



Ministério da
**Ciência, Tecnologia
e Inovação**



iconet.com.br/banon/2009/09.09.22.01-RPQ

IDENTIFICADOR COM BASE NA INTERNET (IBI): SISTEMA DE IDENTIFICAÇÃO

Gerald Jean Francis Banon

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

INPE
São José dos Campos
2011

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 INTELLECTUAL 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)

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, Tecnologia
e Inovação**



iconet.com.br/banon/2009/09.09.22.01-RPQ

IDENTIFICADOR COM BASE NA INTERNET (IBI): SISTEMA DE IDENTIFICAÇÃO

Gerald Jean Francis Banon

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

INPE
São José dos Campos
2011

Dados Internacionais de Catalogação na Publicação (CIP)

B227i Banon, Gerald Jean Francis.
Identificador com base na Internet (IBI): Sistema de identificação / Gerald Jean Francis Banon. – São José dos Campos : INPE, 2011.
40 p. ; (iconet.com.br/banon/2009/09.09.22.01-RPQ)

1. Identificador. 2. Vínculo persistente. 3. Internet. 4. Organização de arquivos. 5. URL. 6. Identifier. I. Título.

CDU 004.451.53

Copyright © 2011 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente com o propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright © 2011 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming, or otherwise, without written permission from INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

AGRADECIMENTOS

Agradecemos aqui aos membros da Comissão de Estudo em Sistemas Espaciais de Transferência de Dados e Informação (CE-08:010.70) do Comitê Brasileiro de Aeronáutica e Espaço (ABNT/CB-08), e em particular a Eduardo W. Bergamini (INPE), Regina M. Silveira (LARC/USP) e Maurício G. V. Ferreira (INPE), pelas sugestões de melhoria na primeira versão deste texto. Estamos particularmente gratos ao Eduardo W. Bergamini pelo incentivo recebido para que, com base no presente relatório, seja montada uma proposta de normalização junto à ABNT, assim como à Regina M. Silveira pela sua sugestão de inclusão das frações de segundos no IBI.

Agradecemos aqui também, o convite feito por Helio Kuramoto do Instituto Brasileiro de Informação em Ciência e Tecnologia (IBICT) para participarmos do grupo de discussão sobre Identificadores Persistentes que promoveu, entre abril de 2008 e abril de 2009, na forma de troca de e-mails, video conferências e interações via Wiki. Foi com base em comentários formulados por Helio Kuramoto e João Alberto de Oliveira Lima que introduzimos, nessa época, novas versões do IBI.

RESUMO

Este relatório apresenta duas formas em que um identificador global pode ser criado para identificar e prover acesso consistente e perene a diversos tipos de itens de informação (documentos, mapas, imagens, etc.) armazenados em acervos como os encontrados em repositórios digitais, em arquivos, ou em outra entidade de informação. A implantação desse identificador global requer, de uma forma indireta, infra-estrutura já existente e facilmente disponível da Internet. Portanto, sem custo adicional, quanto a este aspecto. Esse identificador global pode ser utilizado em associação com o processo de armazenamento de informação em acervos. O que também torna simples a criação de cópias em acervos distintos, incluindo a própria migração de itens de informação entre tais acervos. As diversas aplicações de um identificador global desta natureza são também de particular interesse para uso em sistemas de dados espaciais e de informação.

INTERNET BASED IDENTIFIER - DRAFT STANDARD

ABSTRACT

This report presents a procedure that leads to the creation of two versions of a global identifier which is intended, in a long term, to consistently and compactly identify and to provide a convenient access to various kinds of information items (documents, maps, images, etc.) which are typically stored in collections, as found in digital repositories, in archives, or elsewhere. The practical deployment of this global identifier conveniently and essentially solely requires, at no additional cost, the widely, already available infrastructure of the Internet. This global identifier can be used in combination with information storage systems which deal with collections and which, in this way, may enable remarkable simplicity in the processes dedicated to the creation of copies in different collections, also including simplicity in the migration of information items among such collections. In particular, a variety of convenient applications of a global identifier of this nature in space data and information systems are envisioned.

LISTA DE FIGURAS

	<u>Pág.</u>
3.1 Sistema para geração de IBI	8

LISTA DE TABELAS

	<u>Pág.</u>
4.1 Regras definindo a formação de um nome de repositório uniforme . . .	15
4.2 Exemplificação do funcionamento de um distribuidor temporal com granularidade de um segundo.	22
5.1 Regras definindo a formação de um IBIp	27
5.2 Exemplos de conversão utilizando a rotina <code>CONVERTERPARADECIMAL</code> .	31
5.3 Exemplos de conversão utilizando a rotina <code>CONVERTERDEDECIMAL</code> . .	31
5.4 Tabela de conversão de decimal para IPv4	33
5.5 Tabela de conversão de decimal para IPv6	33
5.6 Tabela de conversão de decimal para IBIp	34

SUMÁRIO

	<u>Pág.</u>
1 Introdução	1
2 Terminologia	3
3 Descrição do sistema para geração de IBI	5
4 Regras de construção do rótulo com base no nome de domínio .	13
5 Regras de construção do rótulo com base no IP	27
REFERÊNCIAS BIBLIOGRÁFICAS	35
APÊNDICE A - DEFINIÇÕES E PROPRIEDADES	37

1 Introdução

Objetivo

Este relatório contém uma proposta de um sistema de identificação que associa a cada item (de informação) a ser identificado, um rótulo gerado com base na Internet e utilizado como identificador desse item. As regras de construção de duas formas de apresentação do rótulo são estabelecidas.

