



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

iconet.com.br/banon/2009/09.09.22.01

**IDENTIFICADOR COM BASE NA INTERNET:
PROJETO DE NORMA ABNT**

Gerald Jean Francis Banon

URL do documento original:
<<http://urlib.net/LK47B6W/362SFKH>>

INPE
São José dos Campos

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
Fax: (012) 3208-6919
E-mail: pubtc@sid.inpe.br

**CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO
INTELECTUAL DO INPE (RE/DIR-204):**

Presidente:

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Membros:

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr^a Regina Célia dos Santos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr. Horácio Hideki Yanasse - Centro de Tecnologias Especiais (CTE)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Deicy Farabello - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Vivéca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

iconet.com.br/banon/2009/09.09.22.01

**IDENTIFICADOR COM BASE NA INTERNET:
PROJETO DE NORMA ABNT**

Gerald Jean Francis Banon

URL do documento original:
<<http://urlib.net/LK47B6W/362SFKH>>

INPE
São José dos Campos

RESUMO

Neste projeto de norma, são apresenta as várias formas em que um identificador global pode vir a ser utilizado para identificar e prover acesso consistente e perene à diversos tipos de itens de informação (documentos, mapas, imagens, etc.) armazenados em acervos, sejam eles repositórios ou arquivos digitais. A implantação e a utilização deste identificador global requerem, de uma forma indireta, a infraestrutura já existente da Internet, portanto, sem custo adicional, neste aspecto. O identificador global pode ser utilizado em associação com o processo de armazenamento de informação em acervos. O que torna simples a criação de cópias em acervos distintos e, também, a própria migração de itens de informação entre os mesmos. As diversas aplicações de um identificador global desta natureza são de particular interesse em sistemas de dados espaciais e de informação.

INTERNET BASED IDENTIFIER - DRAFT STANDARD

ABSTRACT

In this draft standard, are shown the various ways in which a global identifier might be used to identify and provide consistent and enduring access to various kinds of information items (documents, maps, images, etc.) stored in collections, are they digital repositories or archives. The deployment and use of this global identifier require, in an indirect manner, the existing infrastructure of the Internet, therefore, without no additional cost, in this regard. The global identifier can be used in combination with the process of information storage in collections. What makes it simple to create copies in different collections and also migrate information items between them. The various applications of a global identifier of this nature are of particular interest in space data and information systems.

SUMÁRIO

	<u>Pág.</u>
1 Introdução	1
1.1 Objetivo	1
1.2 Justificativa	1
2 Terminologia	2
3 Descrição do sistema de identificação em dois níveis	3
4 Regras de construção do rótulo como nome de repositório uniforme	10
5 Regras de construção do rótulo com base no IP	19
REFERÊNCIAS BIBLIOGRÁFICAS	28
APÊNDICE A - DEFINIÇÕES E PROPRIEDADES	29

LISTA DE FIGURAS

	<u>Pág.</u>
3.1 Sistema de identificação em dois níveis	6

LISTA DE TABELAS

	<u>Pág.</u>
4.1 Regras definindo a formação de um nome de repositório uniforme de um item	12
4.2 Exemplificação do funcionamento de um distribuidor temporal	14
5.1 Regras definindo a formação de um IBIp	20
5.2 Exemplos de conversão utilizando a rotina CONVERTERPARADECIMAL	24

5.3	Exemplos de conversão utilizando a rotina CONVERTERDEDECIMAL . . .	24
5.4	Tabela de conversão de decimal para IPv4	26
5.5	Tabela de conversão de decimal para IPv6	26
5.6	Tabela de conversão de decimal para IBIp	27

1 Introdução

1.1 Objetivo

Este projeto de norma descreve um sistema de identificação com base na Internet que associa a cada item (de informação) a ser identificado, um rótulo utilizado como identificador deste item. As regras de construção de duas formas de apresentação do rótulo são estabelecidas.

Uma vez os itens de informação identificados, estes podem ser recuperados por meio de um sistema de resolução de identificação. No entanto, a descrição deste segundo sistema não faz parte deste projeto de norma.

ser

1.2 Justificativa

Os hipervínculos (*hyperlinks*) ou simplesmente vínculos ou ponteiros, elementos essenciais na navegação entre itens de informação (documentos, mapas, imagens, etc.) disponíveis na Internet devem ter seu funcionamento preservado por longo prazo.

A solução para tornar os ponteiros persistentes passa pelo uso de um sistema de identificação global.

O sistema de endereçamento físico de um item de informação na Web por meio de uma URL (Uniform Resource Locator), não é um sistema de identificação, pois, com o tempo, um determinado item de informação pode mudar de localização, fazendo com que a associação (item de informação) → URL não fique permanente.

"item de informação" (entre aspas)

Uma vez escolhido um sistema de identificação e por meio dele atribuído rótulos à itens de informação, o problema da construção de ponteiros persistentes pode ser solucionado por meio do uso de um sistema de resolução (ou resolvidor) de

identificação, cujo papel consiste em redicionar cada URL, agora contendo apenas o identificador de um item de informação, para a URL contendo o seu endereço físico.

O sistema de identificação descrito neste projeto de norma apresenta-se como uma alternativa simples, quando comparada a outras soluções como aquelas usadas, por exemplo, no PURL (WEIBEL *et al.*, 2010) ou no Handle System[®] (CNRI, 2010). O sistema apresentado está sendo extensivamente utilizado, desde 1995, na plataforma *URLib*.

comparado aoao

A principal diferença (com o) PURL ou (o) Handle System[®] é que, enquanto estes têm um sistema próprio de cadastro de seus usuários, o sistema proposto reaproveita indiretamente o sistema já existente de cadastro dos atores da Internet detenedores de nomes de domínio (*fully qualified domain names*), portanto, sem custo adicional, neste aspecto.

2 Terminologia

Distribuidor espacial: função entre um conjunto de **itens** e um conjunto de **sub-sistema de identificação**, distribuindo cada **item** à um determinado **subsistema de identificação** tornando este **subsistema de identificação** responsável pela identificação deste **item**.

Distribuidor temporal: função entre um conjunto de **itens** e um conjunto de par de datas, expressas em segundo, distribuindo cada **item** num espaço temporal (ver detalhes na Definição 2 na Seção 4).

Gerador de rótulo: função injetora utilizado por um **sistema de identificação** para gerar o **identificador de um item**.

IBI de um item: o **rótulo** atribuído à um **item** por um **sistema de identificação em dois níveis** utilizando o endereçamento por nome de domínio ou IP.

IBIp de um item: o **rótulo** atribuído à um **item** por um **sistema de identificação em dois níveis** utilizando o endereçamento por IP.

Identificador de um item: o **rótulo** atribuído à um **item** por um **sistema de identificação**.

Instalador: função entre um conjunto de **subsistemas de identificação** e um conjunto de pares de cadeia de caracteres informando o nome ou IP do computador, e a porta de acesso, onde foi instalado o **gerador de rótulo** utilizado por um determinado **subsistema de identificação**.

Item: qualquer objeto a ser identificado.

Item de informação: qualquer **item** consistindo exclusivamente em dados digitais, isto é quaisquer dados digitais a serem identificados. Por exemplo: documentos, mapas, imagens, etc. no formato digital.

