



ABNT: ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS
CB: 08 COMITÊ BRASILEIRO DE AERONÁUTICA E ESPAÇO
SC: 08.001 SUBCOMITÊ DE ATIVIDADE ESPACIAL
CE: 08:010.70 COMISSÃO DE ESTUDO EM SISTEMAS ESPACIAIS DE
TRANSFERÊNCIA DE DADOS E DE INFORMAÇÃO

Sistemas Espaciais de Transferência de Dados e de Informação

Sistema para resolução de IBI

Relatório Técnico da Comissão – 15 (RTC-15)
– Versão Editorial – 2: 5 de Outubro de 2015 –

CE 08:010.70 Comissão de Estudos em Sistemas Espaciais
de Transferência de Dados e de Informação

**SISTEMAS ESPACIAIS DE TRANSFERÊNCIA DE DADOS E DE INFORMAÇÃO
SISTEMA PARA RESOLUÇÃO DE IBI**

COLABORADORES DESTA EDIÇÃO

A editoração deste documento interno, denominado Relatório Técnico da Comissão N° 15, ou RTC – 15, contou com a participação seguintes MEMBROS da Comissão de Estudo de Sistemas Espaciais de Transferência de Dados e de Informação – CE 08:010.70, da ABNT:

Cíntia Borges Margi	Membro	LARC/EPUSP
Danilo C. Carvalho	Membro	ANATEL
Eduardo W. Bergamini	Coordenador	INPE/MCTI
Gerald J. F. Banon	Membro	INPE/MCTI
João Manoel R. Zaninotto	Membro	EMBRAER
Leandro Vaz Barros Reis	Membro	ANATEL
Marco Antonio Grivet M. Maia	Membro	PUC-RIO
Marília Vidigal da Costa Souza	Membro	EMBRAER
Maurício G. Vieira Ferreira	Membro	INPE/MCTI
Regina M. Silveira	Observadora	LARC/EPUSP
Reginaldo Palazzo Júnior	Membro	UNICAMP
Sérgio Costa	Membro	IAE/DCTA
Valéria Cristina M. N. Leite	Membro	IAE/DCTA
Rogério Pirk	Membro	IAE/DCTA
Lília de Sá Silva	Membro	INPE
Sergio Fugivara	Membro	IAE/DCTA
Walter Abrahão dos Santos	Membro	INPE
Roberto Roma de Vasconcellos	Membro	IAE/DCTA
Mateus Mosca Viana	Membro	UNIFOR
Marcello Silva de Abreu	Membro	EMBRAER
Antônio Cassiano Filho	Membro	INPE

A Secretária do órgão RME/TEC do INPE que secretaria esta Comissão de Estudo da ABNT, realizou extenso trabalho de apoio de editoração e de correspondência, na elaboração deste documento:

Helen Joyce Aparecida	Secretária	RME/TEC/INPE
-----------------------	------------	--------------

São José dos Campos. Maio de 2015.

NOTA DO AUTOR

Este documento reflete o resultado de um longo processo de amadurecimento de uma proposta de definição de um sistema de identificação de itens de informação por meio dos chamados IBI (Identificadores com base na Internet), e de um sistema de resolução dos identificadores desses itens.

A idéia original do sistema de identificação, concebida em 1995, procura aproveitar a própria infra-estrutura já existente, oferecida pela Internet.

Desde a sua concepção, a realização dessa idéia foi sendo, aos poucos, aperfeiçoada. Deste processo todo, resultou agora a presente proposta de um sistema de resolução de IBI, que já se encontra amplamente testada numa plataforma computacional chamada de *URLib*. A plataforma *URLib* e aquela adotada pelo INPE para hospedar sua Memória Científica.

Vale mencionar que, embora esse sistema de resolução tenha sido primariamente concebido para resolver os IBI de itens de informação armazenados na plataforma *URLib*, no caso, sob um caráter normalizado, o sistema de resolução tal como definido neste documento, pode vir a ser também utilizado para resolver os IBI de itens armazenados em qualquer outra plataforma, portanto, também de forma normalizada, em princípio.

O documento original que deu origem a esta Norma que está sendo recomendada possui o título: “Identificador com base na Internet (IBI): Sistema de resolução”. Ele foi publicado pelo INPE na forma de relatório de pesquisa, sob o código de identificação: urlib.net/www/2014/03.25.23.20-RPQ. Ele está acessível a partir de apontamento no endereço URL: <http://urlib.net/J8LNKB5R7W/3G2EKR5>.

© CE 08:010.70/ABNT, INPE/MCT, Sao Jose dos Campos, SP, Brasil – Maio de 2015.

ABSTRACT

This standard complements an earlier standard entitled “Sistema para geração de IBI” (System for generating IBI). The first standard 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 standard 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 standard also presents a mini-language to 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 standard is given to the design of the Archives services aimed at the IBI resolution, bringing recommendations of how these services should be built.

LISTA DE FIGURAS

	<u>Pág.</u>
5.1 Componentes e atores do Sistema para resolução de IBI	9
5.2 Diagrama de casos de uso do Sistema para resolução de IBI	10
5.3 Diagrama de classes	11
5.4 Classe ItemIdentificado como classe-mãe	13
7.1 Diagrama de sequência do processo de inclusão/exclusão de Arquivo . .	21
8.1 Diagrama de sequência do processo de resolução de IBI	33

LISTA DE TABELAS

	<u>Pág.</u>
6.1 Codificação hexadecimal de caracteres especiais.	16
6.2 Regras definindo a formação de uma <code>lista-de-pares</code>	18
7.1 Regras definindo a formação de uma chave de cadastro	19
7.2 Pares <code><nome>=<valor></code> a serem anexados à URL de base para as mensagens “Incluir Arquivo” (2) e “Excluir Arquivo” (7) endereçadas ao resolvedor	22
8.1 Regras definindo a formação de uma <code>URL-persistente</code>	26
8.2 Modificadores de URL persistente	28
8.3 Pares <code><nome>=<valor></code> interpretados pelo sistema para resolução de IBI	29
8.4 Regras definindo a formação de uma <code>lista-de-verbos</code>	30
8.5 Verbos atuando como modificador elementar de URL persistente	30
8.6 Dados capturados pelo resolvedor ao receber uma URL persistente	34
8.7 Dados sobre o usuário.	35
8.8 Dados extraídos da URL persistente	35
8.9 Pares <code><nome>=<valor></code> a serem anexados à URL de base para a mensagem “Solicitar URL” (3) endereçada ao Arquivo	36
8.10 Regras definindo a formação de uma <code>lista-de-IP</code>	36
8.11 Pares <code><nome>=<valor></code> a serem anexados à URL de base para a mensagem “Agradecer” (5) endereçada ao Arquivo	38
9.1 Componentes do Algoritmo 1 (MONTARRESPOSTA).	42
9.2 Componentes do Algoritmo 2 (MONTARLISTADEURL).	49
9.3 Descrição da função <code>< x, i ></code>	52
9.4 Regras definindo a formação de <code>formas-de-ibi</code>	53
9.5 Regras definindo a formação de uma <code>relação</code>	54
B.1 Regras definindo a formação de <code>idioma e país</code>	75

SUMÁRIO

	<u>Pág.</u>
1 Introdução	1
2 Escopo	3
3 Justificativa	5
4 Termos e definições	7
5 Descrição do sistema para resolução de IBI	9
5.1 Os componentes e os atores do sistema	9
5.2 Itens originais e cópias	11
6 Protocolo de comunicação	15
6.1 Formato de solicitação de serviço	15
6.2 Formato de resposta de serviço	17
7 Inclusão e exclusão de Arquivo	19
7.1 Cadastrar um Arquivo	19
7.2 Incluir/excluir um Arquivo	20
8 Resolução de IBI	25
8.1 URL persistente	25
8.2 Processo de resolução	32
9 Serviço de Arquivo	41
9.1 Solicitações	41
9.2 Respostas	41
10 Exemplos de comunicação entre o resolvedor e os Arquivos	65
10.1 Exemplo sem modificador de IBI	65
10.2 Exemplo com modificador de IBI	66

REFERÊNCIAS BIBLIOGRÁFICAS	71
APÊNDICE A - Definição de relação binária funcional e esquerda- total	73
APÊNDICE B - Regras idioma e país	75

1 Introdução

Esta norma complementa uma norma anterior intitulado “Sistema para geração de IBI”. A primeira norma 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.

Esta segunda norma 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. A norma ainda apresenta uma mini-linguagem para definir “modificadores” de IBI que estendem os acessos à 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 armazenem os itens de informação e de um resolvedor que intermedia a comunicação entre os navegadores e os Arquivos. A ênfase nessa norma é dada à concepção dos serviços de Arquivos voltados para a resolução e IBI, trazendo recomendações de como esses serviços devem ser construídos.

As diversas aplicações de um sistema de resolução de um identificador global desta natureza são também de particular interesse para uso em sistemas de dados espaciais e de informação.

2 Escopo

Esta norma descreve e permite a uma entidade de armazenamento de itens de informação, chamado de Arquivo, participar do sistema de resolução que associa a cada IBI de um item de informação, sua localização na Internet. O protocolo de comunicação entre o resolvidor, que recebe as solicitações de resolução de URL persistente, e os Arquivos é apresentado neste documento.

