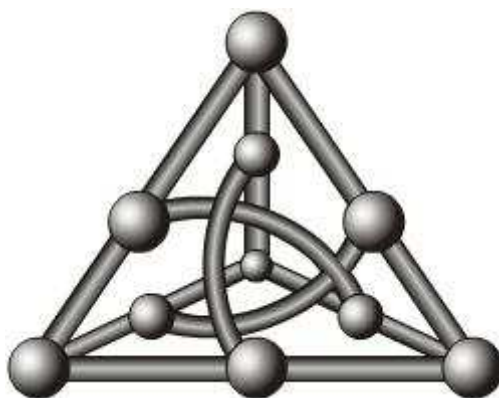

Hidra - Uma Biblioteca Java para o Desenvolvimento de Repositórios de Ativos de Software Orientado a Serviços baseada no modelo RAS

Danielli Urbieto e Pedro Souza

Trabalho de Conclusão de Curso apresentada à
Faculdade de Computação da
Universidade Federal de Mato Grosso do Sul



Orientador: Prof. MSc. Geraldo Barbosa Landre

Campo Grande, Junho de 2015

Resumo

Escreva o resumo aqui.

Palavras-chave: Java, SOA, Webservices, Repositório, Ativos Reusáveis de Software.

Abstract

Write the abstract here.

Keywords: Java, SOA, Webservices, Repository, Software Reusable Assets.

Sumário

1	Introdução	1
1.1	Seção	1
2	Embasamento Teórico e Trabalhos Relacionados	2
2.1	Seção	2
3	Tecnologias e Ferramentas	3
3.1	Seção	3
4	Hidra	4
4.1	Requisitos Funcionais e Não Funcionais	4
4.1.1	Derivação dos Requisitos da Biblioteca Hidra a partir dos Requisitos Arquiteturais da Cambuci	4
4.1.2	Requisitos Funcionais Hidra	7
4.1.3	Requisitos Não Funcionais Hidra	11
4.2	Modelo de Especificação de Ativos Reutilizáveis	12
4.2.1	Considerações Iniciais	12
4.2.2	Definições e Especificações de Ativos de Software	12
4.2.3	Ativos de Software	13
4.2.4	Considerações Finais	15
5	Conclusão	16

Lista de Figuras

4.1	Ilustração de um ativo de software	13
4.2	Estrutura de um Ativo de Softawe [1]	14
4.3	Estrutura de um ativo de Software (OMG, 2005)	15

Lista de Tabelas

4.1	Tabela de Requisitos Hidra	5
4.2	Requisitos Funcionais Hidra	7
4.3	Requisitos Não-Funcionais Hidra	11

Lista de Algoritmos

Capítulo 1

Introdução

Texto.

1.1 Seção

Caso seja necessário dividir em seções.

Capítulo 2

Embasamento Teórico e Trabalhos Relacionados

Texto.

2.1 Seção

Caso seja necessario dividir o texto do capitulo em secoes.

Capítulo 3

Tecnologias e Ferramentas

Texto.

3.1 Seção

Caso seja necessario dividir o texto do capitulo em secoes.

Capítulo 4

Hidra

Texto.

4.1 Requisitos Funcionais e Não Funcionais

4.1.1 Derivação dos Requisitos da Biblioteca Hidra a partir dos Requisitos Arquiteturais da Cambuci

Os requisitos funcionais e não-funcionais da biblioteca *Hidra* foram elicitados a partir da derivações dos requisitos arquiteturais da arquitetura de referência Cambuci [2], bem como, a partir da identificação de novos requisitos voltados diretamente para a definição da biblioteca *Hidra*.

A tabela a seguir apresenta os requisitos arquiteturais da arquitetura de referência Cambuci (colunas ID e Requisito Original), juntamente com os requisitos respectivamente derivados à biblioteca *Hidra* (coluna Requisitos Derivados). Os requisitos específicos da biblioteca *Hidra* foram identificados adotando como padrão as siglas RF para os requisitos funcionais e RN para a requisitos não-funcionais.

