

```

/*
| Tipo de Dado      | Descrição                                     |      |
| Tamanho           |                                               |      |
|-----|-----|-----|
|-----|-----|-----|
| bit               | Armazena valores 0 ou 1                     | 1     |
| bit               |                                               |      |
| tinyint           | Armazena inteiros de 0 a 255                 | 1     |
| byte              |                                               |      |
| smallint          | Armazena inteiros de -32.768 a 32.767        | 2     |
| bytes             |                                               |      |
| int               | Armazena inteiros de -2.147.483.648 a 2.147.483.647 | 4     |
| bytes            |                                               |      |
| bigint            | Armazena inteiros de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 | 8 bytes |
| decimal/numeric   | Armazena números decimais com precisão e escala definidas | 5 a 17 bytes, dependendo da precisão |
| smallmoney        | Armazena valores monetários de -214.748,3648 a 214.748,3647 | 4 bytes |
| money             | Armazena valores monetários de -922.337.203.685.477,5808 a 922.337.203.685.477,5807 | 8 bytes |
| float             | Armazena números de ponto flutuante          | 4     |
| bytes (precisão 1-24) ou 8 bytes (precisão 25-53) |
| real              | Armazena números de ponto flutuante com precisão menor | 4     |
| bytes             |                                               |      |
| date              | Armazena uma data (ano, mês, dia)            | 3     |
| bytes            |                                               |      |
| datetime          | Armazena data e hora, precisa até 1/300 de segundo | 8     |
| bytes            |                                               |      |
| datetime2         | Armazena data e hora com precisão estendida a 8 bytes, dependendo da precisão | 6     |
| smalldatetime     | Armazena data e hora com precisão de minutos | 4     |
| bytes            |                                               |      |
| time              | Armazena apenas a hora do dia                | 3     |
| a 5 bytes, dependendo da precisão |
| char(n)           | Armazena uma cadeia de caracteres de comprimento fixo | n     |
| bytes            |                                               |      |
| varchar(n)        | Armazena uma cadeia de caracteres de comprimento variável | Até n bytes + 2 bytes de overhead |
| nchar(n)          | Armazena uma cadeia de caracteres Unicode de comprimento fixo | 2n bytes |
| nvarchar(n)       | Armazena uma cadeia de caracteres Unicode de comprimento variável | Até 2n bytes + 2 bytes de overhead |
| binary(n)         | Armazena dados binários de comprimento fixo | n     |
| bytes            |                                               |      |
| varbinary(n)      | Armazena dados binários de comprimento variável | Até n bytes + 2 bytes de overhead |
| uniqueidentifier  | Armazena um identificador exclusivo global (GUID) | 16    |
| bytes            |                                               |
*/

```

```

/*
Estrutura para declaração de variáveis no T-SQL

```

```
DECLARE <nome_da_variavel> <tipo_da_variavel> [opcional] = <valor_inicial>;
*/
```

```
DECLARE @i INT = 0;
DECLARE @k INT;
```

```
/*
```

É possível declarar mais de uma variável por vez, mas o tipo de cada uma deve ser especificado: ➤

```
DECLARE var_1 <tipo_1> [inicialização],
        var_2 <tipo_2> [inicialização],
        ...
        var_n <tipo_n> [inicialização];
*/
```

```
DECLARE @r FLOAT = 0,
        @PI FLOAT = 3.14159265,
        @nome VARCHAR(1000) = 'Alexandre Toledo',
        @data_de_inicio DATETIME;
```

```
/*
```

As variáveis podem ser visualizadas com seleções ou com impressões pela função PRINT(.). ➤

```
*/
```

```
PRINT('Nome declarado: ' + @nome);
```

```
/*
```

Variáveis de Sistema (@@)

```
*/
```

```
PRINT('Versão do sistema:' + @@VERSION);
PRINT('Nome do servidor:' + @@SERVERNAME);
PRINT('ID da seção atual:' + @@SPID);
PRINT('Total de conexões no BD:' + @@CONNECTIONS);
PRINT('Número de transações ativas:' + @@TRANCOUNT);
PRINT('Idioma padrão:' + @@LANGUAGE);
PRINT('Tempo limite para bloqueio: ' + @@LOCK_TIMEOUT);
PRINT('Linhas afetadas pela última operação de leitura ou escrita no BD:
      '+@@IDENTITY);
```

```
/*
```

É possível converter uma variável de um tipo para o outro no T-SQL com a função CONVERT(<tipo_destino>, <variavel>). ➤

Obs.: Existe a função CAST(<variavel> AS <tipo_destino>) que tem o mesmo efeito.

```
*/
```

```
PRINT('Versão do sistema:' + @@VERSION);
PRINT('Nome do servidor:' + @@SERVERNAME);
PRINT('ID da seção atual:' + CONVERT(VARCHAR,@@SPID));
PRINT('Total de conexões no BD:' + CONVERT(VARCHAR,@@CONNECTIONS));
PRINT('Número de transações ativas:' + CONVERT(VARCHAR,@@TRANCOUNT));
PRINT('Idioma padrão:' + @@LANGUAGE);
PRINT('Tempo limite para bloqueio: ' + CONVERT(VARCHAR,@@LOCK_TIMEOUT));
PRINT('Linhas afetadas pela última operação de leitura ou escrita no BD:
      '+CONVERT(VARCHAR,@@ROWCOUNT));
```

```
/*
Toda e qualquer atribuição de variáveis fora do escopo de definição da mesma se
faz em um ambiente de configuração (SET).
*/
```

```
DECLARE @i INT = 0;

SET @i = @i + 1;

PRINT(@i);

DECLARE @nome VARCHAR(8000) = 'Bia';

SET @nome = @nome + ' Falcão';

DECLARE @msg VARCHAR(8000) = 'Nome: ' + @nome;

PRINT(@msg);
```