3 Justificativa

Os hipervínculos (*hyperlinks*) ou simplesmente vínculos ou ponteiros, elementos essenciais na navegação entre itens de informação (documentos, mapas, imagens, programas de computadores, etc.) disponíveis na Internet, devem ter 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 à 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 em dois níveis descrito em [Comissão-de-Estudos ABNT/CB08/SC010/CE70 \(2011\)](#) junto ao sistema de resolução descrito nesta norma, ambos extensivamente utilizados na plataforma *URLib*, o primeiro desde 1995 e o segundo desde 1998, apresentam-se como uma alternativa simples, quando comparada 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, 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 detedores 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. Pois, em caso de um original ser reivindicado por mais de

um dono ou detendor¹, 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 8.2).

Quanto à autoria de uma obra inédita depositada em um Arquivo desse 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 mesmo IBI, acessos persistentes a itens de informação futuros, como novas traduções ou a última edição de um item de informação já existente.

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

4 Termos e definições

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

Arquivo: 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**, e 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**, e usado na comunicação entre o **resolvedor** o **Arquivo**.

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

IBI: sigla para “Identificador com Base na Internet”. Qualquer **rótulo** gerado pelo **sistema de identificação em dois níveis**.

IBIp: qualquer **rótulo** gerado pelo **sistema de identificação em dois níveis**, 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 de 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 de como estão armazenados os **itens identificados**.

Nome de repositório uniforme: qualquer **rótulo** gerado pelo **sistema de identificação em dois níveis**, utilizando o endereçamento por nome de domínio.

Resolver: componente do **sistema para resolução de IBI** que recebe as **URL persistentes** e as redirecionam 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: o serviço *Web* oferecido por um **Arquivo** ao **sistema para resolução de IBI**. Em cada **Arquivo**, esse serviço é único, intransferível e identificado por um **IBI**.

Serviço de resolver: o 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 desse item**.

Sistema de identificação em dois níveis: **sistema de identificação** objeto da norma pela [Comissão-de-Estudos ABNT/CB08/SC010/CE70 \(2011\)](#).

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 esse **rótulo**.

Sistema para resolução de IBI: **sistema de resolução** objeto dessa norma. Inverso do **Sistema de identificação em dois níveis**.

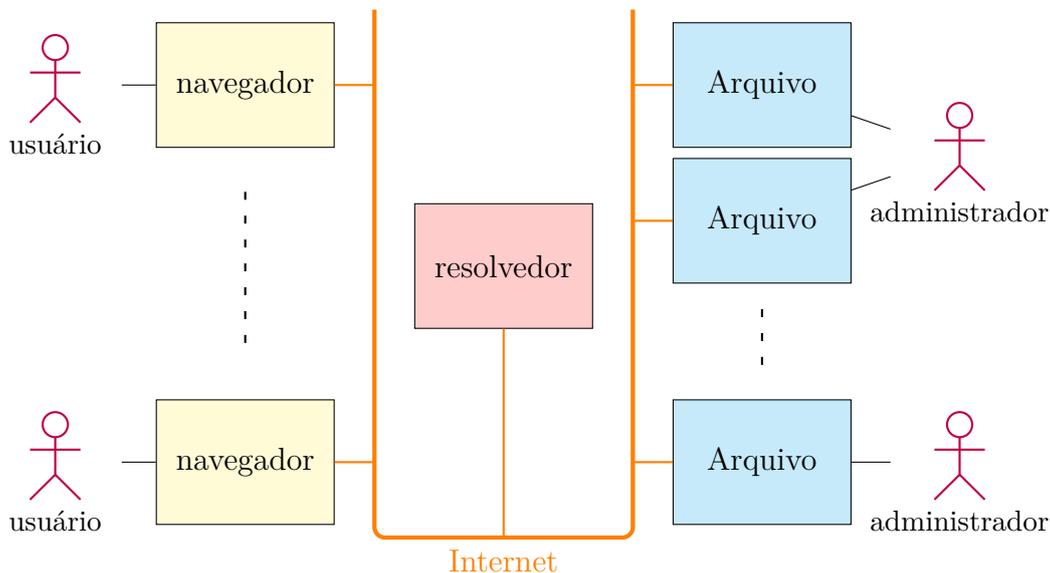
URL persistente: URL que dá acesso a um **item de informação**, independentemente de sua localização na *Web*.

5 Descrição do sistema para resolução de IBI

5.1 Os componentes e os 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 5.1.

Figura 5.1 - 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 objetos a serem encontrados com base nos seus **IBI**. Os **Arquivos** estão sob a responsabilidade dos seus administradores.

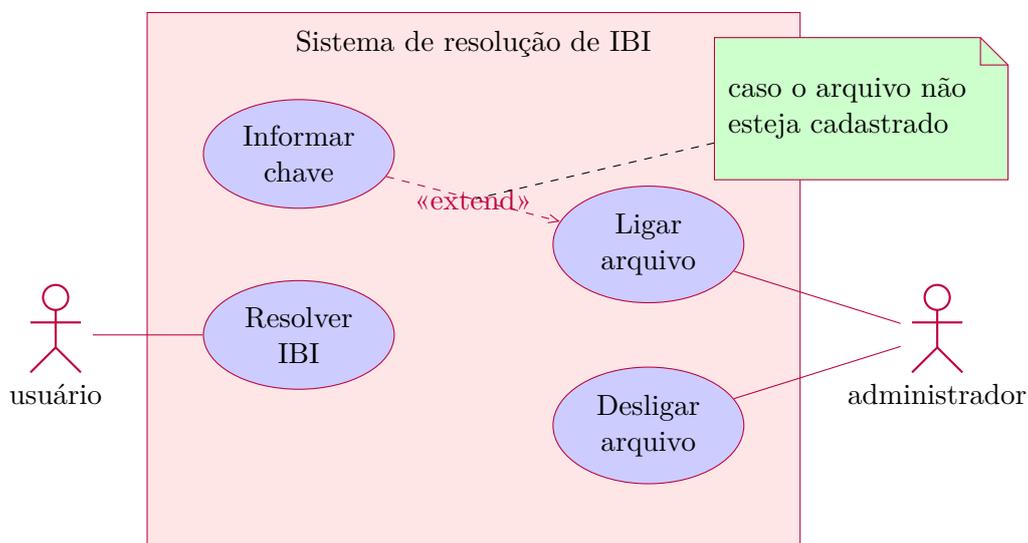
O **resolvedor** é a entidade que apenas intermedia a comunicação entre os navegadores e os **Arquivos**. O **resolvedor** pode ou não armazenar a informação sobre a

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

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

A Figura 5.2 apresenta o **sistema para resolução de IBI** e seus atores por meio de um diagrama de casos de uso UML (GUEDES, 2011)¹.

Figura 5.2 - 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 maiúscula na palavra **Arquivo**, reservando esta notação para representar a classe dos arquivos.

dastró (cf. detalhes na Seção 7.1).

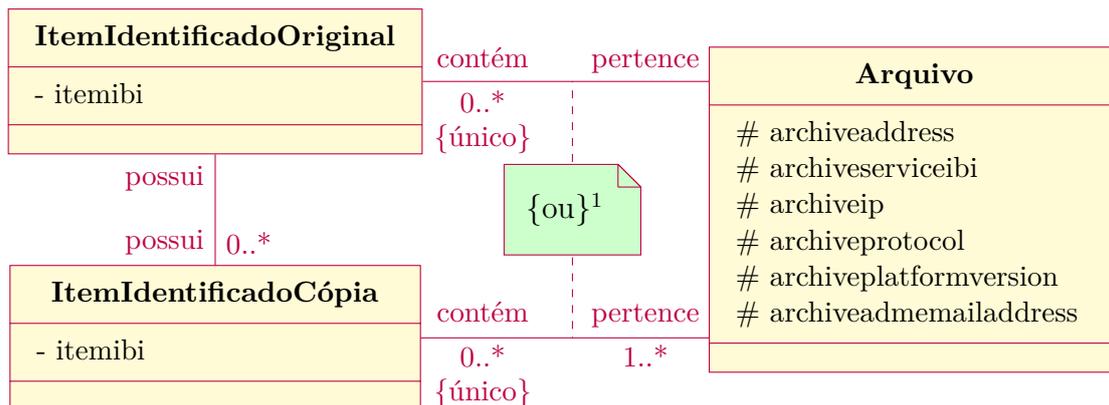
A hospedagem dos Arquivos em servidores distintos permite que administradores independentes participem do processo de resolução.

5.2 Itens originais e 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 armazenada em) apenas um único **Arquivo**, eventuais cópias de um mesmo original podem pertencer a (ou estar armazenada em) mais de um **Arquivo** desde que cada um contenha apenas um **item identificado** com o mesmo **IBI**, seja ele considerado original ou cópia.

Essa configuração está ilustrada na Figura 5.3 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**.

Figura 5.3 - Diagrama de classes



As três classes e suas associações.

¹A restrição “ou” (exclusivo) é relativa a associação “contém”, e aplica-se somente a objetos possuindo o mesmo **IBI**.

Fonte: Produção do autor.

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 original desde que apenas um seja identificado pelo mesmo **IBI** (cf. a primeira restrição {único} na Figura 5.3).

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 apenas um seja identificado pelo mesmo **IBI** (cf. a segunda restrição {único} na Figura 5.3).

