



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

urlib.net/www/2014/03.25.23.20-RPQ

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

Gerald Jean Francis Banon

URL do documento original:

[<http://urlib.net/J8LNKB5R7W/3G2EKR5>](http://urlib.net/J8LNKB5R7W/3G2EKR5)

INPE
São José dos Campos
2017

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GBDIR)

Serviço de Informação e Documentação (SESID)

CEP 12.227-010

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/7348

E-mail: pubtc@inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE - CEPPII (PORTARIA Nº 176/2018/SEI-INPE):**Presidente:**

Dr. Marley Cavalcante de Lima Moscati - Centro de Previsão de Tempo e Estudos Climáticos (CGCPT)

Membros:

Dra. Carina Barros Mello - Coordenação de Laboratórios Associados (COCTE)

Dr. Alisson Dal Lago - Coordenação-Geral de Ciências Espaciais e Atmosféricas (CGCEA)

Dr. Evandro Albiach Branco - Centro de Ciência do Sistema Terrestre (COCST)

Dr. Evandro Marconi Rocco - Coordenação-Geral de Engenharia e Tecnologia Espacial (CGETE)

Dr. Hermann Johann Heinrich Kux - Coordenação-Geral de Observação da Terra (CGOBT)

Dra. Ieda Del Arco Sanches - Conselho de Pós-Graduação - (CPG)

Silvia Castro Marcelino - Serviço de Informação e Documentação (SESID)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon

Clayton Martins Pereira - Serviço de Informação e Documentação (SESID)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Simone Angélica Del Ducca Barbedo - Serviço de Informação e Documentação (SESID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SESID)

EDITORAÇÃO ELETRÔNICA:

Ivone Martins - Serviço de Informação e Documentação (SESID)

Cauê Silva Fróes - Serviço de Informação e Documentação (SESID)



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

urlib.net/www/2014/03.25.23.20-RPQ

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

Gerald Jean Francis Banon

URL do documento original:

[<http://urlib.net/J8LNKB5R7W/3G2EKR5>](http://urlib.net/J8LNKB5R7W/3G2EKR5)

INPE
São José dos Campos
2017

Dados Internacionais de Catalogação na Publicação (CIP)

Banon, Gerald Jean Francis.

B227i Identificador com base na Internet (IBI): Sistema de resolução / Gerald Jean Francis Banon. – São José dos Campos : INPE, 2017.
76 p. ; (urlib.net/www/2014/03.25.23.20-RPQ)

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

CDU 004.451.53

*“Ce que l’on conçoit bien s’énonce clairement,
Et les mots pour le dire viennent aisément”.*

NICOLAS BOILEAU

em “*L’Art poétique*”, 1674

(*Alexandrin* ensinado ao autor por seu pai RENÉ PAUL BANON)

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), em particular, a Eduardo Whitaker Bergamini (INPE), responsável e animador da Comissão, os quais acompanharam os progressos deste trabalho ao longo dos anos de 2013 a 2015. Somos particularmente gratos ao Eduardo W. Bergamini pelo incentivo recebido, para que, com base no presente relatório, fosse montada uma nova proposta de normalização junto à ABNT.

Agradecemos também a toda comunidade do INPE que, a partir de 1995, compôs os primeiros usuários do sistema de resolução de IBI, e em particular, à Marciana Leite Ribeiro do Serviço de Informação e Documentação (SID) do INPE pelo incentivo na construção de uma biblioteca digital que serviu de plataforma de teste para o desenvolvimento desse sistema.

Finalmente, gostaríamos de agradecer a Benício Pereira de Carvalho Filho (INPE), por ter autorizado em 2007 a hospedagem no INPE do resolvidor `urlib.net` e a Gabriela Paola Ribeiro Banon pela sugestões de melhoria do texto, tanto em termos de clareza, quanto na revisão da língua portuguesa.

RESUMO

Este relatório complementa um relatório anterior intitulado “Identificador com base na Internet (IBI): Sistema de identificação”. O primeiro relatório apresentou duas formas em que um identificador global denominado IBI pode ser criado para identificar 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. Este segundo relatório introduz um sistema de resolução de URL persistentes para prover um acesso consistente e perene aos itens de informação, uma vez identificados por IBI. O modo de armazenamento dos itens de informação em Arquivos, que privilegia os conceitos de original e cópias, é descrito formalmente. O relatório ainda apresenta uma minilinguagem para definir “modificadores” de IBI que estendem os acessos a itens de informação relacionados como traduções, última edição ou metadados. Além dos navegadores utilizados pelos usuários, o sistema de resolução é constituído dos Arquivos que armazenam os itens de informação e de um resolvidor que intermedia a comunicação entre os navegadores e os Arquivos. A ênfase neste relatório é dada à concepção dos serviços de Arquivo voltados para a resolução de IBI, trazendo recomendações de como esses serviços devem ser construídos.

INTERNET BASED IDENTIFIER - DRAFT STANDARD

ABSTRACT

This report complements an earlier report entitled “Identificador com base na Internet (IBI): Sistema de identificação” (Internet based identifier (IBI): identification system). The first report presented two ways in which a global identifier called IBI can be created to identify different types of information items (documents, maps, images, etc.) stored in collections like those found in digital repositories, Archives, or elsewhere. This second report introduces a system for the resolution of persistent URL to provide a consistent and enduring access to information items once identified by IBI. The storage model of the information items in Archive which privileges the concepts of original and copies, is formally described. The report also presents a mini-language to define IBI “modifier” that extend the access to related information items like translations, the latest edition or metadata. Besides the browsers used by the users, the resolution system consists of the Archives that store the information items and a resolver that facilitates the communication between browsers and Archives. The emphasis in this report is given to the design of the Archive services aimed at the IBI resolution, bringing recommendations of how these services should be built.

LISTA DE FIGURAS

	<u>Pág.</u>
3.1 Componentes e atores do Sistema para resolução de IBI	7
3.2 Diagrama de casos de uso do Sistema para resolução de IBI	8
3.3 Diagrama de classes	10
3.4 Classe ItemIdentificado como classe-mãe	11
5.1 Diagrama de sequência do processo de inclusão/exclusão de Arquivo . .	21
6.1 Diagrama de sequência do processo de resolução de IBI	33

LISTA DE TABELAS

	<u>Pág.</u>
4.1 Codificação hexadecimal de caracteres especiais.	16
4.2 Regras definindo a formação de uma lista-de-pares	18
5.1 Regras definindo a formação de uma chave de cadastro	19
5.2 Pares <nome>=<valor> a serem anexados à URL de base para as mensagens “Incluir Arquivo” (2) e “Excluir Arquivo” (7), endereçadas ao resolvedor por um Arquivo	22
6.1 Regras definindo a formação de uma URL-persistente	26
6.2 Modificadores de URL persistente	27
6.3 Pares <nome>=<valor> da URL persistente a serem interpretados pelo sistema para resolução de IBI	30
6.4 Regras definindo a formação de uma lista-de-verbos	30
6.5 Verbos atuando como modificador elementar de URL persistente	31
6.6 Dados capturados pelo resolvedor ao receber uma URL persistente	34
6.7 Dados sobre o usuário.	35
6.8 Dados extraídos da URL persistente	35
6.9 Pares <nome>=<valor> a serem anexados à URL de base para a mensagem “Solicitar URL” (3) endereçada aos Arquivos pelo resolvedor	36
6.10 Regras definindo a formação de uma lista-de-IP	37
6.11 Pares <nome>=<valor> a serem anexados à URL de base para a mensagem “Agradecer” (5) endereçada ao Arquivo pelo resolvedor	38
7.1 Componentes do Algoritmo 1 (MONTARRESPOSTA).	42
7.2 Componentes do Algoritmo 2 (MONTARLISTADEURL).	49
7.3 Descrição da função < x, i >.	52
7.4 Regras definindo a formação de formas-de-ibi	53
7.5 Regras definindo a formação de uma relação	54
A.1 Regras definindo a formação de idioma e país	75

SUMÁRIO

	<u>Pág.</u>
1 Introdução	1
2 Terminologia	5
3 Descrição do sistema para resolução de IBI	7
3.1 Componentes e atores do sistema	7
3.2 Itens originais e cópias	9
4 Protocolo de comunicação	15
4.1 Formato de solicitação de serviço	15
4.2 Formato de resposta a uma solicitação de serviço	17
5 Inclusão e exclusão de Arquivo	19
5.1 Cadastrar um Arquivo	19
5.2 Incluir/excluir um Arquivo	20
6 Resolução de IBI	25
6.1 URL persistente	25
6.2 Processo de resolução	32
7 Serviço de Arquivo	41
7.1 Solicitações	41
7.2 Respostas	41
8 Exemplos de comunicação entre o resolvedor e os Arquivos . . .	67
8.1 Exemplo sem modificador de IBI	67
8.2 Exemplo com modificador de IBI	68
REFERÊNCIAS BIBLIOGRÁFICAS	73
APÊNDICE A - Regras idioma e país	75

1 Introdução

Objetivo

Num universo de objetos (chamado aqui de itens ou itens de informação), existe sempre o problema de como encontrá-los. Uma solução consiste em identificá-los (associando univocamente a cada um deles um nome ou rótulo) e manter junto a cada identificador de um item a informação de sua localização no universo.

Num relatório anterior intitulado “Identificador com base na Internet (IBI): Sistema de identificação” (BANON, 2011), foi apresentado as regras gramaticais e os algoritmos para criar o IBI de um item, isto é, torná-lo um item identificado.

O presente relatório expõe as recomendações sobre um sistema de resolução de IBI que associa univocamente a cada IBI a localização do item identificado. Considerando os itens de informação acessíveis via Internet, o sistema de resolução retornará cópias digitais dos itens identificados. Neste aspecto, os sistemas de resolução e de identificação de IBI são mutualmente inversos.

O modo de armazenamento dos itens de informação em Arquivos, que privilegia os conceitos de original e cópias, é descrito formalmente.

O protocolo de comunicação entre o resolvidor, que recebe as solicitações de resolução de IBI, e os Arquivos que armazenam os itens de informação é apresentado, assim como os serviços desempenhados por estes.

O relatório ainda apresenta uma minilinguagem para definir “modificadores” de IBI que estendem os acessos à item de informações relacionados como traduções, última edição ou metadados

Justificativa

Parte-se do princípio que 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 o seu funcionamento preservado por longo prazo.

A solução para tornar os ponteiros persistentes consiste no uso combinado de um sistema de identificação global e um sistema de resolução associado.

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 a itens de informação, o problema da construção de ponteiros persistentes pode ser solucionado com o uso de um sistema de resolução de identificador. O sistema de resolução deve ter o propósito básico de redirecionar cada URL persistente, 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 em [Banon \(2011\)](#) junto ao sistema de resolução descrito neste relatório, ambos extensivamente utilizados na plataforma *URLib*, o primeiro desde 1995 e o segundo desde 1998, apresenta-se como uma alternativa simples, quando comparadas a outras soluções como as usadas, por exemplo, no PURL ([WEIBEL et al., 2015](#)) ou no Handle System[®] ([CNRI, 2015](#)).

Na fase de identificação, a principal diferença do sistema, quando comparado ao PURL ou ao Handle System[®], é que, enquanto estes têm um sistema próprio de cadastro de seus usuários, o sistema de identificação por IBI reaproveita indiretamente o sistema já existente de cadastro dos atores da Internet detentores de nomes de domínio (*fully qualified domain names*), portanto, sem custo adicional, nesse aspecto.

Por distinguir entre os conceitos de original e cópias, o modo de armazenamento dos itens de informação em Arquivos do sistema para resolução de IBI, permite, mesmo em presença de cópias, autenticar *a posteriori* a posse de obras inéditas depositadas em Arquivos desse sistema, isto é, reconhecer *a posteriori* quem é o dono ou detentor dos direitos patrimoniais. Isto porque, em caso de um original ser reivindicado por mais de um dono ou detentor¹, o sistema para resolução de IBI emite um alerta informando que existem dois ou mais Arquivos sob suspeita e que uma investigação é necessária para reconhecer o verdadeiro dono ou detentor (cf. Seção 6.2).

Quanto à autoria de uma obra inédita depositada em um Arquivo deste sistema, o modo de armazenamento permite que ela seja atestada *a posteriori* pelo dono do Arquivo que armazenou sua versão original.

Finalmente, o uso de “modificadores” de IBI permite estabelecer, a partir de um

¹O dono ou detentor de um original é o dono do Arquivo onde ele está armazenado.

mesmo IBI, acessos persistentes a itens de informação futuros, como novas traduções ou uma nova edição de um item de informação já existente.

2 Terminologia

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

Arquivo: (com A maiúscula para diferenciar de arquivo de computador) componente do **sistema para resolução de IBI** que armazena **itens identificados**.

Chave de cadastro: código de segurança próprio a um **Arquivo**, informado ao solicitar seu cadastro no **sistema para resolução de IBI**, usado nas comunicações entre o **Arquivo** e o **resolvedor**.

Chave de URL: código de segurança criado pelo **Arquivo** ao retornar a URL de acesso ao **item de informação** apontado por uma **URL persistente**, usado na comunicação entre o **resolvedor** e o **Arquivo**.

Estado: atributo que define o estado de um **item identificado**, o qual assume o valor “Original”, caso seja tido como original, ou “Copy”, caso seja considerado cópia, ou ainda “Deleted” caso tenha sido removido.

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

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

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

Item: qualquer objeto a ser identificado.

Item de informação: qualquer **item**, consistindo exclusivamente em dados digitais. Por exemplo: documentos, mapas, imagens, programas de computador, serviço web, etc. no formato digital.

Item identificado: qualquer **item** identificado por **IBI**.

Modo de armazenamento: maneira como estão armazenados os **itens identificados**.

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

Resolver: componente do **sistema para resolução de IBI** que recebe as **URL persistentes** e as redireciona para as URL que contêm as localizações físicas dos respectivos **itens de informação**.

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

Serviço de Arquivo: serviço *Web* oferecido por um **Arquivo** ao **sistema para resolução de IBI** que, em cada **Arquivo**, é único, intransferível e identificado por um **IBI**.

Serviço de resolver: serviço *Web* oferecido pelo **resolver** ao **sistema para resolução de IBI** e identificado pelo **IBI** J8LNKB5R7W/3FUQHC5.

Sistema de identificação: qualquer função injetora entre um conjunto de **itens** e um conjunto de **rótulos**, que associa a cada **item** o **identificador deste item**.

Sistema de resolução: qualquer função injetora entre um conjunto de **rótulos** (usados como **identificadores de itens**) e um conjunto de **itens**, que associa a cada **rótulo** o **item identificado** por este **rótulo**.

Sistema para geração de IBI: **sistema de identificação** objeto do relatório por [Banon \(2011\)](#). Inverso do **sistema para resolução de IBI**.

Sistema para resolução de IBI: **sistema de resolução** objeto deste relatório. Inverso do **sistema para geração de IBI**.

URL: sigla em inglês para *Uniform Resource Locator*.

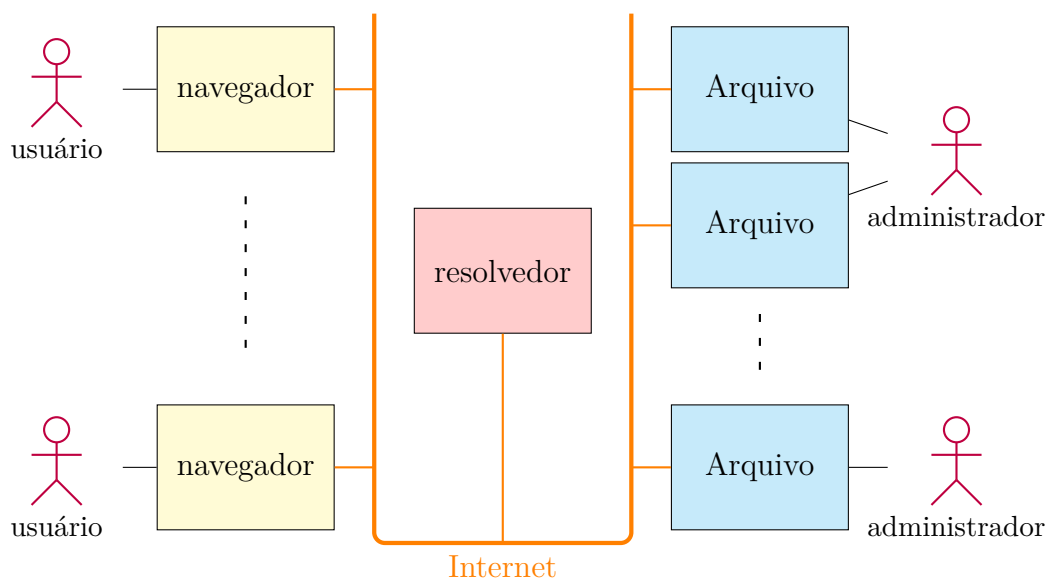
URL persistente: URL que dá acesso a um **item de informação**, independentemente de sua localização na *Web*.

3 Descrição do sistema para resolução de IBI

3.1 Componentes e atores do sistema

O **sistema para resolução de IBI** é composto por três tipos de componente: os navegadores, os **Arquivos** e o **resolvedor**, interligados entre si por meio da Internet; e dois tipos de atores: os usuários e os administradores, como mostrado na Figura 3.1.

Figura 3.1 - Componentes e atores do **Sistema para resolução de IBI**



Os navegadores, o resolvedor e os Arquivos são interligados por meio da Internet. Dois ou mais **Arquivos** podem ter o mesmo administrador.

Fonte: Produção do autor.

Os navegadores são as interfaces entre os usuários e os outros componentes do **sistema para resolução de IBI**.

Os **Arquivos** (com A maiúscula para diferenciar de arquivo de computador) são as entidades que armazenam os **itens identificados**, isto é, os **itens de informação** a serem encontrados com base nos seus **IBI**. Os **Arquivos** estão sob a responsabilidade do seus administradores.

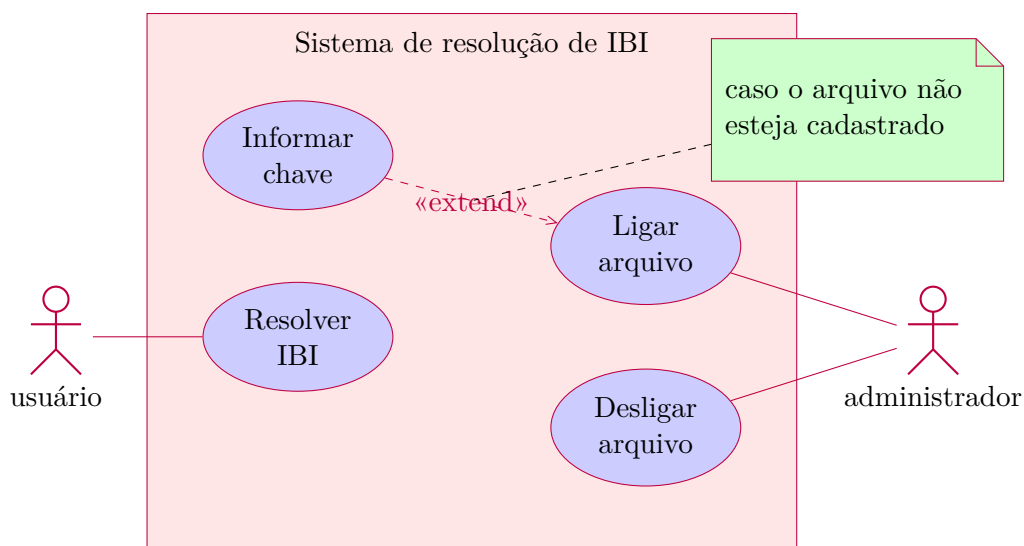
O **resolvedor** é a entidade que apenas intermedeia a comunicação entre os navegadores e os **Arquivos**. O **resolvedor** pode ou não armazenar a informação sobre a

localização, na *Web*, dos **itens de informação**. No entanto, esta informação deve estar armazenada nos respectivos **Arquivos**.

Os atores do **sistema para resolução de IBI** são os usuários do sistema e os administradores dos **Arquivos**.

A Figura 3.2 apresenta o **sistema para resolução de IBI** e seus atores por meio de um diagrama de casos de uso UML (GUEDES, 2011)¹.