Nome do repositório uniforme de um item: Identificador de um item podendo ser utilizado para armazená-lo digitalmente em um sistema de arquivos, caso este seja um **Item de informação**.

Rótulo: qualquer cadeia finita de caracteres escolhidos dentro de um alfabeto finito, utilizada como **identificador de um item**.

Sistema de identificação: qualquer função injetora entre um conjunto de **itens** e um conjunto de **rótulos**, associando a cada **item** o **identificador deste item** (ver detalhes na Seção 3).

Sistema de identificação em dois níveis: qualquer **sistema de identificação** associando um **item** à um **rótulo** obtido a partir de um par de **rótulos**, o primeiro identificando o **subsistema de identificação** responsável pela identificação do **item**, e o segundo sendo o **rótulo** atribuído ao **item** por este **subsistema de identificação** (ver detalhes na Definição 1 na Seção 3).

Subsistema de identificação: qualquer **sistema de identificação** restrito a um subconjunto de **itens**.

3 Descrição do sistema de identificação em dois níveis

Neste projeto de norma, os **itens** (objetos a serem identificados) são considerados formando conjuntos. Por exemplo, um conjunto de pastas.

Por sua vez, os **rótulos** utilizados para identificar os **itens**, são considerados formando conjuntos finitos ou enumeráveis. Por exemplo, o conjunto das cadeias de no

máximo 255 caracteres alfanuméricos, ou ainda o conjunto dos inteiros representando datas expressas em segundo.

Por ser uma função injetora, um **sistema de identificação** associa, de forma permanente, cada **item** à um único **rótulo**, de maneira que, **itens** distintos sejam associados à **rótulos** distintos.

Pela restrição dos conjuntos de **rótulos** serem finitos (resp. enumeráveis), e pela propriedade de **sistema de identificação** ser injetor, os conjuntos dos **itens** devem ser necessariamente finitos (resp. enumeráveis).

O **sistema de identificação**, objeto deste projeto de norma, consiste em quatro principais componentes: um conjunto de **subsistemas de identificação**, um **distribuidor espacial** de **itens**, um **sistema de identificação** dos **subsistemas de identificação** e um **gerador de rótulo**. Juntos eles formam o **sistema de identificação em dois níveis** descrito detalhadamente por meio da definição a seguir.

Definição 1 (sistema de identificação em dois níveis) – Os conjuntos e funções introduzidos a seguir são os componentes de um **sistema de identificação em dois níveis**.

Seja I o conjunto dos **itens** a serem identificados.

Seja S o conjunto dos **subsistemas de identificação**.

Seja R_1 um conjunto finito de **rótulos**.

Seja R_2 um conjunto enumerável de **rótulos**.

Seja R um conjunto enumerável de **rótulos**.

Seja $f : S \rightarrow R_1$ o **sistema de identificação** dos **subsistemas de identificação**

Seja $g : I \rightarrow S$ o **distribuidor espacial** definindo qual é o **subsistema de identificação** responsável pela identificação de cada **item**. Assim, para todos $s \in S$, o subconjunto dos **itens** que estão sob a responsabilidade do **subsistema de identificação** s é $g^*(s)$, a fibra de s através g (ver definição de fibra na Apêndice A).

Seja, para todos $i \in I$, $g(i) : g^*(g(i)) \rightarrow R_2$, o **subsistema de identificação** responsável pela identificação do **item** i .

Seja $h : I \rightarrow R_1 \times R_2$ a função definida por:

$$h(i) \triangleq (f(g(i)), g(i)(i)), \quad \text{para todos } i \in I.$$

Seja $c : R_1 \times R_2 \rightarrow R$ o **gerador de rótulos** concatenando de forma reversível (i.e., c é injetor) os rotulos provenientes de R_1 e R_2 .

Um **sistema de identificação em dois níveis** é a função $s : I \rightarrow R$ definida por: $s \triangleq c \circ h$. □

Na Definição 1, a definição de $g(i)$ como função de $g^*(g(i))$ em R_2 se sustenta desde que $|g(I)| = |g^*(g(I))|$ como consequência de g^* ser injetora (Proposição 3).

Por definição, o **sistema de identificação** f , os **subsistemas de identificação** em $g(I)$ e o **gerador de rótulo** c são funções injetoras. Essa propriedade se estende ao **sistema de identificação em dois níveis** resultante, como enunciado na proposição seguinte.

Proposição 1 (propriedade de um sistema de identificação em dois níveis) – A função $s : I \rightarrow R$ da Definição 1 é injetora. □

Prova. Para todos i e $j \in I$, tem-se:

$$\begin{aligned} s(i) = s(j) &\Leftrightarrow c(h(i)) = c(h(j)) && \text{(Def. 1 e def. de composição)} \\ &\Rightarrow h(i) = h(j) && \text{(} c \text{ é injetora)} \\ &\Rightarrow (f(g(i)), g(i)(i)) = (f(g(j)), g(j)(j)) && \text{(Def. 1)} \\ &\Leftrightarrow f(g(i)) = f(g(j)) \text{ e } g(i)(i) = g(j)(j) && \text{(igualdade de pares)} \\ &\Rightarrow g(i) = g(j) \text{ e } g(i)(i) = g(j)(j) && \text{(} f \text{ é injetora)} \\ &\Rightarrow g(i)(i) = g(i)(j) && \text{(substituição)} \\ &\Rightarrow i = j. && \text{(} g(i) \text{ é injetora)} \end{aligned}$$

Isto é, h é injetora. □

Dado um **item** i em I , um **sistema de identificação em dois níveis** atribui à

i o **rótulo** gerado por c a partir do par $(f(g(i)), g(i)(i))$ constituído, de um lado, pelo **rótulo** $f(g(i))$ atribuído pelo **sistema de identificação** f ao **subsistema de identificação** $g(i)$ (reponsável pela identificação de i dentre do escopo $g^*(g(i))$), e de outro lado, pelo **rótulo** $g(i)(i)$ atribuído pelo **subsistema de identificação** $g(i)$ ao **item** i .

Os componentes do **sistema de identificação em dois níveis** aparecem na Figura 3.1.

