

## 5 Detalhamento da arquitetura para OnOCs

### 5.1 Motivação

A arquitetura para OnOCs descrita no capítulo anterior foi introduzida para facilitar e agilizar o desenvolvimento de catálogos de objetos baseados em ontologias. O capítulo anterior fornece uma visão modular da arquitetura. Neste capítulo é apresentado o detalhamento da arquitetura.

Os diagramas apresentados neste capítulo podem servir como orientação para a organização e a estruturação geral dos catálogos desenvolvidos com base na arquitetura introduzida nesta dissertação.

### 5.2 Casos de uso

Os diagramas de casos de uso são usados para visualizar, especificar, construir e documentar o comportamento dos sistemas baseados na arquitetura proposta. Um caso de uso descreve as interações das entidades externas ao sistema (atores) com o próprio sistema.

Os atores de um OnOC podem ser de três tipos: aplicação cliente, aplicação participante e usuário administrador. A Figura 41 apresenta a hierarquia utilizada para representar os atores de um OnOC.

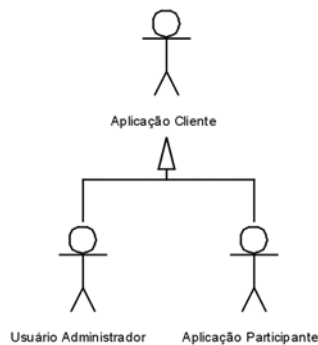


Figura 41 – Atores do OnOC.

As *Aplicações Clientes* são os atores que interagem com o OnOC através de requisições de consulta sem a necessidade de serem usuários cadastrados no sistema. As *Aplicações Participantes* são os atores que interagem com o catálogo com o objetivo de compartilhar objetos com os outros membros da federação. Uma aplicação participante é responsável por uma ou mais fonte de objetos componente da federação. O *Usuário Administrador* ou apenas *Administrador* interage com o sistema com o objetivo de gerenciar a federação. Entre suas tarefas estão a gerencia as permissões de usuários e a criação e atualização da ontologia de referência.

Os diagramas descritos nesta seção foram separados de acordo com a atuação dos atores com o sistema. As interações com o OnOC realizadas pelo ator mais geral, chamado *Aplicação Cliente*, apresentadas na Figura 42, também são realizadas pelos atores especializados chamados *Aplicação Participante* e *Usuário Administrador*, que por sua vez possuem interações específicas com o OnOC.

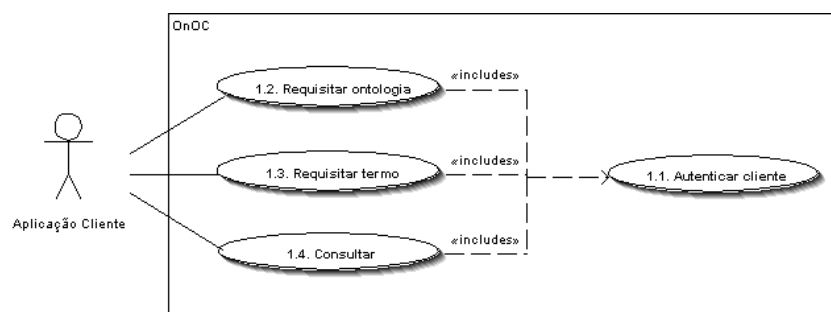


Figura 42 – Casos de uso do ator “Aplicação Cliente”.

A Figura 43 mostra o detalhamento dos casos de uso 1.3 da Figura 42. A aplicação cliente pode requisitar uma classe, uma propriedade ou uma instância das ontologias mantidas pelo catálogo.

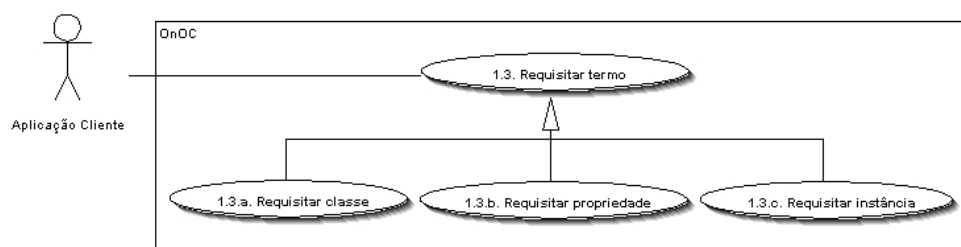


Figura 43 – Detalhamento dos casos de uso 1.3.

O diagrama da Figura 44 mostra as interações específicas com o OnOC realizadas pelo ator *Aplicação Participante*. Vale lembrar que, por ter sido representado como uma especialização do ator *Aplicação Cliente*, toda aplicação participante, além de suas interações específicas, realizará todas as interações descritas para o ator aplicação cliente.

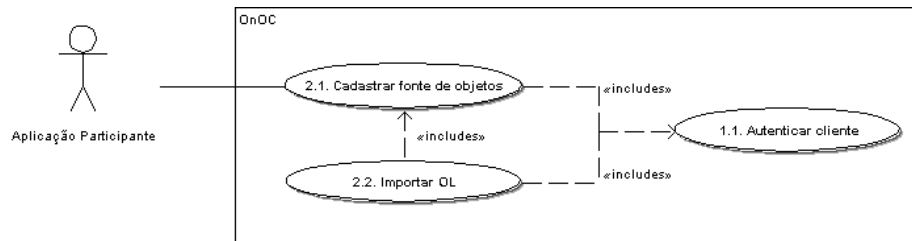


Figura 44 – Casos de uso do ator “Aplicação Participante”.

O diagrama da Figura 45 mostra as interações com o OnOC realizadas pelo ator *Usuário Administrador*. Da mesma forma que as aplicações participantes, todo usuário administrador além de suas interações específicas, realizará todas as interações descritas para o ator aplicação cliente, pois representa uma especialização do ator *Aplicação Cliente*.

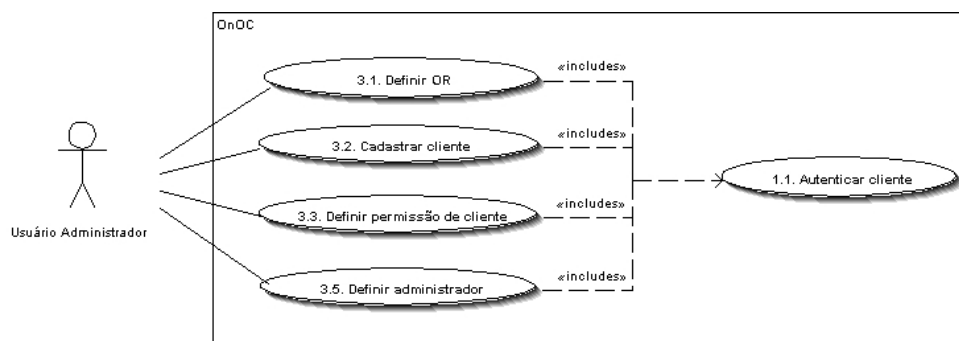


Figura 45 – Casos de uso do ator “Usuário Administrador”.

Com o objetivo de detalhar a especificação dos diagramas apresentados nesta seção, a seguir são apresentadas as descrições típicas dos casos de uso.

**Caso de Uso 1.1 – Autenticar cliente****Ator Primário:** Aplicação Cliente, Aplicação Participante e Usuário Administrador**Pré-condições:** Nenhuma**Fluxo Normal**

1 – O <i>Ator</i> fornece o nome do usuário e senha.
2 – O <i>Sistema</i> verifica a existência do usuário no registro do catálogo
3 – O <i>Sistema</i> permite o acesso do usuário

**Fluxo Alternativo:** usuário inexistente ou senha inválida

2a – Envia mensagem ao usuário informando erro: “Usuário inexistente ou senha inválida”
2b – Finaliza o caso de uso 1.1.

**Caso de Uso 1.2 – Requisitar ontologia****Ator Primário:** Aplicação Cliente**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>Ator</i> solicita uma ontologia ao sistema
2 – O <i>Sistema</i> busca a ontologia no repositório através da sua URI de identificação.
3 – O <i>Sistema</i> devolve a ontologia solicitada

**Fluxo Alternativo:** URI da ontologia não fornecida

2a – O <i>Sistema</i> busca a URI da ontologia de referência no registro do catálogo.
2b – Retorna ao passo 2 passando a URI da ontologia de referência.