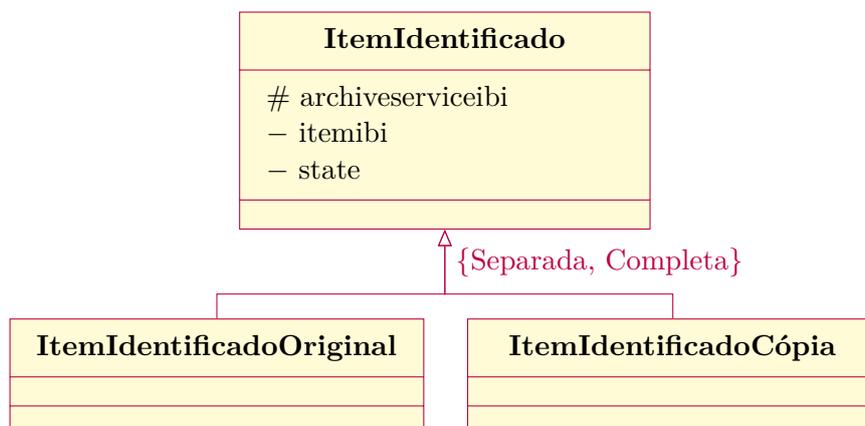
A nota em verde, na Figura 5.3, indica que se um **Arquivo** conter um **item identificado** tido como original, então ele não poderá conter um **item identificado** considerado cópia e possuindo o mesmo **IBI**, e vice e versa.

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 único **item identificado** tido como original.

Um **item identificado** pode ser considerado somente original ou cópia (completude), mas não pode ser considerado as duas coisas juntas (separabilidade). Com essa propriedade, tem-se o conceito de classe-mãe **ItemIdentificado**, como mostrada na Figura 5.4. Nessa classe, aparecem o atributo “archiveserviceibi” que especifica em que **Arquivo** está armazenado um determinado objeto, o atributo “itemibi” que informa o **IBI** do objeto, e o atributo “state” que assumirá o valor “Original”, se o objeto for um **item identificado** tido como original, ou o valor “Copy”, caso ele 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 (“archiveserviceibi”) e por seus **estados** (“state”), os quais satisfazem a relação denominada **modo de armazenamento** e definida a seguir.

Figura 5.4 - Classe **ItemIdentificado** como classe-mãe



Os objetos 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.

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

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

Seja S o conjunto {Original, 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 do **modo de armazenamento**, a relação binária R_{Original} definida como tripleto $(I, A, G_{\text{Original}})$, cujo grafo é o conjunto $G_{\text{Original}} \triangleq \{(i, a) \in I \times A : (i, a, \text{Original}) \in G\}$, é funcional e esquerda-total (Apêndice A).

Em outros termos, 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 nesse resultado, 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, a 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:

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

é também funcional e esquerda-total (Apêndice A).

Assim, 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 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.

6 Protocolo de comunicação

6.1 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 5.1) 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://urllib.net/J8LNKB5R7W/3FUQHC5
```

onde `urllib.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 7.1).

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 [Comissão-de-Estudos ABNT/CB08/SC010/CE70 \(2011\)](#).

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 existem, 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, 1992](#)). Alguns caracteres considerados especiais nos componentes `valor` devem ser codificados, como mostrado na Tabela 6.1. Os caracteres especiais são aqueles reservados para delimitar a URL de base dos componentes anexados à esta (?), como também para delimitar os constituintes desses componentes (= e &), criar espaço em branco (*espaço em branco* e +) ou codificar no sistema hexadecimal caracteres que não sejam ASCII, usando o prefixo percentagem (%).

Tabela 6.1 - Codificação hexadecimal de caracteres especiais.

<i>Caractere</i>	<i>Codificação</i>
<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 + deverá ser codificado: 1997-07-16T19:20%2B01:00.

Exemplo 6.1 (mensagem de solicitação de serviço) – URL:

```
http://mtc-m19.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 6.1, a URL foi a mensagem de solicitação de serviço endereçada ao **Arquivo** cujo endereço *Web* era `mtc-m19.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` definiu o assunto do serviço solicitado (aqui `urlRequest`), e os pares `parsedibiurl.ibi=8JMKD3MGP7W/3EPGUE5` e `clientinformation.ipaddress=150.163.2.175` definiram os parâmetros da solicitação.

6.2 Formato de resposta de serviço

Toda resposta a uma solicitação HTTP feita para um serviço do **sistema para resolução de IBI** deve ser gerada com `Content-Type` assumindo o valor `text/plain`, e deve ter o formato de uma “lista de pares”, cuja formação segue as regras gramaticais definidas na Tabela 6.2. Nessa 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 6.2 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 junto um `par`. Os pares entre si estão separados por `espaço`. O `nome` deve ser uma cadeia de `caractere` sem `espaço`. Por sua vez, o `valor` deve ser uma cadeia de `caractere` sem espaço em branco (SP), a não ser que começa com `{`, e termina com `}`.

Pela regra `caractere` da Tabela 6.2, 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, 1992), e, em consequência disto, serão os únicos caracteres da resposta de qualquer serviço do **sistema para resolução de IBI**.

Se precisar usar caracteres não codificados no sistema ASCII, cada um destes deverá ser representado por um par de algarismos hexadecimais precedido pelo símbolo `%`, sendo que cada par corresponde ao número decimal entre 0 e 255, usado na

¹Nessa norma, a notação BNF é usada com a seguinte alteração: `"|"` é utilizado para representar alternativas no lugar de `"/"`

Tabela 6.2 - 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           = %x0D; nova linha
palavra      = 1*caractere; palavra sem espaço em branco, ex: a1
caractere    = %x21-7A | %x7C | %x7E; caracteres ASCII menos { e }

```

codificação do caractere, no sistema de codificação adotado. Por exemplo, o caractere ó, parte do alfabeto Latino, codificado no sistema iso8859-1 pelo valor decimal 243, será representado pelo par %f3.

Exemplo 6.2 (mensagem de resposta à uma solicitação de serviço) – Lista de pares:

```

archiveaddress  mtc-m19.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-m19.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 6.2, a “lista de pares” foi a resposta à mensagem de solicitação do Exemplo 6.1. Essa 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 codificado no sistema iso8859-1 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 à Tabela 6.1.

7 Inclusão e exclusão de Arquivo

7.1 Cadastrar um Arquivo

Do ponto de vista do funcionamento do **sistema para resolução de IBI**, o papel do administrador de **Arquivo** resume-se em ligá-lo e desligá-lo. No entanto, para que o **Arquivo** possa a passar a integrar o **sistema para resolução de IBI**, o administrador deve inicialmente registrar junto a esse sistema uma **chave de cadastro** de sua escolha.

A Tabela 7.1 contém, conforme à notação BNF (aumentada), as duas 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 7.1 - 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 administrador de **Arquivo** deve encaminhar a **chave de cadastro** escolhida por meio de um *e-mail* endereçado ao administrador do **sistema para resolução de IBI**¹.

Para se identificar, o administrador de **Arquivo** deve encaminhar, junto com a **chave de cadastro**, o **IBI** (Comissão-de-Estudos ABNT/CB08/SC010/CE70, 2011) do **item identificado** tido como original intransferível, e denominado “serviço de **Arquivo** para resolução de **IBI**” ou simplesmente **serviço de Arquivo**, previamente criado nesse **Arquivo** (cf. Seção 9).

Exemplo 7.1 (*e-mail* de cadastro de um **Arquivo**) – Assunto:

Cadastro de um novo Arquivo

Conteúdo:

IBI do serviço de Arquivo:

archiveserviceibi = dpi.inpe.br/banon/1999/01.09.22.14

¹O endereço de *e-mail* do administrador do **sistema para resolução de IBI** é: support@urlib.net

Chave de cadastro:
registrationkey = 1234567890

□

No Exemplo 7.1, o conteúdo do *e-mail* informou que o **Arquivo** com o **serviço de Arquivo**, identificado com **IBI dpi.inpe.br/banon/1999/01.09.22.14**, esteve solicitando seu cadastramento pelo administrador do **sistema para resolução de IBI**, com a **chave de cadastro** constituída pela sequência de dígitos: 1234567890.

7.2 Incluir/excluir um Arquivo

Uma vez cadastrado pelo administrador do **sistema para resolução de IBI**, o administrador do novo **Arquivo** receberá um *e-mail* de confirmação, e o **Arquivo** 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**, i.e., 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** (i.e., o componente <archiveaddress>) poderá assumir, ou não, um novo valor.

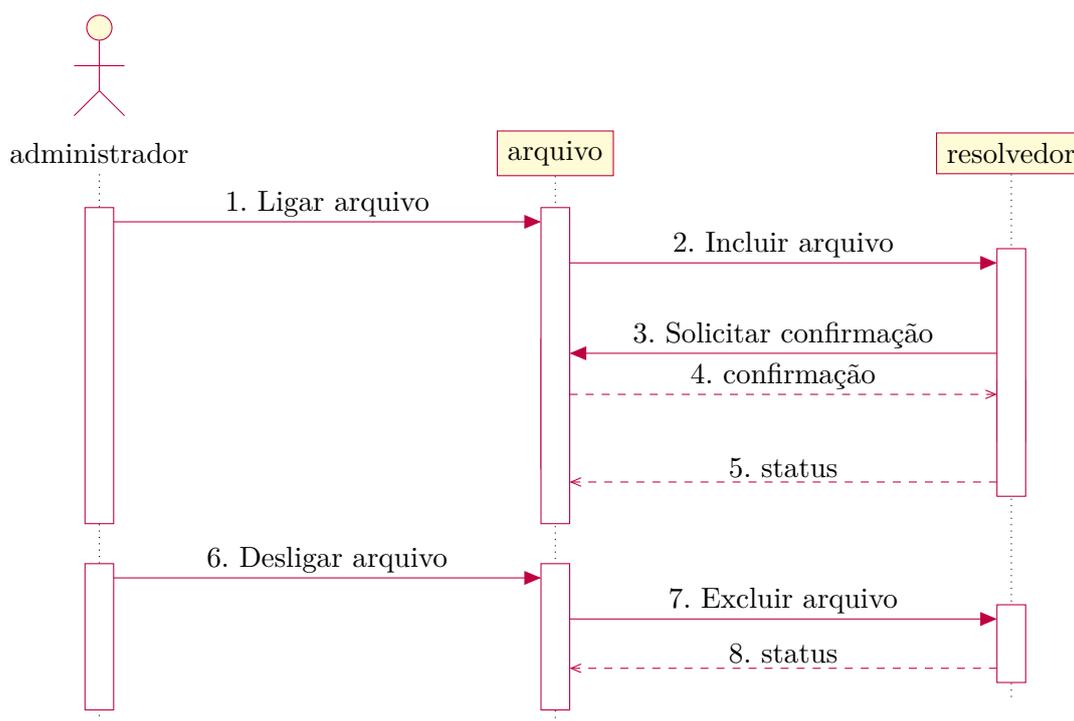
O processo de inclusão/exclusão de **Arquivo** no/do **sistema para resolução de IBI** é detalhado no diagrama de sequência da Figura 7.1. Esse processo é desencadeado pela emissão da mensagem “Ligar arquivo” (1), enviada pelo administrador ao **Arquivo**, e termina pela emissão da mensagem “Desligar arquivo” (6).