Figura 3.2 - Diagrama de casos de uso do **Sistema para resolução de IBI**



Os quatro casos de uso e seus atores.

Fonte: Produção do autor.

Os usuários do sistema são os atores que solicitam a resolução de **IBI**.

Os administradores de **Arquivos** são os atores que os ligam no momento de participar do **sistema para resolução de IBI** e os desligam em caso de manutenção ou de afastamento definitivo do sistema. Um administrador pode ser responsável por mais de um **Arquivo**.

Se o **Arquivo** ainda não estiver cadastrado no **sistema para resolução de IBI**, antes de ligar o **Arquivo**, o seu administrador deverá informar uma chave de ca-

¹Nos diagramas UML deixa-se de usar o A maiúsculo na palavra **Arquivo**, reservando esta notação para representar a classe dos arquivos.

dastró (cf. detalhes na Seção 5.1).

A hospedagem dos Arquivos em servidores distintos permite que administradores independentes participem do processo de resolução.

3.2 Itens originais e cópias

Um Arquivo pode armazenar um número qualquer de itens identificados, mas não pode armazenar dois itens identificados com o mesmo IBI.

Nos Arquivos, distingue-se dois tipos de item identificado: os originais e as cópias.

Como qualquer **item**, os **itens de informação** podem ser reproduzidos por meio de cópias. No **sistema para resolução de IBI**, os **itens identificados**, considerados cópias de um determinado **item identificado** tido como original, mantêm o **IBI** do original. Além disso, enquanto o original deve pertencer a (ou estar armazenado em) apenas um único **Arquivo**, eventuais cópias de um mesmo original podem pertencer a (ou estar armazenadas em) mais de um **Arquivo**.

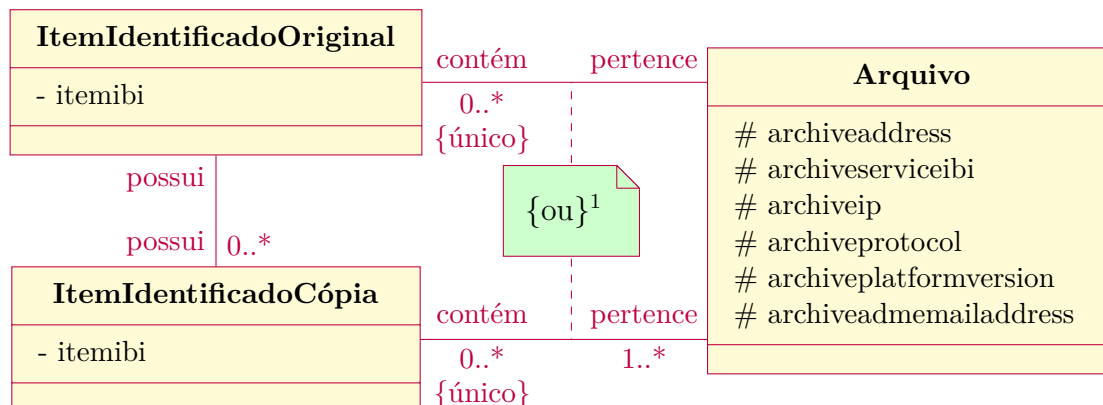
Essa configuração está ilustrada na Figura 3.3 na forma de um diagrama de classes UML (GUEDES, 2011), na qual se destacam três classes e suas associações. **ItemIdentificadoOriginal** é a classe dos **itens identificados** tidos como originais; **ItemIdentificadoCópia**, a classe dos **itens identificados** considerados cópias de alguns originais; e **Arquivo**, a classe dos **Arquivos**.

A associação “pertence”, que liga as classes **ItemIdentificadoOriginal** e **Arquivo**, indica que cada **item identificado** tido como original pertence a um único **Arquivo**. Quanto à associação “contém”, ela indica que cada **Arquivo** pode conter zero ou mais **itens identificados** tidos como originais, desde que apenas um seja identificado pelo mesmo **IBI** (cf. a primeira restrição {único} na Figura 3.3).

A associação “pertence”, que liga as classes **ItemIdentificadoCópia** e **Arquivo**, indica que cada **item identificado** considerado cópia pertence a um ou mais **Arquivos**. Quanto à associação “contém”, ela indica que cada **Arquivo** pode conter zero ou mais **itens identificados** considerados cópias, desde que não haja dois **itens identificados** com o mesmo **IBI** (cf. a segunda restrição {único} na Figura 3.3).

A nota em verde, na Figura 3.3, indica que, se um **Arquivo** contiver um **item identificado** tido como original, então ele não poderá conter um **item identificado** considerado cópia e possuindo o mesmo **IBI**, e vice-versa.

Figura 3.3 - Diagrama de classes



As três classes e suas associações.

¹A restrição “ou” (exclusivo) é relativa à associação “contém” e aplica-se somente aos **itens de informação** possuindo o mesmo **IBI**.

Fonte: Produção do autor.

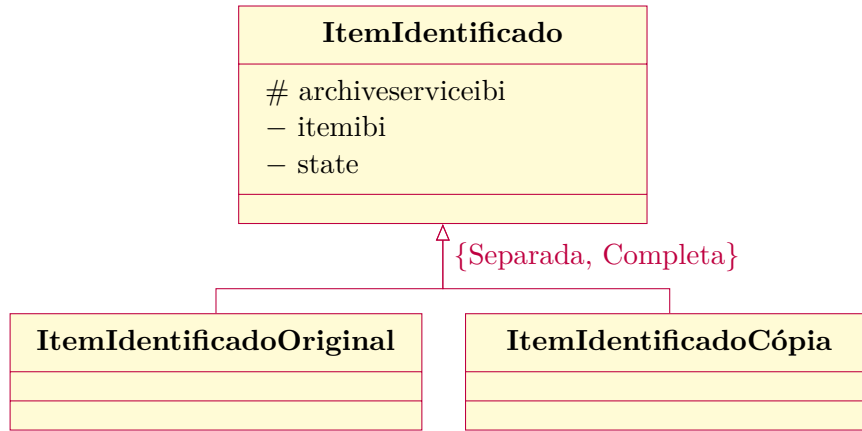
Em consequência disto, as associações “possui” entre as classes **ItemIdentificadoOriginal** e **ItemIdentificadoCópia** indicam que um **item identificado** tido como original pode possuir zero ou mais **itens identificados** considerados cópias, e que cada **item identificado** considerado cópia possui um (e apenas um) **item identificado** tido como original (cf. Proposição 3.2 sobre esta última asserção).

Um **item identificado** deve ser considerado somente original ou cópia (completude)², mas não pode ser considerado as duas coisas juntas (separabilidade). Com estas propriedades, tem-se o conceito de classe-mãe **ItemIdentificado**, como mostrado na Figura 3.4. Nesta classe, aparecem o atributo “archiveserviceibi”, que especifica em que **Arquivo** está armazenado um determinado **item identificado**, o atributo “itemibi”, que informa o **IBI** do **item identificado**, e o atributo “state”, que assumirá o valor “Original”, se o **item identificado** for tido como original, ou o valor “Copy”, se for considerado cópia.

Os conceitos apresentados acima em termo de classes, descrevendo como são armazenados os **itens identificados**, podem ser transpostos em termos matemáticos. Os **itens identificados**, armazenados em **Arquivos** e em determinados **estados**, são caracterizados pelos seus **IBI** (“itemibi”), pelos **Arquivos** onde estão armazenados

²Desconsideram-se aqui os **itens identificados** tido como originais que foram removidos (*deleted*).

Figura 3.4 - Classe **ItemIdentificado** como classe-mãe



Os **itens identificados** da classe-mãe diferenciam-se uns dos outros por meio de três atributos. O atributo “archiveserviceibi” fornece o **IBI** do **Serviço de Arquivo** do **Arquivo** onde o **item identificado** está armazenado; o atributo “itemibi” informa seu **IBI**; e o atributo “state” especifica seu **estado**.

Fonte: Produção do autor.

(“archiveserviceibi”) e por seus **estados** (“state”), os quais satisfazem uma relação terceira como apresentado na definição a seguir.

Definição 3.1 (modo de armazenamento) – Considera-se os conjuntos a seguir.

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

Seja A o conjunto dos **Arquivos** que armazenam os **itens identificados**.

Seja S o conjunto $\{\text{Original}, \text{Copy}\}$ de **estados** definindo a genuinidade dos **itens identificados**.

O **modo de armazenamento** é a relação terceira R definida como o quadrupletto (I, A, S, G) onde G (o grafo de R) satisfaz as seguintes propriedades:

1. Para todo $i \in I$, a_1 e $a_2 \in A$ tem-se:

$$(i, a_1, \text{Original}) \in G \text{ e } (i, a_2, \text{Original}) \in G \Rightarrow a_1 = a_2$$

isto é, o original de um **item identificado** está armazenado em um único arquivo.

2. Para todo $i \in I$, existe um $a \in A$ tal que $(i, a, \text{Original}) \in G$

isto é, há sempre um **Arquivo** para armazenar qualquer **item identificado** tido como original.

$(i, a, s) \in G$ é lido “ i está armazenado em a como s ”.

□

Com base na Definição 3.1, pode-se enunciar uma proposição estabelecendo uma propriedade da relação binária R_{Original} definida como tripleto $(I, A, G_{\text{Original}})$, cujo grafo é o conjunto $G_{\text{Original}} \triangleq \{(i, a) \in I \times A : (i, a, \text{Original}) \in G\}$.

Proposição 3.1 (propriedades da relação entre originais e Arquivos) – A relação binária R_{Original} é uma relação:

1. Funcional, isto é, para todo $i \in I$, a_1 e $a_2 \in A$ tem-se:

$$(i, a_1) \in G_{\text{Original}} \text{ e } (i, a_2) \in G_{\text{Original}} \Rightarrow a_1 = a_2;$$

2. Esquerda-total, isto é, para todo $i \in I$, existe um $a \in A$ tal que $(i, a) \in G_{\text{Original}}$.

□

Prova. A proposição é uma consequência direta das Propriedades 1 e 2 da Definição 3.1, observando que para todo $i \in I$ e $a \in A$ tem-se:

$$(i, a, \text{Original}) \in G \Leftrightarrow (i, a) \in G_{\text{Original}}$$

□

Em outros termos, o grafo de R_{Original} é um grafo-de-um-mapeamento (BANON, 2000): para todo $i \in I$, existe um único $a \in A$ tal que $(i, a) \in G_{\text{Original}}$. Isto é, todo **item identificado** tido como original está armazenado em um único **Arquivo**.

Com base na Proposição 3.1 pode-se estabelecer propriedades importantes de uma relação entre cópias e originais.

Seja R_{Copy} a relação binária definida como tripleto (I, A, G_{Copy}) , cujo grafo é o conjunto $G_{\text{Copy}} \triangleq \{(i, a) \in I \times A : (i, a, \text{Copy}) \in G\}$. Então, com base na Proposição 3.1, pode-se enunciar mais uma proposição estabelecendo uma propriedade da relação binária $\mathcal{R}$ definida como o tripleto $(G_{\text{Copy}}, G_{\text{Original}}, \mathcal{G})$, cujo grafo $\mathcal{G}$ é, por sua

vez, definido, para todo i e $j \in I$ e a e $b \in A$, por:

$$((i, a), (j, b)) \in \mathcal{G} \Leftrightarrow i = j$$

Proposição 3.2 (propriedades da relação entre cópias e originais) – A relação binária $\mathcal{R}$ é uma relação funcional e esquerda-total. $\square$

Prova. 1. Para todo $(i, a) \in G_{\text{Copy}}$, $(j_1, b_1) \in G_{\text{Original}}$ e $(j_2, b_2) \in G_{\text{Original}}$, tem-se:

$$\begin{aligned} ((i, a), (j_1, b_1)) \text{ e } ((i, a), (j_2, b_2)) \in \mathcal{G} &\Rightarrow j_1 = i && \text{(Def. de } \mathcal{G}) \\ &i = j_2 && \text{(Def. de } \mathcal{G}) \\ &b_1 = b_2 \quad (j_1 = j_2 \text{ e } G_{\text{Original}} \text{ é funcional),} \end{aligned}$$

isto é, $((i, a), (j_1, b_1)) \text{ e } ((i, a), (j_2, b_2)) \in \mathcal{G} \Rightarrow (j_1, b_1) = (j_2, b_2)$.

Em outras palavras, $\mathcal{R}$ é funcional.

2. Para todo $(i, a) \in G_{\text{Copy}}$, existe um $b \in A$ tal que $(i, b) \in G_{\text{Original}}$ (pois, R_{Original} é esquerda-total), isto é, pela definição de $\mathcal{G}$, $((i, a), (i, b)) \in \mathcal{G}$. Ou seja, para todo $(i, a) \in G_{\text{Copy}}$, existe $(j, b) \in G_{\text{Original}}$ (basta escolher $j = i$) tal que $((i, a), (j, b)) \in \mathcal{G}$. Em outras palavras, $\mathcal{R}$ é esquerda-total também. $\square$

Assim pela Proposição 3.2, cada **item identificado** armazenado como cópia possui um único **item identificado** armazenado como original (i.e., não há cópia sem original e este é único).

Finalmente, os **itens identificados** tidos como originais são divididos em transferíveis e intransferíveis. No primeiro caso, eles poderão ser transferidos (migrados) de um **Arquivo** para outro, enquanto os intransferíveis deverão permanecer nos **Arquivos** em que foram originalmente armazenados. O **item identificado** denominado **serviço de Arquivo** das duas próximas seções é um exemplo de **item identificado** intransferível.

4 Protocolo de comunicação

4.1 Formato de solicitação de serviço

O funcionamento do **sistema para resolução de IBI** depende de serviços executados tanto pelos **Arquivos** quanto pelo **resolvedor**, assim como da comunicação entre o **resolvedor** de um lado e os **Arquivos** do outro.

Os serviços executados pelos **Arquivos** são denominados “serviços de **Arquivo** para resolução de **IBI**” ou simplesmente **serviços de Arquivo**. Cada **Arquivo** deve ter um **serviço de Arquivo** identificado pelo seu **IBI**. O serviço executado pelo **resolvedor** é denominado **serviço de resolvedor**.

A comunicação entre os componentes do **sistema para resolução de IBI** (cf. Seção 3.1) deve ocorrer na forma de mensagens HTTP (*Hypertext Transfer Protocol*).

Toda solicitação HTTP deve utilizar o método GET. Dessa forma, qualquer mensagem de solicitação torna-se uma URL.

A URL de base das mensagens de solicitação de serviço endereçadas ao **resolvedor** deve ser:

`http://urlib.net/J8LNKB5R7W/3FUQHC5`

onde `urlib.net` é o endereço Web (nome de domínio) do resolvedor e `J8LNKB5R7W/3FUQHC5` é o **IBI** do **serviço de resolvedor**.

A URL de base das mensagens de solicitação de serviço endereçadas a um **Arquivo** deve ser do tipo:

`http://<archiveaddress>/<archiveserviceibi>`

onde o valor `<archiveaddress>` do componente `archiveaddress` é o endereço Web do **Arquivo**, e o valor `<archiveserviceibi>` do componente `archiveserviceibi` é o **IBI** do **serviço de Arquivo**, informado no momento de cadastrar o **Arquivo** (cf. Seção 5.1).

A formação de `<archiveaddress>` deve seguir a regra gramatical denominada `hostport` na Seção 3.2.2 intitulada *Server-based Naming Authority* em [Berners-Lee et al. \(1998\)](#), em que as regras `host` e `port` são as definidas, respectivamente, nas Seções 3.2.2 e 3.2.3 em [Berners-Lee et al. \(2005\)](#).

Por sua vez, a formação de `<archiveserviceibi>` deve seguir as regras gramaticais `repositório` ou `IBIp` para um **IBI**, como definidas em [Banon \(2011\)](#).

Além da URL de base, toda mensagem de solicitação de serviço consiste em uma lista de pares `<nome>=<valor>` que deve ser anexada à URL de base e separada dela por `?`. Múltiplos pares podem aparecer em qualquer ordem e são separados por `&`.

Em todo tipo de solicitação deve constar o par `servicesubject=<valor>`, em que `<valor>` é o assunto do serviço solicitado. Quando existirem, os outros pares definem os parâmetros da solicitação.

Os componentes `valor` dos pares `<nome>=<valor>` devem consistir em sequências de caracteres codificados no sistema ASCII (do Inglês *American Standard Code for Information Interchange*) ([Comissão-de-Estudos ABNT/CB08/SC010/CE70, 2010](#)). Alguns caracteres considerados especiais nos componentes `valor` devem ser codificados, como mostrado na Tabela 4.1. Os caracteres especiais são aqueles reservados para delimitar a URL de base dos pares anexados a esta (?), como também para delimitar os pares entre si (&) e os constituintes destes pares entre si (=), assim como criar espaço em branco (*espaço em branco* e +) ou codificar no sistema hexadecimal caracteres que não sejam ASCII, usando o prefixo percentagem (%).

Tabela 4.1 - Codificação hexadecimal de caracteres especiais.

Caractere	Codificação
<i>espaço em branco</i>	%20
%	%25
&	%26
+	%2B
=	%3D
?	%3F

Por exemplo, o valor `1997-07-16T19:20+01:00` contendo o caractere especial `+` deve ser codificado: `1997-07-16T19:20%2B01:00`.

Exemplo 4.1 (mensagem de solicitação de serviço) – URL:

```
http://mtc-m16d.sid.inpe.br/sid.inpe.br/mtc-m19@80/2009/08.21.17.02?
servicesubject=urlRequest&
clientinformation.ipaddress=150.163.2.175&
parsedibiurl.ibi=8JMKD3MGP7W/3EPGUE5
```

□

No Exemplo 4.1, a URL é a mensagem de solicitação de serviço endereçada ao **Arquivo** cujo endereço Web é `mtc-m16d.sid.inpe.br`, sendo `sid.inpe.br/mtc-m19@80/2009/08.21.17.02` o IBI do seu **serviço de Arquivo**. Anexada à URL de base, encontra-se uma lista de três pares `<nome>=<valor>`, sendo que o par `servicesubject=urlRequest` define o assunto do serviço solicitado (aqui `urlRequest`), e os pares `clientinformation.ipaddress=150.163.2.175` e `parsedibiurl.ibi=8JMKD3MGP7W/3EPGUE5` definem os parâmetros da solicitação.

4.2 Formato de resposta a uma solicitação de serviço

Toda resposta a uma solicitação HTTP feita para um serviço do **sistema para resolução de IBI** deve ser gerada com o `Content-Type` assumindo o valor `text/plain`.

Por sua vez, todo conteúdo deve ter o formato de uma “lista de pares”, cuja formação siga as regras gramaticais definidas na Tabela 4.2. Nesta tabela, as regras gramaticais são especificadas por meio do uso da notação BNF – *Backus Normal Form* ou *Backus-Naur Form* – (aumentada) (CROCKER, 1982; CROCKER; OVERELL, 2008)¹.

As regras em maiúsculas SP (espaço em branco) e CRLF (nova linha) são as mesmas que em Crocker e Overell (1997).

A regra `lista-de-pares` da Tabela 4.2 define uma lista formada por uma alternância de `nome` e de `valor`. Um `nome` e um `valor` consecutivos estão separados entre si por pelo menos um espaço em branco (SP), e formam juntos um `par`. Os pares entre si estão separados por `espaço`. O `nome` deve ser uma `palavra`, isto é, uma concatenação de `caractere`, observa-se que a regra `caractere` não contempla o espaço em branco (SP). Por sua vez, o `valor` deve ser uma `palavra` ou uma sequência dessas separadas por um espaço em branco (SP), neste último caso o `valor` deve começar com `{`, e terminar com `}`.

Pela regra `caractere` da Tabela 4.2, observa-se que os caracteres da “lista de pares” são os codificados no sistema ASCII (do Inglês *American Standard Code for Information Interchange*) (Comissão-de-Estudos ABNT/CB08/SC010/CE70, 2010) e, em consequência disto, serão os únicos caracteres da resposta a uma solicitação de qualquer serviço do **sistema para resolução de IBI**.

Se for necessário usar caracteres não codificados no sistema ASCII, cada um destes deverá ser representado por um par de algarismos hexadecimais, precedido pelo

¹Neste relatório, a notação BNF é usada com a seguinte alteração: “|” é utilizado para representar alternativas no lugar de “/”.