**Fluxo Alternativo:** ontologia não encontrada

2a – Envia mensagem ao usuário informando erro: “Ontologia inexistente no catálogo”
2b – Finaliza caso de uso 1.2.

**Caso de Uso 1.3 – Requisitar termo****Ator Primário:** Aplicação Cliente**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>Ator</i> informa a URI de identificação do termo ao <i>Sistema</i>
2 – O <i>Sistema</i> busca o termo solicitado
3 – O <i>Sistema</i> devolve a descrição do termo solicitado

**Fluxo Alternativo:** termo não encontrado

2a – Envia mensagem ao usuário informando erro: “Termo inexistente no catálogo”
2b – Finaliza caso de uso 1.3.

**Caso de Uso 1.4 – Consultar****Ator Primário:** Aplicação Cliente**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>Ator</i> fornece uma consulta em RDQL ao <i>Sistema</i>
2 – O <i>Sistema</i> busca as ontologias necessárias para atender a consulta
3 – O <i>Sistema</i> faz a consulta solicitada
4 – O <i>Sistema</i> devolve a resposta da consulta solicitada

**Fluxo Alternativo:** erro na consulta

3a – Envia mensagem ao usuário informando erro: “Erro na consulta”
3b – Finaliza caso de uso 1.4.

**Caso de Uso 2.1 – Cadastrar fonte de objetos****Ator Primário:** Aplicação Participante**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>Ator</i> solicita o cadastramento de uma nova fonte de objetos ao <i>Sistema</i>
2 – O <i>Sistema</i> inicia o caso de uso 2.2.
3 – O <i>Sistema</i> faz o cadastramento da nova fonte de objetos no registro do catálogo
4 – O <i>Sistema</i> define a permissão do <i>ator</i> à ontologia local da nova fonte cadastrada
6 – O <i>Sistema</i> informa o sucesso no cadastramento da fonte

**Fluxo Alternativo:** erro na importação da ontologia local

2a – Envia mensagem ao usuário informando erro: “Erro na importação da ontologia”
2b – Finaliza caso de uso 2.1.

**Fluxo Alternativo:** erro no cadastramento da fonte de objetos no registro do catálogo

3a – O <i>Sistema</i> remove a ontologia local do repositório de ontologias
3b – Finaliza caso de uso 2.1.

**Caso de Uso 2.2 – Importar ontologia****Ator Primário:** Aplicação Participante**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>Ator</i> fornece a URI de localização da nova ontologia e a nova URI de identificação
2 – O <i>Sistema</i> importa a nova ontologia da URI de localização para memória
3 – O <i>Sistema</i> cadastra a nova ontologia no repositório de ontologias
4 – O <i>Sistema</i> informa o sucesso na importação da ontologia

**Fluxo Alternativo:** erro na importação da ontologia

2a – Envia mensagem ao usuário informando erro: “Erro na importação da ontologia”
2b – Finaliza o caso de uso 2.2.

**Caso de Uso 3.1 – Definir ontologia de referência****Ator Primário:** Usuário Administrador**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>ator</i> informa a URI e a localização da ontologia de referência
2 – O <i>Sistema</i> carrega a ontologia de referência da localização dada
3 – O <i>Sistema</i> armazena a ontologia de referência no repositório de ontologias
4 – O <i>Sistema</i> cadastra a URI da ontologia de referência no registro do catálogo e a permissão do usuário administrador para leitura e escrita nesta ontologia

**Fluxo Alternativo:** erro ao carregar ontologia

2a – Envia mensagem ao usuário informando erro: “Erro ao carregar ontologia”
2b – Finaliza o caso de uso 3.1.

**Caso de Uso 3.2 – Cadastrar cliente****Ator Primário:** Usuário Administrador**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>Ator</i> informa o nome de usuário, senha e tipo de usuário a ser cadastrado.
2 – O <i>Sistema</i> verifica se o nome de usuário não existe
3 – O <i>Sistema</i> cadastra o novo cliente
4 – O <i>Sistema</i> informa o sucesso no cadastramento

**Fluxo Alternativo:** nome de usuário já cadastrado

2a – Envia mensagem ao usuário informando erro: “Nome de usuário já cadastrado”
2b – Finaliza o caso de uso 3.2.

**Caso de Uso 3.3 – Definir permissão de aplicação****Ator Primário:** Usuário Administrador**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)**Fluxo Normal**

1 – O <i>ator</i> informa o nome de usuário, o tipo de permissão e a ontologia para a qual deseja definir a permissão
2 – O <i>Sistema</i> cadastra no registro do catálogo a permissão para o usuário

**Fluxo Alternativo:** erro no cadastramento de permissão

2a – Envia mensagem ao usuário informando erro: “Erro no cadastramento de permissão”
2b – Finaliza o caso de uso 3.3

**Caso de Uso 3.4 – Definir administrador**

**Ator Primário:** Usuário Administrador  
**Pré-condições:** Autenticar cliente (Caso de Uso 1.1)

**Fluxo Normal**

1 – O <i>ator</i> envia nome de usuário que será novo administrador
2 – O <i>Sistema</i> atualiza o tipo do usuário do administrador atual para <i>Aplicação Cliente</i>
3 – O <i>Sistema</i> atualiza o tipo do usuário do novo administrador para <i>Administrador</i>

**Fluxo Alternativo:** erro na atualização de administrador

2a – Envia mensagem ao usuário informando erro: “Erro na atualização de administrador”
2b – Finaliza o caso de uso 3.4

**5.3**  
**Diagramas de pacotes**

O diagrama de pacotes mostrado na Figura 46 mostra o pacote principal da arquitetura, chamado *OnOC*. Este pacote inclui os cinco módulos apresentados no capítulo anterior. Cada módulo será detalhado a seguir.

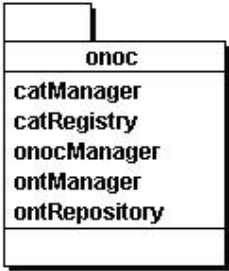


Figura 46 – Pacote *OnOC*.

A Figura 47 mostra o detalhamento do pacote *OnOC*, incluindo os pacotes que compõem a arquitetura. Cada pacote reflete um componente da arquitetura mostrada no Capítulo 4. As dependências entre os pacotes são indicadas pelas setas tracejadas.

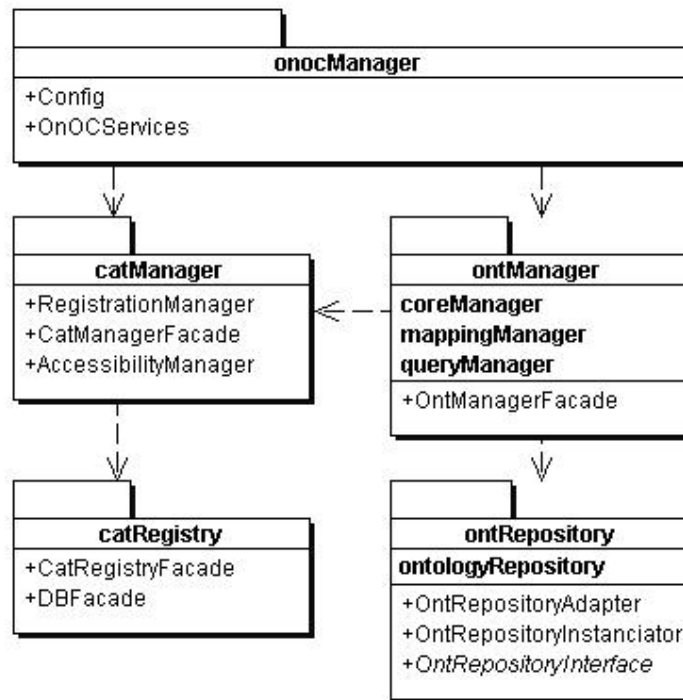


Figura 47 – Diagrama de pacotes do protótipo implementado.

## 5.4

### Diagramas de classes

Esta seção apresenta os diagramas de classe para cada pacote da arquitetura mostrando a dependência entre as classes do mesmo pacote e de pacotes diferentes.

