O USO DE ANS.1 PARA ESPECIFICAÇÃO FORMAL DE PROTOCOLOS

1. A necessidade de uma notação formal

Num sistema de informação distribuído, aplicações remotamente localizadas têm requisitos para transferência de informação (semântica), mas não são envolvidas na determinação dos detalhes da representação de suas informações (sintaxe de transferência) ou codificação durante a comunicação. É o nível de apresentação que provê as funções que definem, através do uso de regras de codificação a sintaxe de transferência para a informação sendo transportada durante a comunicação.

Por exemplo, uma determinada aplicação poderia representar dados tais como "registro-de-pessoal", "relatórios-companhia", "catálogos-biblioteca", ou fragmentos disso, de alguma forma adequada a linguagem de programação usada para escrever o manipulador do protocolo de aplicação. Isto envolveria, tipicamente, *pointers* para uma árvore com uma estrutura, e não uma simples sequência de valores.

O acionamento de uma primitiva P_DATA (que requisita ao nível de apresentação o envio de dados para uma entidade remota), envolveria a passagem de um *pointer* para esta estrutura de dados, que seria manipulada pelo nível de apresentação para codificar os valores e acionar uma primitiva S-DATA (que requisita ao nível de sessão o envio de um conjunto de octetos para a entidade par). Para fazê-lo, o módulo de a presentação também iria requerer, um *pointer* para uma estrutura de dados ou arquivo contendo as definições de tipos.

Por outro lado, o nível de apresentação também provê um serviço comum a todas as aplicações, de negociação de um consenso sobre a sintaxe de transferência da informação que as aplicações desejam comunicar. Esta sintaxe de transferência tem que ser negociada porque somente conhecendo-a, a entidade receptora dos dados terá condições de isolar as unidades de informação (ou campos de um registro ou tela sendo transmitidos). Assim, o nível de apresentação pode receber os dados do nível de aplicação estruturados de acordo com uma dada sintaxe de apresentação, restruturá-los de acordo com outra sintaxe de apresentação (a sintaxe de transferência) e o nível de apresentação receptor, pode, ainda, tornar a promover uma restruturação dos dados para apresentá-los ao seu nível de aplicação de uma forma compreensível para este. No caso mais simples, a entidade de aplicação que está enviando os dados não os e strutura de forma alguma (eles consistem apenas de um único *string* de caracteres do código padrão (ISO646), os dados são transferidos da mesma forma e entregues para o nível de aplicação do mesmo formato. Neste caso, diz-se que o nível de apresentação é nulo.

Numa situação em que a sintaxe de apresentação dos dados fosse diferente para as duas entidades de aplicação interagindo, teria que ser possível negociar a sintaxe de transferência a ser usada. Para negociar uma sintaxe é preciso poder, no mínimo, referí-la por um identificador ou ainda, definí-la e associar-lhe um identificador ou nome. Esta associação entre um conjunto de valores de dados e sua representação constitui o que é denominado um contexto de apresentação definido.

Os nomes e a definição associada devem ser registrados junto a alguma organização de Registro, designada pela ISO. Tal organização manterá registro dos nomes de sintaxe alocados e da definição da sintaxe associada com cada nome específico. Assim, quando em uma negociação de sintaxe de transferência for referenciado um nome registrado, estará especificado, sem ambiguidades, as regras de estruturação a serem usadas durante a transferência. A negociação de contexto somente e usada no nível de apresentação porque nos níveis inferiores do modelo OS, cada parâmetro de dados do usuário, numa primitiva de serviço, e especificado como um valor binário numa sequência de octetos.

Propriedades adicionais de uma sintaxe podem também ser registradas (por exemplo, poderia ser especificado que uma particular sintaxe de transferência e constituída sempre por um número inteiro de octetos).

A ISO definiu uma notação (ISO 8824 e ISO 8825, derivada da recomendação CCITT X.409) que permite difinir tipos de dados simples e complexos, bem como os valores que tais tipos podem assumir. Esta notação e denominada Notação para Sintaxe Abstrata Um (ASN.1 Abstract Sintax Notation One). Esta notação que não indica o valor dos dados, apenas sua forma. Além disso existem algoritmos, denominados Regras Básicas de Codificação (Basic Encoding Rules) que determinam o valor dos octetos representando tais valores e que serão passados para o nível de sessão (isto e denominado a sintaxe de transferência de dados).

Este trabalho pretende apresentar de forma tutorial e didática, os princípios da ASN.1. A sessão 2 deste trabalho, apresentara a definição da notação para representação da sintaxe dos dados e a sessão 3 as regras básicas de codificação.