Tabela 4.2 - Regras definindo a formação de uma lista-de-pares

```

lista-de-pares = par *(espaço par); ex: a1 {a2 a3} a4 a5
    par = nome 1*SP valor; ex: a1 {a2 a3}
        nome = palavra; ex: a1
        valor = palavra | ("{" lista "}"); ex: {a2 a3}
        lista = palavra 1*(SP palavra);
                lista de palavras, ex: a1 a2
    espaço = SP | CRLF
        SP = %x20; espaço em branco
        CRLF = CR LF
        CR = %x0D; retorno de carro
        LF = %x0A; nova linha
    palavra = 1*caractere; palavra sem espaço em branco, ex: a1
    caractere = %x21-7A | %x7C | %x7E; caracteres ASCII menos { e }

```

símbolo %, sendo que cada par corresponde ao número decimal entre 0 e 255, usado na codificação do caractere, no sistema de codificação adotado. Por exemplo, o caractere ó, parte do alfabeto Latino, codificado no sistema da ISO/IEC 8859-1 pelo valor decimal 243, será representado por %f3.

Exemplo 4.2 (mensagem de resposta à uma solicitação de serviço) – Lista de pares:

```

archiveaddress      mtc-m16d.sid.inpe.br
contenttype         Data
ibi                 {rep sid.inpe.br/mtc-m19/2013/09.04.12.27.57 ibip 8JMKD3MGP7W/3EPGUE5}
ibi.archiveservice  {rep sid.inpe.br/mtc-m19080/2009/08.21.17.02}
ibi.platformsoftware {rep dpi.inpe.br/banon/1998/08.02.08.56}
state               Original
timestamp           2013-10-04T14:32:14Z
url                 http://mtc-m16d.sid.inpe.br/col/sid.inpe.br/mtc-m19/2013/09.04.12.27.57/doc/Relat%f3rio%20Final.pdf
urlkey              1426203276-5985125171467764

```

□

No Exemplo 4.2, a “lista de pares” é a resposta à mensagem de solicitação do Exemplo 4.1. Esta lista é constituída por pares nome/valor, dispostos ao longo de 9 linhas (um em cada linha), sendo o nome e o valor de cada par separados por espaços em branco. Observa-se que, na URL (oitavo par), o caractere ó, parte da palavra **Relatório**, por não fazer parte dos caracteres codificados em ASCII, é representado pelo par de algarismos hexadecimais %f3, sendo o sistema da ISO/IEC 8859-1, o adotado no momento da criação do nome de arquivo “**Relatório Final.pdf**”, e que o espaço em branco nesse nome é representado pelo par de algarismos hexadecimais %20, conforme a Tabela 4.1.

5 Inclusão e exclusão de Arquivo

5.1 Cadastrar um Arquivo

Do ponto de vista do funcionamento do **sistema para resolução de IBI**, o papel do administrador de um **Arquivo** resume-se em ligá-lo e desligá-lo (cf. Seção 3.1). No entanto, para que o **Arquivo** possa, ao ligá-lo, passar a ser incluído no **sistema para resolução de IBI** (cf. Seção 5.2), o administrador deve inicialmente cadastrá-lo junto a este sistema. Por motivo de segurança no processo de inclusão/exclusão de um **Arquivo**, o administrador deverá informar uma **chave de cadastro** de sua escolha no momento do cadastramento (cf. a nota em verde na Figura 3.2).

A Tabela 5.1 contém, conforme a notação BNF (aumentada), as três regras para a formação da **chave de cadastro**. Observa-se que a chave consiste simplesmente em uma, ou opcionalmente duas, sequência(s) de 10 ou mais dígitos.

Tabela 5.1 - Regras definindo a formação de uma **chave de cadastro**

```
chave = número ["-" número]; ex: 1234567890-1234567890
número = 10*DÍGITO; ex: 1234567890
DÍGITO = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

O cadastramento de um **Arquivo** é realizado com base numa solicitação de cadastro feita pelo seu administrador por meio de um *e-mail* endereçado a uma entidade responsável pelo cadastro de **Arquivos** que desejam participar do **sistema para resolução de IBI**.

A solicitação de cadastro de um **Arquivo** deve conter a identificação do Arquivo e sua **chave de cadastro**.

O identificador do Arquivo deve ser o **IBI** (BANON, 2011) do **item identificado**, tido como original intransferível, denominado “serviço de **Arquivo** para resolução de **IBI**”, ou simplesmente **serviço de Arquivo**, previamente criado e armazenado neste **Arquivo** (cf. Seção 7).

Exemplo 5.1 (*e-mail* de solicitação de cadastro de um **Arquivo**) – Assunto:

Solicitação de cadastro de um novo Arquivo

Conteúdo:

IBI do serviço de Arquivo:
archiveserviceibi = dpi.inpe.br/banon/1999/01.09.22.14

Chave de cadastro:
registrationkey = 1234567890

□

No Exemplo 5.1, o conteúdo do *e-mail* informou que o administrador do **Arquivo**, com o **serviço de Arquivo** identificado pelo **IBI** dpi.inpe.br/banon/1999/01.09.22.14, solicitou seu cadastramento no **sistema para resolução de IBI**, com a **chave de cadastro** constituída pela sequência de dígitos: 1234567890.

5.2 Incluir/excluir um Arquivo

Uma vez o **Arquivo** cadastrado no **sistema para resolução de IBI**, seu administrador receberá um *e-mail* de confirmação. Então, o **Arquivo** que implementar esta norma passará a ser automaticamente incluído no **sistema para resolução de IBI** no momento em que for ligado pelo seu administrador.

Ligar um **Arquivo** significa expô-lo na *Web* e, em particular, expor seu **serviço de Arquivo**. Ao desligar o **Arquivo**, isto é, deixar de expô-lo na *Web*, este será automaticamente excluído do **sistema para resolução de IBI**, no entanto, seu cadastro será mantido para poder ser incluído novamente quando religado.

No momento de ser religado, o endereço *Web* do **Arquivo**, isto é, o componente **archiveaddress** introduzido na Seção 4.1, poderá assumir, ou não, um novo valor.

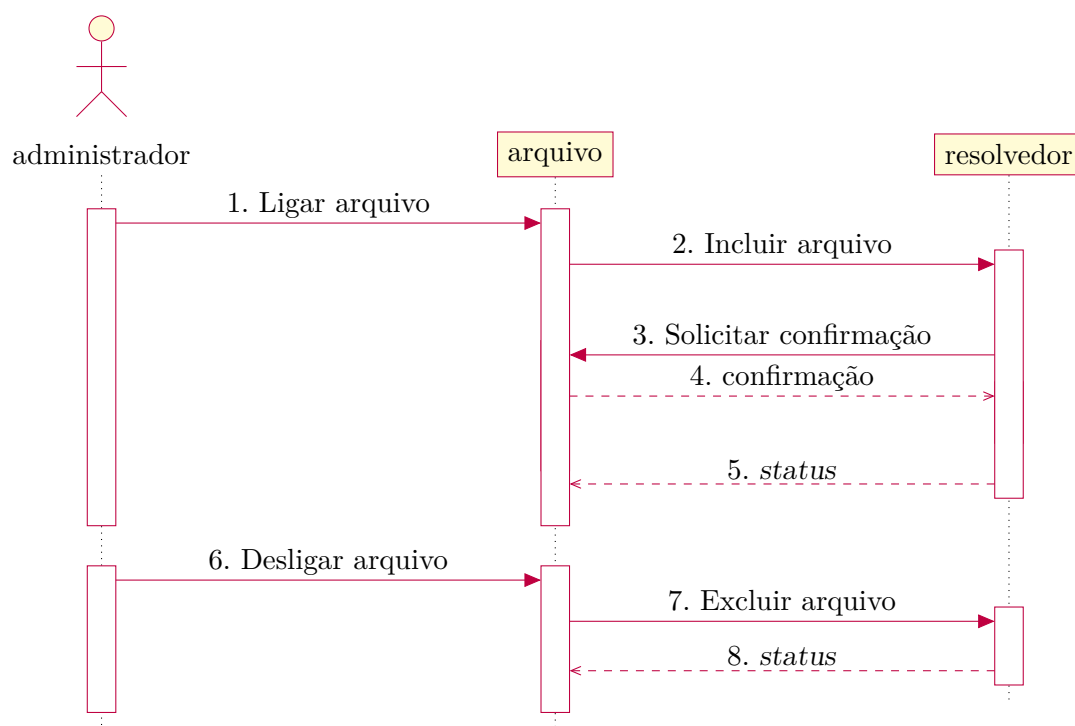
O processo de inclusão/exclusão de **Arquivo** no/do **sistema para resolução de IBI** é detalhado no diagrama de sequência da Figura 5.1. Este processo é desencadeado pela emissão da mensagem “Ligar arquivo” (1), enviada pelo administrador ao **Arquivo**, e termina pela emissão da mensagem “Desligar arquivo” (6).

Ao ser ligado, o **Arquivo** deverá emitir a mensagem “Incluir Arquivo” (2) endereçada ao **resolvedor**. Da mesma forma, ao ser desligado, o **Arquivo** deverá emitir a mensagem “Excluir Arquivo” (7).

Em uma solicitação de inclusão, o par **servicesubject=inclusionRequest** deve ser anexado à URL de base; em uma solicitação de exclusão, é o par **servicesubject=exclusionRequest** que deve ser anexado.

Junto a este par, uma solicitação de inclusão ou exclusão deve conter mais outros

Figura 5.1 - Diagrama de sequência do processo de inclusão/exclusão de **Arquivo**



Ao receber de um **Arquivo** a mensagem “Incluir arquivo” (2), o **resolvedor** envia a mensagem “Solicitar confirmação” (3) a este mesmo **Arquivo** que deve retornar uma “confirmação” (4); conforme a presença ou ausência de resposta do **Arquivo**, o **resolvedor** retorna uma mensagem “status” (5) indicando que o **Arquivo** foi incluído e se a mensagem de confirmação foi ou não recebida. Ao receber de um **Arquivo** a mensagem “Excluir arquivo” (7), o **resolvedor** retorna uma mensagem “status” (8) indicando que o **Arquivo** foi excluído.

Fonte: Produção do autor.

sete pares, como mostrado na Tabela 5.2. Nesta tabela, a coluna *Domínio* indica o domínio dos valores possíveis para cada par. Os valores são aqueles que podem ser gerados a partir de uma gramática especificada por meio da notação BNF (aumentada). A coluna *Tipo* indica que todos os oito pares são obrigatórios (obr.).

Tabela 5.2 - Pares <nome>=<valor> a serem anexados à URL de base para as mensagens “Incluir Arquivo” (2) e “Excluir Arquivo” (7), endereçadas ao **resolvedor** por um **Arquivo**.

Par			Tipo
Nome	Valor		
	Domínio	Significado	
servicesubject	"inclusionRequest"	em caso de solicitação de inclusão	obr.
	"exclusionRequest"	em caso de solicitação de exclusão	
archiveaddress	regra <code>hostport</code> da Seção 4.1	o endereço <i>Web</i> do Arquivo ex: <code>mtc-m21.sid.inpe.br</code>	obr.
archiveserviceibi	regra <code>repositório</code> ou <code>IBIp</code> em Banon (2011)	o IBI do serviço de Arquivo informado no momento de cadastrar o Arquivo (cf. Seção 5.1)	obr.
archiveip	regra <code>IPv4address</code> ou <code>IPv6address</code> da Seção 3.2.2 em Berners-Lee et al. (2005)	o IP do servidor hospedando o Arquivo ex: <code>150.163.34.239</code>	obr.
archiveprotocol	"HTTP"	o protocolo de comunicação	obr.
archiveplatformversion	sequência ASCII conforme Seção 4.1	a versão da plataforma computacional hospedando o Arquivo ex: <code>2014:11.09.02.16.15</code>	obr.
archiveadmemailaddress	regra <code>authority</code> da Seção 3.2 em Berners-Lee et al. (2005)	o endereço de <i>e-mail</i> do administrador do Arquivo ex: <code>banon@dpi.inpe.br</code>	obr.
registrationkey	regra <code>chave</code> da Tabela 5.1	a chave de cadastro do Arquivo informado no momento de cadastrá-lo (cf. Seção 5.1) ex: <code>1234567890</code>	obr.

Neste relatório, apenas um único protocolo de comunicação está sendo considerado: o protocolo HTTP. Por este motivo, o par com nome `archiveprotocol` deve assumir o único valor permitido HTTP. No futuro, haverá a possibilidade de recomendar outros protocolos. Por exemplo, atualmente está sendo usado, porém ainda não divulgado, o protocolo USP para “*URLibService Protocol*”.

Exemplo 5.2 (mensagem de solicitação de inclusão emitida pelo **Arquivo** ao ser ligado pelo seu administrador) – URL de solicitação de inclusão:

```
http://www.urlib.net/J8LNKB5R7W/3FUQHC5?
servicesubject=inclusionRequest&
archiveaddress=mtc-m21.sid.inpe.br&
archiveserviceibi=sid.inpe.br/mtc-m21/2012/06.05.15.34.39&
archiveip=150.163.34.239&
archiveprotocol=HTTP&
archiveplatformversion=2014:11.09.02.16.15&
archiveadmemailaddress=banon@dpi.inpe.br&
registrationkey=1234567890
```

□

Exemplo 5.3 (mensagem de solicitação de exclusão emitida pelo **Arquivo** ao ser desligado pelo seu administrador) – URL de solicitação de exclusão:

```
http://www.urlib.net/J8LNKB5R7W/3FUQHC5?
servicesubject=exclusionRequest&
archiveaddress=mtc-m21.sid.inpe.br&
archiveserviceibi=sid.inpe.br/mtc-m21/2012/06.05.15.34.39&
archiveip=150.163.34.239&
archiveprotocol=HTTP&
archiveplatformversion=2014:11.09.02.16.15&
archiveadmemailaddress=banon@dpi.inpe.br&
registrationkey=1234567890
```

□

Nos Exemplos 5.2 e 5.3, as URL foram, respectivamente, as mensagens “Incluir Arquivo” (2) e “Excluir Arquivo” (7) endereçadas ao **resolvedor** pelo **Arquivo** de endereço web `mtc-m21.sid.inpe.br` e de **serviço de Arquivo** identificado pelo **IBI** `sid.inpe.br/mtc-m21/2012/06.05.15.34.39`, solicitando, respectivamente, sua inclusão e exclusão. Apenas o valor de `servicesubject` esteve alterado na segunda mensagem em relação a primeira (cf. textos em vermelho).

Ao receber de um **Arquivo** a mensagem “Incluir Arquivo” (2), o **resolvedor** verificará se o **Arquivo** é realmente acessível na *Web* por meio do endereço informado `<archiveaddress>`. Nem sempre o acesso é possível, como, por exemplo, em alguns casos de IP dinâmico.

Para cumprir essa verificação, o **resolvedor** enviará a este mesmo **Arquivo** a mensagem “Solicitar confirmação” (3), na expectativa de receber uma resposta de confirmação.

Apenas o par `servicesubject=inclusionConfirmationRequest` é anexado à URL de base em uma solicitação de confirmação.

Exemplo 5.4 (mensagem de solicitação de confirmação emitida pelo **resolvedor**)
– URL de solicitação de confirmação:

`http://mtc-m21.sid.inpe.br/sid.inpe.br/mtc-m21/2012/06.05.15.34.39?
servicesubject=inclusionConfirmationRequest`

□

No Exemplo 5.4, a URL é a mensagem “Solicitar confirmação” (3) enviada pelo **resolvedor** ao **Arquivo** cujo endereço *Web* é `mtc-m21.sid.inpe.br`, sendo `sid.inpe.br/mtc-m21/2012/06.05.15.34.39` o **IBI** do seu **serviço de Arquivo**.

O valor da mensagem “confirmação” (4) deve ser constante e igual a “`confirmation yes`” (cf. Seção 7).

Ao receber a mensagem “Incluir Arquivo” (2), o **resolvedor** incluirá o **Arquivo** no **sistema para resolução de IBI**, se este já estiver cadastrado e se a chave de cadastro informada na mensagem “Incluir Arquivo” (2) coincidir com a chave informada no momento do cadastro. Além disto, se o **resolvedor** receber também a mensagem “confirmação” (4), ele retornará ao **Arquivo** a mensagem “*status*” (5) com o valor:

`“status.archive included status.confirmation successful”,`

caso contrário, o **resolvedor** retornará ao **Arquivo** a mensagem “*status*” (5) com o valor:

`“status.archive included status.confirmation unsuccessful”.`

Ao receber a mensagem “Excluir Arquivo” (7), se a chave de cadastro informada nesta mensagem coincidir com a chave informada no momento do cadastro, o **resolvedor** excluirá o **Arquivo** do **sistema para resolução de IBI** e retornará ao **Arquivo** a mensagem “*status*” (5) com o valor:

`“status.archive excluded”.`

6 Resolução de IBI

6.1 URL persistente

O papel do **sistema para resolução de IBI** é atender aos usuários do sistema que solicitam a resolução de **IBI** com o intuito de receber de volta os **itens identificados** correspondentes.

A solicitação de resolução de **IBI** é codificada na forma de uma **URL persistente**, contendo o **IBI** do **item identificado** de interesse.

A forma simples de uma **URL persistente** é:

`http://urllib.net/<ibi>`

em que `http` é o esquema (protocolo), `urllib.net` é o endereço *Web* (nome de domínio) do **resolvedor** do **sistema para resolução de IBI** e `<ibi>` é o **IBI** do **item identificado** de interesse, isto é, o **IBI** a ser resolvido.

Exemplo 6.1 (URL persistentes) – URL:

`http://urllib.net/8JMKD3MGP8W/35MME4E`

`http://urllib.net/8JMKD3MGP8W/35MMLL8`

`http://urllib.net/LK47B6W/362SFKH`

`http://urllib.net/iconet.com.br/banon/2009/09.09.22.01`

□

Nas três primeiras **URL persistentes** do Exemplo 6.1, `8JMKD3MGP8W/35MME4E`, `8JMKD3MGP8W/35MMLL8` e `LK47B6W/362SFKH` são **IBI** na forma opaca de **IBIp** (BANON, 2011). Na quarta **URL persistente**, o **IBI** `iconet.com.br/banon/2009/09.09.22.01` está na forma de **nome de repositório uniforme** (BANON, 2011).

No exemplo 6.1, as três primeiras URL apontam para três **itens identificados** distintos. A terceira e quarta URL apontam para o mesmo **item identificado**, isto é, `LK47B6W/362SFKH` e `iconet.com.br/banon/2009/09.09.22.01` constituem um exemplo das duas formas do **IBI** de um mesmo **item identificado**.

A forma completa de uma **URL persistente** é:

`http://urlib.net/<ibi><mdf><caminho>?<pergunta>`

em que **mdf** é um componente opcional, chamado de modificador de **URL persistente**; **caminho** é um componente opcional que especifica um recurso (e.g., arquivo) de interesse, parte do **item identificado**; e **pergunta** é um componente opcional que contém parâmetros de funcionalidade e que deve ser interpretados pelo **sistema para resolução de IBI** ou pelo próprio **item identificado** (quando este é um serviço *Web* por exemplo).

A implementação dos recursos necessários para atender a qualquer forma não simples de uma **URL persistente** é facultativa por parte dos **Arquivos** (cf. implementação mínima na Seção 7.2).

Em termos de linguagem formal, uma **URL persistente** é um caso particular de URI (BERNERS-LEE et al., 2005). A Tabela 6.1 especifica as regras de formação de uma **URL-persistente**. A especificação utiliza a notação BNF (aumentada) e as seguintes regras predefinidas: **repositório** (**IBI** como **nome de repositório uniforme**), **IBIp** (**IBI** como **IBIp**), **ALFAMI** (letras minúsculas) e **ALFAMA** (letras maiúsculas), introduzidas em Banon (2011), assim como a regra **path-absolute** (caminho de arquivo começando com "/"), encontrada em Berners-Lee et al. (2005), e as regras idioma e país da Tabela A.1 do Apêndice A.

Tabela 6.1 - Regras definindo a formação de uma **URL-persistente**