As classes apresentadas nos diagramas a seguir compõem os pacotes que representam os módulos apresentados no capítulo anterior. Os inter-relacionamentos entre estes módulos são evidenciados através das linhas tracejadas.

A Figura 48 mostra o diagrama de classes do pacote *OnOCManager*. Este pacote é composto pelas classes responsáveis pelo acesso e configuração do sistema gerado a partir desta arquitetura. A classe *OnOCServices* é responsável pela implementação dos métodos que disponibilizam as operações da interface para acesso ao sistema. A classe *Config* é responsável pelas constantes de configuração, tais como nome de usuário e senha do banco de dados e do repositório de ontologias.



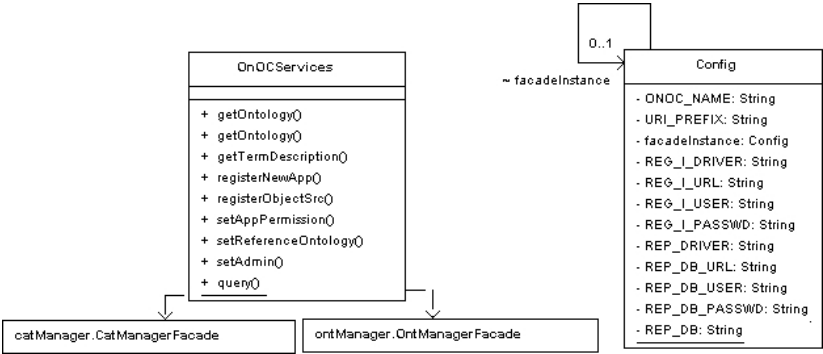


Figura 48 – Diagrama de classes do pacote *OnOCManager*.

A Tabela 8 lista os métodos que materializam os casos de uso descritos na seção 5.2 e os serviços da arquitetura descritos na seção 4.5. Estas operações são disponibilizadas através da classe *OnOCServices* mostrada na Figura 48 do pacote *OnOCManager*.

Tabela 8 – Operações básicas de um OnOC.

Nome da operação	Descrição	Serviços da arquitetura
(O1) <i>getOntology</i>	Solicita a descrição de uma ontologia ao catálogo	(S7)
(O2) <i>getTermDescription</i>	Solicita a descrição de um termo (classe, propriedade ou instância) de uma ontologia ao catálogo	(S8) (S9) (S10)
(O3) <i>query</i>	Consulta as ontologias do catálogo usando uma linguagem de consulta	(S11)
(O4) <i>registerObjectSrc</i>	Cadastra uma nova fonte de objetos	(S6)
(O5) <i>registerNewApp</i>	Cadastra uma nova aplicação no catálogo	(S1)
(O6) <i>setAppPermission</i>	Gerencia as permissões de acesso das aplicações	(S2)
(O7) <i>setReferenceOntology</i>	Define a ontologia de referência do catálogo	(S5)
(O8) <i>setAdmin</i>	Define o usuário administrador do catálogo	(S3)

O detalhamento de cada operação e os parâmetros são apresentados nas tabelas a seguir.

A Tabela 9 define os parâmetros da operação *getOntology*. Esta operação devolve a descrição completa de uma ontologia do catálogo. O método *getOntology* possui duas implementações indicadas. A implementação sem o parâmetro *URI* é utilizada para requisitar a ontologia de referência. A outra implementação, com o parâmetro *URI*, é utilizada para requisitar uma ontologia local.

Tabela 9 – Parâmetros da operação *getOntology*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
APPID	R	Define o ID para autenticação do cliente que está requisitando o serviço.
PWD	R	Define a senha para autenticação do cliente que está requisitando o serviço.
URI	O	Define a URI de identificação da ontologia requisitada.
OUTPUT	O	Define o formato de visualização da ontologia. Os valores permitidos nesta implementação são: RDF/XML, RDF/XML-ABBREV, N-TRIPLE e N3. Valor default: RDF/XML-ABBREV.

A Tabela 10 define os parâmetros da operação *getTermDescription*. A operação *getTermDescription* devolve a descrição em OWL do termo da ontologia de referência requisitado, seja este termo uma classe, propriedade ou instância.

Tabela 10 – Parâmetros da operação *getTermDescription*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
APPID	R	Define o ID para autenticação do cliente que está requisitando o serviço.
PWD	R	Define a senha para autenticação do cliente que está requisitando o serviço.
URI	R	Define a URI do termo requisitado.
OUTPUT	O	Define o formato de visualização da ontologia. Os valores permitidos nesta implementação são: RDF/XML, RDF/XML-ABBREV, N-TRIPLE e N3. Valor default: RDF/XML-ABBREV.

A Tabela 11 define os parâmetros da operação *query*. Esta operação é usada para consultar o catálogo usando a linguagem de consulta.

Tabela 11 – Parâmetros da operação *query*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
APPID	R	Define o ID para autenticação da aplicação.
PWD	R	Define a senha para autenticação da aplicação.
QUERY	R	Define a consulta em uma linguagem de consulta.
OUTPUT	O	Define o formato de visualização da ontologia. Os valores permitidos nesta implementação são: RDF/XML, RDF/XML-ABBREV, N-TRIPLE e N3. Valor default: RDF/XML-ABBREV.
INF	O	Define o uso de inferência no modelo consultado. Os valores permitidos para este parâmetro são: <i>True</i> = com inferência, <i>False</i> = sem inferência. Valor default: TRUE.

Incluindo o parâmetro *INF* com valor *true* a operação *query* realiza a consulta numa ontologia gerada a partir do cruzamento das descrições da ontologia consultada e seus mapeamentos com a ontologia de referência.

A Tabela 12 define os parâmetros da operação *registerObjectSrc*. Esta operação é utilizada por uma aplicação participante para cadastrar uma nova fonte de objetos no catálogo.

Tabela 12 – Parâmetros da operação *registerObjectSrc*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
APPID	R	Define o ID para autenticação da aplicação participante responsável pela fonte de objetos.
PWD	R	Define a senha para autenticação da aplicação participante responsável pela fonte de objetos.
URI	R	Define a URI de identificação da ontologia local no catálogo.
LOCATION	R	Define a URI de localização da ontologia local contendo o documento com a descrição da fonte de objetos a ser cadastrada (local ou na Web).

A Tabela 13 define os parâmetros da operação *registerNewApp*. Esta operação é utilizada pelo usuário administrador para o cadastramento de novas aplicações.

Tabela 13 – Parâmetros da operação *registerNewApp*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
ADMINID	R	Define o ID para autenticação do usuário administrador.
ADMINPWD	R	Define a senha para autenticação do usuário administrador.
APPID	R	Define o ID da nova aplicação a ser cadastrada.
APPPWD	R	Define a senha de autenticação da aplicação.

A Tabela 14 define os parâmetros da operação *setAppPermission*. Esta operação é utilizada pelo usuário administrador do catálogo para definir a permissão de acesso das aplicações às ontologias do catálogo (locais ou de referência).

Tabela 14 – Parâmetros da operação *setAppPermission*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
ADMINID	R	Define o ID para autenticação do usuário administrador.
PWD	R	Define a senha para autenticação do usuário administrador.
URI	R	Define a URI de identificação da ontologia em questão.
APPID	R	Define o ID da aplicação participante em questão.
PERMISSION	R	Define a permissão de acesso. Os valores permitidos são: R (somente leitura) e W (leitura e escrita).

A Tabela 15 define os parâmetros da operação *setReferenceOntology*. Esta operação é utilizada pelo administrador do catálogo para definir a ontologia de referência do catálogo.

Tabela 15 – Parâmetros da operação *setReferenceOntology*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
ADMINID	R	Define o ID para autenticação do usuário administrador.
PWD	R	Define a senha para autenticação do usuário administrador.
URI	R	Define a URI de identificação da ontologia de referência.

A Tabela 16 define os parâmetros da operação *setAdmin*. Esta operação é utilizada pelo usuário administrador para definição de um novo usuário administrador.

Tabela 16 – Parâmetros da operação *setAdmin*.

Parâmetro de requisição	Obrigatório (R) Opcional (O)	Descrição
ADMINID	R	Define o ID para autenticação do usuário administrador.
PWD	R	Define a senha para autenticação do usuário administrador.
APPID	R	Define o ID do novo usuário administrador.