---

2. A Notação para Sintaxe Abstrata Um

ASN.1 ou Abstract Syntax Notation One é uma notação que permite definir tipos de dados simples e complexos e especificar valores que estes tipos podem assumir.

Os valores que são transmitidos podem ser de diversos tipos. Existem os tipos simples e outros, mais complexos, que são formados de vários tipos simples combinados. Cada tipo recebe uma denominação que o distingue, de forma inequívoca de todos os demais tipos. Algumas das maneiras de definir novos tipos são:

1. uma sequência (ordenada) de tipos existentes
2. uma sequência não ordenada de tipos existentes
3. uma seleção de um dentre um conjunto de tipos

Estes são denominados tipos estruturados. Cada tipo recebe um rótulo ("tag"). Um mesmo rótulo pode ser atribuído a mais de um tipo cuja particular identificação será decidida pelo contexto em que o rótulo for usado.

Existem quatro classes de rótulos:

- **UNIVERSAL:** pode ser atribuído a um tipo simples ou a um mecanismo de construção, conforme especificado na tabela A.1.
- **APLICAÇÃO:** rótulos atribuídos a tipos por padrões específicos. Num particular padrão os rótulos da classe de APLICAÇ[[Atilde]]O somente podem ser atribuídos a um único valor.
- **PRIVADA:** rótulos usados numa empresa especifica.
- **ESPECIFICADO-POR-CONTEXTO:** interpretado de acordo com o contexto em que é usado

```
/-------------------------------------------------------------------\
|  TABELA A.1 Rotulo atribuídos na classe universal
|                                                                   |
|-------------------------------------------------------------------|
|  UNIVERSAL 1            | tipo booleano                           |
|  UNIVERSAL 2            | tipo inteiro                            |
```

```
|  UNIVERSAL 3            | tipo string de bits                          |
|  UNIVERSAL 4            | tipo string de octetos                       |
|  UNIVERSAL 5            | tipo nulo                                    |
|  UNIVERSAL 6            | tipo identificador de objeto                 |
|  UNIVERSAL 7            | tipo descritor de objeto                     |
|  UNIVERSAL 8            | tipo externo                                 |
|  UNIVERSAL 9-15         | reservados para adendos ao padrão
|  UNIVERSAL 16           | tipo SEQUENCE e SEQUENCE-OF                  |
|  UNIVERSAL 17           | tipo SET e SET-OF                            |
|  UNIVERSAL 18-22, 25-27 | tipos string de conjuntos de caracteres      |
|  UNIVERSAL 23-24        | tipo hora                                    |
|  UNIVERSAL 28-...       | reservados para adendos ao padrão
\-------------------------------------------------------------------/
```

O valor do rótulo é especificado indicando-se sua classe e o número dentro da classe (que deve ser inteiro não negativo), em notação decimal.

As regras de codificação sempre conduzem o rótulo do tipo, explícita ou implicitamente, bem como alguma representação do valor do tipo.

ASN.1 ou Abstract Syntax Notation One e uma notação que permite definir tipos complexos e especificar valores destes tipos. As regras de codificação constituem outro padrão que aplicadas ao valor de um certo tipo definido pela ASN.1 resultam na especificação completa dos valores daquele tipo durante a transferência. As regras de codificação sempre forçam a transmissão do rótulo de um tipo, implícita ou explicitamente, juntamente com a representação do seu valor.

Supondo que a definição de um registro de funcionário tenha sido definido e recebido a denominação de DADOS-PESSOAL. Os campos pertinentes a tal tipo de registro poderiam ser como os seguintes:

Nome: Joao P. Silva
Cargo: Diretor
Numero: 51
Data de admissão: 17 de setembro de 1971
Nome da esposa: Maria Silva
Numero de filhos: 2

Informações sobre filhos
Nome: Rafael Silva
Data de nascimento: 11 de novembro de 1957
Informações sobre filho
Nome: Suzana Silva
Data de nascimento: 17 de julho de 1959

A descrição formal deste registro, usando a notação ASN.1 seria:

```
Registro pessoal::- [APPLICATION 0] IMPLICIT SET
{ Nome,
  Cargo [0] ISO646 String,
  NumeroEmpregado,
  DataIngresso [1] Date,
  NomeEsposa   [2] Name,
  Filhos         [3] IMPLICIT SEQUENCE OF
                     InformaçãoFilho DEFAULT { }}

Informaçãofilho::- SET
{ Nome,
```

```
  DataNascimento [0] Date}

Nome ::- [APPLICATION 1] IMPLICIT SEQUENCE
 { Nome ISO646 String,
   Inicial ISO646 String,
   Sobrenome ISO646 String,

 NumeroEmpregado::- [APPLICATION 2] IMPLICIT INTEGER
 Date ::- [APPLICATION 3] IMPLICIT ISO 646 String -- AAAAMMDD

 O valor ou conteudo de um registro deste tipo seria:
 { nome "Joao", inicial "P", sobrenome "Silva"},
```

Os detalhes desta especificação serão apresentados na próxima sessão.