```
URL-persistente = "http://urlib.net/" IBI [mdf] [path-absolute] ["?" pergunta]
  IBI           = repositório | IBIp; ex: LK47B6W/362SFKH
  mdf           = ([u [t]] | [t [u]]) [m [t]]; modificador, ex: !+:
    u           = "!"; última edição
    t           = "+" ["(" idioma ["-" país] ")"]; tradução, ex: +(pt-BR)
    m           = ":" ["(" formato ")"]; metadados, ex: :(oai_dc)
  formato       = "oai_dc"
  pergunta      = [par ["&" par]]
    par         = nome "=" valor
    nome        = 1*caractere
    valor       = *caractere
  caractere     = %x21-25 | %x27-3C | %x3E-7E; caracteres ASCII menos & e =
```

Um modificador de **URL persistente** é um operador que modifica o processo de resolução de **IBI**, de forma a retornar um **item de informação** que possui um certo relacionamento com o **item identificado**. Este operador é construído por meio da

composição de operadores escolhidos dentre de três operadores (ou modificadores) elementares.

O primeiro operador elementar, denotado pelo símbolo *dois pontos* (:), retorna metadados do **item identificado**. O segundo, denotado pelo símbolo *ponto de exclamação* (!), retorna a última edição do **item identificado**. O terceiro, denotado pelo símbolo *mais* (+), retorna uma dada tradução do **item identificado**.

Na Tabela 6.1, os três modificadores elementares estão especificados por meio das regras **m** (para metadados), **u** (para última edição) e **t** (para tradução).

O modificador que retorna metadados (:) e o que retorna uma tradução (+) podem ser parametrizados.

A Tabela 6.2 descreve todas as composições permitidas de operadores escolhidos dentre dos três modificadores elementares de **URL persistente**.

Tabela 6.2 - Modificadores de **URL persistente**.

<i>Modificador</i>	<i>Retorno</i>
:	metadados
:+	tradução dos metadados
!	última edição
!+	tradução da última edição
!:	metadados da última edição
!+:	metadados da tradução da última edição
!:+	tradução dos metadados da última edição
!+:+	tradução dos metadados da tradução da última edição
+	tradução
+!	última edição da tradução
+:	metadados da tradução
+:!	metadados da última edição da tradução
+:+	tradução dos metadados da tradução
+:+!	tradução dos metadados da última edição da tradução

Sem parâmetros, o primeiro modificador, *dois pontos* (:), retorna os metadados em formato livre; parametrizado, na forma : (oai_dc), retorna os metadados no formato *oai_dc* (LAGOZE et al., 2002).

Exemplo 6.2 (URL persistentes apontando para metadados) – URL:

<http://urlib.net/8JMKD3MGP8W/35MME4E:>

[http://urlib.net/8JMKD3MGP8W/35MME4E:\(oai_dc\)](http://urlib.net/8JMKD3MGP8W/35MME4E:(oai_dc))

□

Ao acrescentar o modificador *dois pontos* (:) à primeira **URL persistente** do Exemplo 6.1, tem-se as URL do Exemplo 6.2. Uma **URL persistente** como a primeira deste exemplo, deve retornar em formato livre os metadados do **item identificado** (aqui, no exemplo, com **IBI 8JMKD3MGP8W/35MME4E**). Uma **URL persistente** como a segunda do exemplo, em que o modificador *dois pontos* (:) está parametrizado na forma `:(oai_dc)`, deve retornar os metadados no formato *oai_dc*.

Sem parâmetros, o segundo modificador, *mais* (+), retorna a tradução no idioma de preferência selecionado pelo usuário em seu navegador; parametrizado, na forma `+(xx-XX)`, retorna a tradução no idioma definido por `xx-XX`.

Exemplo 6.3 (**URL persistentes** apontando para uma tradução) – URL:

<http://urlib.net/8JMKD3MGP8W/35MME4E+>

[http://urlib.net/8JMKD3MGP8W/35MME4E+\(pt\)](http://urlib.net/8JMKD3MGP8W/35MME4E+(pt))

□

Ao acrescentar o modificador *mais* (+) à primeira **URL persistente** do Exemplo 6.1, que aponta para um **item de informação** escrito em inglês, tem-se as URL do Exemplo 6.3. Se o idioma de preferência selecionado pelo usuário em seu navegador for o inglês, então uma **URL persistente** como a primeira deste exemplo, deve retornar o próprio **item identificado** (aqui, no exemplo, com **IBI 8JMKD3MGP8W/35MME4E**), pois este está escrito em inglês. Se o idioma de preferência selecionado pelo usuário em seu navegador for o português, então uma **URL persistente** como esta, deve retornar, se existir, a tradução em português do **item identificado** (aqui, no exemplo, com **IBI 8JMKD3MGP8W/35MME4E**), senão deve retornar simplesmente a versão do item de informação no seu idioma original.

Uma **URL persistente** como a segunda do exemplo, em que o modificador *mais* (+) está parametrizado na forma `+(pt)`, deve retornar, se existir, a tradução em português do **item identificado** (aqui, no exemplo, com **IBI 8JMKD3MGP8W/35MME4E**), independentemente do idioma de preferência selecionado no navegador. Se não exis-

tir, a URL persistente deve retornar um aviso informando que a tradução não existe.

No caso do Exemplo 6.3, por extensão de linguagem, é aceita o uso do modificador +(en) para fazer referência também ao item de informação escrito em inglês, apesar deste não ser uma tradução propriamente dita, pois trata-se da versão original em inglês.

O terceiro modificador, *ponto de exclamação* (!), retorna a última edição.

Exemplo 6.4 (URL persistentes com um ou mais modificadores elementares) – URL:

<http://urlib.net/8JMKD3MGP8W/35MMLL8!>

[http://urlib.net/8JMKD3MGP8W/35MMLL8!:\(oai_dc\)](http://urlib.net/8JMKD3MGP8W/35MMLL8!:(oai_dc))

□

Ao acrescentar o modificador *ponto de exclamação* (!) à segunda **URL persistente** do Exemplo 6.1, tem-se as URL do Exemplo 6.4. Uma **URL persistente** como a primeira deste exemplo, deve retornar a última edição (aqui, no exemplo, publicada em 2012, ou posterior) do **item identificado** (aqui, no exemplo, publicado em 2002 e com **IBI** 8JMKD3MGP8W/35MMLL8).

Ao acrescentar o modificador *dois pontos* com parâmetro *oai_dc* (: (oai_dc)) à primeira **URL persistente** desse exemplo, tem-se a segunda URL. Uma **URL persistente** como esta, deve retornar no formato *oai_dc* os metadados da última edição do **item identificado** (aqui, no exemplo, com **IBI** 8JMKD3MGP8W/35MMLL8).

Pela regra *pergunta* da Tabela 6.1, o valor do componente *pergunta* da **URL persistente** deve ser uma lista de pares <nome>=<valor>. Por sua vez, alguns caracteres considerados especiais nos componentes *valor* devem ser codificados, como mostrado na Tabela 4.1.

Quando o par deve ser interpretado pelo **sistema para resolução de IBI**, o componente *nome* deve se decompor em um prefixo e um sufixo separados por *um ponto* (.), e o prefixo deve ser *ibiurl* (o prefixo *ibiurl* é para lembrar **URL persistente**). Caso contrário, o par será ignorado por este sistema. Os dois pares <nome>=<valor>, atualmente interpretados pelo **sistema para resolução de IBI**, são mostrados na Tabela 6.3. Nela, a coluna *Tipo* indica que os dois pares são facultativos (fac.).

Tabela 6.3 - Pares <nome>=<valor> da URL persistente a serem interpretados pelo sistema para resolução de IBI.

Par			Tipo
Nome	Valor		
	Domínio	Significado	
ibiurl.requireditemstatus	"Original"	solicitação do original	fac.
ibiurl.verblast	regra lista-de-verbos da Tabela 6.4	modificador de URL persistente ex: GetLastEdition	fac.

O primeiro par com nome `ibiurl.requireditemstatus` assume o valor constante `Original` e é usado para solicitar o original de um **item de informação**.

Essa opção é útil toda vez que se queira acessar por segurança o original em vez de uma possível cópia, lembrando que, por exemplo, a manutenção de um **item de informação** deve ser feita no original, e a manutenção das cópias deve consistir apenas em copiar o original, o que pode não ocorrer imediatamente após o término da manutenção do original.

O segundo par com nome `ibiurl.verblast` assume um valor verificando a regra gramatical `lista-de-verbos`. Como mostrado na Tabela 6.4, a `lista-de-verbos` é formada a partir dos quatro verbos: `GetMetadata`, `GetLastEdition`, `GetTranslation` e `GetFileList`. Estes atuam como modificadores elementares da **URL persistente**.

Tabela 6.4 - Regras definindo a formação de uma `lista-de-verbos`

```

lista-de-verbos = [verbo *("+" verbo)]; ex: GetLastEdition+GetMetadata
verbo          = ("GetTranslation" ["(" idioma ["-" país] ")"]) |
                  "GetLastEdition" | ("GetMetadata" ["(" formato ")"]) |
                  "GetFileList"
idioma         = 2ALFAMI; ex: pt
país           = 2ALFAMA; ex: BR
formato        = "oai_dc"

```

Como mostrado na Tabela 6.5, os três primeiros verbos possuem sinônimos, que são os modificadores elementares de **URL persistente** já introduzidos nesta seção. Na prática, estes três verbos atuam da mesma forma que seus sinônimos.

Exemplo 6.5 (**URL persistente** modificada por uma lista de verbos) – URL:

Tabela 6.5 - Verbos atuando como modificador elementar de **URL persistente**.

<i>Verbo</i>	<i>Sinônimo</i>	<i>Retorno</i>
<code>GetMetadata</code>	:	metadados
<code>GetLastEdition</code>	!	última edição
<code>GetTranslation</code>	+	tradução
<code>GetFileList</code>		lista de arquivos

[http://urlib.net/8JMKD3MGP8W/35MMLL8?
ibiurl.verblast=GetLastEdition+GetMetadata\(oai_dc\)](http://urlib.net/8JMKD3MGP8W/35MMLL8?ibiurl.verblast=GetLastEdition+GetMetadata(oai_dc))

□

Ao acrescentar à segunda **URL persistente**, apresentada no Exemplo 6.1, o par `ibiurl.verblast=GetLastEdition+GetMetadata(oai_dc)`, tem-se a **URL persistente** do Exemplo 6.5. Uma **URL persistente** como a deste exemplo, deve retornar no formato *oai_dc* os metadados da última edição do **item identificado** (aqui, neste exemplo, com **IBI** 8JMKD3MGP8W/35MMLL8). Esta **URL persistente** é equivalente à segunda URL do Exemplo 6.4 que faz uso do modificador !: (*oai_dc*).

Enquanto a posição dos três primeiros verbos (`GetMetadata`, `GetLastEdition` e `GetTranslation`) na lista de verbos `ibiurl.verblast` é pertinente, a posição do quarto verbo (`GetFileList`) é indiferente quanto ao resultado do retorno.

Uma **URL persistente** modificada pelo verbo `GetFileList` deve retornar a lista de recursos (arquivos) que compõe o **item de informação** apontado pela mesma **URL persistente** porém sem esse verbo. A lista de recursos (arquivos) pode ser gerada em formato livre.

Exemplo 6.6 (**URL persistente** modificada retornando uma lista de recursos) – URL:

<http://urlib.net/LK47B6W/362SFKH?ibiurl.verblast=GetFileList>

□

Ao acrescentar à terceira **URL persistente**, apresentada no Exemplo 6.1, o par `ibiurl.verblast=GetFileList`, tem-se a URL do Exemplo 6.6. Uma **URL persistente** como a deste exemplo, deve retornar a lista dos recursos (arquivos) que compõem o **item identificado** (aqui, neste exemplo, com **IBI** LK47B6W/362SFKH).

Finalmente, em uma **URL persistente**, o componente **caminho** permite especificar um determinado recurso (arquivo) que compõe um certo **item de informação**.

Exemplo 6.7 (**URL persistente** retornando um recurso específico) – URL:

<http://urllib.net/LK47B6W/362SFKH/reference.bib>

□

Ao acrescentar à terceira **URL persistente**, apresentada no Exemplo 6.1, o caminho `/reference.bib`, tem-se a URL do Exemplo 6.7. Uma **URL persistente** como a deste exemplo, deve retornar apenas o conteúdo do arquivo especificado (aqui, neste exemplo, `/reference.bib`) que compõe o **item identificado** (aqui, neste exemplo, com **IBI** LK47B6W/362SFKH).

Para merecer o qualificativo de “persistent” uma URL persistente apresentando o componente **caminho**, deve apontar para um recurso (arquivo) cujo nome `<caminho>` se mantenha continuamente inalterado.

6.2 Processo de resolução

O processo de resolução de **IBI** é detalhado no diagrama de sequência da Figura 6.1.

```
sequenceDiagram
    actor U as usuário
    participant N as navegador
    participant R as resolvidor
    participant A as arquivo:Arquivo

    U->>N: 1. Resolver IBI
    Note over N: repetir o loop para solicitar a URL de um eventual novo IBI
    N->>R: 2. Resolver IBI
    loop [p/ cada arq.]
        R->>A: 3. Solicitar URL
        A-->>R: 4. lista de URL
    end
    opt [URL existe]
        R->>A: 5. Agradecer
        A-->>R: 6. aviso
    end
    R-->>N: 7. URL/alerta
    opt [URL existe]
        R->>A: 8. Trazer item de informação
        A-->>R: 9. item de informação
    end
    N-->>U: 10. item/alerta
```

Fonte: Produção do autor.

33

após digitar no campo de endereço do seu navegador a **URL persistente**, clica no botão de envio da URL. Esta emissão pode ainda ocorrer quando o usuário clica num ponteiro persistente¹, parte de um hipertexto.

Em consequência da ativação de uma **URL persistente** pelo usuário, o navegador emite a mensagem “Resolver IBI” (2) endereçada ao **resolvedor** `urlib.net` por meio do protocolo HTTP, reproduzindo a própria **URL persistente** de solicitação de resolução do **IBI** escolhido pelo usuário.

Ao receber a mensagem “Resolver IBI” (2) na forma de uma **URL persistente**, o **resolvedor** captura dados sobre o usuário e dados extraídos da **URL persistente**, como apresentado na Tabela 6.6.

Tabela 6.6 - Dados capturados pelo **resolvedor** ao receber uma **URL persistente**.

<i>Nome</i>	<i>Significado</i>	<i>Repassado</i>
<code>clientinformation.ipaddress</code>	IP do usuário	sim
<code>clientinformation.languagepreference</code>	preferência de idioma do usuário	não
<code>parsedibiurl.filepath</code>	caminho até o recurso (arquivo)	sim
<code>parsedibiurl.ibi</code>	IBI objeto de resolução	sim
<code>parsedibiurl.requireditemstatus</code>	valor solicitado para o estado do item de informação	não
<code>parsedibiurl.verblast</code>	lista de verbos	sim

Os valores das variáveis `parsedibiurl.filepath`, `parsedibiurl.ibi`, `parsedibiurl.requireditemstatus` e `parsedibiurl.verblast` são os valores extraídos da **URL persistente**. Precisamente, o valor de `parsedibiurl.filepath` é o valor do componente `caminho`; o valor de `parsedibiurl.ibi`, o valor do componente `ibi`; o valor de `parsedibiurl.requireditemstatus`, o valor do par com nome `ibiurl.requireditemstatus` do componente `pergunta`; e o valor de `parsedibiurl.verblast`, a união (sem repetição e sem modificação de ordem) dos verbos cujos sinônimos constam do componente `mdf` e dos verbos que constam no par com nome `ibiurl.verblast` do componente `pergunta`.

Exemplo 6.8 (Dados capturados pelo **resolvedor** ao receber uma **URL persis-**

¹Um ponteiro persistente é um ponteiro cujo valor é uma **URL persistente**.

tente) – **URL persistente**:

```
http://urllib.net/LK47B6W/362SFKH?  
ibiurl.requireditemstatus=Original&  
ibiurl.verblast=GetMetadata
```

Ao receber a solicitação de resolução desta URL persistente, o revolvedor capturou os dados da Tabela 6.7 sobre o usuário (navegador) que emitiu esta solicitação.

Tabela 6.7 - Dados sobre o usuário.

<i>Nome</i>	<i>Valor</i>
<code>clientinformation.ipaddress</code>	150.163.2.175
<code>clientinformation.languagepreference</code>	pt-br,fr;q=0.8,en;q=0.5,pt;q=0.3

□

A Tabela 6.7 do Exemplo 6.8 mostra os dois pares nome/valor que informaram sobre o usuário, e a Tabela 6.8, os três que foram extraídos da **URL persistente** exemplificada.

Tabela 6.8 - Dados extraídos da **URL persistente**.

<i>Nome</i>	<i>Valor</i>
<code>parsedibiurl.filepath</code>	(indefinido)
<code>parsedibiurl.ibi</code>	LK47B6W/362SFKH
<code>parsedibiurl.requireditemstatus</code>	Original
<code>parsedibiurl.verblast</code>	GetTranslation GetMetadata

Na etapa seguinte, somente parte dos dados da Tabela 6.6, como indicado na coluna três, será repassada aos **Arquivos**.

Por razões de privacidade, a preferência de idioma que consta no navegador do usuário (`clientinformation.languagepreference`) não está sendo repassada aos **Arquivos**; esta é tratada somente pelo **resolvedor**.

Por razões de simplicidade na concepção do **serviço de Arquivo** e de segurança do funcionamento do **sistema para resolução de IBI**, o *status* do **item de informação** solicitado (`parsedibiurl.requireditemstatus`), também não está sendo

repassado aos **Arquivos**. Quando o *status* de original é solicitado, um **Arquivo** mal-intencionado, que informar irregularmente estar hospedando um original do **item de informação**, poderá ser descoberto pelo **resolvedor** e, por isto, os **Arquivos** não podem tomar conhecimento dessa solicitação.

Em seguida, então, o **resolvedor** endereça a mensagem “Solicitar URL” (3) a todos os **Arquivos** cadastrados e incluídos no **sistema para resolução de IBI** (cf. Seção 5.2). Essa solicitação é feita conforme o formato de solicitação de serviço da Seção 4.1, anexando à URL de base todos os pares, ou parte deles, apresentados na Tabela 6.9.

Tabela 6.9 - Pares <nome>=<valor> a serem anexados à URL de base para a mensagem “Solicitar URL” (3) endereçada aos **Arquivos** pelo **resolvedor**.

Par			Tipo
Nome	Valor		
	Domínio	Significado	
servicesubject	"urlRequest"	solicitação de URL	obr.
clientinformation.ipaddress	regra lista-de-IP da tabela 6.10	endereço IP do usuário solicitando a resolução de um IBI ex: 150.163.2.175	obr.
parsedibiurl.filepath	regra path-absolute em Berners-Lee et al. (2005)	caminho do recurso (arquivo) ex: /reference.bib	fac.
parsedibiurl.ibi	regra repositório ou IBIP em Banon (2011)	IBI objeto de resolução ex: LK47B6W/362SFKH	obr.
parsedibiurl.verblast	regra lista-de-verbos da Tabela 6.4	modificador de URL persistente	fac.

A Tabela 6.10 especifica as regras de formação da regra `lista-de-IP` usada na definição do domínio dos valores do par com nome `clientinformation.ipaddress` da Tabela 6.9. A especificação utiliza a notação BNF (aumentada) e as regras pre-definidas: IPv4 e IPv6, introduzidas em Berners-Lee et al. (2005). Na lista de IP, o primeiro elemento é o IP do usuário. Caso existirem outros elementos, estes são os IP dos seus sucessivos procuradores (*proxy servers*)².

²Em algumas situações, o primeiro elemento da lista de IP pode não ser do usuário, mas sim o do seu procurador final.

Tabela 6.10 - Regras definindo a formação de uma `lista-de-IP`