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

Numa solicitação de inclusão, o par `servicesubject=inclusionRequest` deverá ser anexado à URL de base; numa de exclusão, o par `servicesubject=exclusionRequest`.

Junto a esse par, uma solicitação de inclusão ou exclusão deve conter mais outros sete pares, como mostrado na Tabela 7.2. Nessa tabela, a coluna *Domínio* indica o domínio dos valores possíveis para cada par. Quando os valores não são enumerados explicitamente, pode ser feita referência a uma “regra” gramatical, cuja especificação utiliza a notação BNF (aumentada), para a formação dos possíveis valores. A coluna *Tipo* indica que todos os oito pares são obrigatórios (obr.).

Figura 7.1 - 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 esse 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 que 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.

Tabela 7.2 - Pares <nome>=<valor> a serem anexados à URL de base para as mensagens “Incluir Arquivo” (2) e “Excluir Arquivo” (7) endereçadas ao **resolvedor**.

Nome	Par		Tipo
	Domínio	Valor	
servicesubject	inclusionRequest	em caso de solicitação de inclusão	obr.
	exclusionRequest	em caso de solicitação de exclusão	
archiveaddress	regra hostport da Seção 6.1	o endereço Web do Arquivo ex: mtc-m21.sid.inpe.br	obr.
archiveserviceibi	regra repositório ou IBIP em (Comissão-de-Estudos ABNT/CB08/SC010/CE70, 2011)	o IBI do serviço de Arquivo informado no momento de cadastrar o Arquivo (cf. Seção 7.1)	obr.
archiveip	regra IPv4address ou IPv6address da Seção 3.2.2 em (BERNERS-LEE et al., 2005)	o IP do servidor hospedando o Arquivo ex: 150.163.34.239	obr.
archiveprotocol	HTTP	o protocolo de comunicação	obr.
archiveplatformversion	sequência ASCII conforme Seção 6.1	a versão da plataforma computacional hospedando o Arquivo ex: 2014:11.09.02.16.15	obr.
archiveadmemailaddress	regra authority da Seção 3.2 em (BERNERS-LEE et al., 2005)	o endereço de <i>e-mail</i> do administrador do Arquivo ex: banon@dpi.inpe.br	obr.
registrationkey	regra chave da Tabela 7.1	a chave de cadastro do Arquivo informado no momento de cadastrá-lo (cf. Seção 7.1) ex: 1234567890	obr.

Nesta recomendação, apenas um único protocolo de comunicação está sendo considerado: o protocolo HTTP. Por esse 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 7.2 (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 7.3 (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 7.2 e 7.3, as URL foram, respectivamente, as mensagens “Incluir Arquivo” (2) e “Excluir Arquivo” (7), endereçadas ao **resolvedor** pelo **Arquivo** com **serviço de Arquivo** identificado com **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 à 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 esse 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 numa solicitação de confirmação.

Exemplo 7.4 (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 7.4, a URL foi a mensagem “Solicitar confirmação” (3) enviada pelo **resolvedor** ao **Arquivo** cujo endereço *Web* era `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 9).

Ao receber a mensagem “Incluir Arquivo” (2), o **resolvedor** incluirá o **Arquivo** no **sistema para resolução de IBI** se este já estiver cadastrado e 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 nessa 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”.
```

8 Resolução de IBI

8.1 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 mais simples de uma **URL persistente** é:

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

em que `http` é o esquema (protocolo), `urlib.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 8.1 (URL persistentes) – URL:

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

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

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

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

□

Nas três primeiras **URL persistentes** do Exemplo 8.1, `8JMKD3MGP8W/35MME4E`, `8JMKD3MGP8W/35MMLL8` e `LK47B6W/362SFKH` são **IBI** na forma opaco com base no IP (BANON, 2011). Na quarta **URL persistente**, o **IBI** `iconet.com.br/banon/2009/09.09.22.01` está como nome de repositório (BANON, 2011).

No exemplo 8.1, 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).

Em termos de linguagem formal, uma **URL persistente** é um caso particular de URI (BERNERS-LEE et al., 2005). A Tabela 8.1 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), **IBIp** (**IBI** opaco com base no IP), **ALFAMI** (letras minúsculas) e **ALFAMA** (letras maiúsculas), introduzidas em Comissão-de-Estudos ABNT/CB08/SC010/CE70 (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 B.1 do Apêndice B.

Tabela 8.1 - 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**. Esse operador é construído por meio da composição de operadores escolhidos dentre de três operadores (ou modificadores) elementares. O primeiro, 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**.

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

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

Exemplo 8.2 (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 8.1, tem-se as URL do Exemplo 8.2. Uma **URL persistente**, como a primeira desse 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 (+) 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 8.3 (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 8.1, que aponta para um documento escrito em Inglês, tem-se as URL do Exemplo 8.3. Se o idioma de preferência selecionado pelo usuário em seu navegador for o Inglês,

então uma **URL persistente**, como a primeira desse exemplo, deve retornar o **item identificado** original (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**). 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.

Na Tabela 8.1, os três modificadores elementares estão especificados por meio das regras m (para metadados), u (para última edição) e t (para tradução).

A Tabela 8.2 descreve todas as composições permitidas de operadores escolhidos dentre dos três modificadores elementares de **URL persistente**.

Tabela 8.2 - 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

Exemplo 8.4 (**URL persistentes** com um ou mais modificadores elementares) – URL:

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

`http://urlib.net/8JMKD3MGP8W/35MMLL8!:(oai_dc)`

□

Ao acrescentar o modificador *ponto de exclamação* (!) à segunda **URL persistente** do Exemplo 8.1, tem-se as URL do Exemplo 8.4. Uma **URL persistente**, como a primeira desse exemplo, deve retornar a última edição (aqui nesse exemplo, publicada em 2012, ou posterior) do **item identificado** (aqui nesse 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 nesse exemplo, com **IBI** 8JMKD3MGP8W/35MMLL8).

Pela regra pergunta da Tabela 8.1, 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 6.1.

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 esse sistema. Os dois pares <nome>=<valor>, atualmente interpretados pelo **sistema para resolução de IBI**, são mostrados na Tabela 8.3. Nela, a coluna *Tipo* indica que os dois pares são facultativos (fac.).

Tabela 8.3 - Pares <nome>=<valor> interpretados pelo **sistema para resolução de IBI**.

Nome	Par		Tipo
	Domínio	Significado	
<code>ibiurl.requireditemstatus</code>	Original	solicitação do original	fac.
<code>ibiurl.verblast</code>	regra lista-de-verbos da Tabela 8.4	modificador de URL persistente ex: <code>GetLastEdition</code>	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 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.verblist` assume um valor verificando a regra gramatical `lista-de-verbos`. Como mostrado na Tabela 8.4, 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 8.4 - 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 8.5, os três primeiros verbos possuem sinônimos que são os modificadores elementares de **URL persistente** já introduzidos nessa seção. Na prática, esses três verbos atuam da mesma forma que seus sinônimos.

Tabela 8.5 - 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

Exemplo 8.5 (**URL persistente** modificada por uma lista de verbos) – URL:

```
http://urllib.net/8JMKD3MGP8W/35MMLL8?
ibiurl.verblist=GetLastEdition+GetMetadata(oai_dc)
```

□

Ao acrescentar, à segunda **URL persistente**, apresentada no Exemplo 8.1, o par `ibiurl.verblast=GetLastEdition+GetMetadata(oai_dc)`, tem-se a **URL persistente** do Exemplo 8.5. Uma **URL persistente**, como a desse exemplo, deve retornar no formato `oai_dc` os metadados da última edição do **item identificado** (aqui nesse exemplo, com **IBI** 8JMKD3MGP8W/35MMLL8). Essa **URL persistente** é equivalente à segunda URL do Exemplo 8.4 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.

Exemplo 8.6 (**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 8.1, o par `ibiurl.verblast=GetFileList`, tem-se a URL do Exemplo 8.6. Uma **URL persistente**, como a desse exemplo, deve retornar a lista dos recursos (arquivos) que compõem o **item identificado** (aqui nesse exemplo, com **IBI** LK47B6W/362SFKH).

Finalmente, numa **URL persistente** o componente `<caminho>` permite especificar um determinado recurso (arquivo) que compõe um certo **item de informação**.

Exemplo 8.7 (**URL persistente** modificada retornando um recurso específico) – URL:

`http://urlib.net/LK47B6W/362SFKH/reference.bib`

□

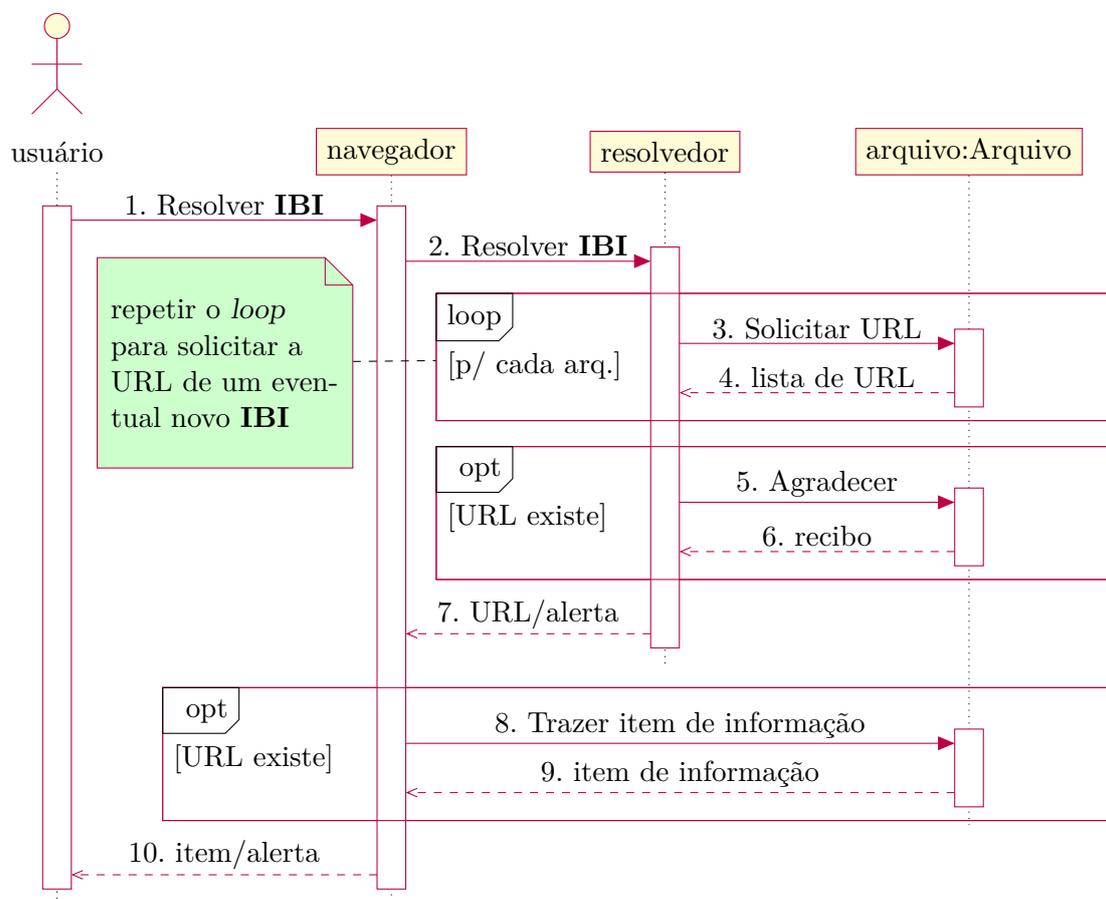
Ao acrescentar, à terceira **URL persistente**, apresentada no Exemplo 8.1, o caminho `/reference.bib`, tem-se a URL do Exemplo 8.7. Uma **URL persistente**, como

a desse exemplo, deve retornar apenas o conteúdo do arquivo especificado (aqui nesse exemplo `/reference.bib`) que compõe o **item identificado** (aqui nesse exemplo, com **IBI** LK47B6W/362SFKH).

8.2 Processo de resolução

O processo de resolução de **IBI** é detalhado no diagrama de sequência da Figura 8.1.

Figura 8.1 - Diagrama de seqüência do processo de resolução de **IBI**



Ao receber a mensagem “Resolver **IBI**” (2), o **resolvedor** endereça a mensagem “Solicitar URL” (3) a todos os **Arquivos** que retornam, cada um, uma “lista de URL” (4) (eventualmente vazia); após análise das respostas, caso a URL solicitada existir, o **resolvedor** emitirá uma mensagem “Agradecer” (5) para o **Arquivo** que a **forceneu**, e retornará essa URL (7), senão emitirá um alerta. No caso da URL existir, ao recebê-la, o navegador a redireciona para esse **Arquivo**, na forma da mensagem “Trazer item de informação” (8). O “item de informação” (9) é retornado ao navegador que o exibe na tela do usuário (10). No caso da URL não existir, o navegador exibe na tela do usuário (10) um alerta indicando que o **item de informação** não foi encontrado.

Fonte: Produção do autor.

O processo de resolução de **IBI** é desencadeado pela emissão da mensagem “Resolver **IBI**” (1), enviada pelo usuário ao seu navegador. Essa mensagem consiste numa **URL persistente**, como descrita na Seção 8.1. A emissão ocorre quando o usuário, após digitar no campo de endereço do seu navegador a **URL persistente**, clicar no

botão de envio da URL. Essa emissão pode ainda ocorrer quando o usuário clicar 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 8.6.

Tabela 8.6 - 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>	o 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**. Mais 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) dos verbos cujos sinônimos constam do componente `mdf` e dos verbos que constam no par com nome `ibiurl.verblast` do componente `pergunta`.

¹Um ponteiro persistente é um ponteiro cujo atributo `href`, na linguagem HTML, tem como valor uma **URL persistente**.

Exemplo 8.8 (Dados capturados pelo **resolvedor** ao receber uma **URL persistente**) – **URL persistente**:

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

Tabela 8.7 - Dados sobre o usuário.

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

Tabela 8.8 - Dados extraídos da **URL persistente**.

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

□

A Tabela 8.7 do Exemplo 8.8 mostra os dois pares nome/valor que informaram sobre o usuário, e a Tabela 8.8 os três que foram extraídos da **URL persistente** exemplificada.

Na etapa seguinte, somente parte dos dados da Tabela 8.6, como indicado na coluna três, será repassada ao **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 e 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, não deve 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 7.2). Essa solicitação é feita conforme o formato de solicitação de serviço da Seção 6.1, anexando à URL de base todos os pares, ou parte deles, apresentados na Tabela 8.9.

Tabela 8.9 - Pares <nome>=<valor> a serem anexados à URL de base para a mensagem “Solicitar URL” (3) endereçada ao **Arquivo**.

Nome	Par		Tipo
	Domínio	Valor	
<code>servicesubject</code>	<code>urlRequest</code>	solicitação de URL	obr.
<code>clientinformation.ipaddress</code>	regra <code>lista-de-IP</code> da tabela 8.10	endereço IP do usuário solicitando a resolução de um IBI ex: 150.163.2.175	obr.
<code>parsedibiurl.filepath</code>	regra <code>path-absolute</code> em (BERNERS-LEE et al., 2005)	caminho do recurso (arquivo) ex: /reference.bib	fac.
<code>parsedibiurl.ibi</code>	regra <code>repositório</code> ou <code>IBIp</code> em (Comissão-de-Estudos ABNT/CB08/SC010/CE70, 2011)	IBI objeto de resolução ex: LK47B6W/362SFKH	obr.
<code>parsedibiurl.verblast</code>	regra <code>lista-de-verbos</code> da Tabela 8.4	modificador de URL persistente	fac.

A Tabela 8.10 especifica as regras de formação da regra `lista-de-IP` usada na definição do domínio do valor do par com nome `clientinformation.ipaddress` da Tabela 8.9. A especificação utiliza a notação BNF (aumentada) e as regras predefinidas: IPv4 e IPv6, introduzidas em Berners-Lee et al. (2005). Na lista de IP, o primeiro elemento é o IP do usuário. Caso existir outros elementos, estes são os IP dos seus sucessivos procuradores (*proxy servers*)².

Tabela 8.10 - 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
```

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

Um exemplo de mensagem do tipo “Solicitar URL” (3) já foi dada no Exemplo 6.1.

Em resposta à solicitação do **resolvedor**, os **Arquivos** retornam, cada um, uma “lista de URL” (4) (eventualmente vazia) conforme descrito na Seção 9.

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

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, senão, 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** seleciona a primeira resposta não vazia que satisfaz a mensagem “Solicitar URL” (3). Nesses casos, se a resposta não contiver a URL para o **item de informação** solicitado, então o **resolvedor** endereça uma nova mensagem “Solicitar URL” (3) a todos os **Arquivos** (cf. a nota em verde ligado ao fragmento *loop* da Figura 8.1). Nessa nova mensagem, o valor do par com nome `parsedibiurl.ibi` é, dentre dos **IBI** contidos na resposta selecionada, aquele que atende à lista de verbos que consta no par `parsedibiurl.verblast` capturado da **URL persistente** (cf. Tabela 8.6). O processo se repete até que a resposta selecionada pelo **resolvedor** contenha a URL para o **item de informação** solicitado (cf. o exemplo da Seção 10.2).

Após ter encontrado a URL apropriada à solicitação de resolução de **IBI**, o **resolvedor**, emite uma mensagem “Agradecer” (5) para o **Arquivo** que forceneu essa URL. A mensagem é construída conforme o formato de solicitação de serviço da Seção 6.1, anexando à URL de base todos os pares apresentados na Tabela 8.11.