A Figura 49 mostra o diagrama de classes do pacote *CatManager*. Este pacote é responsável pelo gerenciamento do catálogo, incluindo o controle de acesso dos clientes do catálogo.

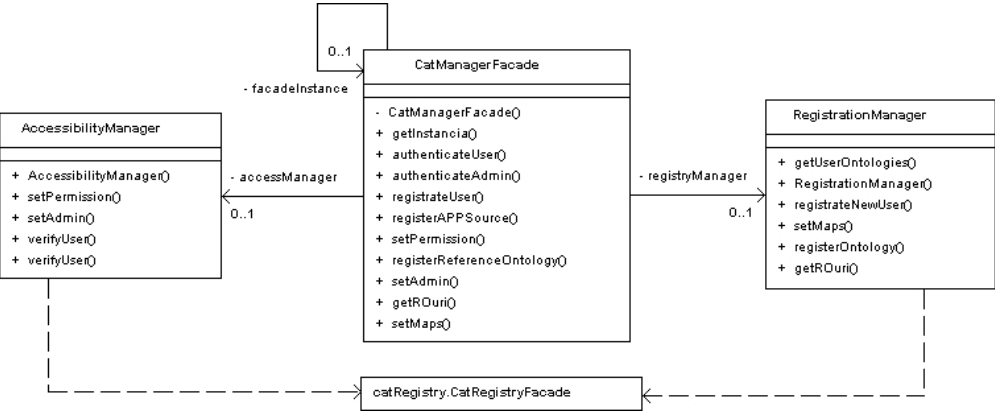


Figura 49 – Diagrama de classes do pacote *CatManager*.

A Figura 50 mostra o diagrama de classes do pacote *CatRegistry*. Este pacote é responsável pelas classes de acesso ao banco de dados de cadastramento das informação mantidas pelo catálogo, incluindo informações

das ontologias armazenadas no repositório, informações de aplicações cadastradas e permissões de acesso. A classe *DBFacade* faz o acesso ao banco de dados utilizado. Os parâmetros de configuração do banco de dados são constantes definidas através da classe *Config* do pacote *OnOCManager*.

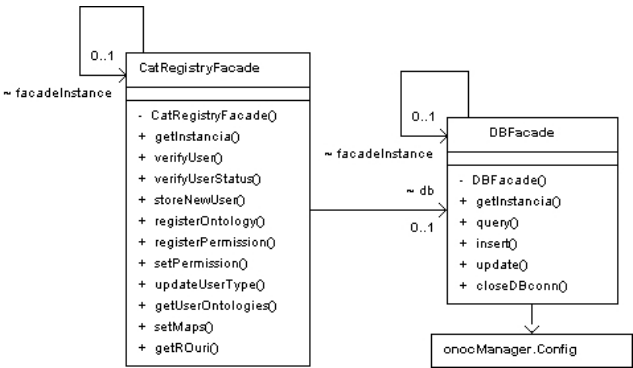


Figura 50 – Diagrama de classes do pacote *CatRegistry*.

A Figura 51 mostra o diagrama de classes do pacote *ontManager*. Este pacote é composto por três pacotes. O pacote *CoreManager* é responsável pelo cadastramento e recuperação das ontologias do repositório. O pacote *QueryManager* é responsável pela consulta às ontologias do catálogo. O pacote *MappingManager* é responsável pelo mapeamento das ontologias locais com a ontologia de referência do catálogo.

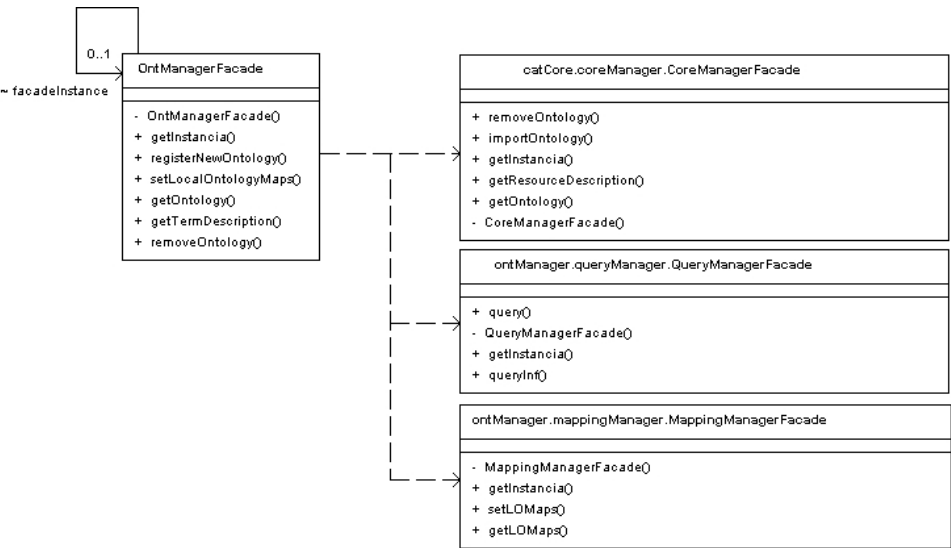


Figura 51 – Diagrama de classes do pacote *OntManager*.

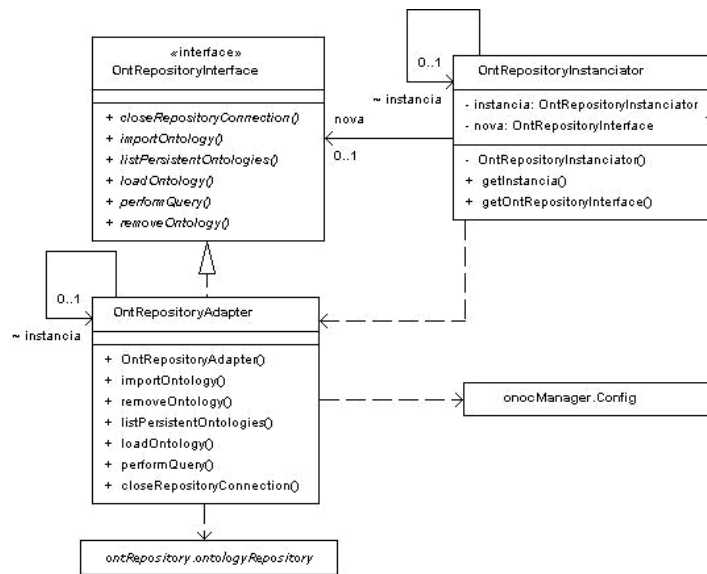


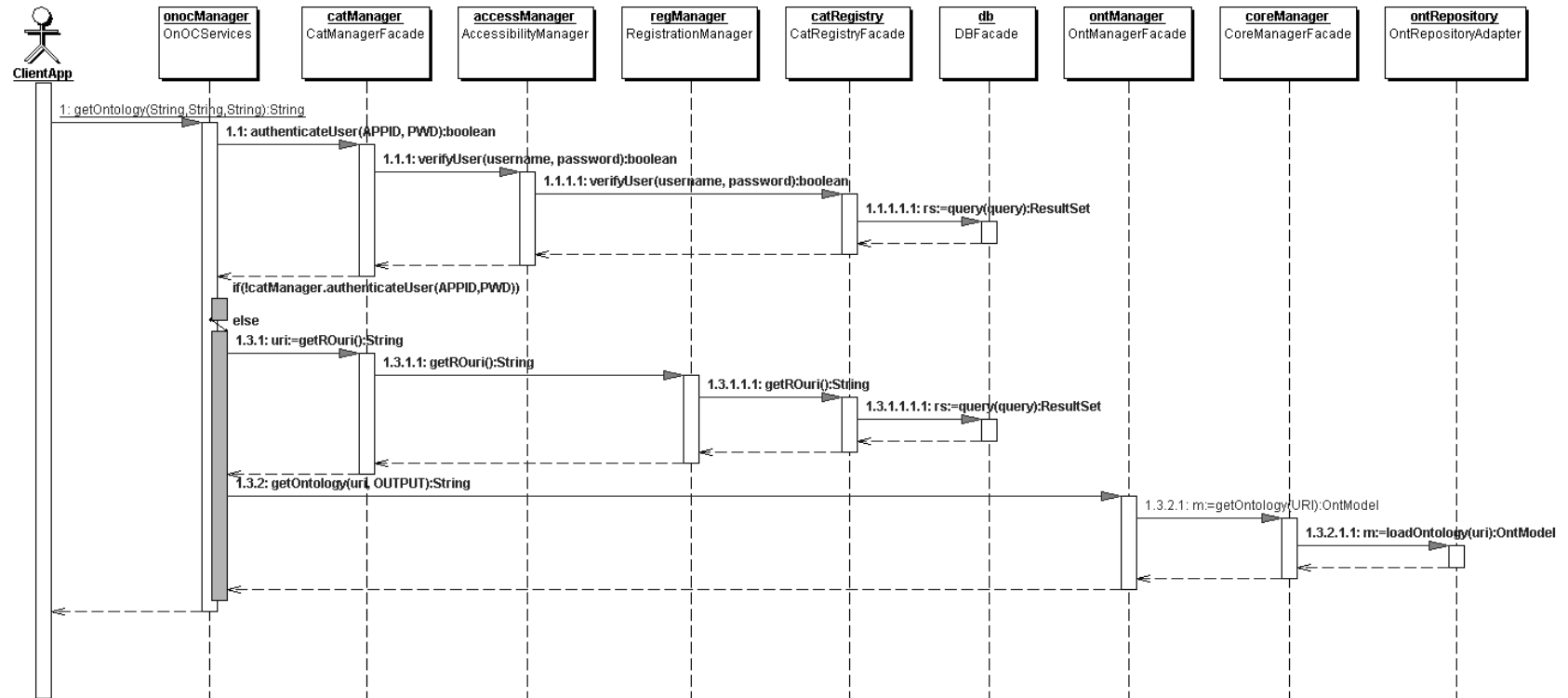
Figura 52 – Diagrama de classes do pacote *OntRepository*.