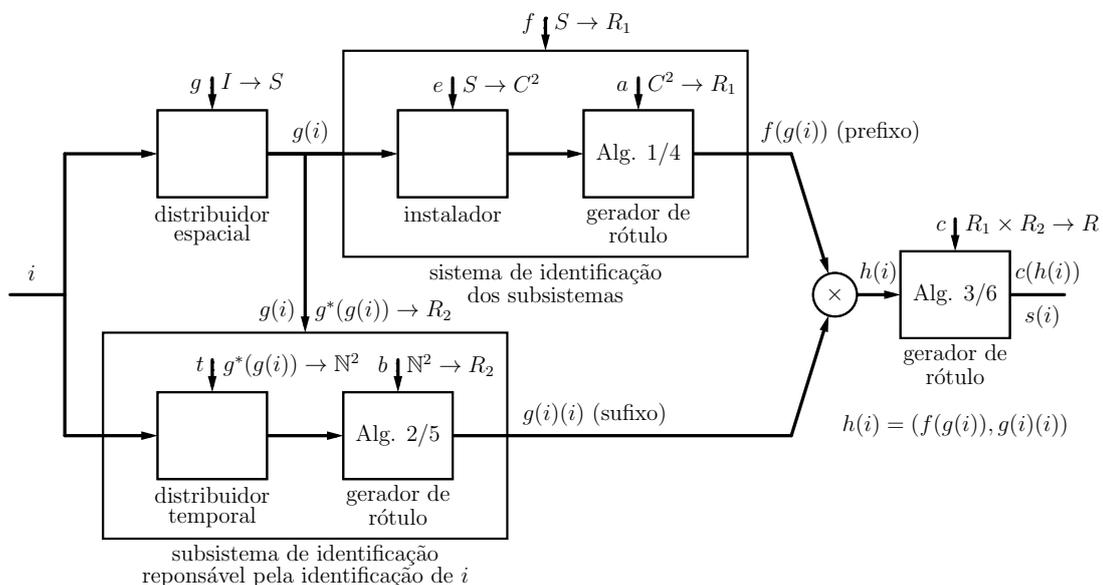


Figura 3.1 - Sistema de identificação em dois níveis

Na Figura 3.1, cada bloco representa uma função, por exemplo g (indicada por uma setinha acima do bloco), com sua entrada (ao lado esquerdo), por exemplo i , e sua saída (ao lado direito), por exemplo $g(i)$.

Enquanto $g(i)(i)$, o **identificador do item** i , fornecido pelo **subsistema de identificação** $g(i)$, tem validade apenas dentre do escopo $g^*(g(i))$ deste subsistema, $s(i)$, o **identificador deste item**, fornecido pelo **subsistema de identificação em dois níveis** s , tem validade dentre do escopo global I .

dentro

dentro

O Handle System[®], por exemplo, possui estes dois níveis de identificação. Os atores no Handle System[®] são o CNRI (*Corporation for National Research Initiatives*) e

os RSP (*Resolution Service Providers*). Na terminologia deste projeto de norma, os RSP são os **subsistemas de identificação** e o CNRI é o **sistema de identificação** dos RSP.

No primeiro nível, o CNRI (f) registra os RSP e fornece, a cada um, um rótulo denominado “prefixo”. No segundo nível, um RSP ($g(i)$) fornece, a cada **item** (i) sob sua responsabilidade, um rótulo denominado “sufixo”.

Usando a terminologia do Handle System[®], o **identificador de um item** é a concatenação de um “prefixo” identificando um **subsistema de identificação** e de um “sufixo” identificando um **item** no escopo deste subsistema.

A solução apresentada neste projeto de norma, consiste em reaproveitar a infraestrutura já existente da Internet para identificar os **subsistemas de identificação**, portanto, sem custo adicional, neste aspecto.

Mais precisamente, o **identificador de um item** será obtido como resultado de uma solicitação a um servidor associado a um determinado **subsistemas de identificação**, hospedado exclusivamente em computador possuindo nome de domínio (*fully qualified domain names*). Assim, cada um destes **subsistemas de identificação** é identificado globalmente pelo seu nome de domínio ou seu IP (*Internet Protocol*) nesta rede, pertimindo assim **de** construir, de forma simples, o prefixo.

Afim de desburocratizar ainda mais o **sistema de identificação** como um todo, o sufixo, fornecido por um **subsistema de identificação**, segue (diferentemente do Handle System[®]) uma regra comum a todos os subsistemas, e consiste na data e hora da associação do **item** ao sufixo. Esta escolha facilita o reuso de um **identificador de subsistema de identificação** quando este passa sob o controle de um novo ator.

Na solução objeto deste projeto de norma, dois tipos de prefixo herdado da Internet são considerados.

O primeiro tipo consiste em construir o **identificador de um subsistema de identificação**, isto é, o prefixo, com base no nome de domínio do computador¹ que hospeda o servidor associado à este subsistema, assim como a porta de acesso a este

¹O nome de domínio pode se referir eventualmente ao nome de domínio de um computador virtual (*virtual host*).

servidor.

No segundo tipo, o prefixo é obtido com base no IP do computador, no lugar do nome de domínio.

Os exemplos reais a seguir antecipam alguns dos detalhes sobre a formação dos identificadores que serão dados nos dois próximos capítulos.

Exemplo 1 (identificador com base no nome de domínio) – A associação de um **item** com um sufixo, ocorrida em 16 de fevereiro de 2009 às 17 horas 46 minutos, resultou no sufixo:

2009/02.16.17.46

O servidor emitindo este sufixo era hospedado em um computador com nome de domínio `mtc-m18.sid.inpe.br`, e acessível a partir da porta 80, levando ao uso do prefixo:

`sid.inpe.br/mtc-m18@80`

Desta forma, o **identificador para o item** passou a ser:

`sid.inpe.br/mtc-m18@80/2009/02.16.17.46`

Usando, por exemplo, o resolvidor de identificação `urlib.net`, o ponteiro (URL) persistente para este **item** ficou:

<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/02.16.17.46>

Adicionalmente, o ponteiro (URL) persistente para os metadados deste **item** ficou:

<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/02.16.17.46??>

□

Observa-se, que mesmo que o nome de domínio `mtc-m18.sid.inpe.br` do Exemplo 1 passe a ser abandonado ou muda de dono, isto não inviabiliza o **identificador criado para este item**. O importante, apenas, é que estes dados eram pertinente no contexto da Internet na data e hora da associação entre o **item** e seu **rótulo**. Esta observação vale também para o segundo exemplo a seguir apontando para o

mesmo **item de informação**.

Exemplo 2 (identificador opaco com base no IP) – A associação de um **item** com um sufixo, ocorrida em 16 de fevereiro de 2009 às 17 horas 46 minutos, resultou no sufixo opaco:

34PGRBS

O servidor emitindo este sufixo era hospedado em um computador com IP 150.163.34.243, e acessível a partir da porta 800, levando ao uso do prefixo opaco:

8JMKD3MGP8W

Desta forma, o **identificador para o item** passou a ser:

8JMKD3MGP8W/34PGRBS

Usando, por exemplo, o resolvedor de identificação `urlib.net`, o ponteiro (URL) persistente para este **item** ficou:

<http://urlib.net/8JMKD3MGP8W/34PGRBS>

Adicionalmente, o ponteiro (URL) persistente para os metadados deste **item** ficou:

<http://urlib.net/8JMKD3MGP8W/34PGRBS??>

□

Observa-se que a granularidade do prefixo é extremamente fina já que os **subsistemas de identificação** são atrelados à números de porta de computador com **de** nomes de domínio (*fully qualified domain names*). São atualmente mais de 200 milhões de nomes de domínio (VERISIGN, 2010).

Quanto a granularidade do sufixo ela pode ser aumentada, acrescentando por exemplo os segundos.

Os dois tipos de **sistemas de identificação em dois níveis** são apresentados a seguir em detalhe. No primeiro, o **identificador de um item**, exibindo o nome de domínio, e é chamado de **nome de repositório uniforme do item**. No segundo tipo, o **identificador de um item**, construído com base no IP, é chamado de **IBIp**

do item.