2.1 A notação usada em ASN.1

ASN.1 utiliza alguns tipos primitivos (elementos de dados) com os quais podem ser compostas estruturas mais complexas. Os tipos primitivos ou básicos são:

- BOOLEAN
- INTEGER
- BITSTRING
- OCTETSYTRING
- NULL

Estruturas complexas são definidas agregando-se tais tipos primitivos de algumas formas previstas na ASN.1. As principais formas de estruturação de tipos compostos ou "construídos" são:

- SEQUENCE-lista ordenada de tipos
- SEQUENCE OF-iteração ilimitada de um único tipo Mecanismos de
- SET-lista não ordenada de tipos estruturação
- SET OF-interação ilimitada de um único tipo (a ordem não e importante)
- CHOICE-um campo que consiste de uma valor dentre os tipos listados.

ASN.1 define uma notação para especificação de valores e para definição de tipos. A definição de um tipo consiste numa coleção de campos que, no mais baixo nível, consiste de um identificador, uma possível etiqueta (rótulo ou "tag"), uma referência e uma possível indicação de que aquele campo e opcional (pode ser omitido).

No menor nível os campos da definição são combinados usando mecanismos de estruturação que começa com o nome do tipo da estrutura e então, em geral, uma lista de campos separados por vírgulas e envolvidos em chaves "{}". É válido usar aninhamento de estruturas.

O identificador compreensível (letra minúscula) serve apenas para auxiliar a compreensão mas é ignorado pelas regras de codificação básicas.

O rótulo ("tag") permite que dois tipos aparentemente idênticos sejam separados tal que seu uso possa ser distinguido em construções tal como CHOICE ou SET.

Assim, um tipo será definido, segundo a ASN.1 usando-se uma das sequências de tipo válidas, que são:

```
Type::= BuiltinType|DefinedType

BuiltinType::= BooleanType|
               IntegerType|
               BitStringType|
               OctetStringType|
               NullType|
               SequenceType|
               SequenceOfType|
               SetTypeType|
               SetOfType|
               ChoiceType|
               TaggedType|
               AnyType|
               CharacterSetType|
               UsefulType|
```

O DefinedType especifica sequências externas usadas para referir definições de tipos e valores.

Os mais significativos tipos de primitivos serão definidos a seguir:

- O tipo **BOOLEAN** e usado para representar valores de variáveis lógicas (isto é que somente podem assumir dois estados). Quando escolher um nome representar uma variável boolena deve ser escolhido um que descreva o estado verdadeiro, por exemplo:

  Casado ::= **BOOLEAN**

-

  O tipo **INTEGER** e usado para modelar os valores designativos de ordem (cardinal) ou inteiros. Exemplo:

  SaldoDaConta ::= **INTEGER** -- em centavos; negativo significa debito

  Pode ser definido o valor mínimo e máximo, por exemplo:

  DiaDoMes ::= **INTEGER {primeiro(1),ultimo(31)}**

  Também pode-se usar inteiros como símbolos:

  DiaDaSemana ::= **INTEGER {Domingo(1), Segunda(2), Terça(3),**

  Quarta(4), Quinta(5), Sexta(6), Sábado(7)}

  EstadoMarital ::= INTEGER {solteiro(0),casado(1),viúvo(2)}

- Usa-se o tipo **BITSTRING** para modelar dados binários de formato e comprimento qualquer (inclusive não multiplo do 8). Poe-se usar este tipo para representar um mapa em que se indica pela posição dos bits ligados a ocorrência de algum fato, por exemplo:

  DiasComSolNoMes::= **BITSTRING{primeiro(1),ultimo(31)}** -- Um dia foi ensolarado se o bit correspondente a ele no mapa tem valor 1

-

O tipo **OCTET STRING** é usado para modelar dados de qualquer formato e comprimento múltiplo de 8 bits. Exemplo:

CorpoDoPacote ::= **OCTET STRING**

OU

Sobrenome ::= PrintableString

A segunda forma é preferível, quando possível.