A Figura 52 mostra o diagrama de classes do pacote *OntRepository*. Para modelar este pacote foi utilizado o padrão de projeto *Adapter* (vide seção 4.4.2.3). Este padrão viabiliza a utilização de qualquer repositório de ontologias. Para criar uma particular implementação baseada nesta arquitetura, é necessário criar a classe de adaptação para o repositório de ontologias a ser utilizado. Para isso, basta criar a nova classe *OntRepositoryAdapter* que adapte os métodos utilizados pela interface do novo repositório de ontologias (*ontologyRepository*).

## 5.5

### Diagramas de seqüência

Esta seção apresenta os diagramas de seqüência modelados para cada método da classe *OnOCServices*. Cada diagrama mostra a seqüência de operações que deve ser executada para cada operação.

Figura 53 – Diagrama de sequência da operação *getOntology* (ontologia de referência).

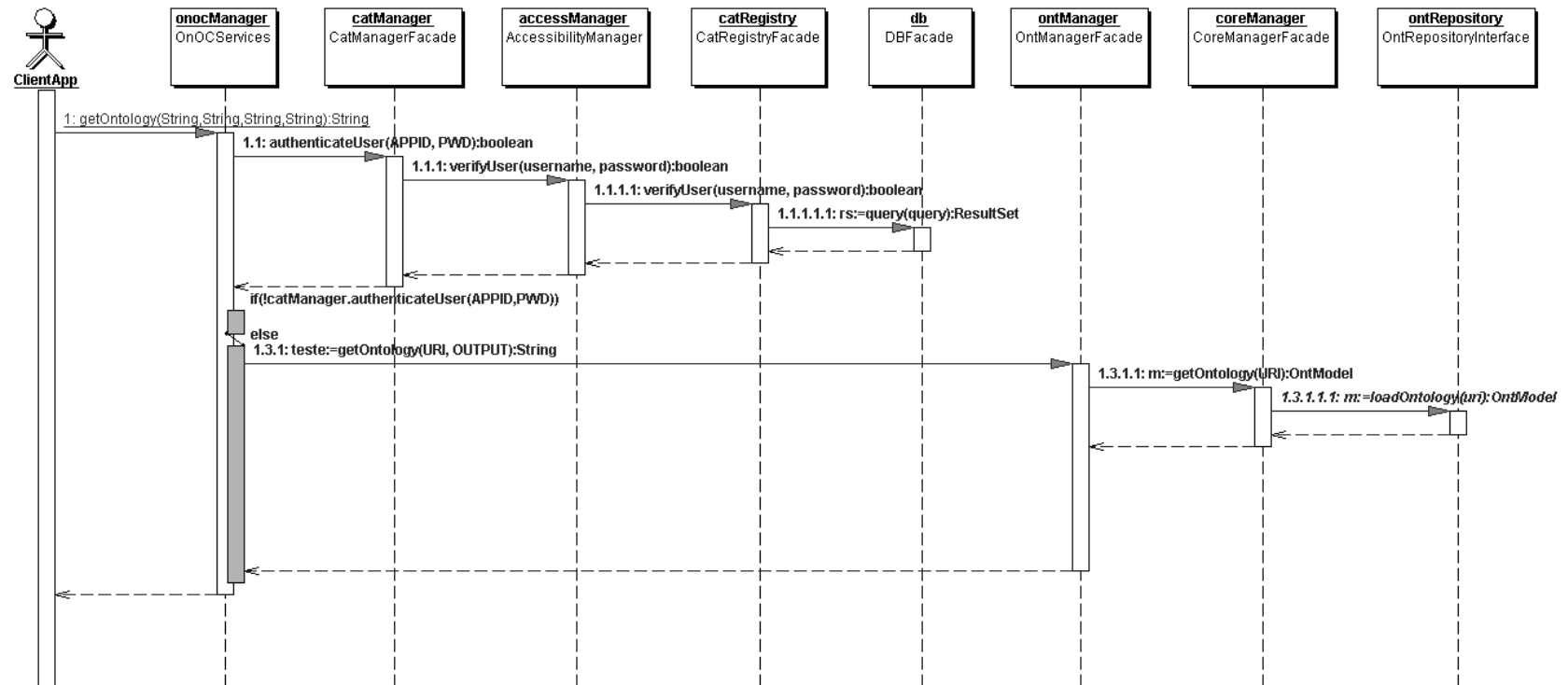
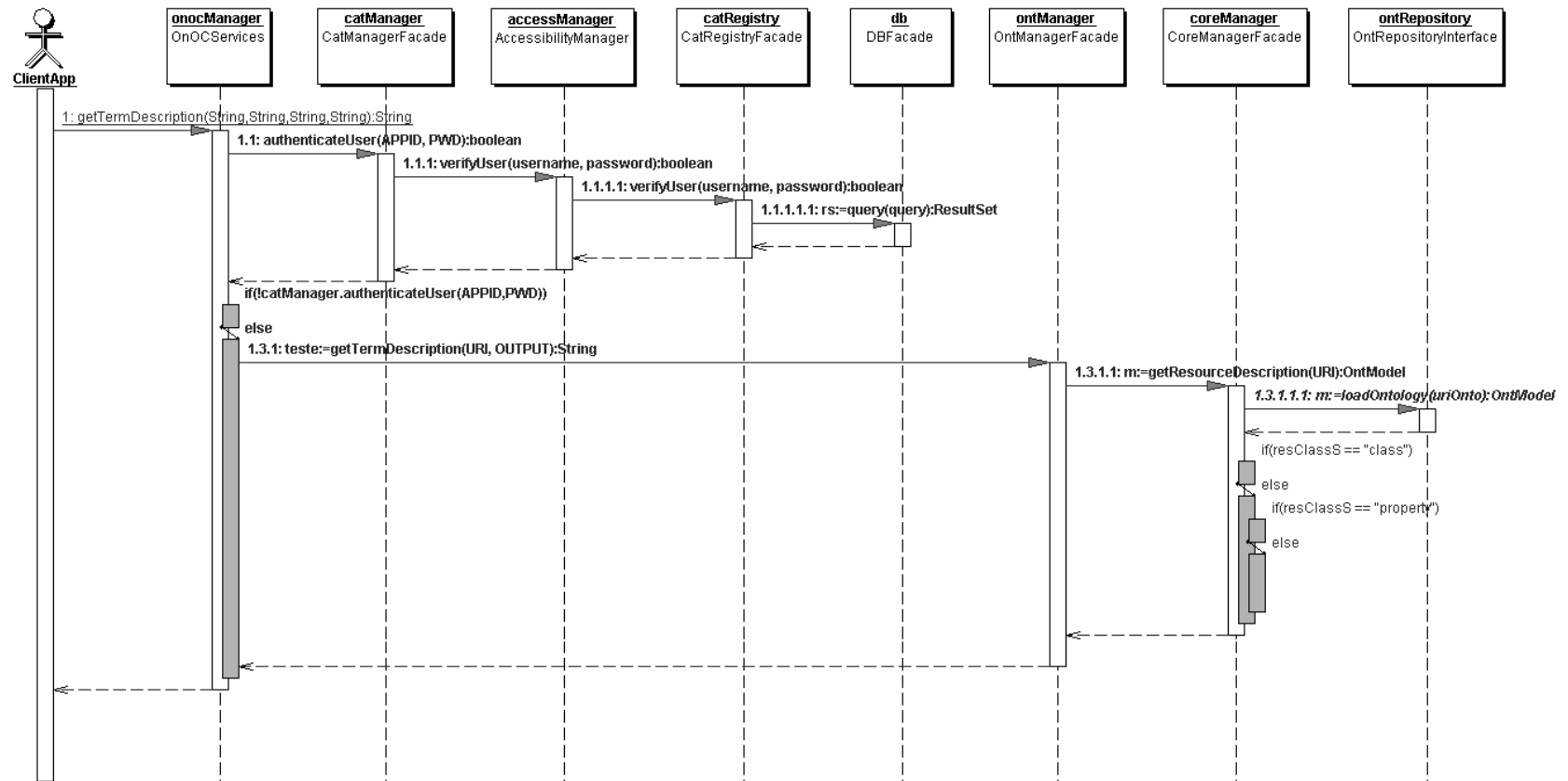
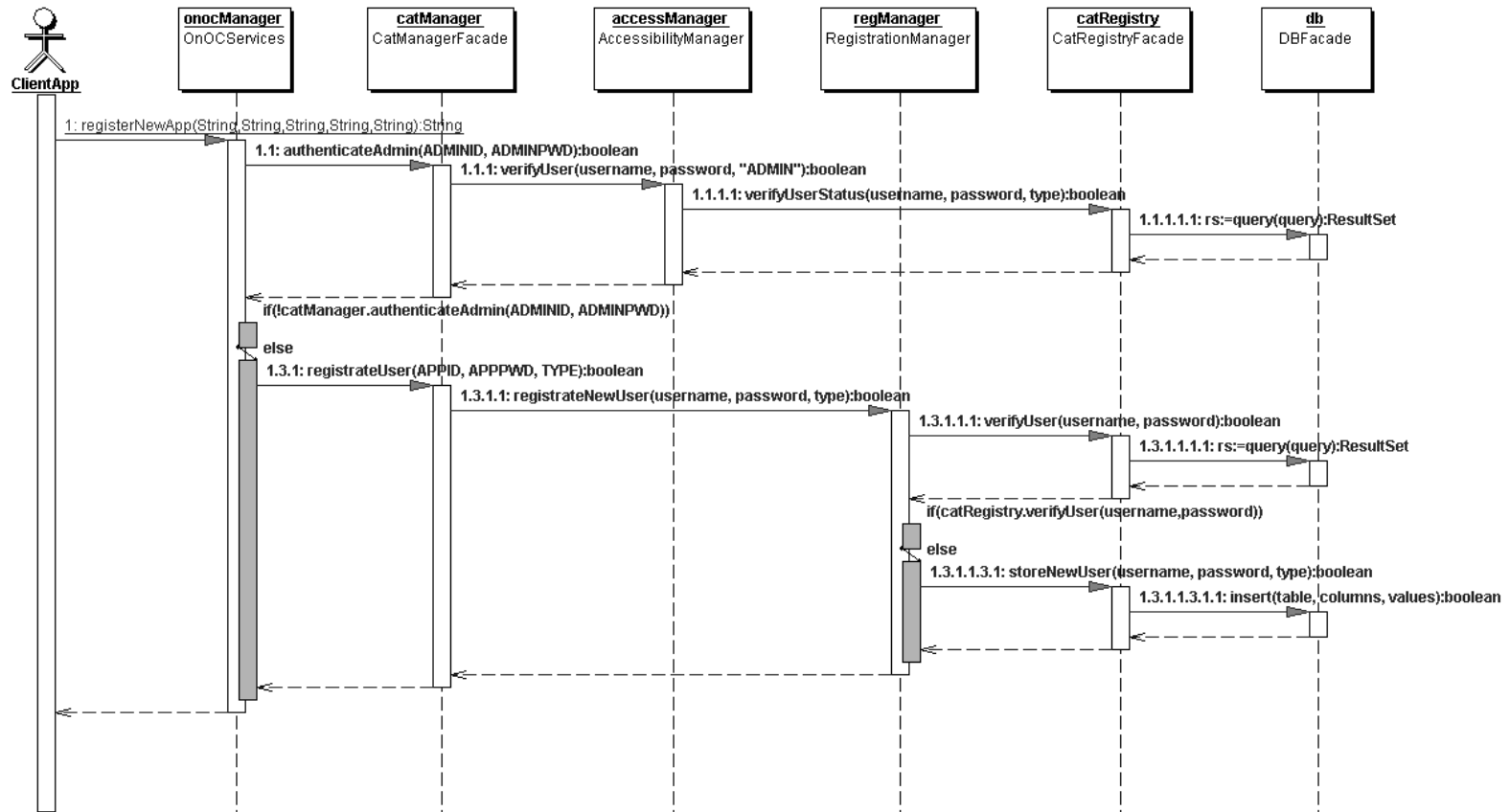
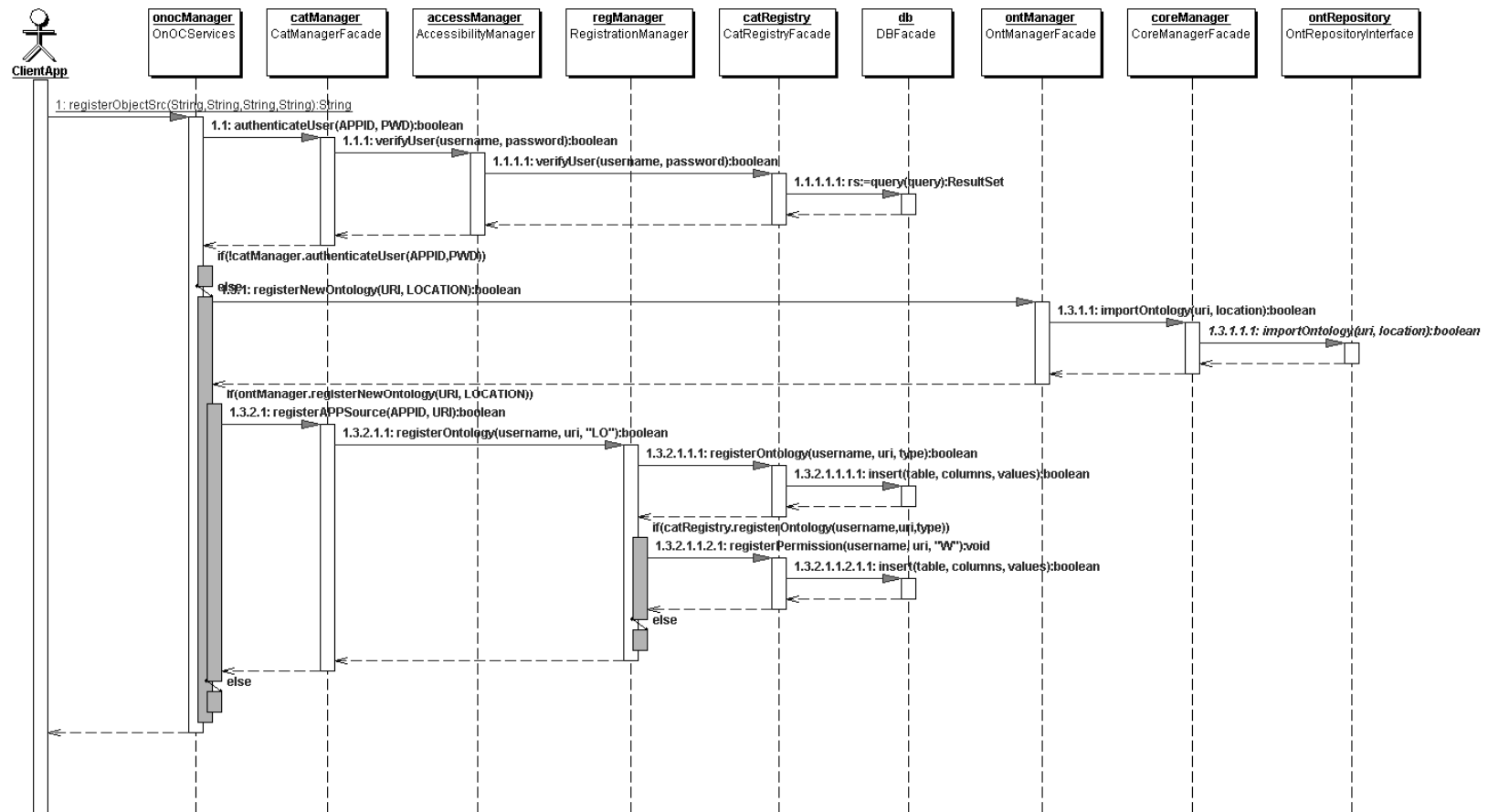