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 retornado ao usuário do **sistema para resolução de IBI**.

Tabela 8.11 - Pares <nome>=<valor> a serem anexados à URL de base para a mensagem “Agradecer” (5) endereçada ao **Arquivo**.

Nome	Par		Tipo
	Valor		
	Domínio	Significado	
servicesubject	acknowledgment	agradecimento	obr.
clientinformation.ipaddress	regra IPv4 ou IPv6 em (BERNERS-LEE et al., 2005)	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 9.4	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 8.1	URL persistente na origem da solicitação de resolução de IBI	obr.
urlkey	regra chave da Tabela 7.1	chave de URL constando da mensagem “lista de URL” (4) recebida do Arquivo	obr.

Exemplo 8.9 (mensagem de agradecimento emitida pelo **resolvedor**) – URL:

```
http://mtc-m19.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-m19.sid.inpe.br/col/sid.inpe.br/mtc-m19/2013/09.04.12.27.57/doc/
Relat%3rio%20Final.pdf&
url.persistent=http://mtc-m19.sid.inpe.br:80/8JMKD3MGP7W/3EPGUE5&
urlkey=1424636122-5331490054869684
```

□

No Exemplo 8.9, a URL foi a mensagem “Agradecer” (5) enviada pelo **resolvedor** ao **Arquivo** com endereço *Web* `mtc-m19.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 8.11; o par `clientinformation.ipaddress=150.163.2.175` informou que a **URL persistente** foi emitida pelo usuário com IP 150.163.2.175, o par `contenttype=Data` indicou que o **item de informação** selecionado é do tipo dados (e não metadados), o par com nome `ibi` indicou 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 de **IBIp**), finalmente, os três últimos pares com nomes `url`, `url.persitent` e `urlkey` informaram, 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 apropriada à solicitação de resolução de **IBI** 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 exibição do mesmo (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 usuário do **sistema para resolução de IBI** uma mensagem de alerta (7).

9 Serviço de Arquivo

9.1 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 7.1 da Seção 7.2. Nesse caso, o `servicesubject` é `inclusionConfirmationRequest`, e apenas o par `servicesubject=inclusionConfirmationRequest` é anexado à URL de base (cf. Exemplo 7.4).

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, ao emitir a mensagem “Solicitar URL” (3) do Diagrama 8.1 da Seção 8.2. Nesse caso, o `servicesubject` é `urlRequest` e os parâmetros da solicitação são os quatro últimos pares da Tabela 8.9 (cf. Exemplo 6.1).

Finalmente, a terceira, quando o **resolvedor** emite a mensagem “Agradecer” (5) do Diagrama 8.1 da Seção 8.2 para o **Arquivo** que possua o **item de informação** cujas especificações atendem 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 8.11 (cf. Exemplo 8.9).

9.2 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: rotinas e outro algoritmo (cf. Tabela 9.1).

Tabela 9.1 - Componentes do Algoritmo 1 (MONTARRESPOSTA).

<i>Componente</i>	<i>Página</i>
Rotina <code>DECODQS</code>	44
Rotina <code>FILTRARLISTADEURL</code>	47
Algoritmo 2 (MONTARLISTADEURL)	50

Algoritmo 1 – MONTARRESPOSTA.

GLOBAL: `clientinformation.ipaddress` (*IP usuário – criado nesse algoritmo – usada por MONTARLISTADEURL*),
`parsedibiurl.filepath` (*caminho de recurso – criado nesse algoritmo – usada 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), apresentado acima, 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 9.1 (Funcionamento da rotina DECODQS) – A execução de:

```
DECODQS(qs, "servicesubject")
```

onde `qs` foi a cadeia de caracteres:

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

resultou na cadeia de caracteres:

```
urlRequest
```

□

No Exemplo 9.1, a rotina DECODQS foi usada para extrair, da cadeia de caracteres `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 `qs` escolhida foi a lista de pares <nome>=<valor> anexados à URL de base da mensagem de solicitação de serviço do Exemplo 6.1. Esta seria uma possível mensagem “Solicitar URL” (3) da Figura 8.1.

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**. Esse 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 ainda, 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) e na Rotina FILTRARLISTADEURL.

Conforme à Tabela 8.9, os dois pares com nome `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 primeiro argumento da rotina. Dentre essas propriedades, podem constar ou não, em particular, as URL de acesso aos **itens de informação** relacionados ao **item identificado** com esse **IBI**.

Exemplo 9.2 (Funcionamento da rotina MONTARLISTADEURL) – A execução de:

```
MONTARLISTADEURL("8JMKD3MGP8W/35MME4E")
```

resultou numa cadeia de caracteres formando a lista de pares:

<code>archiveaddress</code>	<code>mtc-m18.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-m18@80/2009/07.21.13.23 ibip 8JMKD3MGP8W/35MME4E}</code>
<code>ibi.archiveservice</code>	<code>{rep sid.inpe.br/mtc-m18@80/2008/03.17.15.17}</code>
<code>ibi.lastedition</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23 ibip 8JMKD3MGP8W/35MME4E}</code>
<code>ibi.lastedition.metadata</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23.47}</code>
<code>ibi.lastedition.metadata(oai_dc)</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23.47}</code>
<code>ibi.metadata</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23.47}</code>
<code>ibi.metadata(oai_dc)</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23.47}</code>
<code>ibi.metadata.translation</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23.47}</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-m18@80/2009/07.21.13.23}</code>
<code>ibi.translation(pt)</code>	<code>{rep sid.inpe.br/mtc-m18@80/2009/08.25.19.43}</code>
<code>state</code>	<code>Original</code>
<code>state.lastedition</code>	<code>Original</code>
<code>state.lastedition.metadata</code>	<code>Original</code>

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-m18.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-m18.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-m18.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-m18...br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi?choice=oai_dc
url.metadata	http://mtc-m18.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi
url.metadata(oai_dc)	http://mtc-m18...br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi?choice=oai_dc
url.metadata.translation	http://mtc-m18.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.13.23.47/doc/metadata.cgi
url.translation(en)	http://mtc-m18.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-m18.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 9.2 ilustra um conjunto de propriedades disponíveis a respeito do **IBI 8JMKD3MGP8W/35MME4E**. 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`. Num 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` (no exemplo, o décimo primeiro par) informou o **IBI (8JMKD3MGP8W/35MME4E)** cujas propriedades estiveram sendo exibidas. O par com nome `archiveaddress` (o primeiro par) informou qual é o **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 cujo prefixo foi `url`, e cujos valores foram 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 9.5 e Exemplo 9.5).

No conjunto de propriedades do Exemplo 9.2, observa-se também que os valores dos

pares com nome com prefixo qualificado como `.lastedition` são os mesmos que dos pares com nome 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 nome com prefixo qualificado como `.lastedition.metadata` são os mesmos que dos pares com nome com prefixo qualificado como `.metadata`. Observa-se ainda que os valores dos pares com nome com prefixo qualificado como `.metadata(oai_dc)` são os mesmos que dos pares com nome com prefixo qualificado como `.metadata`, a não ser para os pares com nome com prefixo `url`. Isto ocorre porque o acesso a estes metadados é feito a partir de um mesmo script CGI que exhibe 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 nenhuma 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 facultativa. Seu papel é diminuir o tamanho da resposta “lista de URL” (4). Mesmo que essa rotina não elimine nenhum par sem utilidade, o resultado do processo de resolução de **IBI** não será alterado, apenas o desempenho do processo não será o mais eficiente.

Exemplo 9.3 (Funcionamento da rotina **FILTRARLISTADEURL**) – A execução de:

```
FILTRARLISTADEURL(resposta, "GetTranslation")
```

onde `resposta` foi a lista de pares do Exemplo 9.2, resultou numa cadeia de caracteres formando a lista de pares:

```
archiveaddress          mtc-m18.sid.inpe.br
contenttype.translation(en)  Data
contenttype.translation(pt)  Data
ibi                      {rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23 ibip 8JMKD3MGP8W/35MME4E}
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}
ibi.translation(en)      {rep sid.inpe.br/mtc-m18@80/2009/07.21.13.23}
ibi.translation(pt)      {rep sid.inpe.br/mtc-m18@80/2009/08.25.19.43}
```

<code>state.translation(en)</code>	Original
<code>state.translation(pt)</code>	Original
<code>timestamp.translation(en)</code>	2009-07-21T13:23:45Z
<code>timestamp.translation(pt)</code>	2011-09-22T14:45:11Z
<code>url.translation(en)</code>	http://mtc-m18.sid.inpe.br/col/sid.inpe.br/mtc-m18@80/2009/07.21.13.23/doc/CCSDS%20643.0-B-1.pdf
<code>url.translation(pt)</code>	http://mtc-m18.sid.inpe.br/col/sid.inpe.br/mtc-m18@80/2009/08.25.19.43/doc/RTC-07.pdf
<code>urlkey</code>	1426387214-9164566186556927

□

No Exemplo 9.3, a rotina `FILTRARLISTADEURL` foi 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, foi 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 consistiu em preservar apenas os pares cujo nome possuíam um prefixo qualificado como `.translation` (com ou sem parâmetro). Observa-se que da lista de nove URL do Exemplo 9.2, 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 8.4, 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 9.3, 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 da 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 nome `clientinformation.languagepreference` da Tabela 8.6.

O **Algoritmo** 2 (`MONTARLISTADEURL`), usado a Linha 16 do Algoritmo 1 (`MONTARRESPOSTA`), para gerar as propriedades do **IBI** objeto de resolução, é, junto às rotinas que o compõem (cf. Tabela 9.2), a peça principal do **serviço de Arquivo**.