- O tipo **NULL** é usado para indicar ausência de algum elemento numa sequência, tal como no exemplo seguinte:

```
IdentificaçãoPaciente ::= SEQUENCE
    { nome         OCTET STRING,
      numeroQuarto CHOICE
             {INTEGER,
              NULL --se o paciente já saiu do hospital-- }}
```

- O tipo **SEQUENCE OF** é usado para modelar uma coleção de variáveis cujo tipo é o mesmo, cujo número e grande e imprevisível e cuja ordem e significativa. Exemplo:

```
NomeNaçõesMembros ::= SEQUENCE OF ISO646String
  -- na ordem em que se integraram
```

- O tipo **SEQUENCE** é usado para modelar uma coleção de variáveis cujo tipo e o mesmo, cujo número e conhecido e modesto e cuja ordem e significante, desde que a composição do conjunto dificilmente varie de uma versão do protocolo para outra. Exemplo:

```
NomeDosFuncionarios ::= SEQUENCE
      {presidente     ISO646String,
       vicePresidente ISO646String,
       secretaria     ISU646String }
```

Também pode ser usado o tipo **SEQUENCE** quando o tipo dos integrantes difere, mas o número e conhecido e modesto e cuja ordem e significante, desde que a composição do conjunto dificilmente mude de uma versão do protocolo para outra. Exemplo:

```
Credenciais ::= SEQUENCE
  {nomeUsuario ISO646String,
   password    ISO646String,
   numeroConta INTEGER }
```

Se os elementos de uma sequência são em número fixo mas de vários tipos, um nome de referência deve ser atribuído a cada elemento cujo objetivo não seja completamente evidente de seu tipo. Exemplo:

```
Arquivo ::= SEQUENCE
  {  TipoConteudo,
     outros       AtributosArquivo,
     conteudo     ANY   }
```

- O tipo **SET** é usado para representar uma coleção de variáveis cujo número e conhecido e modesto e cuja ordem não e significante. Neste caso, cada variável deve ser rotulada neste contexto. Exemplo:

```
NomeUsuario ::= SET
 { nomePessoal     [0] IMPLICIT ISO646String,
   nomeOrganização [1] IMPLICIT ISO646String,
   nomePais        [2] IMPLICIT ISO646String }
```

A palavra **OPTIONAL** pode ser colocada após uma variável do conjunto cujo aparecimento e opcional. Se os membros de um tipo SET são em número fixo, também deve ser atribuído um nome a cada membro cujo fim não e evidente de seu tipo. Exemplo:

```
AtributosArquivo ::= SET
  {proprietario               [0] IMPLICIT NomeUsuario,
   tamanhoDoConteudoEmOctetos [1] IMPLICIT INTEGER,
                              [2] IMPLICIT ControleAcesso  }
```

- O tipo **SET OF** pode ser usado para representar uma coleção de variáveis cujos tipos são os mesmos e cuja ordem não e significante.Ex:

PalavrasChave ::= SET OF ISO646String --em qualquer ordem

- O tipo rotulado é usado para definir um tipo de dados de uso geral, independente de aplicação que deve ser distinguível de qualquer outro, por meio de sua representação. Exemplo:

ChaveCriptografia ::= [UNIVERSAL 24] IMPLICIT OCTET STRING – sete octetos

Também podem ser usados tipos rotulados para definir um tipo de dados de uso amplo e disperso, num particular contexto de apresentação e que deve ser distinguível (por meio de sua representação) de todos os outros tipos de dados usados naquele contexto de apresentação. Exemplo:

```
NomeArquivo ::= [APPLICATION 8] IMPLICIT SEQUENCE
 { nomeDiretorio ISO646String,
   nomeRelativoArquivoDiretorio ISO646String }
```

Os rótulos específicos de um contexto são usados para distinguir os membros de um conjunto. Os rótulos numéricos devem começar de zero se sua única meta e distinguir os tipos. Exemplo:

```
RegistroCliente ::= SET
 {nome           [0] IMPLICIT ISO646String,
  enderecoPostal [1] IMPLICIT ISO646String,
  numeroConta    [2] IMPLICIT INTEGER,
  debito         [3] IMPLICIT INTEGER --em centavos-- }
```

Quando um particular membro do conjunto recebeu um rótulo a nível de

aplicação não necessita receber rótulo especifico de contexto.Ex:

```
RegistroProduto ::= SET
 {       CodigoUniforme,
  descrição     [0] IMPLICIT ISO646String,
  numeroEstoque [1] IMPLICIT INTEGER }
CodigoUniforme ::= [APPLICATION 13] IMPLICIT INTEGER
```

Rotulação especifica de contexto também e usada para distinguir as alternativas de uma escolha (CHOICE). Rótulos numéricos começam em zero se seu único propósito e distinguir os diversos tipos.