```
lista-de-IP = IP *(SP IP); ex: 172.16.44.200 150.163.68.1
IP         = IPv4 | IPv6
```

Um exemplo de mensagem do tipo “Solicitar URL” (3) já foi dada no Exemplo 4.1.

Em resposta à solicitação do **resolvedor**, cada **Arquivo** retorna uma “lista de URL” (4) (eventualmente vazia), conforme descrito na Seção 7.

Ao receber as respostas dos **Arquivos**, o **resolvedor** executa a análise delas.

Se a variável `parsedibiurl.requireditemstatus` existir (seu valor será **Original**), então o **resolvedor** analisará todas as respostas que satisfazem a mensagem “Solicitar URL” (3). Se todas as respostas não vazias contiverem a URL para o **item de informação** solicitado, então o **resolvedor** verificará se não existem duas ou mais respostas reivindicando o **item de informação** como original. Se houver uma única resposta reivindicando o **item de informação** como original, então o **resolvedor** selecionará essa resposta, caso contrário, emitirá um alerta informando que existem dois ou mais **Arquivos** sob suspeita e que uma investigação é necessária.

Em todos os outros casos, o **resolvedor** selecionará a primeira resposta não vazia que satisfaz a mensagem “Solicitar URL” (3). Nesses casos, se a resposta não contiver ainda a URL para o **item de informação** solicitado, então o **resolvedor** endereçará uma nova mensagem “Solicitar URL” (3) a todos os **Arquivos** (cf. a nota em verde ligado ao fragmento *loop* da Figura 6.1). Nessa nova mensagem, o valor do par com nome `parsedibiurl.ibi` será, entre os **IBI** contidos na resposta selecionada, aquele que atende à lista de verbos que consta no par `parsedibiurl.verblast` capturado da **URL persistente** (cf. Tabela 6.6). O processo se repetirá até que a resposta selecionada pelo **resolvedor** contenha a URL para o **item de informação** solicitado (cf. o exemplo da Seção 8.2).

Após ter encontrado a URL apropriada à solicitação de resolução de **IBI**, o **resolvedor**, emite uma mensagem “Agradecer” (5) para o **Arquivo** que forneceu essa URL. A mensagem é construída conforme o formato de solicitação de serviço da Seção 4.1, anexando à URL de base todos os pares apresentados na Tabela 6.11.

Nessa tabela os valores dos pares com nome `contenttype`, `ibi`, `state`, `url` e `urlkey` são os valores dos pares com os mesmos nomes recebidos do próprio Arquivo (ver Seção 7.2). O par com nome `ibi` é do tipo facultivo (fac.), porque um Arquivo não

é obrigado a fornecer o IBI de todos itens de informação apontados por uma URL persistente modificada (por exemplo, no caso em que os metadados não possuem IBI). Para mais detalhes, ver a rotina CRIARPARIBI na Seção 7.2.

Tabela 6.11 - Pares <nome>=<valor> a serem anexados à URL de base para a mensagem “Agradecer” (5) endereçada ao **Arquivo** pelo **resolvedor**.

Par			Tipo
Nome	Valor		
	Domínio	Significado	
servicesubject	"acknowledgment"	agradecimento	obr.
clientinformation.ipaddress	regra lista-de-IP da tabela 6.10	endereço IP do usuário que solicitou a resolução de um IBI ex: 150.163.2.175	obr.
contenttype	"Data" "Metadata"	tipo de conteúdo do item de informação apontado pela URL persistente	obr.
ibi	regra formas-de-ibi da Tabela 7.4	formas de IBI identificando o item de informação apontado pela URL persistente	fac.
state	"Original" "Copy"	estado do item de informação apontado pela URL persistente	obr.
url	regra URI em Seção 3 de (BERNERS-LEE et al., 2005) com scheme valendo http	URL de acesso ao item de informação apontado pela URL persistente	obr.
url.persistent	regra URL-persistente da Tabela 6.1	URL persistente na origem da solicitação de resolução de IBI	obr.
urlkey	regra chave da Tabela 5.1	chave de URL constando da mensagem “lista de URL” (4) recebida do Arquivo	obr.

A mensagem de agradecimento pode ser útil para os **Arquivos** que queiram, por exemplo, manter estatísticas de acesso aos **itens identificados**, pois, quando recebida por um **Arquivo**, ela indica que um determinado **item de informação** armazenado nele foi selecionado no processo de resolução de **IBI** e deverá ser retor-

nado ao usuário do **sistema para resolução de IBI** na fase final do processo de resolução.

Exemplo 6.9 (mensagem de agradecimento emitida pelo **resolvedor**) – URL:

```
http://mtc-m16d.sid.inpe.br/sid.inpe.br/mtc-m19@80/2009/08.21.17.02?
servicesubject=acknowledgment&
clientinformation.ipaddress=150.163.2.175&
contenttype=Data&
ibi=rep%20sid.inpe.br/mtc-m19/2013/09.04.12.27.57%20ibip%208JMKD3MGP7W/3EPGUE5&
url=http://mtc-m16d.sid.inpe.br/col/sid.inpe.br/mtc-m19/2013/09.04.12.27.57/doc/
Relat%3rio%20Final.pdf&
url.persistent=http://mtc-m16d.sid.inpe.br:80/8JMKD3MGP7W/3EPGUE5&
urlkey=1424636122-5331490054869684
```

□

No Exemplo 6.9, a URL é a mensagem “Agradecer” (5) enviada pelo **resolvedor** ao **Arquivo** com endereço Web `mtc-m16d.sid.inpe.br` e com **serviço de Arquivo** identificado pelo **IBI** `sid.inpe.br/mtc-m19@80/2009/08.21.17.02`. À URL de base estão anexados os pares definidos na Tabela 6.11.

O par `clientinformation.ipaddress=150.163.2.175` informa que a **URL persistente** foi emitida pelo usuário com IP 150.163.2.175. O par `contenttype=Data` indica que o **item de informação** selecionado é do tipo dados (e não metadados). O par com nome `ibi` indica que seu `ibi` é `sid.inpe.br/mtc-m19/2013/09.04.12.27.57` (na forma de **nome de repositório uniforme**) e `8JMKD3MGP7W/3EPGUE5` (na forma opaca de **IBIp**). Finalmente, os três últimos pares com nomes `url`, `url.persistent` e `urlkey` informam, respectivamente, a URL de acesso ao **item de informação**, a **URL persistente** que deu origem à solicitação de resolução de **IBI** e a chave de URL fornecido pelo **Arquivo** no momento de retornar a mensagem “lista de URL” (4).

Após emitir a mensagem de agradecimento, o **resolvedor** retorna a URL de acesso ao **item de informação**, resultado do processo de resolução, ao navegador do usuário. Ao receber a URL (7), o navegador emite a mensagem “Trazer item de informação” (8) endereçada ao **Arquivo** selecionado, reproduzindo a URL recebida. O **Arquivo** então retorna o **item de informação** (9) ao navegador do usuário para sua exibição (10).

Se a análise das respostas recebidas pelo **resolvedor** indicar que o **IBI** objeto da resolução não existe ou foi removido, então o **resolvedor** não emitirá nenhuma mensagem de agradecimento, e retornará para o navegador do usuário uma mensagem

de alerta (7).

7 Serviço de Arquivo

7.1 Solicitações

O “serviço de **Arquivo** para resolução de **IBI**”, ou simplesmente **serviço de Arquivo**, é um programa de computador, geralmente um *script* CGI (*Common Gateway Interface*), que recebe solicitações da *Web* e retorna respostas. Cada **Arquivo** deve ter seu próprio **serviço de Arquivo** e torná-lo um **item identificado** tido como original intransferível.

O **serviço de Arquivo** é solicitado pelo **resolvedor** em três circunstâncias.

A primeira, quando o **resolvedor** solicita a confirmação de inclusão do **Arquivo** no **sistema para resolução de IBI**, ao emitir a mensagem “Solicitar confirmação” (3) do Diagrama 5.1 da Seção 5.2. Nesse caso, o `servicesubject` é `inclusionConfirmationRequest`, e apenas o par `servicesubject=inclusionConfirmationRequest` é anexado à URL de base (cf. Exemplo 5.4).

A segunda, quando o **resolvedor** solicita, a todos os **Arquivos** incluídos no **sistema para resolução de IBI**, a URL de acesso ao **item de informação** apontado pela **URL persistente**, isto é, quando emitir a mensagem “Solicitar URL” (3) do Diagrama 6.1 da Seção 6.2. Nesse caso, o `servicesubject` é `urlRequest` e os parâmetros da solicitação são os quatro últimos pares da Tabela 6.9 (cf. Exemplo 4.1).

Finalmente, a terceira, quando o **resolvedor** emite a mensagem “Agradecer” (5) do Diagrama 6.1 da Seção 6.2 para o **Arquivo** que possua o **item de informação** cujas especificações atendam a uma solicitação de resolução de **IBI**. Nesse caso, o `servicesubject` é `acknowledgment` e os parâmetros do agradecimento são os sete últimos pares da Tabela 6.11 (cf. Exemplo 6.9).

7.2 Respostas

As regras gramaticais e as descrições semânticas que devem guiar a resposta dada à solicitação dos três serviços mencionados na seção anterior são apresentadas por meio do Algoritmo 1 (MONTARRESPOSTA) e de seus componentes: as rotinas apresentadas na Tabela 7.1. Nesta seção, para distinguir entre nomes de variável e seus valores, os valores serão denotados entre aspas (“”).

Tabela 7.1 - Componentes do Algoritmo 1 (MONTARRESPOSTA).

<i>Componente</i>	<i>Página</i>
Rotina DECODQS	44
Rotina MONTARLISTADEURL	45
Rotina FILTRARLISTADEURL	47

Algoritmo 1 – MONTARRESPOSTA.

GLOBAL: *clientinformation.ipaddress* (*IP usuário – criado nesse algoritmo – usado por MONTARLISTADEURL*),
 parsedibiurl.filepath (*caminho de recurso – criado nesse algoritmo – usado por MONTARLISTADEURL*),
 parsedibiurl.verblast (*lista de verbos – criada nesse algoritmo – usada por MONTARLISTADEURL e por FILTRARLISTADEURL*).

ENTRADA: *qs* (*lista de pares <nome>=<valor> anexados à URL de base da mensagem “Solicitar URL” (3)*).

SAÍDA: *resposta* (*lista de pares*).

```
1.  servicesubject ← DECODQS(qs, "servicesubject")
2.  Se servicesubject = "inclusionConfirmationRequest" Então
3.    | resposta ← "confirmation yes"
4.    | Retorne
5.  Se servicesubject = "urlRequest" Então
6.    | clientinformation.ipaddress ← DECODQS(qs, "clientinformation.ipaddress")
7.    | Se "parsedibiurl.filepath" presente em qs Então
8.    |   | parsedibiurl.filepath ← DECODQS(qs, "parsedibiurl.filepath")
9.    | Senão
10.   | | parsedibiurl.filepath ← ""
11.   | parsedibiurl.ibi ← DECODQS(qs, "parsedibiurl.ibi")
12.   | Se "parsedibiurl.verblast" presente em qs Então
13.   |   | parsedibiurl.verblast ← DECODQS(qs, "parsedibiurl.verblast")
14.   | Senão
15.   |   | parsedibiurl.verblast ← ""
16.   | resposta ← MONTARLISTADEURL(parsedibiurl.ibi)
17.   | resposta ← FILTRARLISTADEURL(resposta, parsedibiurl.verblast)
18.   | Retorne
19. Se servicesubject = "acknowledgment" Então
20.   | resposta ← "notice {acknowledgment received}"
21.   | Retorne
```

O Algoritmo 1 (MONTARRESPOSTA) usa a rotina DECODQS (Decodificação da *Query String*). Essa rotina extrai da cadeia de caracteres informada em seu primeiro argumento a cadeia de caracteres correspondente ao valor do par <nome>=<valor>, cujo nome é informado em seu segundo argumento.

Exemplo 7.1 (Funcionamento da rotina DECODQS) – A execução de:

DECODQS(qs, "servicesubject")

onde o valor da variável `qs` era a cadeia de caracteres:

```
servicesubject=urlRequest&  
clientinformation.ipaddress=150.163.2.175&  
parsedibiurl.ibi=8JMKD3MGP7W/3EPGUE5
```

resultou na cadeia de caracteres:

urlRequest

□

No Exemplo 7.1, a rotina DECODQS é usada para extrair da cadeia de caracteres em `qs` a cadeia de caracteres que especificou o assunto da solicitação de serviço enviada ao **Arquivo** (aqui, nesse exemplo, `urlRequest`). No exemplo, a cadeia de caracteres em `qs` escolhida é a lista de pares <nome>=<valor> anexados à URL de base da mensagem de solicitação de serviço do Exemplo 4.1. Esta é uma possível mensagem “Solicitar URL” (3) da Figura 6.1.

A primeira etapa do Algoritmo 1 (MONTARRESPOSTA) consiste em extrair (Linha 1) da variável de entrada `qs` o valor do par `servicesubject=<valor>` anexado à URL de base da mensagem endereçada ao **Arquivo**. Este valor, dado por `DECODQS(qs, "servicesubject")`, é guardado na variável `servicesubject`.

Se o valor da variável `servicesubject` for `"inclusionConfirmationRequest"`, então a resposta deverá ser a constante `"confirmation yes"` (Linha 3).

Se o valor da variável `servicesubject` for `"acknowledgment"`, então a resposta deverá ser a constante `"notice {acknowledgment received}"` (Linha 20).

Se o valor da variável `servicesubject` for `"urlRequest"`, então a resposta deverá ser a da rotina `MONTARLISTADEURL` (Linha 16) ou, de modo opcional, sua versão filtrada dada pela rotina `FILTRARLISTADEURL` (Linha 17).

Para o funcionamento dessas rotinas, o Algoritmo 1 (MONTARRESPOSTA) deve capturar (Linhas 6 a 15), por meio da rotina DECODQS, os valores de mais quatro possíveis pares <nome>=<valor> anexados à URL de base da mensagem endereçada ao **Arquivo**. Os valores capturados são guardados nas variáveis: `clientinformation.ipaddress`, `parsedibiurl.filepath`, `parsedibiurl.ibi` e `parsedibiurl.verblast`. Das quatro variáveis, a primeira, a segunda e a última, são globais e serão usadas no Algoritmo 2 (MONTARLISTADEURL)¹, sendo a última também usada como argumento na Rotina FILTRARLISTADEURL.

Conforme a Tabela 6.9, os dois pares com nomes `parsedibiurl.filepath` e `parsedibiurl.verblast` são facultativos na mensagem “Solicitar URL” (3). Por este motivo, antes de capturar os valores desses pares por meio da rotina DECODQS, a sua presença é testada (Linhas 7 e 12). Se estiverem ausentes, as variáveis correspondentes `parsedibiurl.filepath` e `parsedibiurl.verblast` receberão uma cadeia de caracteres vazia (Linhas 10 e 15).

A Linha 16 do Algoritmo 1 (MONTARRESPOSTA), a rotina MONTARLISTADEURL (Algoritmo 2) retorna, na forma de uma lista de pares, um conjunto de propriedades do **IBI** informado no único argumento da rotina. Entre essas propriedades, podem constar ou não, em particular, as URL de acesso aos **itens de informação** relacionados ao **item identificado** com esse **IBI**.

Exemplo 7.2 (Funcionamento da rotina MONTARLISTADEURL) – A execução de:

MONTARLISTADEURL("8JMKD3MGP8W/35MME4E")

resultou em uma cadeia de caracteres formando a lista de pares:

<code>archiveaddress</code>	<code>mtc-m16c.sid.inpe.br</code>
<code>contenttype</code>	<code>Data</code>
<code>contenttype.lastedition</code>	<code>Data</code>
<code>contenttype.lastedition.metadata</code>	<code>Metadata</code>
<code>contenttype.lastedition.metadata(oai_dc)</code>	<code>Metadata</code>
<code>contenttype.metadata</code>	<code>Metadata</code>
<code>contenttype.metadata(oai_dc)</code>	<code>Metadata</code>
<code>contenttype.metadata.translation</code>	<code>Metadata</code>
<code>contenttype.translation(en)</code>	<code>Data</code>
<code>contenttype.translation(pt)</code>	<code>Data</code>
<code>ibi</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23 ibip 8JMKD3MGP8W/35MME4E}</code>
<code>ibi.archiveservice</code>	<code>{rep sid.inpe.br/mtc-m18080/2008/03.17.15.17}</code>
<code>ibi.lastedition</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23 ibip 8JMKD3MGP8W/35MME4E}</code>
<code>ibi.lastedition.metadata</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23.47}</code>
<code>ibi.lastedition.metadata(oai_dc)</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23.47}</code>
<code>ibi.metadata</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23.47}</code>
<code>ibi.metadata(oai_dc)</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23.47}</code>
<code>ibi.metadata.translation</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23.47}</code>
<code>ibi.platformsoftware</code>	<code>{rep dpi.inpe.br/banon/1998/08.02.08.56}</code>
<code>ibi.translation(en)</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23}</code>

¹Essas três variáveis foram definidas como globais, unicamente por uma razão editorial, evitando assim a definição e uso de rotinas com muitos argumentos.

ibi.translation(pt)	{rep sid.inpe.br/mtc-m18080/2009/08.25.19.43}
state	Original
state.lastedition	Original
state.lastedition.metadata	Original
state.lastedition.metadata(oai_dc)	Original
state.metadata	Original
state.metadata(oai_dc)	Original
state.metadata.translation	Original
state.translation(en)	Original
state.translation(pt)	Original
timestamp	2009-07-21T13:23:45Z
timestamp.lastedition	2009-07-21T13:23:45Z
timestamp.lastedition.metadata	2014-04-04T17:39:54Z
timestamp.lastedition.metadata(oai_dc)	2014-04-04T17:39:54Z
timestamp.metadata	2014-04-04T17:39:54Z
timestamp.metadata(oai_dc)	2014-04-04T17:39:54Z
timestamp.metadata.translation	2014-04-04T17:39:54Z
timestamp.translation(en)	2009-07-21T13:23:45Z
timestamp.translation(pt)	2011-09-22T14:45:11Z
url	http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23/doc/CCSDS%20643.0-B-1.pdf
url.lastedition	http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23/doc/CCSDS%20643.0-B-1.pdf
url.lastedition.metadata	http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi
url.lastedition.metadata(oai_dc)	http://mtc-m16c ... br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi?choice=oai_dc
url.metadata	http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi
url.metadata(oai_dc)	http://mtc-m16c ... br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi?choice=oai_dc
url.metadata.translation	http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi
url.translation(en)	http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23/doc/CCSDS%20643.0-B-1.pdf
url.translation(pt)	http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/08.25.19.43/doc/RTC-07.pdf
urlkey	1426387214-9164566186556927

□

A lista de pares do Exemplo 7.2 ilustra um conjunto de propriedades disponíveis a respeito do **IBI 8JMKD3MGP8W/35MME4E** informado no único argumento da rotina. Nessa lista, o nome de alguns pares consiste em uma sequência de palavras separadas por *pontos* (.). Por definição, a primeira palavra é chamado de prefixo do nome. Omitindo o prefixo, o resto do nome é chamado de qualificador de prefixo. Por exemplo, no nome `ibi.lastedition`, `ibi` é o seu prefixo e `.lastedition` o qualificador do prefixo `ibi`. Em um exemplo como este, para se referir ao nome `ibi.lastedition`, usa-se a expressão “nome com prefixo `ibi` qualificado como `.lastedition`”.

Na lista de pares do exemplo, o par com nome `ibi` (o décimo primeiro par) informa as duas formas desse IBI. O par com nome `archiveaddress` (o primeiro par) informa qual foi o endereço *web* do **Arquivo** que gerou esse conjunto de propriedades. No conjunto, observa-se o par com nome `url`, cujo valor foi a URL de acesso ao **item identificado** com **IBI 8JMKD3MGP8W/35MME4E**. Além desse par, nesse exemplo, existem outros oito pares com nomes cujos prefixos foram `url`, e com valores sendo URL de acesso a **itens de informação** relacionados ao **item identificado** com **IBI 8JMKD3MGP8W/35MME4E**. O qualificador de prefixo do nome desses pares indica o tipo de relação com o **item identificado**. Por exemplo, o qualificador `.metadata` em `url.metadata` indica que a URL é a dos metadados do **item identificado** objeto da resolução (cf. Tabela 7.5 e Exemplo 7.5).

No conjunto de propriedades do Exemplo 7.2, observa-se também que os valores dos pares com nomes com prefixo qualificado como `.lastedition` são os mesmos que os dos pares com nomes sem qualificador de prefixo. Isto ocorre porque o **item identificado** com **IBI** 8JMKD3MGP8W/35MME4E não possui (ainda) uma próxima edição. Nesse caso, a última edição do **item identificado** é o próprio **item**. Pelo mesmo motivo, os valores dos pares com nomes com prefixo qualificado como `.lastedition.metadata` são os mesmos que os dos pares com nomes com prefixo qualificado como `.metadata`.

Observa-se ainda que os valores dos pares com nomes com prefixo `ibi` qualificado como `.metadata(oai_dc)` são os mesmos que os dos pares com nomes com prefixo `ibi` qualificado como `.metadata`. Isto ocorre, nse exemplo, porque o acesso a estes metadados é feito a partir de um mesmo *script* CGI que gera os metadados em diversos formatos.

Dependendo do conhecimento ao alcance do **Arquivo**, o conjunto das propriedades de um **IBI** pode ser eventualmente grande (cf. as numerosas chamadas às rotinas CRIARPAR e JUNTARPAR no Algoritmo 2 (MONTARLISTADEURL)).

A Linha 17 do Algoritmo 1 (MONTARRESPOSTA), a rotina FILTRARLISTADEURL, com base na lista de verbos (`parsedibiurl.verblast`) informada em seu segundo argumento, elimina, da resposta informada em seu primeiro argumento, os pares que não terão qualquer utilidade na finalização do processo de resolução de **IBI** pelo **resolvedor**. Em outros termos, considerando a lista de verbos, a rotina FILTRARLISTADEURL preserva apenas as propriedades de interesse ao processo de resolução de **IBI**.

O uso da rotina FILTRARLISTADEURL é facultativo. Seu papel é diminuir o tamanho da resposta “lista de URL” (4). Mesmo que essa rotina não elimine par algum sem utilidade, o resultado do processo de resolução de **IBI** não será alterado, apenas o desempenho do processo não será tão eficiente.

Exemplo 7.3 (Funcionamento da rotina FILTRARLISTADEURL) – A execução de:

FILTRARLISTADEURL(*resposta*, "GetTranslation")

onde o valor de *resposta* é a lista de pares do Exemplo 7.2, resultou em uma cadeia de caracteres formando a lista de pares:

<code>archiveaddress</code>	<code>mtc-m16c.sid.inpe.br</code>
<code>contenttype.translation(en)</code>	<code>Data</code>
<code>contenttype.translation(pt)</code>	<code>Data</code>
<code>ibi</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23 ibip 8JMKD3MGP8W/35MME4E}</code>

<code>ibi.archiveservice</code>	<code>{rep sid.inpe.br/mtc-m18080/2008/03.17.15.17}</code>
<code>ibi.platformsoftware</code>	<code>{rep dpi.inpe.br/banon/1998/08.02.08.56}</code>
<code>ibi.translation(en)</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/07.21.13.23}</code>
<code>ibi.translation(pt)</code>	<code>{rep sid.inpe.br/mtc-m18080/2009/08.25.19.43}</code>
<code>state.translation(en)</code>	<code>Original</code>
<code>state.translation(pt)</code>	<code>Original</code>
<code>timestamp.translation(en)</code>	<code>2009-07-21T13:23:45Z</code>
<code>timestamp.translation(pt)</code>	<code>2011-09-22T14:45:11Z</code>
<code>url.translation(en)</code>	<code>http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23/doc/CCSDS%20643.0-B-1.pdf</code>
<code>url.translation(pt)</code>	<code>http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/08.25.19.43/doc/RTC-07.pdf</code>
<code>urlkey</code>	<code>1426387214-9164566186556927</code>

□

No Exemplo 7.3, a rotina `FILTRARLISTADEURL` é usada para simplificar (filtrar) a lista de pares especificada na variável `resposta`, em seu primeiro argumento. Nesse exemplo, a lista de verbos, no segundo argumento da rotina, é formado pelo único verbo `GetTranslation`. Com base nesse verbo, a lista de pares pôde ser simplificada sem prejuízo à finalização do processo de resolução de **IBI**, a ser executado posteriormente no **resolvedor**. A simplificação consiste em preservar apenas os pares cujos nomes possuem um prefixo qualificado como `.translation` (com ou sem parâmetro). Observa-se que, da lista de nove URL do Exemplo 7.2, ficaram apenas duas, relativas às traduções em inglês e português, nos pares com nomes `url.translation(en)` e `url.translation(pt)`, respectivamente.

Conforme especificado na Tabela 6.4, o verbo `GetTranslation` pode ou não estar acompanhado de um parâmetro indicando o idioma (e o país) da tradução solicitada (e.g., `GetTranslation(pt)`). Na ausência de parâmetro, como no Exemplo 7.3, a seleção da URL de acesso ao **item de informação** no idioma apropriado será feita pelo **resolvedor**. Por esse motivo, as várias opções de URL apontando, cada uma para uma tradução em um idioma diferente, como inglês (`en`) e português (`pt`), devem constar na lista de pares a ser encaminhada ao **resolvedor**. Ao receber a resposta dos **Arquivos**, o **resolvedor** escolherá a URL que corresponde ao idioma que melhor atende as preferências do usuário, como especificado em seu navegador. Para isso, o **resolvedor** fará uso do par com o nome `clientinformation.languagepreference` da Tabela 6.6.

O Algoritmo 2 (`MONTARLISTADEURL`), usado na Linha 16 do Algoritmo 1 (`MONTARRESPOSTA`), para gerar as propriedades do **IBI** objeto de resolução, é, junto às rotinas que o compõem (cf. Tabela 7.2), a peça principal do **serviço de Arquivo**.

Tabela 7.2 - Componentes do Algoritmo 2 (MONTARLISTADEURL).

<i>Componente</i>	<i>Página</i>
Rotina CRIARPAR	51
Rotina CRIARPARSTATE	56
Rotina JUNTAR	56
Rotina CRIARPARNEXTEDITION	57
Rotina CRIARPARIBI	58
Rotina CRIARPARURL	60

Algoritmo 2 – MONTARLISTADEURL.

GLOBAL: relações (*subconjunto de relações verificando a regra relação da Tabela 7.5*),
 clientinformation.ipaddress (*IP do usuário – usado por CRIARPARURL*),
 parsedibiurl.filepath (*caminho de um recurso – usado por CRIARPARURL*),
 parsedibiurl.verblist (*lista de verbos – usada por CRIARPARURL*).
ENTRADA: parsedibiurl.ibi (*IBI objeto da resolução*).
SAÍDA: resposta (*lista de pares*).

