrito. 0: Modo desativado 1: Modo Ativado

#### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

#### **Declaração:**

```
Function _ImprimeTexto180 (sTexto: AnsiString; iPosicao: Integer;  
iFonte: Integer; iEspacoLinha; iNegrito: Integer): Integer;  
stdcall; external 'UranoPos.dll';
```

#### **Exemplo:**

```
procedure TForm1.Button14Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
        //String, Posicao, Fonte, EspacLinha, Negrito  
iRetorno:= _ImprimeTexto180('Texto impresso',1,1,2,1);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ***ImprimeAutoTeste***

Imprime um cupom de autoteste na impressora desde que esta possua esta opção.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Nenhum	Nenhum	Nenhum

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeAutoTeste(): Integer; stdcall; external  
'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button15Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
iRetorno:= _ImprimeAutoTeste();  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ***ImprimeQRCODE\_M1***

Imprime um código de barras no formato QRCode modelo 1.

Esta função é compatível com quase todos os modelos de impressoras da Urano.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Alinhamento	Integer	Posição em que o texto deve ser impresso.
Posição		0: Esquerda. 1: Centro. 2: Direita.
Modelo	Integer	Define o modelo do QRCode que será utilizado. 1~2
Tamanho	Integer	Define o tamanho para o QRCode. 1~7
Nível de Correção	Integer	Define o nível de correção de erro para o QRCode. 0: 7%   1: 15%   2: 25%   3: 30%
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeQRCODE_M1(iAlinQR: Integer; iModelo: Integer;  
iTamanhoQR: Integer; iNCorrecao: Integer; sTextoQR: AnsiString):  
Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button16Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //Posicao, Modelo, Tamanho, NivelCorreção, Texto  
    iRetorno:= _ImprimeQRCODE_M1(1,1,4,0, 'Texto impresso');  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## **ImprimeQRCODE\_M2** (\*Consulte Anexo I para mais informações)

Imprime um código de barras no formato QRCode modelo 2.  
Esta função é compatível com apenas alguns modelos de impressoras da Urano.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Alinhamento	Integer	Posição em que o texto deve ser impresso.
Posição		0: Esquerda. 1: Centro. 2: Direita.
Modelo	Integer	Define o modelo do QRCode que será utilizado. 1~3
Tamanho	Integer	Define o tamanho para o QRCode. 1~7
Nível de Correção	Integer	Define o nível de correção de erro para o QRCode. 0: 7%   1: 15%   2: 25%   3: 30%
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.

*\*Consulte Anexo I para mais informações.*

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeQRCODE_M2(iAlinQR: Integer; iModelo: Integer;  
iTamanhoQR: Integer; iNCorrecao: Integer; sTextoQR: AnsiString):  
Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button17Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //Posicao, Modelo, Tamanho, NívelCorreção, Texto  
    iRetorno:= _ImprimeQRCODE_M2(1,1,4,0, 'Texto impresso');  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## **ImprimePDF417\_M2** (\*Consulte Anexo I para mais informações)

Imprime um código de barras no formato PDF417 modelo 2.  
Esta função é compatível com apenas alguns modelos de impressoras da Urano.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Alinhamento	Integer	Posição em que o texto deve ser impresso.
Posição		0: Esquerda. 1: Centro. 2: Direita.
Caracteres Linha	Integer	Define a quantidade de caracteres por linha. 1~30
Relação X Y	Integer	Indica a ampliação longitudinal. 1~6 *
Nível de Correção	Integer	Define o nível de correção. 0~8 *
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.

\*Consulte Anexo I para mais informações.

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimePDF417_M2(iAlinPDF: Integer; iNCaracLinha: Integer; iRelX_Y: Integer; iNCorrecao: Integer; sTextoPDF: AnsiString): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button18Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //Posicao, CaracteresLinha, Relação XY, NívelCorreção, Texto  