A duplicação, em servidores distintos, de componentes desse sistema deve permitir que atores independentes participem do processo de identificação global.

Uma vez os itens de informação identificados e depositados em servidores interligados na Internet, estes podem ser recuperados por meio de um sistema de resolução de identificação. No entanto, a descrição desse segundo sistema não faz parte deste relatório.

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 persistente, pois, com o tempo, um determinado item de informação pode mudar de localização, fazendo com que a associação: “item de informação” \mapsto URL não fique permanente.

Uma vez escolhido um sistema de identificação e por meio dele atribuídos 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, que deve ter o propósito básico de redirecionar cada URL, agora contendo apenas o identificador de um item de informação, para a URL que, efetivamente, contém o seu endereço físico.

O sistema de identificação descrito neste relatório, apresenta-se como uma alternativa simples, quando comparada a outras soluções como aquelas usadas, por exem-

plo, no PURL (WEIBEL et al., 2010) ou no Handle System[®] (CNRI, 2010). O sistema apresentado está sendo extensivamente utilizado, desde 1995, na plataforma *URLib*.

A principal diferença, comparado ao PURL ou ao 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 detedores de nomes de domínio (*fully qualified domain names*), portanto, sem custo adicional, nesse aspecto.

2 Terminologia

Para os efeitos deste documento, aplicam-se os seguintes termos e definições.

Distribuidor espacial: função entre um conjunto de **itens** e um conjunto de **subsistema de identificação**, distribuindo cada **item** à um determinado **subsistema de identificação** tornando esse **subsistema de identificação** responsável pela identificação desse **item**.

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

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

IBI: sigla para “Identificador com Base na Internet”. Qualquer **rótulo** gerado pelo **sistema para geração de IBI**.

IBI de um item: **rótulo** atribuído à um **item** pelo **sistema para geração de IBI** utilizando o endereçamento por nome de domínio ou IP.

IBIp: qualquer **rótulo** gerado pelo **sistema para geração de IBI** utilizando o endereçamento por IP.

IBIp de um item: **rótulo** atribuído à um **item** pelo **sistema para geração de IBI** utilizando o endereçamento por IP.

Identificador com base na Internet: qualquer **rótulo** gerado pelo **sistema de identificação em dois níveis** objeto deste relatório.

Identificador de um item: **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 de repositório uniforme: qualquer **rótulo** gerado pelo **sistema para geração de IBI** utilizando o endereçamento por nome de domínio.

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** atribuído à um **item** pelo **sistema para geração de IBI** utilizando o endereçamento por nome de domínio.

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 desse item**.

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 esse **subsistema de identificação** (ver detalhes na Definição 1 na Seção 3).

Sistema para geração de IBI: **sistema de identificação em dois níveis** objeto deste relatório.

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

3 Descrição do sistema para geração de IBI

Neste relatório, 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 relatório, 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 um **sistema de identificação em dois níveis** como 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

todo $s \in S$, o subconjunto dos **itens** que estão sob a responsabilidade do **subsistema de identificação** s é $g^*(s)$, a fibra de s através de g (ver definição de fibra no Apêndice A).

Seja, para todo $i \in I$, $g(i) : g^*(g(i)) \rightarrow R_2$, o **subsistema de identificação** responsável pela identificação do **item** i , no escopo desse subsistema.

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

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

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

Um **sistema de identificação em dois níveis** é a função $s : I \rightarrow R$ definida como a composição de h e c , i.e., 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), ou seja, à dois elementos distintos s_1 e s_2 de $g(I)$ podem corresponder duas funções, respectivamente, com domínio $g^*(s_1)$ e $g^*(s_2)$ que serão também distintos por g^* ser injetora.

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 todo i e $j \in I$, tem-se:

$$\begin{aligned}
s(i) = s(j) &\Leftrightarrow c(h(i)) = c(h(j)) && \text{(Def. 1 e def. de composi\c{c}\~{a}o)} \\
&\Rightarrow h(i) = h(j) && \text{(}c\text{ \c{e} injetora)} \\
&\Rightarrow (f(g(i)), g(i)(i)) = (f(g(j)), g(j)(j)) && \text{(Def. 1)} \\
&\Leftrightarrow f(g(i)) = f(g(j)) \text{ e } g(i)(i) = g(j)(j) && \text{(igualdade de pares)} \\
&\Rightarrow g(i) = g(j) \text{ e } g(i)(i) = g(j)(j) && \text{(}f\text{ \c{e} injetora)} \\
&\Rightarrow g(i)(i) = g(i)(j) && \text{(substitui\c{c}\~{a}o)} \\
&\Rightarrow i = j. && \text{(}g(i)\text{ \c{e} injetora)}
\end{aligned}$$

Isto \c{e}, h \c{e} injetora.