Exemplo:

```
AtributoDeCliente ::= CHOICE
 {nome           [0] IMPLICIT ISO646String,
  enderecoPostal [1] IMPLICIT ISO646String,
  numeroDaConta  [2] IMPLICIT INTEGER,
  debito         [3] IMPLICIT INTEGER -- em centavos--}
```

Os rótulos de uso privativo são usados para rotular tipos de dados que são de uso apenas no escopo de uma particular organização ou pais e que devem ser distinguiveis de todos os demais tipos de dadso usados por aquela organização ou pais. Exemplo:

NumeroCartaoAcme ::= [PRIVATE 2] IMPLICIT INTEGER

- O tipo **CHOICE** (seleção) e usado para representar uma variável que e selecionada dentre uma coleção de variáveis cujo número e conhecido e modesto. Cada variável do conjunto deve ser identificada por um rótulo especifico do contexto. Exemplo:

```
IdentificadorArquivo ::= CHOICE
  { nomeRelativo  [0] IMPLICIT ISO646String, -- nome do arquivo
    nomeAbsoluto  [1] IMPLICIT ISO646String, -- nome do arquivo e
do
                                                diretorio que o
contem
    numeroSerial  [2] IMPLICIT INTEGER
     --identificador atribuido pelo sistema }
```

O tipo seleção e usado para representar uma variável cujo tipo e um dentre alguma particular alternativa de um **CHOICE** previamente definido.

Exemplo: a definição seguinte é possível

```
AtributosCorrentes ::= SEQUENCE
  { data-ultimo-uso   AtributoArquivo,
    nome-arquivo      AtributoArquivo }
```

se existir a seguinte definição:

```
AtributoArquivo ::= CHOICE
  { data-ultimo-uso   INTEGER,
    nome-arquivo  ISO646String  }
```

- O tipo **ANY** é usado para modelar uma variável cujo tipo não e especificado ou e especificado em outro ponto usando ASN.1.Exemplo:

  ConteudoMensagem ::= ANY

  – um elemento de dados cujo tipo e especificado na norma XXX

- O tipo **EXTERNAL** é usado para referenciar uma variável cujo tipo não é especificado ou que é especificado em outro local usando ASN.1 .

---

3. Basic Encoding Rules

As "Basic Encoding Rules" provêem um algoritmo que especifica como um valor de qualquer estrutura (tipo) definida usando ASN.1 deve ser codificada para transmissão. O uso do algoritmo no sentido contrário permite que qualquer receptor que tinha conhecimento da definição do tipo ASN.1, possa decodificar os bits que chegam em um valor daquele tipo.

A codificação de um tipo de dados ASN.1 (usando as Basic Encoding Rules) permite que um receptor, sem conhecimento da definição de tipo, reconheça o inicio e o fim das construções (SEQUENCE, SET, etc) e os octetos representando os tipos básicos de dados (BOOLEAN, INTEGER). No uso mais simples da notação, e também possível determinar, a partir da codificação, as construções efetivamente usadas e os tipos de dados básicos.)

Quando o rótulo ASN.1 ("[]") e empregado, a codificação porta tanto o valor do rótulo como o valor identificando a construção do tipo de dado básico que foi rotulado.

Também e possível, na notação ASN.1, usar o que e chamado rotulação IMPLÍCITA a qual indica que, nas situações em que e usada, a rotulação não e necessária durante a transferência dos dados, pois pode ser percebida

pelo contexto e assim, e minimizada a quantidade de octetos transferida. A alternativa IMPLICIT não deve ser usada com os tipos CHOICE, ANY ou EXTERNAL. Exemplo:

```
ExemploTipo ::= CHOICE
    { primeiraEscolha [0] IMPLICIT INTEGER,
      segunda-escolha [1] IMPLICIT BOOLEAN }
```

Neste caso, o valor do rótulo (zero ou um) estará presente na codificação gerada pela regras de codificação mas os códigos para identificar que os tipos são INTEGER ou BOOLEAN serão suprimidos. Evidentemente, o receptor devera conhecer a definição dos tipos para saber, pelo rótulo, qual o tipo recebido.

Cada valor de dados a ser codificado terá uma representação consistindo de:

- T tipo
- C comprimento - quando indefinido, termina com 2 octetos nulos
- V valor - pode ser em si uma série de componentes TCV

As regras de codificação sempre conduzem o rótulo do tipo, explícita ou implicitamente, bem como alguma representação do valor do tipo. Estas regras constituem outro padrão que aplicadas ao valor de umcerto tipo definido pela ASN.1 resultam na especificação completa dos valores daquele tipo durante a transferência.