4 Regras de construção do rótulo como nome de repositório uniforme

No sistema de identificação em dois níveis apresentado nesta seção, o **identificador de um item** é chamado também de **nome do repositório uniforme do item** porque ele pode ser utilizado para definir uma sequência de quatro diretórios servindo para armazenar, num sistema de arquivos, o **item** sendo identificado, caso este esteja do tipo **item de informação**.

seja

Os repositórios são chamados de uniforme porque, por meio destes, qualquer **item de informação** pode ser armazenado em qualquer sistema de arquivos, debaixo de um mesmo diretório, sem conflito de nome quando considerados outros **itens de informação**, facilitando assim o depósito de cópias em sistema de arquivos distintos e ainda a migração de **itens de informação** entre os mesmos.

No **nome de repositório uniforme de um item**, o prefixo e o sufixo são separados por "/" e cada um é, por sua vez, subdividida em duas partes separadas também por "/". Assim, os **rótulos** são constituídos de quatro partes, que podem se tornar uma sequência de quatro diretórios.

Como anunciado, as duas partes do prefixo são construídas a partir de um nome de domínio de computador (*hostname*) e eventualmente de um número de porta. Quanto ao sufixo, as duas partes são construídas a partir de uma informação de data e hora.

Assim, as quatro partes do **nome do repositório uniforme de um item** são formadas por, nesta ordem:

- a) um nome de subdomínio,
- b) uma palavra¹ de domínio, e eventualmente um número de porta, separados por "." ou por "@",
- c) um ano e
- d) um mês, dia, hora, minuto, e eventualmente segundo, separados por ".".

¹Um nome de domínio é constituído de palavras separadas por pontos.

Estas quatro partes são reconhecíveis no Exemplo 1 do capítulo anterior, onde o **nome do repositório uniforme do item** era:

sid.inpe.br/mtc-m18@80/2009/02.16.17.46

Para definir precisamente a sintaxe do **nome do repositório uniforme de um item**, neste projeto de norma, usa-se uma gramática BNF – *Backus Normal Form* ou *Backus-Naur Form* – (aumentada) (CROCKER, 1982; CROCKER; OVERELL, 2008) com a seguinte alteração: "|" é utilizado para alternativas no lugar de "/".

A sintaxe da parte relativa ao prefixo incorpora as regras própria à formação de “nome de domínio” (*domain name*) como definidas na Seção 3.1 intitulada *Name space specifications and terminology* por Mockapetris (1987), e de “nome de domínio de um computador” (*hostname*) como definidas na Seção 3.2.2 intitulada *Server-based Naming Authority* por Berners-Lee et al. (1998).

A Tabela 4.1 contém as regras para a formação de um **nome de repositório uniforme de um item**. contém

Ao acrescentar a porta à palavra no prefixo, o separador pode ser o símbolo "." ou o símbolo "@". Recomenda-se usar apenas o símbolo ".". O uso do símbolo "@" era necessário em implementações do IBI anteriores a agosto de 2010. O inconveniente deste símbolo é que ele induza certos aplicativos a interpretar o **identificador de um item** como um endereço de *e-mail*.

A regra <subdomínio> é denotada <hostname> em Berners-Lee et al. (1998).

O nome de domínio de um computador sendo insensível a maiúscula e minúscula, esta propriedade se estende ao **nome do repositório uniforme de um item**. Assim, sid.inpe.br/mtc-m18@80/2009/02.16.17.46 e sid.INPE.br/MTC-m18@80/2009/02.16.17.46 são equivalentes. Por este motivo, em todo rigor, o **sistema de identificação dos subsistemas de identificação** não aponta para um **rótulo** mas sim, para uma classe de equivalência de **rótulos**. Na prática, recomenda-se utilizar apenas letras minúsculas na geração do prefixo.

Além das regras sintáticas da Tabela 4.1, um **nome de repositório uniforme de um item** deve verificar as regras semânticas definidas por meio dos Algoritmos 1, 2 e 3. Por sua vez, por construção, o Algoritmo 3 gera um **rótulo** que verifica as

Tabela 4.1 - Regras definindo a formação de um **nome de repositório uniforme de um item**

repositório	=	prefixo "/" sufixo ; ex: sid.inpe.br/mtc-m19/2010/08.25.12.38
prefixo	=	subdomínio "/" palavra [("." "@") porta] ; ex: sid.inpe.br/mtc-m19
subdomínio	=	*(palavra ".") última-palavra ["."]; ex: dpi.inpe.br
palavra	=	ALFANUM (ALFANUM *(ALFANUM "-") ALFANUM); ex: sid
ALFANUM	=	ALFA DÍGITO
ALFA	=	ALFAMI ALFAMA
ALFAMI	=	"a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
ALFAMA	=	"A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
DÍGITO	=	"0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
última-palavra	=	ALFA (ALFA *(ALFANUM "-") ALFANUM); ex: br
porta	=	1*DÍGITO; ex: 80
sufixo	=	ano "/" mês "." dia "." hora "." minuto ["."] segundo] ; ex: 2010/08.25.12.38
ano	=	4*DÍGITO; ex: 2010
mês	=	2DÍGITO; ex: 08
dia	=	2DÍGITO; ex: 25
hora	=	2DÍGITO; ex: 12
minuto	=	2DÍGITO; ex: 38
segundo	=	2DÍGITO

regras sintáticas da Tabela 4.1.

O Algoritmo 1, é a descrição biunívoca do **gerador de rótulo**, denotado a na Figura 3.1, utilizado pelo **sistema de identificação** f para a identificação dos **subsistemas de identificação**.

O Algoritmo 2, é a descrição biunívoca do **gerador de rótulo**, denotado b na Figura 3.1, utilizado por qualquer **subsistema de identificação**.

O Algoritmo 3 é a descrição biunívoca do **gerador de rótulo**, denotado c na Figura 3.1, utilizado pelo **sistema de identificação em dois níveis** s .

O **sistema de identificação em dois níveis** funciona de forma **distribuída**, um servidor associado a cada **subsistema de identificação**. Os servidores são hospedados em computadores possuindo nomes de domínio, e o acesso aos servidores sendo feito via portas.

Ao receber uma solicitação de identificação de um **item** i o servidor associado ao **subsistema de identificação** $g(i)$ executa os Algoritmos 1, 2 e 3 e retorna o **identificador do item**.

Para o correto funcionamento, as entradas do Algoritmo 1 devem ser: o nome de domínio do computador (*hostname*) ou do computador virtual (*virtual host*) que hospede o servidor, e a porta que dá acesso a este servidor.

O par formado pelo nome de domínio (em minúsculo) do computador e a porta de acesso referentes ao **subsistema de identificação** $g(i)$ é interpretado, na Figura 3.1 como a saída do chamado **instalador** recebendo como entrada o **subsistema de identificação** $g(i)$.

O nome de domínio do computador (*hostname*) pode ser obtido, por exemplo, por meio do comando `nslookup`.

Quanto ao Algoritmo 2, as entradas devem ser: as datas, arredondadas em segundo, da geração do **rótulo** (sufixo) corrente e do último **rótulo** (sufixo) pelo **gerador de rótulo** b utilizado pelo **subsistema de identificação** $g(i)$.

O par formado pelas datas de geração do sufixo corrente e do último sufixo é interpretado, na Figura 3.1 como a **saída** de um **distribuidor temporal** recebendo como entrada o próprio **item** i .

O papel do **distribuidor temporal** utilizado pelo **subsistema de identificação** $g(i)$ é **distribuir** numa grade bidimensional com granularidade de um segundo, as solicitações de identificações referentes aos **itens** do conjunto $g^*(g(i))$.