□

Dado um **item** i em I , um **sistema de identifica\c{c}\~{a}o em dois n\~{i}veis** atribui \c{a} i o **r\~{o}tulo** gerado por c a partir do par $(f(g(i)), g(i)(i))$ constitu\~{i}do, de um lado, pelo **r\~{o}tulo** $f(g(i))$ atribuido pelo **sistema de identifica\c{c}\~{a}o** f ao **subsistema de identifica\c{c}\~{a}o** $g(i)$ (repons\~{a}vel pela identifica\c{c}\~{a}o de i dentro do escopo $g^*(g(i))$), e de outro lado, pelo **r\~{o}tulo** $g(i)(i)$ atribuido pelo **subsistema de identifica\c{c}\~{a}o** $g(i)$ ao **item** i .

Enquanto $g(i)(i)$, o **identificador do item** i , fornecido pelo **subsistema de identifica\c{c}\~{a}o** $g(i)$, tem validade apenas dentro do escopo $g^*(g(i))$ desse subsistema, $s(i)$, o **identificador desse item**, fornecido pelo **sistema de identifica\c{c}\~{a}o em dois n\~{i}veis** s , tem validade dentro do escopo global I .

O Handle System[®], por exemplo, possui esses dois n\~{i}veis de identifica\c{c}\~{a}o. Os atores no Handle System[®] s\~{a}o o *Corporation for National Research Initiatives* (CNRI) e os *Resolution Service Providers* (RSP). Na terminologia deste relat\~{o}rio, os RSP s\~{a}o os **subsistemas de identifica\c{c}\~{a}o** e o CNRI \c{e} o **sistema de identifica\c{c}\~{a}o** dos RSP.

No primeiro n\~{i}vel, o CNRI (f) registra os RSP e fornece, a cada um, um r\~{o}tulo denominado “prefixo”. No segundo n\~{i}vel, um RSP ($g(i)$) fornece, a cada **item** (i) sob sua responsabilidade, um r\~{o}tulo denominado “sufixo”.

Usando a terminologia do Handle System[®], o **identificador de um item** \c{e} a concatena\c{c}\~{a}o de um “prefixo” identificando um **subsistema de identifica\c{c}\~{a}o** e de

um “sufixo” identificando um **item** no escopo desse subsistema.

O **sistema para geração de IBI**, objeto deste relatório, é também um **sistema de identificação em dois níveis** cujos componentes aparecem na Figura 3.1.

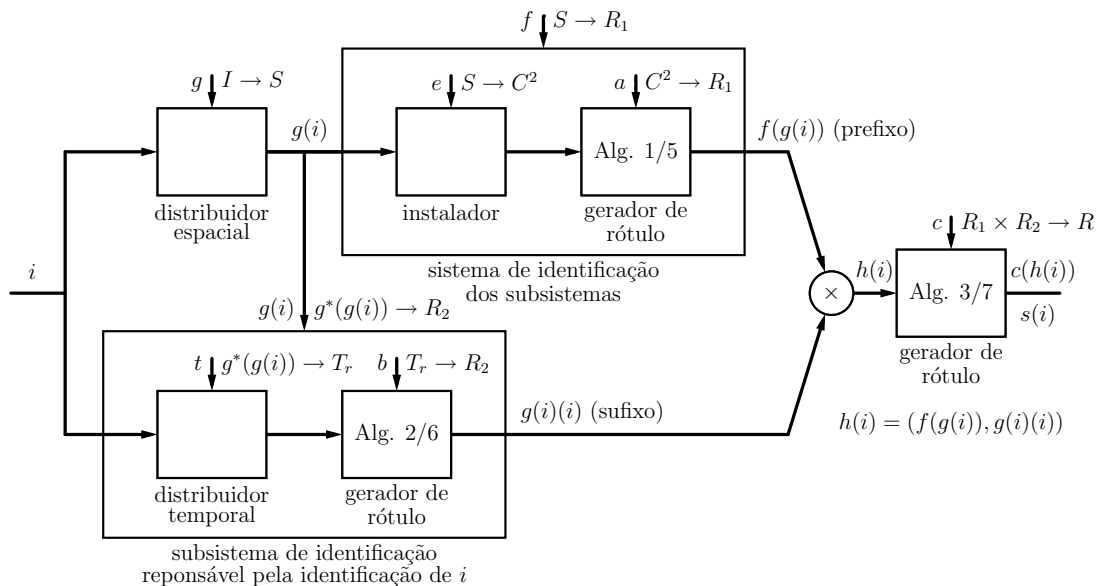


Figura 3.1 - Sistema para geração de IBI

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 $g(i)$.

No **sistema para geração de IBI**, o primeiro nível f consiste (diferentemente do Handle System[®]) em reaproveitar a infra-estrutura já existente da Internet para identificar os **subsistemas de identificação**, portanto, sem custo adicional, nesse aspecto.

Mais precisamente, o **identificador de um item** será obtido como resultado de uma solicitação a um servidor responsável por um determinado **subsistema de identificação**, hospedado exclusivamente em computador possuindo nome de domínio (*fully qualified domain names*). Nesse contexto, cada um dos **subsistemas de identificação** é identificado globalmente por um nome de domínio ou um IP (*Internet Protocol*) na Internet (e uma porta), permitindo assim construir, de forma simples, o prefixo.