    iRetorno:= _ImprimePDF417_M2(1,20,4,0, 'Texto impresso');  
  
    Case iRetorno of  
  
    1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## **ImprimeDataMatrix\_M2** (\*Consulte Anexo I para mais informações)

Imprime um código de barras no formato DataMatrix modelo 2.  
Esta função é compatível com apenas alguns modelos de impressoras da Urano.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Alinhamento	Integer	Posição em que o texto deve ser impresso.
Posição		0: Esquerda. 1: Centro. 2: Direita.
Altura Módulo	Integer	Define a altura do módulo 0~4
Largura Módulo	Integer	Define a largura do módulo. 0~4
Tamanho Módulo	Integer	Define o tamanho do módulo. 1~7
Texto	String	Texto a ser impresso no formato de uma String.
	AnsiString	Ex: 'Texto impresso'.

*\*Consulte Anexo I para mais informações.*

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeDataMatrix_M2 (iAlinDM: Integer; iAlturaMod: Integer; iLarguraMod: Integer; iTamanhoMod: Integer; sTextoDM: AnsiString): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button19Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
        //Posicao, Altura, Largura, Tamanho, Texto  
iRetorno:= _ImprimeDataMatrix_M2(1,4,4,5, 'Texto impresso');  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## **ImprimeCode128C**

Imprime um código de barras no formato code128C.

Esta função está configurada para imprimir os 22 pares da chave de acesso formatado para atender padrão CFe-Sat.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Chave de Acesso	String AnsiString	Deve conter os 22 pares separados por vírgula no formato de uma String. Ex: <code>'43,16,07,90,18,06,21,00,01,97,65,03,00,00,02,23,35,10,00,22,33,50'</code> .
Alinhamento Posição	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Altura Módulo	Integer	Define a altura do módulo 0~2.

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeCode128C(ChaveNum: AnsiString; iPosicao: Integer;  
iAlturaMod: Integer): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button20Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
                //Chave, Posição, Altura  
iRetorno:= _ImprimeCode128C('43,16,07,90,18,06,21,00,01,97,  
65,03,00,00,02,23,35,10,00,22,33,50', 1, 1);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## **ImprimeCode128\_C**

Imprime um código de barras no formato code128C.

Esta função está configurada para imprimir os 22 pares da chave de acesso divididos em 2 módulos contendo 10 pares cada módulo no formato para atender padrão CFe-Sat.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Chave de Acesso	String AnsiString	Deve conter os 22 pares separados por vírgula no formato de uma String. Ex: '43,16,07,90,18,06,21,00,01,97,65,03,00,00,02,23,35,10,00,22,33,50'.
Alinhamento Posição	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Altura Módulo	Integer	Define a altura do módulo. 0~2
Largura Módulo	Integer	Define a largura do módulo. 0~3

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeCode128_C(ChaveNum: AnsiString; iPosicao: Integer;  
iAlturaMod: Integer; iLarguraMod: Integer): Integer; stdcall;  
external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button21Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
                //Chave, Posição, Altura, Largura  
iRetorno:= _ImprimeCode128_C('43,16,07,90,18,06,21,00,01,97,  
65,03,00,00,02,23,35,10,00,22,33,50',1,1,1);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ImprimeCodigoBarras

Imprime um código de barras no padrão 1D no modelo informado junto ao comando. Para mais informações consulte Anexo1.

### Parâmetro(s):

Nome	Tipo	Descrição
Alinhamento	Integer	Posição em que o texto deve ser impresso.
Posição		0: Esquerda. 1: Centro. 2: Direita.
Modelo	Integer	Define o modelo do código de barras. 0~6 *
Altura Módulo	Integer	Define a altura do módulo. 0~2
Largura Módulo	Integer	Define a largura do módulo. 0~3
Leitura Humana	Integer	Define a posição para leitura humana. 0~3 *
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.

\*Consulte Anexo I para mais informações.

### Retorno:

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### Declaração:

```
Function _ImprimeCodigoBarras(iPosicao: Integer; iModelo: Integer;  
iAltura: Integer; iLargura: Integer; iLH: Integer; sTexto:  
AnsiString): Integer; stdcall; external 'UranoPos.dll';
```

### Exemplo:

```
procedure TForm1.Button22Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //Posição, Modelo, Altura, Largura, Leitura humana, texto  
    iRetorno:= _ImprimeCodigoBarras(1,1,1,1,2, 'Texto impresso');  
  
    Case iRetorno of  
  
    1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ***ImprimeCodigoBarras\_C (Versão Alternativa)***

Imprime um código de barras no padrão 1D de acordo com modelo informado junto ao comando. Para mais informações consulte Anexo1.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Alinhamento	Integer	Posição em que o texto deve ser impresso.
Posição		0: Esquerda. 1: Centro. 2: Direita.
Modelo	Integer	Define o modelo do código de barras. 0~9 *
Altura Módulo	Integer	Define a altura do módulo. 0~2
Largura Módulo	Integer	Define a largura do módulo. 0~3
Leitura Humana	Integer	Define a posição para leitura humana. 0~3 *
Texto	String AnsiString	Texto a ser impresso no formato de uma String. Ex: 'Texto impresso'.

*\*Consulte Anexo I para mais informações.*

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeCodigoBarras_C(iPosicao: Integer; iModelo:  
Integer; iAltura: Integer; iLargura: Integer; iLH: Integer; sTexto:  
AnsiString): Integer; stdcall; external 'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button23Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //Posição, Modelo, Altura, Largura, Leitura humana, texto  
    iRetorno:= _ImprimeCodigoBarras_C(1,1,1,1,2, 'Texto impresso');  
  
    Case iRetorno of  
  
    1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ***ImprimeLogoNV***

Imprime uma imagem previamente armazenada em uma determinada posição da memória interna da impressora.

### **Parâmetro(s):**

<b>Nome</b>	<b>Tipo</b>	<b>Descrição</b>
Posição Memória	Integer	Define a posição da memória que será impresso. 1~5 *
Alinhamento	Integer	Posição em que o texto deve ser impresso. 0: Esquerda. 1: Centro. 2: Direita.
Tamanho	Integer	Define o tamanho da imagem que será impresso. 0~3

*\*Consulte Anexo I para mais informações.*

### **Retorno:**

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### **Declaração:**

```
Function _ImprimeLogoNV(iNLogo: Integer; iPosicao: Integer;  
iTamanho: Integer): Integer; stdcall; external  
'UranoPos.dll';
```

### **Exemplo:**

```
procedure TForm1.Button24Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
  
        //Posicao da memória, Posicao, Tamanho  
iRetorno:= _ImprimeLogoNV(1,1,2);  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## ComandoTx

Permite enviar comandos para a impressora no formato de uma string representando um valor do tipo decimal da tabela ASCII. Cada valor representado deve estar separado por vírgula.

### Parâmetro(s):

Nome	Tipo	Descrição
ComandoTx	String AnsiString	Deve conter os valores representados em decimal separados por vírgula no formato de uma String Ex: '27,105'.

### Retorno:

Retorna um valor do tipo inteiro contendo o seguinte significado:

- 1: Sucesso.

### Declaração:

```
Function _ComandoTx(sTexto: AnsiString): Integer; stdcall; external  
'UranoPos.dll';
```

### Exemplo:

```
procedure TForm1.Button25Click(Sender: TObject);  
var  
iRetorno: Integer;  
  
begin  
    //Comando em decimal representado no formato string  
    iRetorno:= _ComandoTx('27,105');  
  
Case iRetorno of  
  
1: ShowMessage('Comando executado com sucesso.');
```

**end;**

**end;**

## Informações Adicionais sobre a DLL

- ❑ Nome do arquivo DLL: UranoPos.dll
- ❑ Local de instalação: Na pasta SYSTEM do sistema operacional Windows (normalmente a pasta é a '\WINDOWS\SYSTEM') ou na pasta onde o aplicativo executará.
- ❑ Versão da DLL: deve ser obtida executando a função *Le\_VersaoDLL()*.
- ❑ Observações gerais:
  - A DLL foi desenvolvida em C++ Builder, sendo que todos os retornos de suas funções são do tipo inteiro, com exceção da função *Le\_VersaoDLL()* que terá como retorno uma *string*.

## Anexo I

### - Sobre os parâmetros das funções:

Página	Nome da função	Parâmetro	Significado dos parâmetros
18	<i>ImprimeTxFormatado</i>	Formato	Formato em que o texto será impresso: 1: Fonte-B   2: Fonte-A   3: Largura dupla 4: Altura dupla   5: Altura/Largura dupla 6: Sublinhado
28	<i>ImprimeCodigoBarras</i>	Modelo	0: UPC-A   1: UPC-E   2: EAN-13   3: EAN-8 4: CODE39   5: ITF-25   6: CODABAR
28	<i>ImprimeCodigoBarras</i>	L. humana	0: Não   1: Acima   2: Abaixo   3: Ambos
29	<i>ImprimeCodigoBarras_C</i>	Modelo	0: UPC-A   1: UPC-E   2: EAN-13   3: EAN-8 4: CODE39   5: ITF-25   6: CODABAR   7: CODE93   8: COD128   9: EAN128
29	<i>ImprimeCodigoBarras_C</i>	L. humana	0: Não   1: Acima   2: Abaixo   3: Ambos

### - Uso Exclusivo:

Página	Nome da função	Equipamentos compatíveis:
23	<i>ImprimeQRCODE_M2</i>	ZP80-UBT   ZP80-UW   ZP220-UBT
24	<i>ImprimePDF417_M2</i>	ZP80-UBT   ZP80-UW   ZP220-UBT
25	<i>ImprimeDataMatrix_M2</i>	ZP80-UBT   ZP80-UW   ZP220-UBT

### - Reservado:

Nome da função	Obs.
<i>ImprimePDF417_M1</i>	Reservado para modelos novos.
<i>ImprimeDataMatrix_M1</i>	Reservado para modelos novos.