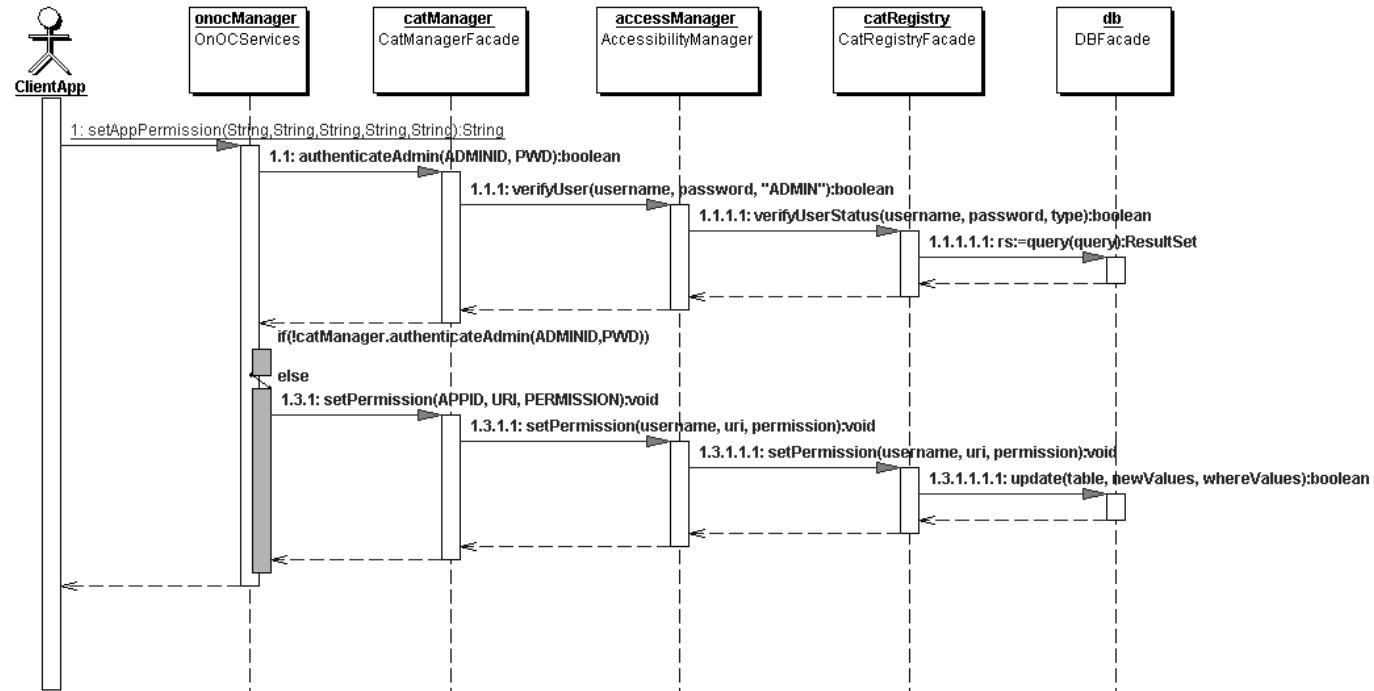
Figura 54 – Diagrama de sequência da operação *getOntology* (ontologias locais).