O segundo nível $g(I)$, consiste (diferentemente do Handle System[®]) em adotar uma regra para geração do sufixo, comum a todos os **subsistemas de identificação**. Essa regra é construída com base na data e hora da associação do **item** ao **rótulo**. Essa escolha facilita o reuso de um **identificador de subsistema de identificação** quando este passa sob o controle de um novo ator.

Com essas escolhas, qualquer **rótulo** gerado pelo **sistema de identificação em dois níveis** s recebeu o nome de “Identificador com Base na Internet” ou **IBI**, e o **rótulo** $s(i)$ atribuído ao **item** i por s é o **IBI do item** i .

Na solução objeto deste relatório, 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 responsável por esse subsistema, assim como a porta de acesso a esse 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 próximas seções.

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 minutos², resultou no sufixo:

2009/02.16.17.46

O servidor emitindo esse 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:

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

²Data e hora expressas em Tempo Universal Coordenado (em inglês: *Coordinated Universal Time* (UTC)).

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

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

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

Adicionalmente, o ponteiro (URL) persistente para os metadados desse **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 esse item**. O importante, apenas, é que esses 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 ilustrando a formação de um identificador com base no IP.

Exemplo 2 (identificador opaco com base no IP) – A associação de um **item** com um sufixo, ocorrida na segunda 1234806360 em *POSIX time* (correspondendo a data de 16 de fevereiro de 2009 às 17 horas 46 minutos), resultou no sufixo opaco:

34PGRBS

O servidor emitindo esse 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 esse **item** ficou:

`http://urlib.net/8JMKD3MGP8W/34PGRBS`

Adicionalmente, o ponteiro (URL) persistente para os metadados desse **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 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 ou ainda as frações de segundos.

Os dois tipos de **sistema para geração de IBI** são apresentados a seguir em detalhe. No primeiro, o **identificador de um item**, exibindo o nome de domínio, 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 com base no nome de domínio

No sistema para geração de IBI 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 seja do tipo **item de informação**.

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**, 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 expressa em Tempo Universal Coordenado (em inglês *Coordinated Universal Time* (UTC)).

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 segundo², separados por ".".

Estas quatro partes são reconhecíveis no Exemplo 1 da seção anterior, onde o **nome**

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

²ou fração de segundos.

do repositório uniforme do item era:

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

O formato do sufixo é uma adaptação da norma ISO 8601 onde os separadores foram trocados de forma a se adequar as normas próprias para caminhos em sistema de arquivos. Por exemplo, a data e hora: 16 de fevereiro de 2009 às 17 horas 46 minutos, na norma ISO 8601 fica:

2009-02-16T17:46Z

ou ainda:

20090216T1746Z

enquanto o sufixo fica:

2009/02.16.17.46

No sufixo, os separadores foram escolhidos para melhorar a legibilidade do **nome do repositório uniforme de um item** e fazer com que o ano corresponde a um diretório e o resto da data a um outro.

Para definir precisamente a sintaxe do **nome do repositório uniforme de um item**, neste relatório, 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 próprias 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**.

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 desse símbolo é que ele induza certos aplicativos a interpretar o **identificador de**

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

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	=	inteiro ["." fração]	
inteiro	=	2DÍGITO	
fração	=	1*DÍGITO	

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 esse motivo, em todo rigor, o **sistema para geração de IBI** 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 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 para geração de IBI** s .

O **sistema para geração de IBI** funciona de forma distribuída, um servidor para cada **subsistema de identificação**. Os servidores estão 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 responsável pelo **subsistema de identificação** $g(i)$ executa os Algoritmos 1, 2 e 3 e retorna o **identificador do item**.

Algoritmo 1 – MONTARPREFIXODONOMEDEREPOSITÓRIODEUMITEM.

ENTRADA: computador (*cadeia de caracteres representando o nome de domínio (em minúsculo) do computador (eventualmente virtual) que hospeda o servidor responsável pelo subsistema de identificação*),
porta (*inteiro decimal representando o número da porta de acesso ao servidor responsável pelo subsistema de identificação*).

SAÍDA: prefixo (*cadeia de caracteres*).

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

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* (*racional decimal produzido pelo distribuidor temporal e usado na geração do rótulo corrente – sufixo corrente*),

SAÍDA: *sufixo* (*cadeia de caracteres*).

AUXILIARES: *ano* (*inteiro decimal*),
mês (*inteiro decimal*),
dia (*inteiro decimal*),
hora (*inteiro decimal*),
minuto (*inteiro decimal*),
segundo (*inteiro decimal*),
fração-de-segundo (*inteiro decimal*).