O **distribuidor temporal** é descrito detalhadamente por meio da definição a seguir.

Definição 2 (distribuidor temporal) – Os conjuntos e expressões introduzidos a seguir são os componentes de um **distribuidor temporal**.

Seja I' um conjunto de **itens**.

Seja \mathbb{N}^2 o produto Cartesiano de \mathbb{N} por ele mesmo.

Seja $t_i \in \mathbb{R}^+$ a data expressa em segundos² da solicitação de identificação do **item** i (supõe-se que $i \mapsto t_i$ é injetora).

²Mais precisamente em *Unix time* ou *POSIX time*.

Seja $[i] \in \{1, 2, \dots, |I'|\}$ o valor indicando que o **item** i foi o $[i]$ ésimo **item** a solicitar uma identificação, i.e., $[i]$ é dado por:

$$[i] \triangleq \sum_{j \in I'} \begin{cases} 1 & \text{se } t_j < t_i, \\ 0 & \text{caso contrário,} \end{cases} \quad \text{para todos } i \in I'.$$

Seja $]k[\in I'$ o valor indicando que o k ésimo **item** a solicitar uma identificação é o **item** $]k[$, i.e., $]k[$ é dado por:

$$]k[\triangleq i \Leftrightarrow k = [i] \quad \text{para todos } k \in \{1, 2, \dots, |I'|\}.$$

Seja $t'_i \in \mathbb{N}$ a data, arredondada em segundo, para a geração do rótulo utilizado na identificação do **item** i , definida por:

$$t'_i \triangleq \begin{cases} \text{int}(t_i) & \text{para } [i] = 1, \\ \max(t'_{]i[-1[} + 1, \text{int}(t_i)) & \text{para } [i] = 2, \dots, |I'|. \end{cases}$$

Na expressão acima, $]i[-1[$ é o **item** imediatamente anterior ao **item** i , considerando as datas de solicitação de identificação.

Um **distribuidor temporal** é a função $t : I' \rightarrow \mathbb{N}^2$ definida por:

$$t(i) \triangleq \begin{cases} (t'_i, 0) & \text{para } [i] = 1, \\ (t'_i, t'_{]i[-1[}) & \text{para } [i] = 2, \dots, |I'|. \end{cases}$$

□

A Tabela 4.2 contem os dados de um exemplo de funcionamento de um **distribuidor temporal**.

Tabela 4.2 - Exemplificação do funcionamento de um **distribuidor temporal**

i	t_i (segundos)	$[i]$	t'_i	$t(i)$
d	1287587646,394023	1	1287587646	(1287587646, 0)
b	1287588012,2930	2	1287588012	(1287588012, 1287587646)
a	1287588115,186234	3	1287588115	(1287588115, 1287588012)
c	1287588115,3462	4	1287588116	(1287588116, 1287588115)
g	1287588115,99623	5	1287588117	(1287588117, 1287588116)
f	1287588116,72	6	1287588118	(1287588118, 1287588117)
e	1287588539,788342	7	1287588539	(1287588539, 1287588118)

Na Tabela 4.2, as linhas foram ordenadas por datas crescentes de solicitação de identificação (coluna t_i). Os valores de $[i]$ indicam a ordem das solicitações. Observe, por exemplo, que o **item g** apesar de ter solicitado sua identificação numa data anterior à 1287588116 segundos, recebeu como data para a geração do **rótulo** o valor de 1287588117 segundos. Isto ocorreu porque os **itens a, c e g** fizeram os três a solicitação de identificação dentro do período de um segundo. Conseqüentemente o **itens g** (haverá) que aguardar que os **itens a e c** recebam primeiro suas identificações para receber a sua. terá

Na prática o **distribuidor temporal** pode ser implementado usando os conceitos de fila de espera e de temporizador para redistribuir as datas de solicitação de identificação numa grade bidimensional com granularidade de um segundo.

Algoritmo 1 – MONTARPREFIXODO NOME DE REPOSITÓRIO DE UM ITEM.

ENTRADA: computador (*nome de domínio (em minúsculo) do computador (eventualmente virtual) hospedando o servidor associado ao subsistema de identificação*),
porta (*número da porta de acesso ao servidor associado ao subsistema de identificação*).

SAÍDA: prefixo (*prefixo*).

AUXILIARES: parte (*inteiro*),
aux (*cadeia de caracteres*),
subdomínio (*cadeia de caracteres*),
primeira-palavra (*cadeia de caracteres*),
palavra-porta (*cadeia de caracteres*),
c (*caracter*).

1. aux \leftarrow computador.
2. parte \leftarrow 2.
3. **Enquanto** aux \neq "", **Faça**
4. c \leftarrow SAIRFILA(aux).
5. **Se** c = "." **Então**
6. └ parte \leftarrow 1.
7. **Senão**
8. **Se** parte = 1 **Então**
9. └ ENTRARFILA(subdomínio, c).
10. **Senão**
11. └ ENTRARFILA(primeira-palavra, c).
12. **Se** porta = 80 **Então**
13. └ palavra-porta \leftarrow primeira-palavra.
14. **Senão**
15. └ palavra-porta \leftarrow CONCATENAR(primeira-palavra, ".", porta).
16. prefixo \leftarrow CONCATENAR(subdomínio, "/", palavra-porta).

Algoritmo 2 – MONTARSUFIXODO NOME DE REPOSITÓRIO DE UM ITEM.

ENTRADA: *data-corrente* (*data*, arredondada em segundo, da geração do rótulo corrente – *sufixo corrente*),
última-data (*data*, arredondada em segundo, da geração do último rótulo – *último sufixo*).

SAÍDA: *sufixo* (*sufixo*).

AUXILIARES: *data-em-segundo-arredondada-em-minuto* (*inteiro*),
data (*inteiro*),
ano (*inteiro*),
mês (*inteiro*),
dia (*inteiro*),
hora (*inteiro*),
minuto (*inteiro*),
segundo (*inteiro*).

1. $data-em-segundo-arredondada-em-minuto \leftarrow 60 * INT(data-corrente/60)$.
2. **Se** *última-data* < *data-em-segundo-arredondada-em-minuto* **Então**
3. $data \leftarrow data-em-segundo-arredondada-em-minuto$.
4. **Senão**
5. $data \leftarrow data-corrente$.
6. $ano \leftarrow EXTRAIR(data, ano)$.
7. $mês \leftarrow EXTRAIR(data, mês)$.
8. $dia \leftarrow EXTRAIR(data, dia)$.
9. $hora \leftarrow EXTRAIR(data, hora)$.
10. $minuto \leftarrow EXTRAIR(data, minuto)$.
11. $segundo \leftarrow EXTRAIR(data, segundo)$.
12. $sufixo \leftarrow CONCATENAR(ano, "/", mês, ".", dia, ".", hora, ".", minuto)$.
13. **Se** *segundo* \neq "00" **Então**
14. $sufixo \leftarrow CONCATENAR(sufixo, ".", segundo)$.

Algoritmo 3 – MONTARNOMEDEREPOSITÓRIODEUMITEM.