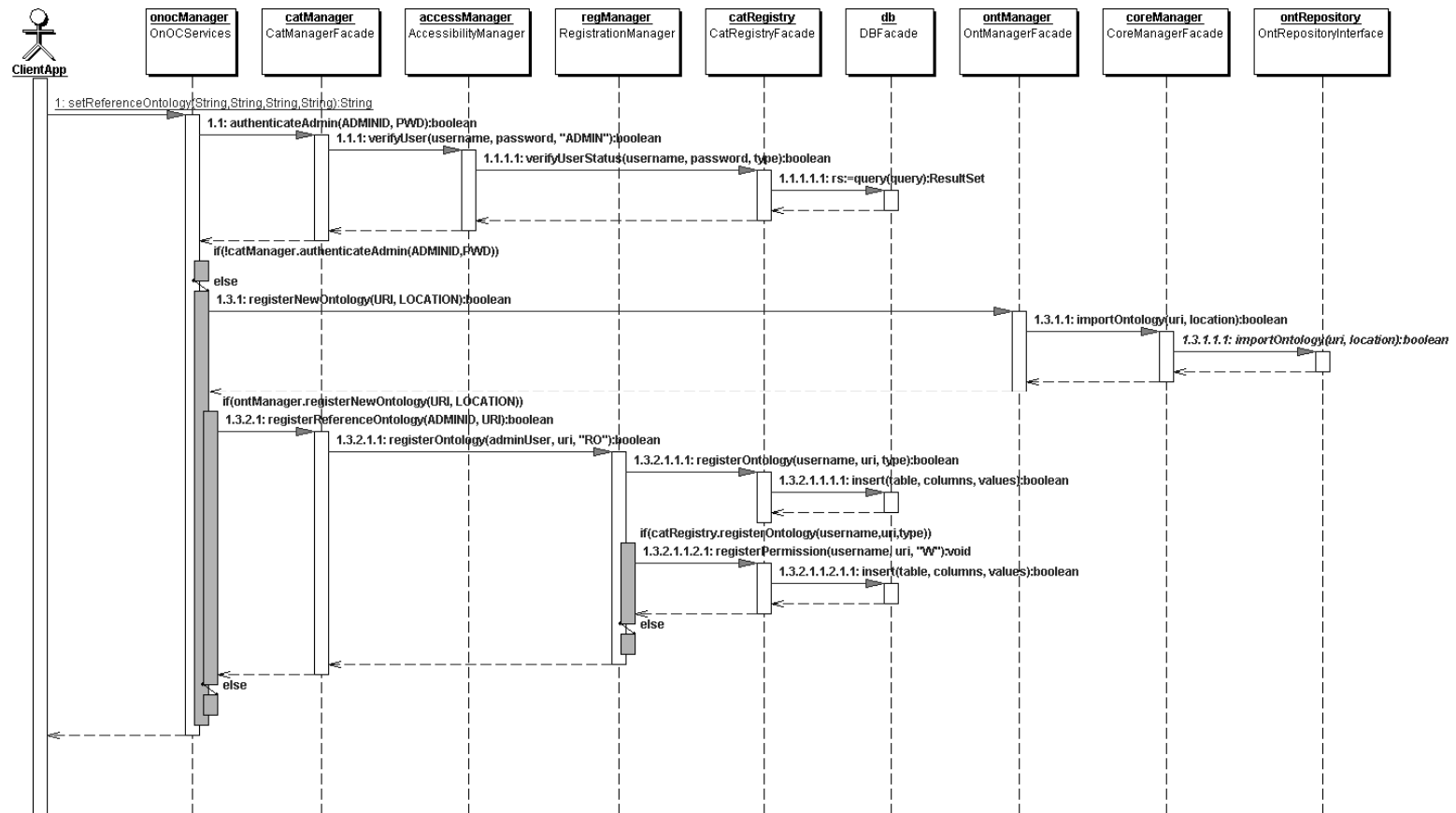


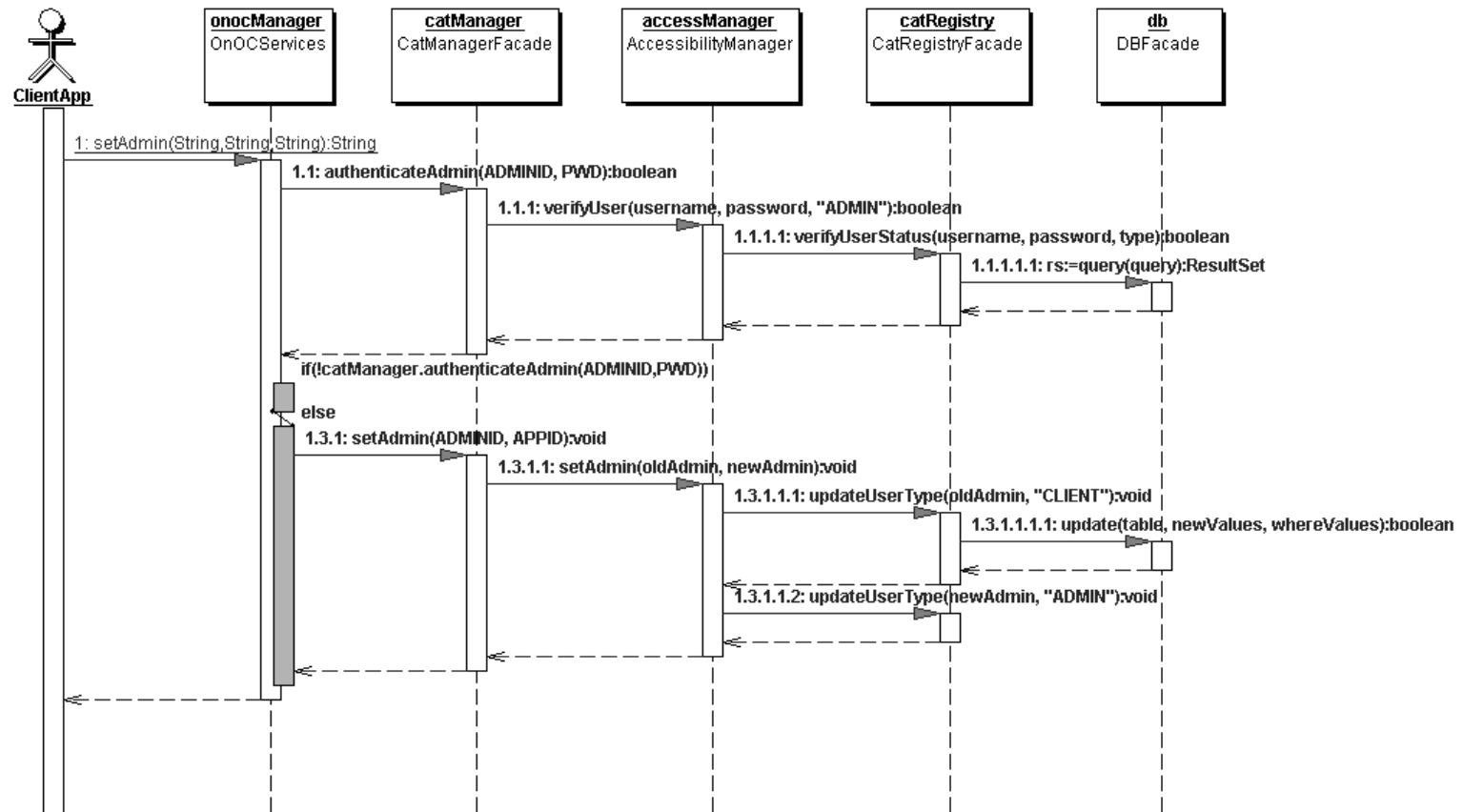
Figura 55 – Diagrama de sequência da operação *getTermDescription*.

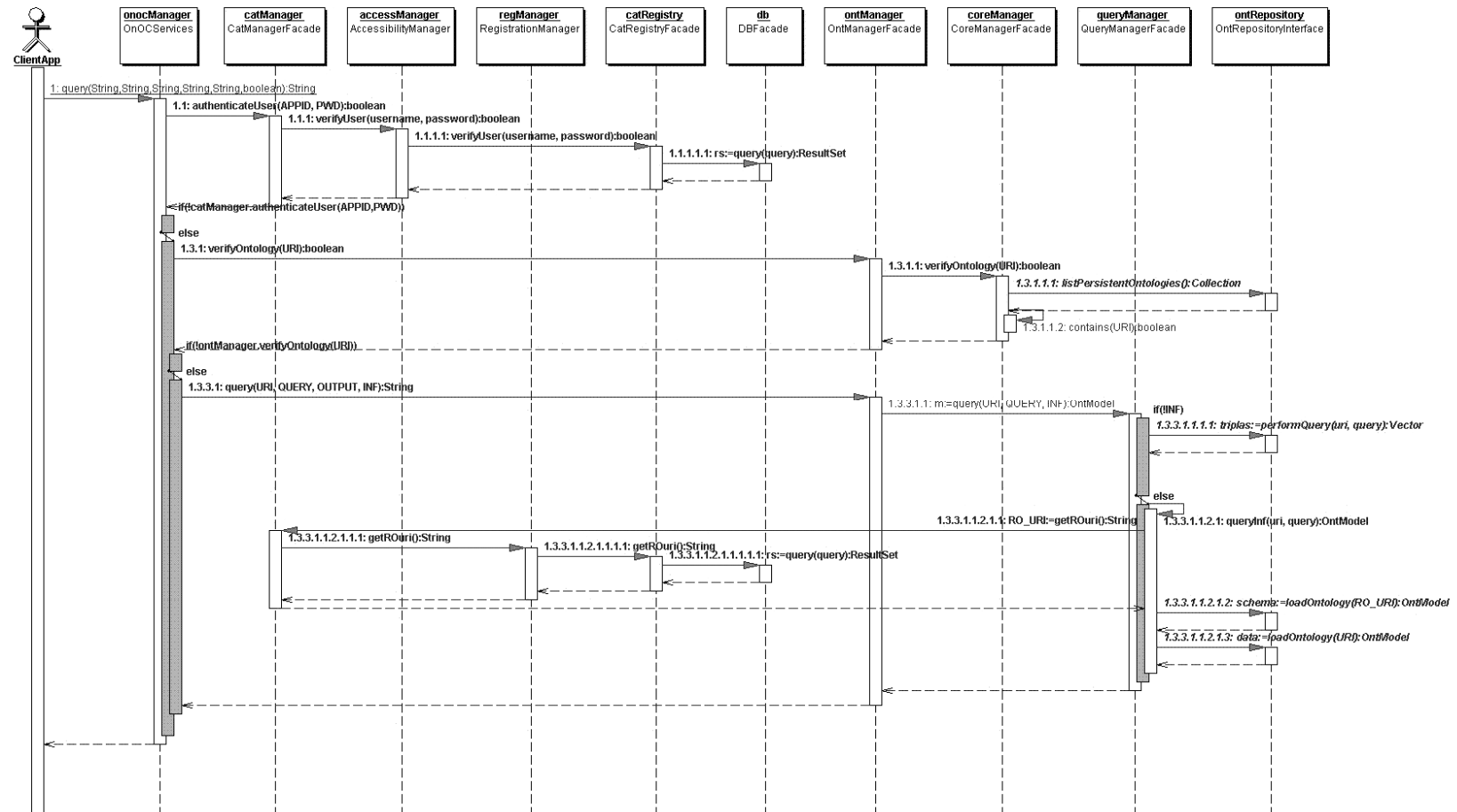
Figura 56 – Diagrama de sequência da operação *registerNewApp*.

Figura 57 – Diagrama de sequência da operação *registerObjectSrc*.

Figura 58 – Diagrama de sequência da operação *setAppPermission*.

Figura 59 – Diagrama de seqüência da operação *setReferenceOntology*.

Figura 60 – Diagrama de seqüência da operação *setAdmin*.

Figura 61 – Diagrama de sequência da operação *query*.