Exemplo de uso da ASN.1 para descrição de uma estrutura de dados concernente a um registro de pessoa. O registro contem os seguintes campos:

Nome: John T. Smith
Cargo: Director
Numero: 51
Data de admissao: 17 de setembro 1971
Nome da esposa: Mary Smith
Numero de filhos: 2

Informações sobre filhos

Nome: Ralph T. Smith
Data de nascimento: 11 de novembro 1957

Informações sobre filho
Nome: Susan B. Jones
Data de nascimento: 17 de julho de 1959

A descrição formal deste registro, usando a notação padronizada seria:

```
Registro pessoal::- [APPLICATION 0] IMPLICIT SET
{ Nome,
  cargo [0] ISO646 String,
  numero NumeroEmpregado,
  dataDeIngresso [1] Date,
  nomeDaEsposa   [2] Name,
  filhos         [3] IMPLICIT SEQUENCE OF
                     InformaçãoFilho DEFAULT { }}
InformaçãoFilho::- SET
{ Nome,
  dataNascimento [0] Date}
Nome ::- [APPLICATION 1] IMPLICIT SEQUENCE
 { nome ISO646 String,
   inicial ISO646 String,
   sobrenome ISO646 String,

 NumeroEmpregado::- [APPLICATION 2] IMPLICIT INTEGER
 Date ::- [APPLICATION 3] IMPLICIT ISO 646 String -- AAAAMMDD
```

O valor ou conteúdo de um registro deste tipo seria:

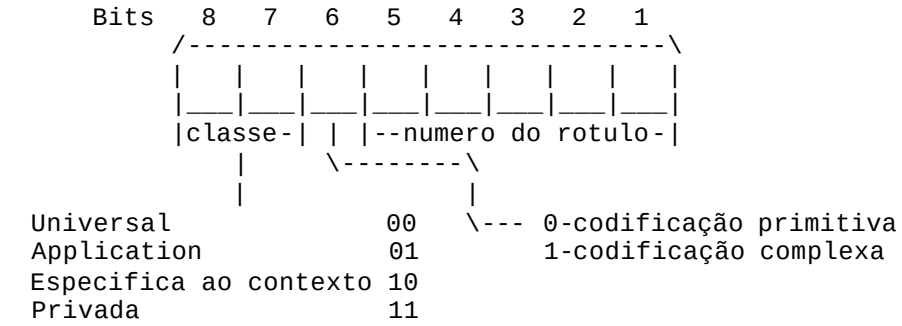
```
{ nome "John", inicial "T", sobrenome "Smith"},
  titulo "Director",
  numero 51,
  dataDeIngresso "19710917",
  nomeDaEsposa {nome "Mary", inicial "T", sobrenome "Smith"},
  filhos
   {{{nome "Ralph",inicial "T",sobrenome "Smith} ,
      dataDeNascimento "19571111"  },
    {{nome "Susan", inicial "B",sobrenome "Jones"},
     dataDeNascimento =19590717"    }}
```

3.1 Regras gerais para codificação

A codificação de um valor de todos os tipos, exceto os externos deve- ra consistir de 4 componentes que deverão aparecer na seguinte ordem:

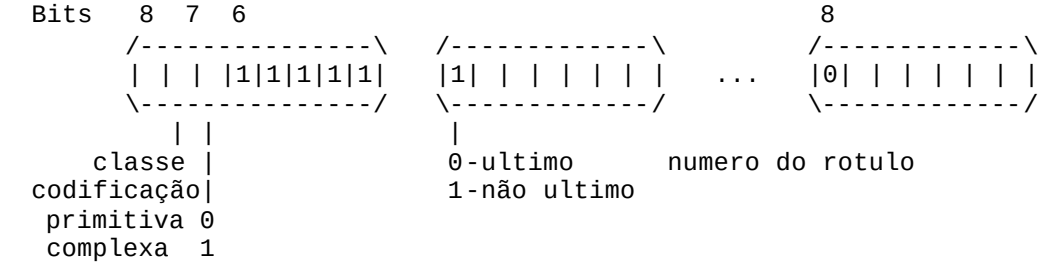
- octetos de identificação
- octetos de comprimento
- octetos de conteúdo
- octeto de fim-de-conteudo

3.1.1. Os octetos de identificação indicam a classe e número:



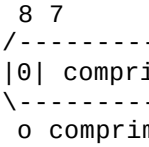
octetos subsequentes

Quando o número do rótulo não puder ser expresso com apenas 5 bits,
usa-se a forma seguinte:

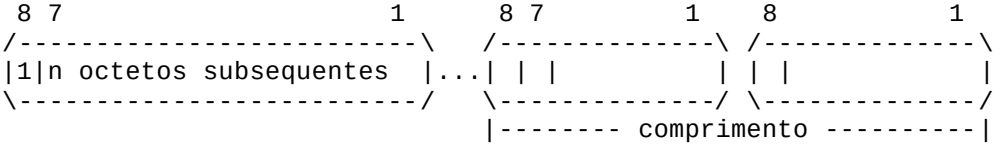


3.1.2. Octetos de comprimento Poderão ser indicados de duas maneiras:

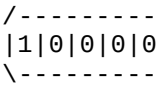
1. um ou mais octetos indicando o comprimento do conteúdo
   Neste caso, se o comprimento for menor do que 127 o formato será:



se o comprimento for maior do que 127



2. indicando apenas o final do campo, da seguinte maneira:



Isto indica que o final do conteudo sera sinalizado por um octeto de fim-de-conteudo.