1. *ano* ← EXTRAIR(*data*, *ano*)
2. *mês* ← EXTRAIR(*data*, *mês*)
3. *dia* ← EXTRAIR(*data*, *dia*)
4. *hora* ← EXTRAIR(*data*, *hora*)
5. *minuto* ← EXTRAIR(*data*, *minuto*)
6. *segundo* ← EXTRAIR(*data*, *segundo*)
7. *fração-de-segundo* ← EXTRAIR(*data*, *fração de segundo*)
8. *sufixo* ← CONCATENAR(*ano*, "/", *mês*, ".", *dia*, ".", *hora*, ".", *minuto*)
9. **Se** *fração-de-segundo* = 0 **Então**
10. **Se** *segundo* ≠ "00" **Então**
11. *sufixo* ← CONCATENAR(*sufixo*, ".", *segundo*)
12. **Senão** *sufixo* ← CONCATENAR(*sufixo*, ".", *segundo*, ".", *fração-de-segundo*)

Algoritmo 3 – MONTARNOMEDEREPOSITÓRIODEUMITEM.

ENTRADA: **prefixo** (*cadeia de caracteres obtida em saída do Algoritmo 1: MONTARPREFIXODO NOMEDE REPOSITÓRIODEUMITEM*),
sufixo (*cadeia de caracteres obtida em saída do Algoritmo 2: MONTARSUFIXODO NOMEDE REPOSITÓRIODEUMITEM*).
SAÍDA: **repositório** (*cadeia de caracteres*).

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 da cadeia de caracteres informada no primeiro argumento, mais o caractere informado no segundo argumento.

A rotina SAIRFILA retira o primeiro caractere da cadeia de caracteres informada no seu argumento, e retorna esse caractere.

A rotina EXTRAIR retorna, no formato compatível com as regras da Tabela 4.1, o número decimal referente a unidade informada no segundo argumento, quando a data em segundo ou fração de segundo, informada no primeiro argumento, é convertida para o Tempo Universal Coordenado (em inglês: *Coordinated Universal Time* (UTC)).

Utiliza-se o padrão UTC de forma a permitir a continuação do funcionamento dos **subsistemas de identificação** mesmo em caso de entrada/saída do horário de verão ou de troca de computadores situados em longitudes distintas.

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.

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,

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

o prefixo dos **nomes de repositório uniforme de um item** fica assim geralmente mais curto.

O Algoritmo 2 gera um **rótulo** (sufixo) com base no valor da **data** fornecido pelo **distribuidor temporal**.

No caso de necessitar de uma resposta mais rápida do **subsistema de identificação**, basta escolher uma granularidade r menor. No entanto, a possibilidade de atender as solicitações de identificação por meio de um grande número de **subsistemas de identificação** (lembrando que a granularidade do prefixo é extremamente fina) constitui uma outra solução para minimizar o problema de uma alta frequência de solicitações.

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.

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 hospeda o servidor responsável pelo **subsistema de identificação**, e a porta que dá acesso a esse 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, em C^2 , do chamado **instalador** e , recebendo como entrada, em S , 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, a entrada deve ser a data $t(i)$, interpretada, na Figura 3.1 como a saída, em T_r , de um distribuidor temporal, recebendo como entrada, em $g^*(g(i))$, o próprio **item** i .

Para todo $i \in I$, o papel do **distribuidor temporal** utilizado pelo **subsistema de identificação** $g(i)$ é distribuir numa grade T_r , com granularidade de r segundos, as solicitações de identificações referentes aos **itens** do conjunto $g^*(g(i))$.

O **distribuidor temporal** é descrito detalhadamente por meio da definição a se-

guir, onde I' representa o domínio $g^*(g(i))$ de um determinado **subsistema de identificação** $g(i)$.

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{Q}^+ o conjunto dos racionais positivos.

Seja \mathbb{R}^+ o conjunto dos reais positivos.

Seja $G \triangleq \{60, 1, 1/10, 1/100, \dots\}$ o conjunto de racionais, definindo as possíveis granularidades temporais: minuto, segundo e frações de segundos.

Seja $t_i \in \mathbb{R}^+$ a data, expressa em fração de segundos⁴, da solicitação de identificação do **item** $i \in I'$ (supõe-se que $i \mapsto t_i$ é injetora).

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

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

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:

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

Seja $t^{(r)} \in \mathbb{Q}^+$ a data t arredondada em r segundos, i.e., $t^{(r)}$ é dado por:

$$t^{(r)} \triangleq \text{rint}(t/r), \quad \text{para todo } t \in \mathbb{R}^+ \text{ e } r \in G.$$

Seja $T_r = \{t^{(r)} : t \in \mathbb{R}^+\}$ o conjunto das datas arredondadas em r segundos, sendo $r \in G$.

Seja $t'_i \in T_r$, com $r \in G$ e $i \in I'$, a data, arredondada em r segundos, dada por:

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

Na expressão acima, $t(]i] - 1])$ é a data fornecida pelo **distribuidor temporal** referente ao **item** $]i] - 1[$ imediatamente anterior ao **item** i , considerando as datas de solicitação de identificação. A data t'_i será considerada a data na qual o **distribuidor temporal** fornece sua resposta $t(i)$ usada na

⁴Mais precisamente em *Unix time* ou *POSIX time*.

geração do rótulo utilizado na identificação do **item** i . Em outros termos, t'_i será considerada a data de geração do rótulo do **item** i , enquanto $t(i)$ é data utilizada pelo gerador de rótulo b para montar o sufixo do identificador de i .