ENTRADA: **prefixo** (*gerado por* MONTARPREFIXODO NOMEDE REPOSITÓRIODE-
UMITEM),
sufixo (*gerado por* MONTARSUFXODO NOMEDE REPOSITÓRIODE-
UMITEM).
SAÍDA: **repositório** (*nome do repositório de um item*).

1. **repositório** ← CONCATENAR(**prefixo**, "/", **sufixo**).

Os algoritmos apresentados acima usam as seguintes rotinas.

A rotina CONCATENAR concatena as cadeias de caracteres informadas no seus argumentos.

A rotina ENTRARFILA concatena a direita de uma cadeia de caracteres mais um caracter.

A rotina SAIRFILA retira o primeiro caracter de uma cadeia de caracteres e retorna este caracter.

A rotina INT retorna a parte inteira do valor real do argumento.

A rotina EXTRAIR retorna, no formato compatível com as regras da Tabela 4.1, o número inteiro de unidades de tempo, com base na unidade informada no segundo argumento, expresso em UTC (*Coordinated Universal Time*), correspondente à data em segundo, informada no primeiro argumento.

Utiliza-se o padrão UTC de forma a permitir a continuação do funcionamento do **sistema de identificação em dois níveis** mesmo em caso de saída do horário de verão ou de troca de computadores situados em longitudes distintas, e referentes a um mesmo **subsistema de identificação**.

O Algoritmo 1 separa a primeira palavra do nome de domínio³ do computador (eventualmente virtual) do subdomínio sem esta palavra, dividindo assim o prefixo em duas partes, a primeira parte contendo o subdomínio, e a segunda, a primeira palavra.

³Um nome de domínio é constituído de palavras separadas por pontos.

Pelo Algoritmo 1, verifica-se, que quando o número de porta é 80, este é omitido na segunda parte do prefixo.

Como a porta 80 é a porta geralmente utilizada pelos servidores HTTP que rodam os scripts CGI que implementam os Algoritmos 1, 2 e 3, os **nomes de repositório uniforme de um item** ficam assim geralmente mais curto.

O Algoritmo 2 tenta gerar o **rótulo** (sufixo) mais curto possível, fazendo, quando possível, a economia da exibição dos dois dígitos representativos dos segundos. Para conseguir este resultado, a **data-em-segundo-arredondada-em-minuto** é calculada (cf., Linha 1) com base no valor da **data-corrente** (primeiro elemento do par fornecido pelo **distribuidor temporal**) e caso a **última-data** (segundo elemento do par fornecido pelo **distribuidor temporal**) esteja inferior à **data-em-segundo-arredondada-em-minuto**, esta data, arredondada em minuto, prevalece, e com isto, os dois dígitos representativos dos segundos podem ser omitidos.

O Algoritmo 3 concatena o prefixo e o sufixo e interpõe entre estes o símbolo "/". A presença, nesta posição, de um símbolo que não pertence aos alfabetos usados na geração do prefixo e do sufixo, torna a concatenação reversível, pois com sua presença é possível reconhecer, sem ambiguidade, o prefixo do sufixo após a concatenação.

5 Regras de construção do rótulo com base no IP

No **sistema de identificação em dois níveis** apresentado nesta seção, o **identificador de um item** é opaco, construído com base no IP, e chamado **IBIp daquele item**.

Num **IBIp de um item**, o prefixo e o sufixo são também separados por "/".

Como anunciado, o prefixo é construído a partir da informação de um IP (*Internet Protocol*) de computador e eventualmente de um número de porta. Quanto ao sufixo, este é construído a partir de uma informação de data e hora como no capítulo anterior. A opacidade é obtida codificando estas informações.

A concatenação do prefixo e do sufixo são reconhecíveis no Exemplo 2 do Capítulo 3, onde o **IBIp do item** era:

8JMKD3MGP8W/34PGRBS

A Tabela 5.1 contem as regras para a formação de um **IBIp**.

Tabela 5.1 - Regras definindo a formação de um **IBIp**

IBIp	=	palavra "/" palavra ; ex: 8JMKD3MGP7W/385N5PE
palavra	=	1*ALFANUM; ex: 385N5PE
ALFANUM	=	ALFA DÍGITO
ALFA	=	ALFAMI ALFAMA
ALFAMI	=	"a" "b" "c" "d" "e" "f" "g" "h" "j" "k" "l" "m" "n" "p" "q" "r" "s" "t" "u" "w" "x"
ALFAMA	=	"A" "B" "C" "D" "E" "F" "G" "H" "J" "K" "L" "M" "N" "P" "Q" "R" "S" "T" "U" "W" "X"
DÍGITO	=	"2" "3" "4" "5" "6" "7" "8" "9"

Pela Tabela 5.1, observa-se que os caracteres "0", "O", "1", "I", "V", "Y" e "Z" foram excluídos. Os cinco primeiros para eliminar possíveis dúvidas na leitura destes caracteres quando se faz uso de certos fontes. Os dois últimos foram reservados caso se faz necessário definir no futuro um sistema de identificação sensível a maiúscula e minúscula.

Como no caso do **nome do repositório uniforme de um item** do capítulo anterior, o **IBIp de um item** é insensível a maiúscula e minúscula. Assim 8JMKD3MGP8W/34PGRBS e 8jmkd3mgp8w/34pgrbs são equivalentes. Por este motivo, em todo rigor, o **sistema de identificação em dois níveis** não aponta para um **rótulo** mas sim, para uma classe de equivalência de **rótulos**. Na prática, recomenda-se utilizar apenas letras maiúsculas na geração do **rótulo**.

Além das regras sintáticas da Tabela 5.1, um **IBIp de um item** deve verificar as regras semânticas definidas por meio dos Algoritmos 4, 5 e 6. Por sua vez, por construção, o Algoritmo 6 gera um **rótulo** que verifica as regras sintáticas da Tabela 5.1.

O Algoritmo 4, é a descrição biunívoca do **gerador de rótulo**, denotado a na Figura 3.1, utilizado pelo **sistema de identificação** f para a identificação dos **subsistemas de identificação**.

O Algoritmo 5, é a descrição biunívoca do **gerador de rótulo**, denotado b na

Figura 3.1, utilizado por qualquer **subsistema de identificação**.

O Algoritmo 6 é a descrição biunívoca do **gerador de rótulo**, denotado c na Figura 3.1, utilizado pelo **sistema de identificação em dois níveis** s .

Algoritmo 4 – MONTARPREFIXODOIBIPDEUMITEM.

ENTRADA: IP (*IP do computador hospedando o servidor associado ao subsistema de identificação*),
porta (*número da porta de acesso ao o servidor associado ao subsistema de identificação*).

SAÍDA: prefixo (*prefixo*).

GLOBAL: decimal-para-IBIp (*Tabela 5.6 de conversão de decimal para IBIp*),
decimal-para-IPv4 (*Tabela 5.4 de conversão de decimal para IPv4*),
decimal-para-IPv6 (*Tabela 5.5 de conversão de decimal para IPv6*).

AUXILIARES: IP-codificado-decimal (*inteiro*),
IP-codificado (*cadeia de caracteres*),
porta-codificada (*cadeia de caracteres*).