3.1.3. Octetos de conteúdo

Zero ou mais octetos codificando os valores sendo transmitidos.

3.1.4. Octetos de fim-de-conteudo

Dois octetos zero. Este campo somente estará presente quando o comprimento do conteúdo não for conhecido ao ser iniciada sua transmissão; neste caso, no octeto de comprimento será sinalizada esta forma de delimitação de conteúdo (opção b acima)

3.2. Regras de codificação para valores

3.2.1. Booleano (primitivo)

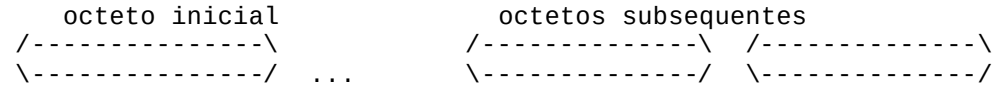
FALSE 0 octeto zero
TRUE qualquer valor não nulo

3.2.2. Integer (primitivo)

Consiste de um ou mais octetos concatenados cujo conteúdo devera ser o complemento de dois do número binário inteiro igual ao valor inteiro. Se mais de um octeto e usado, os primeiros nove bits não deverão ser todos nulos ou todos "1".

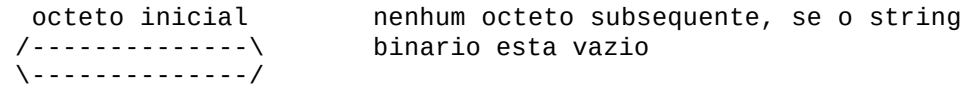
3.2.3.String binário { primário ou composto }

primário



número de bits não usados
nos octetos subsequentes

ou



Composto

Consiste de zero ou mais codificações completas de valores de dados,

cada qual sendo a codificação de um string de bits.

3.2.4.String de octetos { primitivo ou composto }

primitivo

zero ou mais octetos

composto

zero ou mais codificações completas de zero ou mais valores dados

3.2.5.Valor nulo primitivo

nenhum octeto e o octeto de comprimento será zero

3.2.6.Valores sequênciados (composto)

Contem codificação completa de um valor de dados para cada um dos tipos listados na definição ASN.1, na ordem em que aparecem na definição, paramentos que o tipo foi referenciado com a palavra-chave "OPTIONAL" ou a palavra-chave "DEFAULT".

Observação: não e possível transferir codificação de valores de dados posteriores na definição ASN.1 sem que os anteriores sejam conhecidos.

3.2.7.Sequência de valores (composto)

Consiste zero, um ou mais codificações completas de valores com os tipos listados na definição ASN.1. A ordem das codificações dos valores deve ser a mesma que a ordem dos valores a serem codificados ou produzidos por decodificação.

3.2.8.Conjunto de valores diferentes ou iguais (SET ou SET-OFF) composto

Consiste na codificação completa de um valor de dados de cada um dos tipos listados na definição ASN.1, na ordem escolhida pelo emissor.

3.2.9.Escolha (CHOICE)

A codificação de um valor escolhido deve ser a mesma de um valor do tipo escolhido.

3.2.10.Seleção (SELECTION)

A codificação de um valor selecionado devera ser a mesma que a codificação de um valor do tipo selecionado. A codificação pode ser primitiva ou composta, dependendo do tipo selecionado.

3.2.11.Valor rotulado

A codificação de um valor rotulado deve ser derivada da codificação completa do correspondente valor do tipo aparecendo na notação dos tipos rotulados (denominada a base da codificação).