Seja $t'(i) \in T_r^2$, com $r \in G$ e $i \in I'$, o par de datas, arredondadas em r segundos, dado por:

$$t'(i) \triangleq \begin{cases} (t'_i, t'_i - r) & \text{se } [i] = 1, \\ (t'_i, t(\lfloor [i] - 1 \rfloor)) & \text{se } [i] = 2, \dots, |I'|. \end{cases}$$

O mapeamento $i \mapsto t'(i)$ define uma função t' de I' em T_r^2 .

Seja $r_{(t,s)} \in G$, com $t > s$, a maior granularidade r tal que a data t arredondada em r segundos, seja maior do que a data s , i.e., $r_{(t,s)}$ é dado por:

$$r_{(t,s)} \triangleq \max(\{r \in G : t^{(r)} > s\}), \quad \text{para todo } t \text{ e } s \in \mathbb{R}^+ \text{ tal que } t > s.$$

Nota-se que $r_{(t,t-r)} = r$ para todo $t \in \mathbb{R}^+$ e $r \in G$.

Seja $t(i) \in T_{r_{t(i)}} \subset T_r$, com $r \in G$, a data t'_i arredondada em $r_{t'(i)}$ segundos:

$$t(i) \triangleq t'_i{}^{(r_{t'(i)})}, \quad \text{para todo } i \in I'.$$

Nota-se que, para todo $r \in G$, $t(\lfloor 1 \rfloor) = t'_{\lfloor 1 \rfloor}{}^{(r)}$, e que $r_{t(i)} \leq r$, $T_{r_{t(i)}} \subset T_r$ e $t(\lfloor [i] - 1 \rfloor) < t(i) \leq t'_i$, para todo $i \in I'$ tal que $[i] > 1$.

Um **distribuidor temporal**, com granularidade $r \in G$, é a função $t : I' \rightarrow T_r$ tal que $i \mapsto t(i)$. □

A Tabela 4.2 contém os dados de um exemplo de funcionamento de um **distribuidor temporal** com granularidade de 1 segundo ($r = 1$).

Tabela 4.2 - Exemplificação do funcionamento de um **distribuidor temporal** com granularidade de um segundo.

i	t_i (segundos)	$[i]$	$t'(i)$	$r_{t'(i)}$	$t(i)$	sufixo
d	1287587646,394023	1	(1287587646, 1287587645)	1	1287587646	2010/10.20.15.14.06
b	1287588012,2930	2	(1287588012, 1287587646)	60	1287588000	2010/10.20.15.20
a	1287588115,186234	3	(1287588115, 1287588000)	60	1287588060	2010/10.20.15.21
c	1287588115,3462	4	(1287588115, 1287588060)	1	1287588115	2010/10.20.15.21.55
g	1287588115,99623	5	(1287588116, 1287588115)	1	1287588116	2010/10.20.15.21.56
f	1287588116,72	6	(1287588117, 1287588116)	1	1287588117	2010/10.20.15.21.57
e	1287588539,788342	7	(1287588539, 1287588117)	60	1287588480	2010/10.20.15.28

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 $t(i)$, a ser usada na geração do **rótulo**, o valor de 1287588116 segundos. Isto ocorreu porque os **itens a, c e g** fizeram, os três, a solicitação de identificação dentro do período de um segundo. Consequentemente, o **item g** terá que aguardar que os **itens a e c** recebam primeiro suas identificações para receber a sua. Esse “atraso” está se repercutindo para o próximo **item f** que fez sua solicitação de identificação apenas no segundo seguinte (1287588116) ao do **item g** (1287588115).

Na prática, o **distribuidor temporal** poderá ser implementado usando, por exemplo, os conceitos de sala de espera e de temporizador, de forma a associar a cada **item i** uma data de solicitação de identificação t_i (data do começo do atendimento após o aguardo na sala de espera) e transformar esta na data $t(i)$, numa grade temporal com granularidade r , para geração do **rótulo**.

Nesse caso, os **itens a** serem identificados por um determinado **subsistema de identificação** entram numa sala de espera. Quando o servidor responsável pelo **subsistema de identificação** está pronto para identificar um novo **item**, um dos **itens** na sala de espera é sorteado. O instante do sorteio torna-se a data de solicitação de identificação do item sorteado.

O Algoritmo 4 mostra como criar, na prática, a data $t(i)$ utilizada, pelo gerador de rótulo b , para montar o sufixo do identificador de um **item** $i \in I'$.

Para criar a data $t(i)$, esse algoritmo tem que ser executado na data t_i (data de solicitação de identificação), e receber essa data t_i como entrada.

No Algoritmo 4 a entrada, denotada **data-corrente**, corresponde à data t_i , e a saída, denotada **data-para-sufixo**, corresponde à data $t(i)$.

Algoritmo 4 – CRIARDATAPARAMONTARSUFIXODOIBIDEUMITEM.

ENTRADA: `data-corrente` (*racional decimal representando a data de solicitação de uma identificação, i.e., a data de execução do próprio algoritmo*).

SAÍDA: `data-para-sufixo` (*racional decimal*).