Tabela 9.2 - Componentes do Algorímo 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	59

Algoritmo 2 – MONTARLISTADEURL.

GLOBAL: relações (*subconjunto de relações verificando a regra relação da Tabela 9.5*),
clientinformation.ipaddress (*IP do usuário – usado por CRIARPARURL*),
parsedibiurl.filepath (*caminho de um recurso – usado por CRIARPARURL*),
parsedibiurl.verblast (*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.  |   |   |   | resposta ← JUNTAR(resposta, par)
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 6.2. 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.

Mais precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres x e i ,

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

onde SP é a regra definida na Tabela 6.2 e $\langle x, i \rangle$ é a cadeia de caracteres que representa o valor de x referente a i , como apresentado na Tabela 9.3.

Tabela 9.3 - Descrição da função $\langle x, i \rangle$.

x	$\langle x, i \rangle$	
	Domínio	Significado
"archiveaddress"	regra hostport da Seção 6.1	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 9.4	formas do IBI do item identificado em relação com IBI i
"ibi.archiveservice"	regra formas-de-ibi da Tabela 9.4	formas do IBI do serviço de Arquivo informado no momento de cadastrá-lo (cf. Seção 7.1)
"ibi.nextedition"	regra formas-de-ibi da Tabela 9.4	formas do IBI identificando a próxima edição do item identificado com IBI i
"ibi.platformsoftware"	regra formas-de-ibi da Tabela 9.4	formas do IBI identificando a plataforma que hospede 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 7.1	chave de url de acesso ao item de informação relacionado com item identificado com IBI i

A expressão “formas de **IBI**”, na coluna *Significado* da Tabela 9.3, refere-se às várias formas como um **IBI** pode ser escrito (Comissão-de-Estudos ABNT/CB08/SC010/CE70, 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; a segunda, o **IBI** opaco

com base no IP.

A Tabela 9.4 especifica as regras de formação da regra `formas-de-ibi` usada na coluna *Domínio* da Tabela 9.3. A especificação utiliza a notação BNF (aumentada) e as regras predefinidas: `repositório` (**IBI** como nome de repositório) e `IBIp` (**IBI** opaco com base no IP), introduzidas em Comissão-de-Estudos ABNT/CB08/SC010/CE70 (2011).

Tabela 9.4 - 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 9.4 (Funcionamento da rotina `CRIARPAR` para acessar o **estado** de um **item identificado**) – A execução de:

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

resultou numa cadeia de caracteres formando o par:

```
state Original
```

□

No Exemplo 9.4, a rotina `CRIARPAR` foi usada para criar uma cadeia de caracteres que forma um par, conforme a regra `par` da Tabela 6.2. O valor `Original` do par, dado por `< "state", "8JMKD3MGP7W/3EPGUE5" >`, forneceu o **estado** do **item identificado** com **IBI** `8JMKD3MGP7W/3EPGUE5`. Assim, o resultado da execução da rotina `CRIARPAR("state", "8JMKD3MGP7W/3EPGUE5")` transmitiu a informação de que o **item identificado** com **IBI** `8JMKD3MGP7W/3EPGUE5` devia ser tido como original (no **Arquivo** considerado).

Na coluna `x` da Tabela 9.3, 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 9.5.

Tabela 9.5 - 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 9.5, são apresentadas algumas relações verificando a regra `relação` da Tabela 9.5, assim como seus significados.

Exemplo 9.5 (Algumas relações) – Relações e seus significados:

<code>.lastedition</code>	última edição
<code>.translation</code>	tradução
<code>.translation(pt-BR)</code>	tradução em Português do Brasil
<code>.metadata</code>	metadados
<code>.metadata(oai_dc)</code>	metadados no formato <code>oai_dc</code>
<code>.lastedition.translation</code>	tradução da última edição
<code>.lastedition.translation(pt-BR)</code>	tradução em Português do Brasil da última edição
<code>.lastedition.metadata</code>	metadados da última edição
<code>.lastedition.metadata(oai_dc)</code>	metadados no formato <code>oai_dc</code> da última edição
<code>.translation(pt-BR).lastedition</code>	última edição da tradução em Português do Brasil
<code>.translation(en).metadata</code>	metadados da tradução em Inglês
<code>.metadata.translation(en)</code>	tadução em Inglês dos metadados

□

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, a rotina `CRIARPARURL`, a Linha 24 do **Algoritmo 2**, retornaria uma cadeia de caracteres vazia (`""`), já que não haveria nenhum **item de informação** no idioma Valão, o que levaria a processamentos desnecessários. Por outro lado, se o **Arquivo** armazenar **itens de informação** sem especificação de idioma, então ele deverá incluir no subconjunto `relações`, a relação `.translation` sem parâmetro de idioma. Essa opção será útil para o **resolvedor**

poder retornar a URL do **item de informação** que melhor atende à URL **persistente** quando esta contém o modificador *mais* (+) sem parâmetro de idioma também.

Na terceira coluna da Tabela 9.3, 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 acima significará “metadados da última edição do **item identificado com IBI i**”, e o significado de < x, i > na Tabela 9.3 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* é vazio, então a expressão “**item de informação em relação com IBI i**” deve ser entendida simplesmente como “**item identificado com IBI i**”.

Exemplo 9.6 (Funcionamento da rotina CRIARPAR para acessar o **IBI** da última edição de um **item identificado**) – A execução de:

```
CRIARPAR("ibi.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}
```

□

No Exemplo 9.6, a rotina CRIARPAR foi usada para criar uma cadeia de caracteres que forma um par, conforme a regra *par* da Tabela 6.2. O valor {rep sid.inpe.br/mtc-m19/2013/09.04.12.27.57 ibip 8JMKD3MGP7W/3EPGUE5} do par, dado por < “ibi.lastedition”, “8JMKD3MGP7W/3EPGUE5” >, forneceu as formas do **IBI** da última edição do **item identificado com IBI 8JMKD3MGP7W/3EPGUE5**. Assim, o resultado da execução da rotina CRIARPAR(“ibi.lastedition”, “8JMKD3MGP7W/3EPGUE5”) transmitiu a informação de que a última edição do **item identificado com IBI 8JMKD3MGP7W/3EPGUE5** era o próprio **item identificado**, ou seja, nesse caso, o **item identificado com IBI**

8JMKD3MGP7W/3EPGUE5 não possuía (ainda) uma nova edição.

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 6.2. 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.

Mais precisamente, para toda cadeia de caracteres `i`,

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

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 a identificar o **Arquivo**, deverão ser adicionados à variável `resposta`, por meio das rotinas `CRIARPAR` e `JUNTAR` as Linhas 6 a 10.

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.

Mais 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 6.2.

Exemplo 9.7 (Funcionamento da rotina `JUNTAR`) – A execução de:

```
JUNTAR("archiveaddress mtc-m19.sid.inpe.br", "state Original")
```

resultou na cadeia de caracteres:

```
archiveaddress mtc-m19.sid.inpe.br state Original
```

□

No Exemplo 9.7, a rotina `JUNTAR` foi usada para juntar duas cadeias de caracteres que formam, cada um, um par, conforme a regra `par` da Tabela 6.2. O resultado foi uma lista de dois pares que verifica a regra `lista-de-pares` da Tabela 6.2. Observe-se que o segundo par poderia ser o resultado da execução da rotina `CRIARPAR`, como ilustrado no Exemplo 9.4.

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 6.2. 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.

Mais precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres `i`,

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

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 10.2).

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 num *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 9.5. O Exemplo 9.5 ilustra alguns possíveis valores para a variável `relação`.

A Linha 22, já dentre o *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 6.2. 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, 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.

Mais precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres `r` e `i`,

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

onde `SP` é a regra definida na Tabela 6.2 e `\langle r, i \rangle` é a cadeia de caracteres que representa as formas do **IBI** (cf. Tabela 9.4) do **item de informação** relacionado ao **item identificado** com **IBI** `i`, de acordo com o relacionamento definido pela relação `r`.

Exemplo 9.8 (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 numa cadeia de caracteres formando o par:

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

□

No Exemplo 9.8, a rotina CRIARPARIBI foi usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 6.2. O valor `{rep sid.inpe.br/mtc-m19/2013/09.04.12.27.58}` do par, dado por `< ".metadata", "8JMKD3MGP7W/3EPGUE5" >`, forneceu 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") transmitiu a informação de que os metadados do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5 eram, por sua vez, identificados pelo **IBI** sid.inpe.br/mtc-m19/2013/09.04.12.27.58.

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 ao **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 10.2).

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 6.2. 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.verblist`.

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 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 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.

Mais precisamente, usando a notação BNF (aumentada), para toda cadeia de caracteres r que verifica a regra **relação** da Tabela 9.5, e para toda cadeia de caracteres i , y e z , sendo as duas últimas, respectivamente, as variáveis globais `parsedibiurl.filepath` e `parsedibiurl.verblast`,

$$\text{CRIARPARURL}(r, i) \triangleq \begin{cases} "" & \text{se } \langle \cdot \rangle \text{ abaixo} \\ & \text{for desconhecido,} \\ "url" r 1*SP \langle r, i \rangle_1 & \text{se } y = "" \text{ e } z = "", \\ "url" r 1*SP \langle r, i, y \rangle_2 & \text{se } y \neq "" \text{ e } z = "", \\ "url" r 1*SP \langle r, i, z \rangle_3 & \text{se } y = "" \text{ e } z \neq "", \\ "url" r 1*SP \langle r, i, z \rangle_3 & \text{se } y \neq "" \text{ e } z \neq ". \end{cases}$$

onde SP é a regra definida na Tabela 6.2 e

- a) $\langle r, i \rangle_1$ 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) $\langle r, i, y \rangle_2$ 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) $\langle r, i, z \rangle_3$ será 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" $\in z^1$, caso 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 **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.