1. **Se** porta = 800 **Então**
2. └─ porta-codificada ← "".
3. **Senão**
4. └─ porta-codificada ← CONVERTERDEDECIMAL(porta, decimal-para-IBIp).
5. **Se** "." ∈ IP **Então**
6. ┌─ IP-codificado-decimal ← CONVERTERPARADECIMAL(IP, decimal-para-IPv4).
7. ├─ IP-codificado ← CONVERTERDEDECIMAL(IP-codificado-decimal, decimal-para-IBIp).
8. └─ prefixo ← CONCATENAR(IP-codificado, "W", porta-codificada).
9. **Senão**
10. ┌─ IP-codificado-decimal ← CONVERTERPARADECIMAL(IP, decimal-para-IPv6).
11. ├─ IP-codificado ← CONVERTERDEDECIMAL(IP-codificado-decimal, decimal-para-IBIp).
12. └─ prefixo ← CONCATENAR(IP-codificado, "X", porta-codificada).

Algoritmo 5 – MONTARSUFIXODOIBIPDEUMITEM.

ENTRADA: *data-corrente* (*data, arredondada em segundo, da geração do rótulo corrente – sufixo corrente*),
última-data (*data, arredondada em segundo, da geração do último rótulo – último sufixo*).

SAÍDA: *sufixo* (*sufixo*).

GLOBAL: *decimal-para-IBIp* (*Tabela 5.6 de conversão de decimal para IBIp*).

AUXILIARES: *data-em-segundo-arredondada-em-minuto* (*inteiro*),
segundos (*inteiro*),
data (*inteiro*).

1. $data-em-segundo-arredondada-em-minuto \leftarrow 60 * INT(data-corrente/60)$.
2. **Se** $última-data < data-em-segundo-arredondada-em-minuto$ **Então**
3. $data \leftarrow data-em-segundo-arredondada-em-minuto$.
4. **Senão**
5. $data \leftarrow data-corrente$.
6. $segundos \leftarrow data - 807235200$.
7. $sufixo \leftarrow CONVERTERDEDECIMAL(segundos, decimal-para-IBIp)$.

Algoritmo 6 – MONTARIBIPDEUMITEM.

ENTRADA: *prefixo* (*gerado por MONTARPREFIXODOIBIPDEUMITEM*),
sufixo (*gerado por MONTARSUFIXODOIBIPDEUMITEM*).

SAÍDA: *IBIp* (*IBIp de um item*).

1. $IBIp \leftarrow CONCATENAR(prefixo, "/", sufixo)$.

Os algoritmos apresentados acima usam as seguintes rotinas.

A rotina `CONVERTERPARADECIMAL` converte, de um determinado sistema de numeração para o sistema de numeração decimal, uma cadeia de caracteres informada no primeiro argumento, em um número inteiro decimal conforme à tabela fornecida no segundo argumento. Esta conversão utiliza a tabela inversa da tabela fornecida.

Alguns exemplos de utilização da rotina `CONVERTERPARADECIMAL` são apresentados na Tabela 5.2

A rotina `CONVERTERDEDECIMAL` converte, do sistema de numeração decimal para

Tabela 5.2 - Exemplos de conversão utilizando a rotina CONVERTERPARADECIMAL

IP	tabela	saida
150.163.2.174	decimal-para-IPv4	4588904456580
2001:252:0:1::2008:6	decimal-para-IPv6	478239719325051908572237

Tabela 5.3 - Exemplos de conversão utilizando a rotina CONVERTERDEDECIMAL

decimal	tabela	saida
1	decimal-para-IBIp	3
19050	decimal-para-IBIp	U5H
480992662	decimal-para-IBIp	38G3TS3
4588904456580	decimal-para-IBIp	J8LNKAN8P
478239719325051908572237	decimal-para-IBIp	7URMDHLL9SSN2D89M

um outro sistema de numeração, o número inteiro decimal informado no primeiro argumento, em uma cadeia de caracteres conforme à tabela fornecida no segundo argumento.

Alguns exemplos de utilização da rotina CONVERTERDEDECIMAL são apresentados na Tabela 5.3

Nas Linhas 1 a 4 do Algoritmo 4, testa-se se o número de porta é 800, neste caso não há conversão deste número e usa-se uma cadeia vazia, caso contrário o número de porta é codificado usando a Tabela 5.6.

Diferentemente do capítulo anterior, não se considera o número de porta 80, porque pode existir mais de um computador virtual (*virtual host*) usando a mesma porta 80. Em testes deste projeto de norma, utilizou-se números de porta como 800 e 802, para o acesso aos scripts que implementam os **geradores de rótulo** utilizados pelos **sistemas de identificação em dois níveis** hospedados em computadores virtuais distintos (dentro um mesmo computador real, i.e, referente ao mesmo IP). Caso existir apenas um único computador virtual (para o mesmo IP), então privilegiou-se a utilização do número de porta 800. Assim, toda vez que se utiliza o número de porta 800, os **IBIps de um item** ficam mais curto.

Na Linha 5 do Algoritmo 4 testa-se o tipo de IP. Caso o IP for do tipo IPv4, na concatenação da Linha 8 utiliza-se o caracter "W" para separar o IP codificado, do

número de porta codificado. Caso o IP for do tipo IPv6, na Linha 12 utiliza-se o caracter "X" para este propósito.

Como os caracteres "W" e "X" não fazem parte dos grafemas da Tabela 5.6 é possível, caso seja necessário, decodificar o prefixo do **IBIp de um item**.

Para codificar um IP, considera-se que seu valor representa um número dentro de um determinado sistema de numeração. A codificação, consiste então em converter a representação no sistema original para uma nova representação num outro sistema de numeração, denotado aqui IBIp. Como existem dois tipos de IP: IPv4 e IPv6, considera-se dois sistemas de numeração originais, denotados, respectivamente, IPv4 e IPv6.

A converção de um IP de um destes dois sistemas de numeração para o sistema IBIp, é feita no Algoritmo 4, recorrendo a sua representação no sistema decimal. Assim, para converter uma representação no sistema IPv4 (resp., IPv6) para sua representação no sistema IBIp, é feito primeiro a conversão da representação no sistema IPv4 (resp., IPv6) para sua representação no sistema decimal com base na tabela inversa da Tabela 5.4 (resp., 5.5) e em seguida a conversão da sua representação no sistema decimal para sua representação no sistema IBIp com base na Tabela 5.6.

As 5 primeiras linhas do Algoritmo 5 são as mesmas que do Algoritmo 2.

Em seguida, na Linha 6, o Algoritmo 5 calcula a diferença em segundos entre a variável `data` e a constante 807235200. Esta constante é o número de segundos (em *Unix time*) corresponde à data 19950801T000000Z (data no formato ISO 8601) de início do mês durante o qual ocorreu a geração do primeiro **rótulo**, seguindo este projeto de norma. Este modo operante, permite gerar **rótulos** mais curtos, desde que obtidos como resultado da conversão de números menores.

Tabela 5.4 - Tabela de conversão de decimal para IPv4

decimal	IPv4
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	.