GLOBAL: `última-data-para-sufixo` (*racional decimal representando a data criada para o sufixo na execução anterior do próprio algoritmo*),
`r` (*racional decimal representando a granularidade temporal*).

AUXILIARES: `data-arredondada` (*racional decimal*),
`data-de-criação` (*racional decimal*),
`atraso` (*racional decimal*),
`s` (*racional decimal*),
`t` (*racional decimal*),
`data-curta` (*racional decimal*).

1. `data-arredondada` $\leftarrow r * \text{INT}(\text{data-corrente}/r)$
2. **Se** `última-data-para-sufixo` não existe **Então**
3. `última-data-para-sufixo` $\leftarrow \text{data-corrente-arredondada} - r$
4. `última-data-para-sufixo` $\leftarrow r * \text{INT}(\text{última-data-para-sufixo}/r)$
5. `data-de-criação` $\leftarrow \text{MAX}(\text{última-data-para-sufixo} + r, \text{data-arredondada})$
6. `atraso` $\leftarrow \text{data-de-criação} - \text{data-corrente}$
7. **Se** `atraso` > 0 **Então** ESPERAR(`atraso`)
8. `data-curta` $\leftarrow \text{data-de-criação}$
9. `s` $\leftarrow r$
10. **Enquanto** `última-data-para-sufixo` $<$ `data-curta` **Faça**
11. `s` $\leftarrow 10 * s$
12. **Se** `s` = 10 **Então** `s` $\leftarrow 60$
13. `t` $\leftarrow \text{data-curta}$
14. **Se** `s` > 60 **Então Interrompa**
15. `data-curta` $\leftarrow s * \text{INT}(\text{data-de-criação}/s)$
16. `data-para-sufixo` $\leftarrow t$
17. `última-data-para-sufixo` $\leftarrow t$

A variável global `última-data-para-sufixo` tem o papel de uma memória que preserva o último valor de `data-para-sufixo` a ser reaproveitado na criação do próximo sufixo. Caso a variável `última-data-para-sufixo` não exista (por exemplo, na primeira solicitação de identificação), seu valor será escolhido igual à data corrente arredondada em r segundos (`data-arredondada`), menos a granularidade corrente r (Linhas 2 e 3).

A variável global `r` define a granularidade temporal a ser usada na criação do sufixo. Seu valor pode ser 60 (para a granularidade de um minuto), 1 (para segundo) ou 0,1 (para décimo de segundos), 0,01 (para centésimo de segundos), 0,001 (para milésimo de segundos), ...

O valor de `r` pode ser alterado externamente ao algoritmo, de forma a atender novas condições de uso do **subsistema de identificação**.

Na Linha 1, a `data-corrente` é arredondada em r segundos por meio da rotina INT que retorna a parte inteira do valor real do seu argumento.

A Linha 4 serve para reformatar o valor da variável `última-data-para-sufixo`, caso tenha havido uma mudança no valor da granularidade r entre duas criações sucessivas de sufixo.

Na Linha 5, a `data-de-criação` é calculada por meio da rotina MAX que retorna o maior dos seus dois argumentos. Essa variável auxiliar corresponde à data t'_i . Para essa data ser a data de criação do sufixo, uma temporização é introduzida na Linha 7, por meio da rotina ESPERAR que espera pelo tempo especificado na variável `atraso`, toda vez que a `data-de-criação` calculada for maior que a `data-corrente`.

As Linhas 8 a 16 servem para o cálculo da saída `data-para-sufixo` que pode ser uma data menor que a `data-de-criação` caso uma versão mais “curta” dessa data seja compatível com a `última-data-para-sufixo`.

5 Regras de construção do rótulo com base no IP

No sistema para geração de IBI apresentado nesta seção, o **identificador de um item** é opaco, construído com base no IP, e chamado **IBIp** daquele item.

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

Como anunciado, o prefixo é construído a partir da informação de um IP 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 na seção anterior. A opacidade é obtida codificando estas informações.

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

8JMKD3MGP8W/34PGRBS

A Tabela 5.1 contém 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 desses 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 de repositório uniforme** da seção anterior, o **IBIp** é insensível a maiúscula e minúscula. Assim 8JMKD3MGP8W/34PGRBS e 8jmkd3mgp8w/34pgrbs são equivalentes. Por esse motivo, em todo rigor, o sistema

para geração de IBI 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 5, 6 e 7. Por sua vez, por construção, o Algoritmo 7 gera um **rótulo** que verifica as regras sintáticas da Tabela 5.1.

O Algoritmo 5, 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 6, é a descrição biunívoca do **gerador de rótulo**, denotado b na Figura 3.1, utilizado por qualquer **subsistema de identificação**.

O Algoritmo 7 é a descrição biunívoca do **gerador de rótulo**, denotado c na Figura 3.1, utilizado pelo **sistema para geração de IBI** s .

Algoritmo 5 – MONTARPREFIXODOIBIPDEUMITEM.

ENTRADA: IP (*cadeia de caracteres representando o IP do computador que hospeda o servidor responsável pelo subsistema de identificação*),
porta (*inteiro decimal representando o número da porta de acesso ao o servidor responsável pelo subsistema de identificação*).

SAÍDA: prefixo (*cadeia de caracteres*).