```
1.  resposta ← ""
2.  state ← CRIARPARSTATE(parsedibiurl.ibi)
3.  Se state = "" Então
4.    Retorne resposta
5.  Senão
6.    resposta ← JUNTAR(resposta, CRIARPAR("archiveaddress", parsedibiurl.ibi))
7.    par ← CRIARPAR("ibi.archiveservice", parsedibiurl.ibi)
8.    resposta ← JUNTAR(resposta, par)
9.    par ← CRIARPAR("ibi.platformsoftware", parsedibiurl.ibi)
10.   resposta ← JUNTAR(resposta, par)
11.   Se state = "Deleted" Então
12.     resposta ← JUNTAR(resposta, CRIARPAR("ibi", parsedibiurl.ibi))
13.     resposta ← JUNTAR(resposta, state)
14.     resposta ← JUNTAR(resposta, CRIARPAR("timestamp", parsedibiurl.ibi))
15.   Retorne resposta
16.   Senão
17.     par ← CRIARPARNEXTEDITION(parsedibiurl.ibi)
18.     resposta ← JUNTAR(resposta, par)
19.     par ← CRIARPAR("urlkey", parsedibiurl.ibi)
20.     resposta ← JUNTAR(resposta, par)
21.     Para Cada relação ∈ relações Faça
22.       par ← CRIARPARIBI(relação, parsedibiurl.ibi)
23.       resposta ← JUNTAR(resposta, par)
24.       par ← CRIARPARURL(relação, parsedibiurl.ibi)
25.       Se par ≠ "" Então
26.         resposta ← JUNTAR(resposta, par)
27.         par ← CRIARPAR(("contenttype" relação), parsedibiurl.ibi)
28.         resposta ← JUNTAR(resposta, par)
29.         par ← CRIARPAR(("state" relação), parsedibiurl.ibi)
30.         resposta ← JUNTAR(resposta, par)
31.         par ← CRIARPAR(("timestamp" relação), parsedibiurl.ibi)
32.       Retorne resposta
33.   Retorne resposta
```

O Algoritmo 2 (MONTARLISTADEURL), apresentado acima, usa a rotina CRIARPAR. Essa rotina cria uma cadeia de caracteres que forma um par, conforme a regra par da Tabela 4.2. O nome do par é o valor informado no primeiro argumento da rotina. O valor do par é referente ao **item de informação** relacionado ao **item identificado** com **IBI** informado no segundo argumento.

Precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres x da Tabela 7.3 e todo **IBI** i existente,

$$\text{CRIARPAR}(x, i) \triangleq x \text{ 1*SP } \langle x, i \rangle$$

onde SP é a regra definida na Tabela 4.2 e $\langle x, i \rangle$ é a cadeia de caracteres que representa o valor de x referente a i , como apresentado na Tabela 7.3.

Tabela 7.3 - Descrição da função $\langle x, i \rangle$.

x	$\langle x, i \rangle$	
	Domínio	Significado
"archiveaddress"	regra hostport da Seção 4.1	endereço Web do Arquivo ex: marte.sid.inpe.br
"contenttype" relação (relação $\in$ relações)	"Data" "Metadata"	tipo de conteúdo do item de informação em relação com IBI i
"ibi" relação (relação $\in$ relações)	regra formas-de-ibi da Tabela 7.4	formas do IBI do item identificado em relação com IBI i
"ibi.archiveservice"	regra formas-de-ibi da Tabela 7.4	formas do IBI do serviço de Arquivo informado no momento de cadastrá-lo (cf. Seção 5.1)
"ibi.nextedition"	regra formas-de-ibi da Tabela 7.4	formas do IBI identificando a próxima edição do item identificado com IBI i
"ibi.platformsoftware"	formas-de-ibi "{}"	formas do IBI identificando o <i>software</i> que assegura o funcionamento da plataforma hospedando o Arquivo
"state" relação (relação $\in$ relações)	"Original" "Copy" "Deleted"	estado do item de informação em relação com IBI i
"timestamp" relação (relação $\in$ relações)	ISO 8601	data e hora da última atualização ou remoção do item de informação em relação com IBI i ex: 2014-01-02T17:23:57Z
"urlkey"	regra chave da Tabela 5.1	chave de URL de acesso ao item de informação relacionado com o item identificado com IBI i

A expressão “formas de **IBI**”, na coluna *Significado* da Tabela 7.3, refere-se às várias formas como um **IBI** pode ser escrito (BANON, 2011). Por exemplo, sid.inpe.br/mtc-m19/2013/09.04.12.27.57 e 8JMKD3MGP7W/3EPGUE5 são duas formas do **IBI** de um mesmo **item identificado**. A primeira forma é o **IBI** como

nome de repositório uniforme; a segunda, como **IBIp**.

A Tabela 7.4 especifica as regras de formação da regra `formas-de-ibi` usada na coluna *Domínio* da Tabela 7.3. A especificação utiliza a notação BNF (aumentada) e as regras predefinidas: **repositório** (**IBI** como **nome de repositório uniforme**) e **IBIp** (**IBI** como **IBIp**), introduzidas em Banon (2011).

Tabela 7.4 - Regras definindo a formação de `formas-de-ibi`

```
formas-de-ibi = forma-rep | forma-ibip | forma-rep " " forma-ibip
  forma-rep   = "rep " repositório
  forma-ibip  = "ibip " IBIp
```

Exemplo 7.4 (Funcionamento da rotina `CRIARPAR` para acessar o **estado** de um **item identificado**) – A execução de:

```
CRIARPAR("state", "8JMKD3MGP7W/3EPGUE5")
```

resultou em uma cadeia de caracteres formando o par:

```
state Original
```

□

No Exemplo 7.4, a rotina `CRIARPAR` é usada para criar uma cadeia de caracteres que forma um par, conforme a regra `par` da Tabela 4.2. O valor `Original` do par, dado por `< "state", "8JMKD3MGP7W/3EPGUE5" >`, fornece o **estado** do **item identificado** com **IBI** `8JMKD3MGP7W/3EPGUE5`. Assim, o resultado da execução da rotina `CRIARPAR("state", "8JMKD3MGP7W/3EPGUE5")` transmite a informação de que o **item identificado** com **IBI** `8JMKD3MGP7W/3EPGUE5` deve ser tido como original (no **Arquivo** considerado).

Na coluna `x` da Tabela 7.3, aparecem expressões como `"contenttype" relação` em que `relação` $\in$ `relações`. A primeira parte da expressão usa a notação BNF (aumentada). Por exemplo, se `relação` for a cadeia de caracteres `".metadata"`, então `"contenttype" relação` é, por concatenação, a cadeia de caracteres `"contenttype.metadata"`. Na segunda parte, `relações` é a variável global do Algoritmo 2 (`MONTARLISTADEURL`). Ela representa um subconjunto de relações que verificam a regra `relação` da Tabela 7.5.

Tabela 7.5 - Regras definindo a formação de uma **relação**

```

relação = ([u [t]] | [t [u]]) [m [t]]; ex: .lastedition.metadata
u = ".lastedition"; ex: .lastedition
t = ".translation" ["(" idioma ["-" país] ")"];
ex: .translation(en)
idioma = 2ALFAMI; ex: pt
país = 2ALFAMA; ex: BR
m = ".metadata" ["(" formato ")"]; ex: .metadata(oai_dc)
formato = "oai_dc"

```

No Exemplo 7.5, são apresentadas algumas relações verificando a regra **relação** da Tabela 7.5, assim como seus significados.

Exemplo 7.5 (Algumas relações) – Relações e seus significados:

""	do próprio item identificado (ii)
".lastedition"	da última edição do ii
".translation"	da tradução do ii
".translation(pt-BR)"	da tradução em português do Brasil do ii
".metadata"	dos metadados do ii
".metadata(oai_dc)"	dos metadados no formato <i>oai_dc</i> do ii
".lastedition.translation"	da tradução da última edição do ii
".lastedition.translation(pt-BR)"	da tradução em português do Brasil da última edição do ii
".lastedition.metadata"	dos metadados da última edição do ii
".lastedition.metadata(oai_dc)"	dos metadados no formato <i>oai_dc</i> da última edição do ii
".translation(pt-BR).lastedition"	da última edição da tradução em português do Brasil do ii
".translation(en).metadata"	dos metadados da tradução em inglês do ii
".metadata.translation(en)"	da tradução em inglês dos metadados do ii

□

Cada **Arquivo** pode ter seu próprio subconjunto **relações**. Por exemplo, um **Arquivo** que não armazena **itens de informação** no idioma Valão (*Walloon*) não deveria incluir, no subconjunto **relações**, relações contendo a relação elementar **.translation(wa)**. Caso o fizesse, as rotinas **CRIARPARIBI** e **CRIARPARURL**, respectivamente, as Linhas 22 e 24 do Algoritmo 2, retornariam sempre, ambas, uma cadeia de caracteres vazia (""), já que não haveria qualquer **item de informação** no idioma Valão, fazendo com que a resposta criada até o momento seria devolvida inalterada, ou seja, os processamentos realizados referentes ao idioma Valão (*Walloon*) teriam sido totalmente desnecessários.

Por outro lado, se o **Arquivo** armazenar **itens de informação** sem especificação de idioma, então ele deverá incluir no subconjunto **relações**, relações contendo a relação elementar **.translation** (sem parâmetro de idioma). Essa opção será útil

para o **resolvedor** poder retornar a URL do **item de informação** que atende à **URL persistente** quando esta contém o modificador *mais* (+) sem parâmetro de idioma também. Se não fizer, o resolvedor retornará ao usuário, que emitiu a URL persistente, uma mensagem indicando que o item de informação não foi encontrado.

O subconjunto **relações** deve conter pelo menos a relação vazia "" (cadeia de caracteres vazia). Numa implementação mínima do Sistema para resolução de IBI, na parte que cabe um Arquivo, o subconjunto **relações** pode ser reduzido à uma única relação que deve ser a relação vazia. Neste caso, o Arquivo somente poderá atender às solicitações do resolvedor referente às URL persistentes na sua forma simples (cf. 6.1).

Na coluna *Significado* da Tabela 7.3, a expressão “**item de informação em relação com IBI i**” é um atalho para “**item de informação** relacionado ao **item identificado com IBI i**, de acordo com o relacionamento definido pela variável **relação**”.

Assim, se o valor da variável **relação** for `".lastedition.metadata"`, a expressão “**item de informação em relação com IBI i**” significará “metadados da última edição do **item identificado com IBI i**”, e o significado de $\langle x, i \rangle$ na Tabela 7.3 para “`"state" relação`”, ou seja, para `"state.lastedition.metadata"`, será: “**estado** dos metadados da última edição do **item identificado com IBI i**”.

Quando o valor da variável **relação** for vazio (`"`), então a expressão “**item de informação em relação com IBI i**” deve ser entendida simplesmente como “**item identificado com IBI i**”, pois, neste caso, o “**item de informação em relação com IBI i**” é o próprio “**item identificado com IBI i**”.

Exemplo 7.6 (Funcionamento da rotina CRIARPAR para acessar o carimbo de data/hora (*time stamp*) da última edição de um **item identificado**) – A execução de:

```
CRIARPAR("timestamp.lastedition", "8JMKD3MGP7W/3EPGUE5")
```

resultou em uma cadeia de caracteres formando o par:

```
timestamp.lastedition 2013-10-04T14:32:14Z
```

□

No Exemplo 7.6, a rotina CRIARPAR é usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O valor 2013-10-04T14:32:14Z

do par, dado por `< "timestamp.lastedition", "8JMKD3MGP7W/3EPGUE5" >`, fornece o carimbo de data/hora (*time stamp*) da última edição do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5. Assim, o resultado da execução da rotina `CRIARPAR("timestamp.lastedition", "8JMKD3MGP7W/3EPGUE5")` transmite a informação de que carimbo de data/hora da última edição do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5 é 2013-10-04T14:32:14Z.

A primeira etapa do Algoritmo 2 (MONTARLISTADEURL) consiste em encontrar (Linha 2) o valor do **estado** (*state*) do **item identificado** com **IBI** especificado na variável de entrada `parsedibiurl.ibi`. Para isso, o Algoritmo 2 usa a rotina `CRIARPARSTATE`. Se o **IBI** informado em seu único argumento existir, essa rotina deverá criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O nome do par deverá ser **state**; o seu valor, o **estado** do **item identificado** com o **IBI** informado. Se o **IBI** não existir no **Arquivo**, a rotina `CRIARPARSTATE` deverá criar uma cadeia de caracteres vazia.

Precisamente, para todo **IBI** *i*,

$$\text{CRIARPARSTATE}(i) \triangleq \begin{cases} \text{CRIARPAR}(\text{"state"}, i) & \text{se } \mathbf{IBI} \text{ } i \text{ existir,} \\ "" & \text{caso contrário.} \end{cases}$$

A Linha 3 do Algoritmo 2 (MONTARLISTADEURL), o valor da variável **state** é testado. Se o valor for vazio (`""`), então o valor da variável de saída **resposta** deverá ser também vazio (`""`), indicando que o **IBI** informado não existe no **Arquivo**. Senão, os pares com nomes `archiveaddress`, `ibi.archiveservice` e `ibi.platformsoftware`, que servem para identificar o **Arquivo**, deverão ser adicionados à variável **resposta**, por meio das rotinas `CRIARPAR` e `JUNTAR` as Linhas 6 a 10. Caso o *software* que assegura o funcionamento da plataforma hospedando o **Arquivo** não esteja sendo identificado por **IBI**, o valor do par com nome `ibi.platformsoftware` deverá ser vazio, representado pela cadeia de caracteres *abre e fecha chaves* (`{}`).

A rotina `JUNTAR`, usada várias vezes no Algoritmo 2 (MONTARLISTADEURL), junta as cadeias de caracteres informadas em seus argumentos, deixando-as separadas por um **espaço**.

Precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres

x e y ,

$$\text{JUNTAR}(x, y) \triangleq \begin{cases} x \text{ espaço } y & \text{se } x \text{ e } y \neq "", \\ y & \text{se } x = "", \\ x & \text{se } y = "". \end{cases}$$

onde **espaço** é a regra definida na Tabela 4.2.

Exemplo 7.7 (Funcionamento da rotina JUNTAR) – A execução de:

```
JUNTAR("archiveaddress mtc-m16d.sid.inpe.br", "state Original")
```

resultou na cadeia de caracteres:

```
archiveaddress mtc-m16d.sid.inpe.br state Original
```

□

No Exemplo 7.7, a rotina JUNTAR é usada para juntar duas cadeias de caracteres que formam, cada um, um par, conforme a regra **par** da Tabela 4.2. O resultado é uma lista de dois pares que verifica a regra **lista-de-pares** da Tabela 4.2. Nesse exemplo, usou-se como “espaço” o espaço em branco (SP). Observa-se que o segundo par poderia ser o resultado da execução da rotina CRIARPAR, como ilustrado no Exemplo 7.4.

A Linha 11 do Algoritmo 2 (MONTARLISTADEURL), o valor da variável **state** é comparado ao valor "Deleted".

Se a comparação indicar que o **IBI** foi removido do **Arquivo**, então os pares com nomes **ibi**, **state** e **timestamp** deverão ser adicionados à variável **resposta**, como mostrado as Linhas 12 a 14, antes de seu valor ser retornado (Linha 15).

Se a comparação indicar que o **IBI** não foi removido, então os pares que retratam as propriedades do **IBI** deverão ser criados, como mostrado nas linhas seguintes.

A Linha 17, o Algoritmo 2 (MONTARLISTADEURL) usa a rotina CRIARPARNEXTEDITION. Se o **item identificado**, com **IBI** informado em seu único argumento, possuir uma nova edição, essa rotina deverá criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O nome do par será "**ibi.nextedition**"; o seu valor, as formas do **IBI** da nova edição do **item identificado**. Se não houver nova edição, a rotina CRIARPARNEXTEDITION deverá criar uma cadeia de caracteres vazia.

Precisamente, usando a notação BNF (aumentada), para todo **IBI** i existente,

$$\text{CRIARPARNEXTEDITION}(i) \triangleq \begin{cases} \text{CRIARPAR}(\text{"ibi.nextedition"}, i) & \text{se existir nova edição,} \\ "" & \text{caso contrário.} \end{cases}$$

A propriedade que consiste em informar que o **item identificado** possui uma nova edição poderá ser importante quando o processo de resolução de **IBI** deve levar à última edição do **item identificado** (cf. o exemplo da Seção 8.2).

A Linha 19, o Algoritmo 2 (MONTARLISTADEURL) usa a rotina CRIARPAR para criar um par, com nome `urlkey` e cujo valor é uma **chave de URL**. Essa chave poderá ser usada pelo **Arquivo** ao receber a mensagem “Agradecer” (5), enviada pelo **resolvedor**, para autorizar, por exemplo, a incrementação das estatísticas de acesso ao **item de informação** que deverá ser retornado ao usuário do **sistema para resolução de IBI**. Por motivo de segurança, essa chave deveria ser trocada a cada resposta fornecida pelo **Arquivo**.

A Linha 21, o Algoritmo 2 (MONTARLISTADEURL) entra em um *loop* do tipo **Para Cada**, em que a variável **relação** vai assumir todos os valores possíveis dentre um subconjunto de relações verificando a regra **relação** da Tabela 7.5. O Exemplo 7.5 ilustra alguns possíveis valores para a variável **relação**.

A Linha 22, já dentro do *loop*, o Algoritmo 2 (MONTARLISTADEURL) usa a rotina CRIARPARIBI. Se o **IBI** do **item de informação**, relacionado ao **item identificado** com **IBI** informado no segundo argumento da rotina, for conhecido, então essa rotina deverá criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O nome do par será `"ibi" r` (usando a notação BNF (aumentada)), em que o valor da relação r é o valor informado no primeiro argumento da rotina. O valor do par será as formas do **IBI** do **item de informação** relacionado ao **item identificado**, de acordo com o relacionamento definido pela relação r . Se o **IBI** do **item de informação** não for conhecido (por exemplo, no caso em que os metadados não possuem **IBI**, ou seja, não são um **item identificado**), a rotina CRIARPARIBI deverá criar uma cadeia de caracteres vazia.

Precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres r que verifica a regra **relação** da Tabela 7.5 e todo **IBI** i existente,

$$\text{CRIARPARIBI}(r, i) \triangleq \begin{cases} \text{"ibi" } r \text{ 1*SP } \langle r, i \rangle & \text{se } \langle \cdot \rangle \text{ for conhecido,} \\ "" & \text{caso contrário.} \end{cases}$$

onde **SP** é a regra definida na Tabela 4.2 e $\langle r, i \rangle$ é a cadeia de caracteres que representa as formas do **IBI** (cf. Tabela 7.4) do **item de informação** relacionado ao **item identificado** com **IBI** i , de acordo com o relacionamento definido pela relação r .

Exemplo 7.8 (Funcionamento da rotina CRIARPARIBI para acessar o **IBI** de um **item de informação** relacionado a um **item identificado**) – A execução de:

```
CRIARPARIBI(".metadata", "8JMKD3MGP7W/3EPGUE5")
```

resultou em uma cadeia de caracteres formando o par:

```
ibi.metadata {rep sid.inpe.br/mtc-m19/2013/09.04.12.27.58}
```

Ou ainda, a execução de:

```
CRIARPARIBI(".lastedition", "8JMKD3MGP7W/3EPGUE5")
```

resultou numa cadeia de caracteres formando o par:

```
ibi.lastedition {rep sid.inpe.br/mtc-m19/2013/09.04.12.27.57 ibip 8JMKD3MGP7W/3EPGUE5}
```

□

Na primeira parte do Exemplo 7.8, a rotina CRIARPARIBI é usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O valor `{rep sid.inpe.br/mtc-m19/2013/09.04.12.27.58}` do par, dado por $\langle ".metadata", "8JMKD3MGP7W/3EPGUE5" \rangle$, fornece a forma **rep** do **IBI** dos metadados do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5. Assim, o resultado da execução da rotina CRIARPARIBI(".metadata", "8JMKD3MGP7W/3EPGUE5") transmite a informação de que os metadados do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5 é, por sua vez, identificados pelo **IBI** sid.inpe.br/mtc-m19/2013/09.04.12.27.58.

Na segunda parte do Exemplo 7.8, a rotina CRIARPARIBI é usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O valor `{rep sid.inpe.br/mtc-m19/2013/09.04.12.27.57 ibip 8JMKD3MGP7W/3EPGUE5}` do

par, dado por `< ".lastedition", "8JMKD3MGP7W/3EPGUE5" >`, fornece as formas do **IBI** da última edição do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5. Assim, o resultado da execução da rotina `CRIARPARIBI(".lastedition", "8JMKD3MGP7W/3EPGUE5")` transmite a informação de que a última edição do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5 é o próprio **item identificado**, ou seja, nesse caso, o **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5 não possui (ainda) uma nova edição.

De forma geral, a informação retornada pela rotina `CRIARPARIBI` é útil, porque será repassada, na mensagem “Agradecer” (5), para o **Arquivo** que forneceu a URL apropriada à solicitação de resolução de **IBI**. Dessa forma, esse **Arquivo** poderá, por exemplo, manter estatísticas de acesso aos **itens identificados**.

A informação retornada pela rotina `CRIARPARIBI` é, também, essencial para tentar completar o processo de resolução de **IBI** quando, para uma mesma relação **r**, a rotina `CRIARPARURL`, a ser apresentada em seguida, não consegue retornar a URL de interesse. Nesse caso, o **resolvedor** recorrerá ao **IBI** retornado pela rotina `CRIARPARIBI`, para emitir uma nova mensagem “Solicitar URL” (3), na expectativa de receber uma nova resposta que forneça a URL de interesse (cf. o exemplo da Seção 8.2).

A rotina `CRIARPARURL`, usada no Algoritmo 2 (`MONTARLISTADEURL`) Linha 24, cria uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O nome do par é `"url" r` (usando a notação BNF (aumentada)), em que o valor da relação **r** é o valor informado no primeiro argumento da rotina. O valor do par é uma das três URL definidas abaixo, dependendo dos valores das duas variáveis globais `parsedibiurl.filepath` e `parsedibiurl.verblast`.

Se essas duas variáveis globais estiverem vazias (iguais à `""`), o que é o mais usual, então a URL deverá ser simplesmente a de acesso ao **item de informação** relacionado ao **item identificado** com **IBI** **i** informado em seu segundo argumento, de acordo com o relacionamento definido pela relação **r**.

Se a única variável não vazia for `parsedibiurl.filepath` e contiver o nome de um recurso (arquivo) que compõe o **item de informação** relacionado ao **item identificado** com **IBI** **i**, de acordo com o relacionamento definido pela relação **r**, então a URL deverá ser a de acesso a esse recurso (arquivo).

Se a única variável não vazia for `parsedibiurl.verblast` e contiver o verbo

`GetFileList`, então a URL deverá ser a de acesso a uma página de exibição da lista de recursos (arquivos) que compõem o **item de informação** relacionado ao **item identificado** com **IBI** i , de acordo com o relacionamento definido pela relação r .

Se essas duas variáveis não estiverem vazias, então deve prevalecer a segunda (`parsedibiurl.verblast`).

Se a URL de acesso for desconhecida do **Arquivo**, a rotina `CRIARPARURL` deverá criar uma cadeia de caracteres vazia.

Precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres r que verifica a regra **relação** da Tabela 7.5, para todo **IBI** i existente, e para toda cadeia de caracteres y e z , sendo, respectivamente, as variáveis globais `parsedibiurl.filepath` e `parsedibiurl.verblast`,

$$\text{CRIARPARURL}(r, i) \triangleq \begin{cases} "" & \text{se } < \cdot > \text{ abaixo} \\ & \text{for desconhecido,} \\ "url" r 1*SP < r, i >_1 & \text{se } y = "" \text{ e } z = "", \\ "url" r 1*SP < r, i, y >_2 & \text{se } y \neq "" \text{ e } z = "", \\ "url" r 1*SP < r, i, z >_3 & \text{se } y = "" \text{ e } z \neq "", \\ "url" r 1*SP < r, i, z >_3 & \text{se } y \neq "" \text{ e } z \neq ". \end{cases}$$

onde SP é a regra definida na Tabela 4.2 e

- a) $< r, i >_1$ é a cadeia de caracteres que representa a URL de acesso ao **item de informação** relacionado ao **item identificado** com **IBI** i , de acordo com o relacionamento definido pela relação r ,
- b) $< r, i, y >_2$ é a cadeia de caracteres que representa a URL de acesso ao recurso (arquivo) especificado em y e que compõe o **item de informação** relacionado ao **item identificado** com **IBI** i , de acordo com o relacionamento definido pela relação r .
- c) $< r, i, z >_3$ é a cadeia de caracteres representando a URL de acesso à página de exibição da lista de recursos (arquivos) que compõem o **item de informação** relacionado ao **item identificado** com **IBI** i , de acordo com o relacionamento definido pela relação r , se `"GetFileList" ∈ z2`, caso

²Quando, por medida de segurança, a lista dos recursos (arquivos) que compõem um determinado **item identificado** não pode ser divulgada, a página de exibição deve informar explicitamente que o acesso a lista não está permitida.

contrário, será simplesmente a cadeia de caracteres $\langle r, i \rangle_1$.

A critério do **Arquivo**, a rotina CRIARPARURL pode usar ou não a variável global `clientinformation.ipaddress` para determinar qual URL será usada. Por exemplo, um **Arquivo** pode, por razão de licença de uso de um determinado **item de informação**, querer que uma **URL persistente** aponte para o próprio **item de informação** ou para seus metadados, dependendo do IP do usuário. Essa situação ocorre, em particular, para **Arquivos** que formam um repositório digital institucional, nos quais estão armazenados trabalhos já publicados em outros repositórios, como os das editoras científicas.

Exemplo 7.9 (Funcionamento da rotina CRIARPARURL para acessar a URL de um **item identificado**) – Com as variáveis globais `parsedibiurl.filepath` e `parsedibiurl.verblast` vazias, a execução de:

`CRIARPARURL("", "LK47B6W/362SFKH")`

resultou em uma cadeia de caracteres formando o par:

`url http://mtc-m16d.sid.inpe.br/col/iconet.com.br/banon/2009/09.09.22.01/doc/@relatorio.pdf`

□

No Exemplo 7.9, a rotina CRIARPARURL é usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O valor do par, dado por $\langle "", "LK47B6W/362SFKH" \rangle_1$, fornece a URL de acesso ao **item identificado** com **IBI** LK47B6W/362SFKH.

Exemplo 7.10 (Funcionamento da rotina CRIARPARURL para acessar a URL dos metadados de um **item identificado**) – Com as variáveis globais `parsedibiurl.filepath` e `parsedibiurl.verblast` vazias, a execução de:

`CRIARPARURL(".metadata", "8JMKD3MGP7W/3EPGUE5")`

resultou em uma cadeia de caracteres formando o par:

`url.metadata http://mtc-m16d.sid.inpe.br/col/sid.inpe.br/mtc-m19/2013/09.04.12.27.58/doc/metadata.cgi`

□

No Exemplo 7.10, a rotina CRIARPARURL é usada para criar uma cadeia de ca-

racteres que forma um par, conforme a regra **par** da Tabela 4.2. O valor do par, dado por `< ".metadata", "8JMKD3MGP7W/3EPGUE5" >1`, fornece a URL de acesso aos metadados do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5.

Exemplo 7.11 (Funcionamento da rotina `CRIARPARURL` para acessar a URL de um arquivo que compõe um **item identificado**) – Com a variável global `parsedibiurl.filepath` assumindo o valor `"reference.bib"`, a execução de:

```
CRIARPARURL("", "LK47B6W/362SFKH")
```

resultou em uma cadeia de caracteres formando o par:

```
url http://mtc-m16d.sid.inpe.br/col/iconet.com.br/banon/2009/09.09.22.01/doc/reference.bib
```

□

No Exemplo 7.11, a rotina `CRIARPARURL` é usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O valor do par, dado por `< "", "LK47B6W/362SFKH" >2`, fornece a URL de acesso ao arquivo `reference.bib` que compõe o **item identificado** com **IBI** LK47B6W/362SFKH. O nome desse arquivo é especificado na primeira variável global `parsedibiurl.filepath`. Em relação à URL criada no Exemplo 7.9, observa-se que o nome de arquivo `@relatorio.pdf` foi substituído, na URL desse novo exemplo, pelo nome de arquivo `reference.bib`. Com essa opção, foi possível substituir o nome de arquivo `@relatorio.pdf`, considerado arquivo padrão (ou arquivo alvo), por um outro nome de arquivo que compõe o **item identificado**. O exemplo seguinte mostra como acessar a lista de todos os arquivos que compõem o **item identificado**.

Exemplo 7.12 (Funcionamento da rotina `CRIARPARURL` para acessar a URL da página de exibição da lista dos arquivos que compõem um **item identificado**) – Com a variável global `parsedibiurl.verblast` contendo o valor `"GetFileList"`, a execução de:

```
CRIARPARURL("", "LK47B6W/362SFKH")
```

resultou em uma cadeia de caracteres formando o par:

```
url http://mtc-m16d.sid.inpe.br/displaydoccontent.cgi/LK47B6W/362SFKH?displaytype=FileList
```

□

No Exemplo 7.12, a rotina CRIARPARURL é usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 4.2. O valor do par, dado por `< "", "LK47B6W/362SFKH" >`₃, fornece a URL de acesso à página de exibição dos arquivos que compõem o **item identificado** com **IBI** LK47B6W/362SFKH.

Após a execução da rotina CRIARPARURL, sua saída é testada a Linha 25 do Algoritmo 2 (MONTARLISTADEURL). Se essa não for vazia, então as rotinas CRIARPAR, das Linhas 27, 29 e 31, deverão ser executadas. Isto é importante, por exemplo, para que o **resolvedor** seja informado do **estado** (original ou cópia) do **item de informação** apontado por uma determinada URL. Assim, o **sistema para resolução de IBI** poderá atender uma solicitação de **URL persistente** quando o par `ibiurl.requireditemstatus=Original` é anexado à URL de base `http://urllib.net`.

Finalmente, se não existir (ainda) uma próxima edição para o **item identificado** com **IBI** *i*, a execução de CRIARPARIBI(".lastedition" *r*, *i*) e CRIARPARURL(".lastedition" *r*, *i*), deverá retornar o mesmo valor que, respectivamente, CRIARPARIBI(*r*, *i*) e CRIARPARURL(*r*, *i*), para toda relação *r* tal que ".lastedition" *r* (usando a notação BNF (aumentada)) pertença ao subconjunto **relações**, a variável global do Algoritmo 2 (MONTARLISTADEURL).

Nas mesmas condições, a execução de CRIARPAR(*x* ".lastedition" *r*, *i*) deverá retornar o mesmo valor que CRIARPAR(*x* *r*, *i*) para todo *x* assumindo os valores "contenttype", "state" ou "timestamp".

De forma similar, se não existir (ainda) uma próxima edição para uma tradução *t* do **item identificado** com **IBI** *i*, a execução de CRIARPARIBI(*t* ".lastedition" *r*, *i*) e CRIARPARURL(*t* ".lastedition" *r*, *i*), deverá retornar o mesmo valor que, respectivamente, CRIARPARIBI(*t* *r*, *i*) e CRIARPARURL(*t* *r*, *i*), para toda relação *r* tal que "*t* ".lastedition" *r*" (usando a notação BNF (aumentada)) pertença ao subconjunto **relações**.

Nas mesmas condições, a execução de CRIARPAR(*x* *t* ".lastedition" *r*, *i*) deverá retornar o mesmo valor que CRIARPAR(*x* *t* *r*, *i*) para todo *x* assumindo os valores "contenttype", "state" ou "timestamp".

Esta convenção consiste em admitir que, se não existir (ainda) uma próxima edição, a última edição deverá ser simplesmente a edição atual (cf. comentários acompanhando o Exemplo 7.2).

Uma vez processada cada *relação* do *loop Para Cada* do Algoritmo 2 (MONTARLISTADEURL), a variável **resposta** contém, na forma de uma lista de pares, todas as propriedades conhecidas do **IBI** objeto da resolução. Seu valor é então retornado a Linha 33.

8 Exemplos de comunicação entre o resolvedor e os Arquivos

8.1 Exemplo sem modificador de IBI

No exemplo escolhido, considera-se que o **resolvedor** recebeu a mensagem “Resolver IBI” (2) da Figura 6.1, correspondente à **URL persistente**:

<http://urllib.net/8JMKD3MGP8W/35MMLL8>

Essa URL é a segunda do Exemplo 6.1.

Ao receber esta URL, o resolvedor capturou, conforme a Tabela 6.6, os seguintes pares de informação:

```
clientinformation.ipaddress {172.16.44.200 150.163.68.1}  
clientinformation.languagepreference pt-BR,fr;q=0.8,en;q=0.5,pt;q=0.3  
parsedibiurl.ibi 8JMKD3MGP8W/35MMLL8
```

O primeiro par informa o IP do usuário (172.16.44.200) e do seu procurador (150.163.68.1). O segundo par informa a preferência de idioma selecionada pelo usuário em seu navegador (aqui, o primeiro idioma de preferência do usuário é o português (pt) do Brasil (BR)). O terceiro informa o **IBI** da **URL persistente**, objeto da resolução.

Em seguida, o **resolvedor** emitiu a mensagem “Solicitar URL” (3) da Figura 6.1, endereçada a todos os **Arquivos** incluídos no **sistema para resolução de IBI**, anexando à URL de base referente a cada **Arquivo** os pares <nome>=<valor> conforme a Tabela 6.9, parte deles sendo os capturados ao receber a mensagem “Resolver IBI” (2). Por exemplo, a mensagem destinada ao **Arquivo** com endereço Web `mtc-m16c.sid.inpe.br` e **serviço de Arquivo** com **IBI** `sid.inpe.br/mtc-m18@80/2008/03.17.15.17`, foi:

```
http://mtc-m16c.sid.inpe.br/sid.inpe.br/mtc-m18@80/2008/03.17.15.17?  
clientinformation.ipaddress=172.16.44.200%20150.163.68.1&  
parsedibiurl.ibi=8JMKD3MGP8W/35MMLL8&  
servicesubject=urlRequest
```

A primeira resposta “lista de URL” (4) (cf. Figura 6.1) não vazia, recebida dos **Arquivos**, foi a lista de pares:

archiveaddress	mtc-m16c.sid.inpe.br
contenttype	Data
ibi	{rep sid.inpe.br/mtc-m18@80/2009/07.21.14.43 ibip 8JMKD3MGP8W/35MMLL8}
ibi.archiveservice	{rep sid.inpe.br/mtc-m18@80/2008/03.17.15.17}
ibi.platformsoftware	{rep dpi.inpe.br/banon/1998/08.02.08.56}
state	Original
timestamp	2009-07-21T14:43:31Z

url
urlkey

<http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.14.43/doc/CCSDS%20650.0-B-1.pdf>
1427244889-5349022633744855

Essa lista traz as propriedades disponíveis a respeito do **IBI 8JMKD3MGP8W/35MMLL8** mencionado no terceiro par (com nome **ibi**). Nessa lista, o primeiro par indica ao **resolvedor** que foi precisamente o **Arquivo** com endereço *Web* mtc-m16c.sid.inpe.br que possuía informações sobre o **IBI 8JMKD3MGP8W/35MMLL8**. Observa-se que a resposta contém, no oitavo par (com nome **url**), a URL de acesso ao **item identificado** com **IBI 8JMKD3MGP8W/35MMLL8**.

No passo seguinte, o **resolvedor** emitiu a mensagem “Agradecer” (5) endereçada ao **Arquivo** com endereço *Web* mtc-m16c.sid.inpe.br, isto é, ao **Arquivo** que forneceu a URL de acesso ao **item identificado** com **IBI 8JMKD3MGP8W/35MMLL8**. Conforme a Tabela 6.11, a URL correspondente a essa mensagem foi:

```
http://mtc-m16c.sid.inpe.br/sid.inpe.br/mtc-m18080/2008/03.17.15.17?  
clientinformation.ipaddress=172.16.44.200%20150.163.68.1&  
contenttype=Data&  
ibi=rep%20sid.inpe.br/mtc-m18080/2009/07.21.14.43%20ibip%208JMKD3MGP8W/35MMLL8&  
servicesubject=acknowledgment&  
state=Original&  
url=http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.14.43/doc/  
CCSDS%20650.0-B-1.pdf&  
url.persistent=http://urlib.net:80/8JMKD3MGP8W/35MMLL8&  
urlkey=1427244889-5349022633744855
```

A resposta “aviso” (6) (cf. Figura 6.1) do **Arquivo** com endereço *Web* mtc-m16c.sid.inpe.br foi, então, o par:

```
notice {acknowledgment received}
```

Finalmente, o **resolvedor** retornou ao navegador do usuário, para redirecionamento, a URL (7):

```
http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.14.43/doc/  
CCSDS%20650.0-B-1.pdf
```

8.2 Exemplo com modificador de IBI

No exemplo escolhido, considera-se que o **resolvedor** recebeu a mensagem “Resolver IBI” (2) da Figura 6.1, correspondente à **URL persistente**:

```
http://urlib.net/8JMKD3MGP8W/35MMLL8!:\(oai\_dc\)
```

Essa URL é a segunda do Exemplo 6.4. Concatenado ao **IBI 8JMKD3MGP8W/35MMLL8**, o modificador `!(oai_dc)` informou o **resolvedor** que devia retornar no formato *oai_dc*, os metadados da última edição do **item identificado** com **IBI 8JMKD3MGP8W/35MMLL8**.

Ao receber esta URL, o resolvedor capturou os seguintes pares de informação:

```
clientinformation.ipaddress {172.16.44.200 150.163.68.1}
clientinformation.languagepreference pt-BR,fr;q=0.8,en;q=0.5,pt;q=0.3
parsedibiurl.ibi 8JMKD3MGP8W/35MMLL8
parsedibiurl.verblast {GetLastEdition GetMetadata(oai_dc)}
```

O último par informa a lista dos dois verbos que descrevem o modificador `!(oai_dc)` da **URL persistente** (cf. Tabela 6.5).

Em seguida, o **resolvedor** emitiu a mensagem “Solicitar URL” (3) da Figura 6.1, endereçada a todos os **Arquivos** incluídos no **sistema para resolução de IBI**. Por exemplo, a mensagem destinada ao **Arquivo** com endereço *Web* `mtc-m16c.sid.inpe.br` e **serviço de Arquivo** com **IBI** `sid.inpe.br/mtc-m18@80/2008/03.17.15.17` foi:

```
http://mtc-m16c.sid.inpe.br/sid.inpe.br/mtc-m18@80/2008/03.17.15.17?
clientinformation.ipaddress=172.16.44.200%20150.163.68.11&
parsedibiurl.ibi=8JMKD3MGP8W/35MMLL8&
parsedibiurl.verblast=GetLastEdition%20GetMetadata(oai_dc)&
servicesubject=urlRequest
```

A primeira resposta “lista de URL” (4) (cf. Figura 6.1) não vazia, recebida dos **Arquivos**, foi a lista de pares:

<code>archiveaddress</code>	<code>mtc-m16c.sid.inpe.br</code>
<code>ibi</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/07.21.14.43 ibip 8JMKD3MGP8W/35MMLL8}</code>
<code>ibi.archiveservice</code>	<code>{rep sid.inpe.br/mtc-m18@80/2008/03.17.15.17}</code>
<code>ibi.nextedition</code>	<code>{rep sid.inpe.br/mtc-m18/2012/07.12.18.08}</code>
<code>ibi.platformsoftware</code>	<code>{rep dpi.inpe.br/banon/1998/08.02.08.56}</code>

Essa lista traz as propriedades disponíveis a respeito do **IBI 8JMKD3MGP8W/35MMLL8**. Entretanto, observa-se que a resposta não contém, como esperado, o par com nome `url.lastedition.metadata(oai_dc)`, cujo valor seria a URL de acesso aos metadados, no formato *oai_dc*, da última edição do **item identificado** com **IBI 8JMKD3MGP8W/35MMLL8**. Nem contém o par com nome `ibi.lastedition`, cujo valor seria o **IBI** da última edição do **item identificado**. No entanto, a lista contém o par com nome `ibi.nextedition` (quarto par), cujo valor é o **IBI** `sid.inpe.br/mtc-m18/2012/07.12.18.08` de uma nova edição do **item identificado**. Nesse caso então, o **resolvedor** teve que emitir uma nova mensagem

“Solicitar URL” (3) endereçada novamente a todos os **Arquivos** para tentar resolver o **IBI** `sid.inpe.br/mtc-m18/2012/07.12.18.08`. Desta vez, entre todas as mensagens emitidas, a mensagem destinada ao **Arquivo** com endereço *Web* `mtc-m16c.sid.inpe.br` foi:

```
http://mtc-m16c.sid.inpe.br/sid.inpe.br/mtc-m18080/2008/03.17.15.17?
clientinformation.ipaddress=172.16.44.200%20150.163.68.1&
parsedibiurl.ibi=sid.inpe.br/mtc-m18/2012/07.12.18.08&
parsedibiurl.verblast=GetLastEdition%20GetMetadata(oai_dc)&
servicesubject=urlRequest
```

Comparando a primeira solicitação “Solicitar URL” (3) com esta segunda, observa-se que no par com nome `parsedibiurl.ibi`, o valor `sid.inpe.br/mtc-m18/2012/07.12.18.08` (**IBI** da próxima edição) está substituindo o valor `8JMKD3MGP8W/35MMLL8` (**IBI** da edição anterior).

A primeira resposta “lista de URL” (4) (cf. Figura 6.1) não vazia, recebida dos **Arquivos**, foi a lista de pares:

<code>archiveaddress</code>	<code>mtc-m16c.sid.inpe.br</code>
<code>contenttype.lastedition.metadata(oai_dc)</code>	<code>Metadata</code>
<code>ibi</code>	<code>{rep sid.inpe.br/mtc-m18/2012/07.12.18.08 ibip 8JMKD3MGP8W/3C9EP6P}</code>
<code>ibi.archiveservice</code>	<code>{rep sid.inpe.br/mtc-m18080/2008/03.17.15.17}</code>
<code>ibi.lastedition.metadata(oai_dc)</code>	<code>{rep sid.inpe.br/mtc-m18/2012/07.12.18.08.49}</code>
<code>ibi.platformsoftware</code>	<code>{rep dpi.inpe.br/banon/1998/08.02.08.56}</code>
<code>state.lastedition.metadata(oai_dc)</code>	<code>Original</code>
<code>timestamp.lastedition.metadata(oai_dc)</code>	<code>2014-04-04T17:36:01Z</code>
<code>url.lastedition.metadata(oai_dc)</code>	<code>http://mtc-m16c ... br/col/sid.inpe.br/mtc-m18/2012/07.12.18.08.49/doc/metadata.cgi?choice=oai_dc</code>
<code>urlkey</code>	<code>1426286454-36108967764060357</code>

Essa nova lista traz as propriedades disponíveis a respeito do **IBI** `sid.inpe.br/mtc-m18/2012/07.12.18.08`, mencionado no terceiro par (com nome `ibi`). Nessa lista, o primeiro par indica ao **resolvedor** que o **Arquivo** com endereço *Web* `mtc-m16c.sid.inpe.br` possuía informações sobre o **IBI** `sid.inpe.br/mtc-m18/2012/07.12.18.08`. Entretanto, agora, em vez e exibir o **IBI** para uma próxima edição¹, a resposta contém no nono par (com nome `url.lastedition.metadata(oai_dc)`) a URL de acesso aos metadados, no formato *oai_dc*, da última edição do **item identificado** com **IBI** `sid.inpe.br/mtc-m18/2012/07.12.18.08`. O que encerrou a sequência de solicitações de URL, lembrando que a última edição de qualquer edição com **IBI** *i* é também a última edição de qualquer outra edição com **IBI** *j*.

No passo seguinte, o **resolvedor** emitiu a mensagem “Agradecer” (5) endereçada ao **Arquivo** com endereço *Web* `mtc-m16c.sid.inpe.br`, isto é, ao **Arquivo** que

¹Isto indica que o **item identificado** com **IBI** `sid.inpe.br/mtc-m18/2012/07.12.18.08` não possui uma próxima edição, ou seja a última edição deste **item** é ele próprio.

forneceu a URL de acesso aos metadados, no formato *oai_dc*, da última edição do **item identificado** com **IBI** 8JMKD3MGP8W/35MMLL8. A URL correspondente a essa mensagem foi:

```
http://mtc-m16c.sid.inpe.br/sid.inpe.br/mtc-m18080/2008/03.17.15.17?
clientinformation.ipaddress=172.16.44.200%20150.163.68.1&
contenttype=Metadata&
ibi=rep%20sid.inpe.br/mtc-m18/2012/07.12.18.08.49&
servicesubject=acknowledgment&
state=Original&
url=http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18/2012/07.12.18.08.49/doc/
metadata.cgi%3Fchoice%3Doai_dc&
url.persistent=http://urlib.net:80/8JMKD3MGP8W/35MMLL8!:(oai_dc)&
urlkey=1426286454-36108967764060357
```

A resposta “aviso” (6) (cf. Figura 6.1) do **Arquivo** com endereço *Web* *mtc-m16c.sid.inpe.br* foi, então, o par:

```
notice {acknowledgment received}
```

Finalmente, o **resolvedor** retornou ao navegador do usuário, para redirecionamento, a URL (7):

```
http://mtc-m16c.sid.inpe.br/col/sid.inpe.br/mtc-m18/2012/07.12.18.08.49/doc/
metadata.cgi?choice=oai_dc
```


REFERÊNCIAS BIBLIOGRÁFICAS

BANON, G. J. F. **Formal introduction to digital image processing**. 2. ed. São José dos Campos: INPE, 2000. vii + 180 p. Second edition. This material was used in 2000, 2001 and 2002 as class notes for the INPE posgraduate course CAP-404. This work has been supported by CNPq under contract 300966/90-3. Disponível em: <[./ibi:83LX3pFwXQZeBBx/hwiTk](#)>. Acesso em: 07 out. 2014. 12

_____. **Identificador com base na Internet (IBI): Sistema de identificação**. São José dos Campos: INPE, 2011. 40 p. Disponível em: <[./ibi:LK47B6W/362SFKH](#)>. Acesso em: 23 ago. 2014. 1, 2, 6, 16, 19, 22, 25, 26, 36, 52, 53

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. 15

BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. **Uniform Resource Identifiers (URI): Generic syntax**. Washington DC: The Internet Engineering Task Force (IETF), Aug. 2005. 61 p. RFC 3986. Disponível em: <<http://tools.ietf.org/html/rfc3986>>. Acesso em: 28 jan. 2015. 15, 22, 26, 36, 38

Comissão-de-Estudos ABNT/CB08/SC010/CE70. **Inglês Codificado em ASCII - CCSDS 643.0-B-1**. São José dos Campos: [s.n.], 2010. 13 p. Disponível em: <[./ibi:8JMKD3MGP8W/35TE8FS](#)>. Acesso em: 10 jan. 2015. 16, 17

CORPORATION FOR NATIONAL RESEARCH INITIATIVES (CNRI). **Handle System Fundamentals**. 2015. Disponível em: <http://www.handle.net/overviews/system_fundamentals.html>. Acesso em: 31 mar. 2015. 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. 17

CROCKER, D. H.; OVERELL, P. **Augmented BNF for Syntax Specifications: ABNF**. Washington DC: The Internet Society, Nov. 1997. RFC

2234. Disponível em: <<http://tools.ietf.org/html/rfc2234>>. Acesso em: 13 dec. 2014. 17

_____. _____. 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. 17

GUEDES, G. T. A. **UML2: Uma abordagem prática**. 2. ed. São Paulo: Novatec Editora, 2011. 8, 9

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **ISO 639-1: Codes for the representation of names of languages – Part 1: Alpha-2 code**. Geneva, 2002. 75

_____. **ISO 3166-1: Codes for the representation of names of countries and their subdivisions – Part 1: Country codes**. Geneva, 2013. 75

LAGOZE, C.; SOMPEL, H. V. d.; NELSON, M.; WARNER, S. **The Open Archives Initiative Protocol for Metadata Harvesting**. Ithaca, NY: Cornell University, Jun. 2002. Protocol Version 2.0 of 2002-06-14 - Document Version 2015-01-08. Disponível em: <<http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>>. Acesso em: 14 mar. 2015. 27

WEIBEL, S.; JUL, E.; SHAFER, K. **PURLs: Persistent Uniform Resource Locators**. OCLC Online Computer Library Center, 2015. Disponível em: <http://purl.org/docs/brief_intro.html>. Acesso em: 31 mar. 2015. 2

APÊNDICE A - Regras idioma e país

O significado de cada dupla de caracteres pode ser consultado em [ISO \(2002\)](#) para idioma e [ISO \(2013\)](#) para país.

Tabela A.1 - Regras definindo a formação de idioma e país

idioma =	"aa"	"ab"	"ae"	"af"	"ak"	"am"	"an"	"ar"	"as"	"av"	
	"ay"	"az"	"ba"	"be"	"bg"	"bh"	"bi"	"bm"	"bn"	"bo"	
	"br"	"bs"	"ca"	"ce"	"ch"	"co"	"cr"	"cs"	"cu"	"cv"	
	"cy"	"da"	"de"	"dv"	"dz"	"ee"	"el"	"en"	"eo"	"es"	
	"et"	"eu"	"fa"	"ff"	"fi"	"fj"	"fo"	"fr"	"fy"	"ga"	
	"gd"	"gl"	"gn"	"gu"	"gv"	"ha"	"he"	"hi"	"ho"	"hr"	
	"ht"	"hu"	"hy"	"hz"	"ia"	"id"	"ie"	"ig"	"ii"	"ik"	
	"io"	"is"	"it"	"iu"	"ja"	"jv"	"ka"	"kg"	"ki"	"kj"	
	"kk"	"kl"	"km"	"kn"	"ko"	"kr"	"ks"	"ku"	"kv"	"kw"	
	"ky"	"la"	"lb"	"lg"	"li"	"ln"	"lo"	"lt"	"lu"	"lv"	
	"mg"	"mh"	"mi"	"mk"	"ml"	"mn"	"mr"	"ms"	"mt"	"my"	
	"na"	"nb"	"nd"	"ne"	"ng"	"nl"	"nn"	"no"	"nr"	"nv"	
	"ny"	"oc"	"oj"	"om"	"or"	"os"	"pa"	"pi"	"pl"	"ps"	
	"pt"	"qu"	"rm"	"rn"	"ro"	"ru"	"rw"	"sa"	"sc"	"sd"	
	"se"	"sg"	"si"	"sk"	"sk"	"sl"	"sm"	"sn"	"so"	"sq"	
	"sr"	"ss"	"st"	"su"	"sv"	"sw"	"ta"	"te"	"tg"	"th"	
	"ti"	"tk"	"tl"	"tn"	"to"	"tr"	"ts"	"tt"	"tw"	"ty"	
	"ug"	"uk"	"ur"	"uz"	"ve"	"vi"	"vo"	"wa"	"wo"	"xh"	
	"yi"	"yo"	"za"	"zh"	"zu"						
país =	"AD"	"AE"	"AF"	"AG"	"AI"	"AL"	"AM"	"AO"	"AQ"	"AR"	
	"AS"	"AT"	"AU"	"AW"	"AX"	"AZ"	"BA"	"BB"	"BD"	"BE"	
	"BF"	"BG"	"BH"	"BI"	"BJ"	"BL"	"BM"	"BN"	"BO"	"BQ"	
	"BR"	"BS"	"BT"	"BV"	"BW"	"BY"	"BZ"	"CA"	"CC"	"CD"	
	"CF"	"CG"	"CH"	"CI"	"CK"	"CL"	"CM"	"CN"	"CO"	"CR"	
	"CU"	"CV"	"CW"	"CX"	"CY"	"CZ"	"DE"	"DJ"	"DK"	"DM"	
	"DO"	"DZ"	"EC"	"EE"	"EG"	"EH"	"ER"	"ES"	"ET"	"FI"	
	"FJ"	"FK"	"FM"	"FO"	"FR"	"GA"	"GB"	"GD"	"GE"	"GF"	
	"GG"	"GH"	"GI"	"GL"	"GM"	"GN"	"GP"	"GQ"	"GR"	"GS"	
	"GT"	"GU"	"GW"	"GY"	"HK"	"HM"	"HN"	"HR"	"HT"	"HU"	
	"ID"	"IE"	"IL"	"IM"	"IN"	"IO"	"IQ"	"IR"	"IS"	"IT"	
	"JE"	"JM"	"JO"	"JP"	"KE"	"KG"	"KH"	"KI"	"KM"	"KN"	
	"KP"	"KR"	"KW"	"KY"	"KZ"	"LA"	"LB"	"LC"	"LI"	"LK"	
	"LR"	"LS"	"LT"	"LU"	"LV"	"LY"	"MA"	"MC"	"MD"	"ME"	
	"MF"	"MG"	"MH"	"MK"	"ML"	"MM"	"MN"	"MO"	"MP"	"MQ"	
	"MR"	"MS"	"MT"	"MU"	"MV"	"MW"	"MX"	"MY"	"MZ"	"NA"	
	"NC"	"NE"	"NF"	"NG"	"NI"	"NL"	"NO"	"NP"	"NR"	"NU"	
	"NZ"	"OM"	"PA"	"PE"	"PF"	"PG"	"PH"	"PK"	"PL"	"PM"	
	"PN"	"PR"	"PS"	"PT"	"PW"	"PY"	"QA"	"RE"	"RO"	"RS"	
	"RU"	"RW"	"SA"	"SB"	"SC"	"SD"	"SE"	"SG"	"SH"	"SI"	
	"SJ"	"SK"	"SL"	"SM"	"SN"	"SO"	"SR"	"SS"	"ST"	"SV"	
	"SX"	"SY"	"SZ"	"TC"	"TD"	"TF"	"TG"	"TH"	"TJ"	"TK"	
	"TL"	"TM"	"TN"	"TO"	"TR"	"TT"	"TV"	"TW"	"TZ"	"UA"	
	"UG"	"UM"	"US"	"UY"	"UZ"	"VA"	"VC"	"VE"	"VG"	"VI"	
	"VN"	"VU"	"WF"	"WS"	"YE"	"YT"	"ZA"	"ZM"	"ZW"		