Tabela 5.5 - Tabela de conversão de decimal para IPv6

decimal	IPv6
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	a
11	b
12	c
13	d
14	e
15	f
16	:

Tabela 5.6 - Tabela de conversão de decimal para IBIp

decimal	IBIp
0	2
1	3
2	4
3	5
4	6
5	7
6	8
7	9
8	A
9	B
10	C
11	D
12	E
13	F
14	G
15	H
16	J
17	K
18	L
19	M
20	N
21	P
22	Q
23	R
24	S
25	T
26	U

REFERÊNCIAS BIBLIOGRÁFICAS

BERNERS-LEE, T.; FIELDING, R.; IRVINE, U. C.; MASINTER, L. **Uniform Resource Identifiers (URI): Generic syntax**. Washington DC: The Internet Engineering Task Force (IETF), Aug. 1998. 40 p. RFC 2396. Disponível em: <<http://tools.ietf.org/html/rfc2396>>. Acesso em: 19 ago. 2010. 11

CORPORATION FOR NATIONAL RESEARCH INITIATIVES (CNRI). **Handle System Fundamentals**. 2010. Disponível em: <http://www.handle.net/overviews/system_fundamentals.html>. 2

CROCKER, D. H. **Standard for the format of ARPA Internet messages**. Washington DC: The Internet Engineering Task Force (IETF), Aug. 1982. 47 p. RFC 822. Disponível em: <<http://tools.ietf.org/html/rfc822>>. Acesso em: 19 ago. 2010. 11

CROCKER, D. H.; OVERELL, P. **Augmented BNF for Syntax Specifications: ABNF**. Washington DC: The Internet Engineering Task Force (IETF), Jan. 2008. 16 p. RFC 5234. Disponível em: <<http://tools.ietf.org/html/rfc5234>>. Acesso em: 19 ago. 2010. 11

MOCKAPETRIS, P. **Domain names - concepts and facilities**. Washington DC: The Internet Engineering Task Force (IETF), Nov. 1987. 55 p. RFC 1034. Disponível em: <<http://tools.ietf.org/html/rfc1034>>. Acesso em: 19 ago. 2010. 11

VERISIGN. Internet grows to nearly 202 million domain names in third quarter of 2010. 2010. Disponível em: <https://press.verisign.com/easyir/customrel.do?easyirid=AFC0FF0DB5C560D3&version=live&prid=691769&releasejsp=custom_97>. Acesso em: 05 dez. 2010. 9

WEIBEL, S.; JUL, E.; SHAFER, K. **PURLs: Persistent Uniform Resource Locators**. OCLC Online Computer Library Center, 2010. Disponível em: <http://purl.org/docs/brief_intro.html>. Acesso em: 3 dez. 2010. 2

APÊNDICE A - DEFINIÇÕES E PROPRIEDADES

Definição 3 (função) – Sejam X e Y dois conjuntos não vazios, e F um subconjunto do produto cartesiano $X \times Y$. Uma *função f de X em Y* é uma tripla (X, Y, F) que satisfaz os seguintes axiomas:

- (i) para todos $x \in X$, existe um $y \in Y$ tal que $(x, y) \in F$;
- (ii) para todos (x_1, y_1) e $(x_2, y_2) \in F$, $x_1 = x_2 \Rightarrow y_1 = y_2$.

Uma função f de X em Y é denotada por $f : X \rightarrow Y$.

X e Y são, respectivamente o *domínio* e *contradomínio* de f . □

Pelo Axioma (i) da Definição 3, uma função f associa cada elemento x de X à pelo menos um elemento y de Y . Pelo Axioma (ii), este elemento é único. O elemento y associado à x é denotado por $f(x)$ e chamado de *valor de f em x* :

$$f(x) \triangleq y \Leftrightarrow (x, y) \in F, \quad \text{para todos } x \in X \text{ e } y \in Y. \quad (\text{A.1})$$

Com base nesta notação, f satisfaz as seguintes propriedades decorrentes dos axiomas da Definição 3.

Proposição 2 (propriedades de função) – Seja f uma função de X em Y , então:

- (i) para todos $x \in X$, existe um $y \in Y$ tal que $y = f(x)$;
- (ii) para todos x_1 e $x_2 \in X$, $f(x_1) \neq f(x_2) \Rightarrow x_1 \neq x_2$. □

Prova. Propriedade (i) decorre do Axioma (i) da Definição 3 e da definição de $f(x)$ (Equação A.1). Propriedade (ii) decorre do Axioma (ii), pois, para todos x_1 e $x_2 \in X$:

$$\begin{aligned} \text{verdadeiro} &\Leftrightarrow (x_1, f(x_1)) \text{ e } (x_2, f(x_2)) \in F && (\text{Eq. A.1}) \\ &\Rightarrow (x_1 = x_2 \Rightarrow f(x_1) = f(x_2)) && (\text{Axioma (ii) da Def. 3}) \\ &\Leftrightarrow (f(x_1) \neq f(x_2) \Rightarrow x_1 \neq x_2). && (\text{Prop. da implicação/negação}) \end{aligned}$$

□

Definição 4 (imagem, imagem inversa) – Sejam X e Y dois conjuntos não vazios,

e f uma função de X em Y .

A *imagem de um subconjunto A de X através de f* , é o subconjunto de Y denotado por $f(A)$ e dado por:

$$f(A) \triangleq \{f(x) : x \in A\}, \quad \text{para todos } A \subset X.$$

A *imagem inversa de um subconjunto B de Y através de f* , é o subconjunto de X denotado por $f^{-1}(B)$ e dado por:

$$f^{-1}(B) \triangleq \{x \in X : f(x) \in B\}, \quad \text{para todos } B \subset Y.$$

□

Para todos $y \in f(X)$, $f^{-1}(\{y\})$ é chamado de *fibra de y através de f* . Aqui chama-se de *geradora de fibras através de f* , a função de $f(X)$ em $\mathcal{P}(X)$ (o conjunto de todas partes de X), denotada por f^* e dada por:

$$f^*(y) \triangleq f^{-1}(\{y\}), \quad \text{para todos } y \in f(X). \quad (\text{A.2})$$

Com base na definição de imagem inversa (Definição 4) e de geradora de fibras (Equação A.2), tem-se:

$$x \in f^*(y) \Leftrightarrow f(x) = y, \quad \text{para todos } x \in X \text{ e } y \in Y; \quad (\text{A.3})$$

$$y \in f(X) \Leftrightarrow f^*(y) \neq \emptyset, \quad \text{para todos } y \in Y. \quad (\text{A.4})$$

Proposição 3 (propriedade da geradora de fibras) – Sejam X e Y dois conjuntos não vazios, e f uma função de X em Y , então f^* de $f(X)$ em $\mathcal{P}(X)$ é injetora. □

Prova. A função $f^* : f(X) \rightarrow \mathcal{P}(X)$ é injetora, pois, para todos y_1 e $y_2 \in f(X)$:

$$y_1 \neq y_2 \Rightarrow (\forall x_1 \in f^*(y_1) \text{ e } x_2 \in f^*(y_2), f(x_1) \neq f(x_2)) \quad (\text{Eq. A.3})$$

$$\Rightarrow (\forall x_1 \in f^*(y_1) \text{ e } x_2 \in f^*(y_2), x_1 \neq x_2) \quad (\text{Propriedade (ii) da Prop. 2})$$

$$\Leftrightarrow f^*(y_1) \cap f^*(y_2) = \emptyset \quad (\text{Definição de interseção})$$

$$\Rightarrow f^*(y_1) \neq f^*(y_2). \quad (\text{Eq. A.4})$$

□