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

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

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

Algoritmo 6 – MONTARSUFIXODOIBIPDEUMITEM.

ENTRADA: `data-para-sufixo` (*racional decimal produzido pelo distribuidor temporal e usado na geração do rótulo corrente – sufixo corrente*),
SAÍDA: `sufixo` (*cadeia de caracteres*).
GLOBAL: `decim-para-IBIp` (*Tabela 5.6 de conversão de decimal para IBIp*).
AUXILIARES: `parte-inteira` (*inteiro decimal*),
`segundo` (*inteiro decimal*),
`fração` (*inteiro decimal*),
`fração-codificada` (*cadeia de caracteres*).

1. `parte-inteira` \leftarrow INT(`data-para-sufixo`)
2. `segundo` \leftarrow `parte-inteira` – 807235200
3. `fração` \leftarrow `data` – `parte-inteira`
4. `sufixo` \leftarrow CONVERTERDEDECIMAL(`segundo`, `decim-para-IBIp`)
5. **Se** `fração` \neq 0 **Então**
6. `fração-codificada` \leftarrow CONVERTERDEDECIMAL(`fração`, `decim-para-IBIp`)
7. `sufixo` \leftarrow CONCATENAR(`sufixo`, "w", `fração-codificada`)

Algoritmo 7 – MONTARIBIPDEUMITEM.

ENTRADA: `prefixo` (*cadeia de caracteres obtida em saída do Algoritmo 5: MONTARPREFIXODOIBIPDEUMITEM*),
`sufixo` (*cadeia de caracteres obtida em saída do Algoritmo 6: 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 a cadeia de caracteres, informada no primeiro argumento, em um número inteiro decimal, conforme à tabela informada no segundo argumento. Esta conversão utiliza a tabela inversa da tabela informada.

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

A rotina `CONVERTERDEDECIMAL` converte o número inteiro decimal, informado no primeiro argumento, em uma cadeia de caracteres, conforme à tabela informada no

Tabela 5.2 - Exemplos de conversão utilizando a rotina CONVERTERPARADECIMAL

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

segundo argumento.

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

Tabela 5.3 - Exemplos de conversão utilizando a rotina CONVERTERDEDECIMAL

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

A entrada do Algoritmo 5 é a mesma que do Algoritmo 1.

Nas Linhas 1 a 4 do Algoritmo 5, testa-se se o número de porta é 800, nesse caso não há conversão desse número e usa-se uma cadeia vazia, caso contrário o número de porta é codificado usando a Tabela 5.6.

Diferentemente da seção 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 desta proposta para um sistema de identificação, foram utilizados números de porta como 800 e 802, para o acesso à dois servidores hospedados em computadores virtuais distintos, mas dentro um mesmo computador real, portanto, associado ao mesmo IP. Para o caso em que existe apenas um único computador virtual, foi adotado o número de porta 800. Desta forma, toda vez que se utiliza o número de porta 800, o prefixo do **IBIp de um item** torna-se mais curto.

Na Linha 5 do Algoritmo 5 testa-se o tipo de IP. Caso o IP for do tipo IPv4, na concatenação da Linha 8 utiliza-se o caractere "W" para separar o IP codificado, do número de porta codificado. Caso o IP for do tipo IPv6, na Linha 12 utiliza-se o caractere "X" para esse propósito.

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

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 conversão de um IP de um desses dois sistemas de numeração para o sistema IBIp, é feita no Algoritmo 5, 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.

A entrada do Algoritmo 6 é a mesma que do Algoritmo 2.

Na Linha 2, o Algoritmo 6 calcula a diferença em segundos entre a variável **parte-inteira** 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 a presente proposta de identificação. Esse modo operante, permite gerar **rótulos** mais curtos, desde que obtidos como resultado da conversão de números menores.

Na linha 7, utiliza-se o caractere "W" para separar a parte inteira codificada da parte fracionária codificada. Como o caractere "W" não faz parte dos grafemas da Tabela 5.6 é possível, caso seja necessário, decodificar o sufixo do **IBIp de um item**.

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. 14, 15

CORPORATION FOR NATIONAL RESEARCH INITIATIVES (CNRI). **Handle System Fundamentals**. 2010. Disponível em: <http://www.handle.net/overviews/system_fundamentals.html>. Acesso em: 3 dez. 2010. 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. 14

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. 14

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. 14

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. 11

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 todo $x \in X$, existe um $y \in Y$ tal que $(x, y) \in F$;
- (ii) para todo (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), esse elemento é único. O elemento y associado à x é denotado por $f(x)$ e chamado de *valor de f em x* :

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

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 todo $x \in X$, existe um $y \in Y$ tal que $y = f(x)$;
- (ii) para todo 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 todo 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:

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

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:

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

□

A *fibra de y através de f* é o subconjunto de X denotado por $f^*(y)$ e dado por:

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

Aqui chama-se de *geradora de fibras através de f* , a função f^* de $f(X)$ em $\mathcal{P}(X)$ (o conjunto de todas partes de X), dada por:

$$f^* : y \mapsto f^*(y). \tag{A.2}$$

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

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

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

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. \square

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

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

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

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

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

\square