Exemplo 9.9 (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 numa cadeia de caracteres formando o par:

```
url http://mtc-m19.sid.inpe.br/col/iconet.com.br/banon/2009/09.09.22.01/doc/@relatorio.pdf
```

□

No Exemplo 9.9, a rotina CRIARPARURL foi usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 6.2. O valor do par, dado por $\langle "", "LK47B6W/362SFKH" \rangle_1$, forneceu a URL de acesso ao **item identificado** com **IBI** LK47B6W/362SFKH.

Exemplo 9.10 (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(".metadados", "8JMKD3MGP7W/3EPGUE5")
```

¹Quando, por medida de segurança, a lista dos recursos (arquivos) que compõem um certo **item identificado** não deve ser divulgada, a página de exibição deve informar explicitamente que o acesso a lista não está permitida.

resultou numa cadeia de caracteres formando o par:

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

□

No Exemplo 9.10, a rotina CRIARPARURL foi usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 6.2. O valor do par, dado por `< ".metadados", "8JMKD3MGP7W/3EPGUE5" >1`, forneceu a URL de acesso aos metadados do **item identificado** com **IBI** 8JMKD3MGP7W/3EPGUE5.

Exemplo 9.11 (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 numa cadeia de caracteres formando o par:

```
url http://mtc-m19.sid.inpe.br/col/iconet.com.br/banon/2009/09.09.22.01/doc/reference.bib
```

□

No Exemplo 9.11, a rotina CRIARPARURL foi usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 6.2. O valor do par, dado por `< "", "LK47B6W/362SFKH" >2`, forneceu a URL de acesso ao arquivo `reference.bib` que compunha o **item identificado** com **IBI** LK47B6W/362SFKH. O nome desse arquivo era especificado na primeira variável global `parsedibiurl.filepath`. Em relação à URL criada no Exemplo 9.9, 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 qualquer outro nome de arquivo que compunha o **item identificado**. O exemplo seguinte mostra como acessar a lista de todos os arquivos que compõem o **item identificado**.

Exemplo 9.12 (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 numa cadeia de caracteres formando o par:

```
url http://mtc-m19.sid.inpe.br/displaydoccontent.cgi/LK47B6W/362SFKH?displaytype=FileList
```

□

No Exemplo 9.12, a rotina CRIARPARURL foi usada para criar uma cadeia de caracteres que forma um par, conforme a regra **par** da Tabela 6.2. O valor do par, dado por $\langle "", "LK47B6W/362SFKH" \rangle_3$, forneceu a URL de acesso à página de exibição dos arquivos que compunham 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://urlib.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)) pertence ao subconjunto **relações**, 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)) pertence 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 9.2).

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.

10 Exemplos de comunicação entre o resolvedor e os Arquivos

10.1 Exemplo sem modificador de IBI

No exemplo escolhido, considera-se que o **resolvedor** recebeu a mensagem “Resolver IBI” (2) da Figura 8.1, correspondente à **URL persistente**:

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

Essa URL é a segunda do Exemplo 8.1.

Ao receber esta URL, o resolvedor capturou, conforme à Tabela 8.6, 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 informou o IP do usuário (172.16.44.200) e do seu procurador (150.163.68.1). O segundo par informou 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 informou o **IBI** da **URL persistente**, objeto da resolução.

Em seguida, o **resolvedor** emitiu a mensagem “Solicitar URL” (3) da Figura 8.1, 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 8.9, parte deles sendo os capturados ao receber a mensagem “Resolver IBI” (2). Por exemplo, a mensagem destinada ao **Arquivo** com endereço *Web* mtc-m18.sid.inpe.br e **serviço de Arquivo** com **IBI** sid.inpe.br/mtc-m18@80/2008/03.17.15.17, foi:

```
http://mtc-m18.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 8.1) não vazia, recebida dos **Arquivos**, foi a lista de pares:

```
archiveaddress          mtc-m18.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-m18.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 trouxe as propriedades disponíveis a respeito do **IBI 8JMKD3MGP8W/35MMLL8** mencionado no terceiro par (com nome **ibi**). Nessa lista, o primeiro par indicou ao **resolvedor** que foi precisamente o **Arquivo** com endereço *Web* mtc-m18.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-m18.sid.inpe.br, isto é, ao **Arquivo** que forneceu a URL de acesso ao **item identificado** com **IBI 8JMKD3MGP8W/35MMLL8**. Conforme a Tabela 8.11, a URL correspondente à essa mensagem foi:

```
http://mtc-m18.sid.inpe.br/sid.inpe.br/mtc-m18080/2008/03.17.15.17?  
contenttype=Data&  
ibi=rep%20sid.inpe.br/mtc-m18080/2009/07.21.14.43%20ibip%208JMKD3MGP8W/35MMLL8&  
servicesubject=acknowledgment&  
state=Original&  
url=http://mtc-m18.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 “recibo” (6) (cf. Figura 8.1) do **Arquivo** com endereço *Web* mtc-m18.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-m18.sid.inpe.br/col/sid.inpe.br/mtc-m18080/2009/07.21.14.43/doc/  
CCSDS%20650.0-B-1.pdf
```

10.2 Exemplo com modificador de IBI

No exemplo escolhido, considera-se que o **resolvedor** recebeu a mensagem “Resolver IBI” (2) da Figura 8.1, correspondente à **URL persistente**:

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

Essa URL é a segunda do Exemplo 8.4. 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 informou a lista dos dois verbos que descrevem o modificador !:(oai_dc) da **URL persistente** (cf. Tabela 8.5).

Em seguida, o **resolvedor** emitiu a mensagem “Solicitar URL” (3) da Figura 8.1, endereçada a todos os **Arquivos** incluídos no **sistema para resolução de IBI**. Por exemplo, a mensagem destinada ao **Arquivo** com endereço *Web* mtc-m18.sid.inpe.br e **serviço de Arquivo** com **IBI** sid.inpe.br/mtc-m18@80/2008/03.17.15.17 foi:

```
http://mtc-m18.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 8.1) não vazia recebida dos **Arquivos** foi a lista de pares:

archiveaddress	mtc-m18.sid.inpe.br
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.nextedition	{rep sid.inpe.br/mtc-m18/2012/07.12.18.08}
ibi.platformsoftware	{rep dpi.inpe.br/banon/1998/08.02.08.56}

Essa lista trouxe 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, dentre de todas as mensagens emitidas, a mensagem destinada ao **Arquivo** com endereço

Web mtc-m18.sid.inpe.br foi:

```
http://mtc-m18.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 8.1) não vazia recebida dos **Arquivos** foi a lista de pares:

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

Essa nova lista trouxe 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 indicou ao **resolvedor** que o **Arquivo** com endereço `Web mtc-m18.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.

No passo seguinte, o **resolvedor** emitiu a mensagem “Agradecer” (5) endereçada ao **Arquivo** com endereço `Web mtc-m18.sid.inpe.br`, isto é, ao **Arquivo** que forneceu a URL de acesso aos metadados, no formato `oai_dc`, da última edição do **item identificado** com **IBI** `8JMKD3MGP8W/35MMLL8`. A URL correspondente à essa mensagem foi:

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.

```
http://mtc-m18.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-m18.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 “recibo” (6) (cf. Figura 8.1) do **Arquivo** com endereço *Web* `mtc-m18.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-m18.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. **Identificador com base na Internet (IBI): Sistema de identificação**. São José dos Campos: INPE, 2011. 40 p. Disponível em: <<http://urlib.net/LK47B6W/362SFKH>>. Acesso em: 23 ago. 2014. 25
- 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.], 1992. 13 p. Disponível em: <<http://urlib.net/sid.inpe.br/mtc-m18@80/2009/08.25.19.43>>. Acesso em: 10 jan. 2015. 16, 17
- _____. **Sistema para geração de IBI**. São José dos Campos: [s.n.], 2011. 52 p. Disponível em: <<http://urlib.net/8JMKD3MGP8W/3CLC37B>>. Acesso em: 12 maio 2015. 5, 8, 16, 19, 22, 26, 36, 52, 53
- 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. 5
- 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://arran.fi.muni.cz/bnfparsers2/share/rfc2234-6.1.abnf>>. 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. 10, 11

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, 2011. 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. 5

APÊNDICE A - Definição de relação binária funcional e esquerda-total

Sejam A e B dois conjuntos.

A relação binária $\mathcal{R} = (A, B, \mathcal{G})$ é funcional se e somente se seu grafo \mathcal{G} satisfaz:

para todo $a \in A, b_1$ e $b_2 \in B$ tem-se $(a, b_1) \in \mathcal{G}$ e $(a, b_2) \in \mathcal{G} \Rightarrow b_1 = b_2$.

A relação binária $\mathcal{R} = (A, B, \mathcal{G})$ é esquerda-total se e somente se seu grafo \mathcal{G} satisfaz:

para todo $a \in A$, existe um $b \in B$ tal que $(a, b) \in \mathcal{G}$.

APÊNDICE B - Regras idioma e país

O significado de cada dupla de caracteres pode ser consultado em ISO (2002) para idioma e ISO (2011) para país.

Tabela B.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"	"rj"	"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"		