3.3 Codificação do registro de pessoal exemplificado segundo as Basic Encoding Rules

---

```
Pessoal
Registro Comprimento  Conteúdo
60       8185
         Nome  Comprimento  Conteúdo
         61    10
               IA5String    Comprimento   Conteúdo
               16           4             "John"
               IA5String    Comprimento   Conteudo
```

```
            16          1           "T"
            IA5String   Comprimento  Conteudo
            16          5           "Smith"

      Cargo Comprimento  Conteudo
      A0    0A
            IA5String   Comprimento  Conteudo
            16          08          "Director"

      Empregado Comprimento  Conteudo
      Numero  Comprimento   Conteudo
      42      01            33


      Data de Ingresso  Comprimento  Conteudo
      A1                0A

              Data  Comprimento   Conteudo
                43        08        "19710917"

      Nome da esposa    Comprimento  Conteudo
      A2                12
                Nome    Comprimento   Conteudo
                61      10
                        IA5String  Comprimento Conteudo
                        16         04          "Mary"
                        IA5String  Comprimento Conteudo
                        16         01          "T"
                        IA5String  Comprimento Conteudo
                        16         05          "Smith"

[3]   Comprimento  Conteudo
A3    42
      SET     Comprimento  Conteudo
      31      1F

              Nome  Comprimento  Conteudo
              61    11

                    IA5String Comprimento Conteudo
                    16        05          "Ralph"
                    IA5String Comprimento Conteudo
                    16        01          "T"
                    IA5String Comprimento Conteudo
                    16        05          "Smith"
               Data de Nascimento  Comprimento Conteudo
               A0                  0A
                     Data  Comprimento  Conteudo
                     43    08           "19571111"

      SET     Comprimento  Conteudo
      31      1F
              Nome  Comprimento  Conteudo
              61    11

                    IA5String Comprimento Conteudo
                    16        05          "Susan"
```

```
      IA5String Comprimento Conteudo
      16        01          "B"
      IA5String Comprimento Conteudo
      16        05          "Jones"
Data de Nascimento  Comprimento Conteudo
A0                  0A
       Data  Comprimento  Conteudo
       43    08           "19590717"
```

Questões sobre ASN.1

As seguintes questões foram propostas aos alunos da disciplina Redes de Computadores do curso de Pós Graduação da UFRGS:

1. Qual a finalidade de usar rótulos na definição das estruturas ASN.1?
2. Para que serve e como é usada a opção IMPLICIT?
3. Elabore uma definição de uma estrutura para itens numa biblioteca. Considere que nem todos os itens conterão todas as informações. Considere que um item pode estar em várias condições: na prateleira, emprestado, em reserva local, em conserto, reservado, etc...
4. Construa uma instância de sua definição, indicando os valores que seriam colocados em cada tipo primitivo e composto da sua estrutura.
5. Transcreva esta instância usando as Regras de Codificação Básicas produzindo a sequência de bits que seriam transmitidos. Mostre-os de forma compreensiva (agrupando os bits de forma a facilitar sua interpretação.

Respostas

- Grupo 1 *(Afonso J. R. Cardoso, Luciano P. Barreto, Maria de Fátima W. P. Lima, Mauro L. B. Nogueira)*
- Grupo 2 *(Cláudia M. Sbardelloto, Cristiano A. Costa, João A. B. Filho, Luís F. F. Garcia, Márcia Gusmão)*
- Grupo 3 *(Herbert Luna, Joice Lee Otsuka, Jorge Juan Zavaleta Gavidia, Marcelo Moreto)*
- Grupo 4 *(Roberto Bello de Oliveira)*

Classe Universal

Classe universal	
UNIVERSAL 1	Boolean type
UNIVERSAL 2	Integer type
UNIVERSAL 3	Bitstring type
UNIVERSAL 4	Octetstring type
UNIVERSAL 5	Null type
UNIVERSAL 6	Object identifier type
UNIVERSAL 7	Object descriptor type
UNIVERSAL 8	External type
UNIVERSAL 9	Real type
UNIVERSAL 10	Enumerated type
UNIVERSAL 12 à 15	Reserved forfuture versions of this Recommendation
UNIVERSAL 16	Sequence and Sequence-of types
UNIVERSAL 17	Set and Set-of types
UNIVERSAL 18 à 22, 25 à 27	Character string types
UNIVERSAL 23, 24	Time types
UNIVERSAL 28- . . .	Reserved for future versions of this Recommendation

Seqüências de caracteres reservados					
BEGIN	BOOLEAN	END	INTEGER	DEFINITIONS	BIT
EXPLICIT	STRING	ENUMERATED	OCTET	EXPORTS	NULL
IMPORTS	SEQUENCE	REAL	OF	INCLUDES	SET
MIN	IMPLICIT	MAX	CHOICE	SIZE	ANY
FROM	EXTERNAL	WITH	OBJECT	COMPONENT	IDENTIFIER
PRESENT	OPTIONAL	ABSENT	DEFAULT	DEFINED	COMPONENTS
BY	UNIVERSALPLUS-INFINITY	APPLICATION	MINUS-INFINITY	PRIVATE	TAGS
TRUE	FALSE				