Tabela 4.1: Tabela de Requisitos Hydra

ID	Requisito Original	Requisito Derivado
RA-AS[1]	A arquitetura de referência deve possibilitar que repositórios de ativos de software incluam um novo ativo, que pode ser composto por vários artefatos.	RF-01 e RF-02
RA-AS[2]	A arquitetura de referência deve possibilitar que repositórios de ativos de software forneçam mecanismo para aceitação e certificação de ativos.	RF-03 e RF-04
RA-AS[3]	A arquitetura de referência deve possibilitar que repositórios de ativos de software desativem ativos que não serão mais utilizados.	RF-05
RA-AS[4]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam a classificação de um ativo e também informar o contexto de sua utilização.	RF-06 e RF-07
RA-AS[5]	A arquitetura de referência deve possibilitar que repositórios de ativos de software registrem a dependência entre ativos.	RF-08
RA-AS[6]	A arquitetura de referência deve possibilitar que repositórios de ativos de software notifiquem os interessados sobre mudanças que aconteçam no ativo.	RF-09
RA-AS[7]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam realizar buscas e recuperação dos ativos	RF-10, RF-11, RF-12
RA-AS[8]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam a navegação entre ativos	Não derivado para a versão atual da <i>Hydra</i> (Trabalho Futuro).
RA-AS[9]	A arquitetura de referência deve possibilitar que repositórios de ativos de software aceite múltiplas fontes de origem de ativos, com o objetivo de facilitar a integração entre equipes e entre repositórios diferentes.	RN-01
RA-AS[10]	A arquitetura de referência deve possibilitar que repositórios de ativos de software criem e armazenem múltiplas versões de um mesmo ativo.	RN-02
RA-AS[11]	A arquitetura de referência deve possibilitar que repositórios de ativos de software gerencie a configuração, como por exemplo, a definição dos itens do ativo que são configuráveis, o controle de mudanças dos itens do ativo que são configuráveis.	RN-03
RA-AS[12]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permita o registro de impressões dos usuários a respeito da versão do ativo que eles utilizaram.	Não derivado para a versão atual da <i>Hydra</i> (Trabalho Futuro).
RA-AS[13]	A arquitetura de referência deve possibilitar que repositórios de ativos de software registrem métricas coletadas sobre a utilização do ativo.	Fora do escopo da <i>Hydra</i> (Trabalho Futuro).

Continua na página seguinte

Tabela 4.1 – *Tabela de Requisitos Arquiteturais Cambuci*

ID	Requisito Original	Requisito Derivado
RA-AS[14]	A arquitetura de referência deve possibilitar que repositórios de ativos de software ofereçam informações relativas ao reúso, iniciativas de reúso, ativos mais usados, etc.	Fora do escopo da <i>Hidra</i> (Trabalho Futuro).
RA-AS[15]	A arquitetura de referência deve possibilitar que repositórios de ativos de software permitam o acesso de acordo com o papel que o usuário assume.	Não derivado para a versão atual da <i>Hidra</i> (Trabalho Futuro).
RA-AS[16]	A arquitetura de referência deve possibilitar que repositórios de ativos de software garantam a integridade dos ativos, ou seja, que eles não sofram alterações não autorizadas.	RN-04
RA-AS[17]	A arquitetura de referência deve possibilitar que repositórios de ativos de software realizem o gerenciamento de transação, garantindo a atomicidade, consistência, isolamento e durabilidade.	RF-13
RAS[1]	A arquitetura de referência deve possibilitar que repositórios de ativos de software desenvolvidos para persistir diferentes tipos de ativos possam ser facilmente integrados.	RF-14
RAS[2]	A arquitetura de referência deve possibilitar que repositórios de ativos de software implementados em linguagens de programação distintas e sob diferentes plataformas possam ser facilmente integrados.	RF-15
RAS[3]	A arquitetura de referência deve prover mecanismos para que repositórios de ativos de software na forma de serviços possam ser publicados e posteriormente descobertos por aplicações cliente.	RN-01
RAS[4]	A arquitetura de referência deve prover mecanismos para que repositórios de ativos de software orientados a serviço possam ser compostos por processos de negócio ou utilizados por aplicações cliente.	RN-01 e RN-04
RAS[5]	A arquitetura de referência deve viabilizar o desenvolvimento de repositórios de ativos de software que disponibilizem informações sobre suas características e direções normativas de uso, por meio de descrições padronizadas.	RF-16
RAS[6]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que disponibilizem descrições semânticas, permitindo assim sua classificação nos repositórios de serviço.	RF-17
RAS[7]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software que tenham à disposição informações e documentos relacionados às suas características de qualidade.	RF-18

Continua na página seguinte

Tabela 4.1 – Tabela de Requisitos Arquiteturais Cambuci

ID	Requisito Original	Requisito Derivado
RAS[8]	A arquitetura de referência deve prover mecanismos para a captura, monitoramento, registro e sinalização do não cumprimento de requisitos de qualidade estabelecidos entre serviços provedores e serviços clientes.	RF-19
RAS[9]	A arquitetura de referência deve viabilizar o desenvolvimento de repositório de ativos de software escalável, capaz de evoluir de maneira incremental, por meio da composição de novas funcionalidades disponíveis na forma de serviços.	RN-05
RAS[10]	A arquitetura de referência deve possibilitar que serviços de repositório de ativos de software e composições desses serviços sejam tratados uniformemente, ou seja, possam ser publicados, localizados e utilizados da mesma forma.	RN-01
RAS[11]	A arquitetura de referência deve possibilitar que serviços do repositório de ativos de software possam interagir diretamente ou por meio do uso de barramentos de serviço.	RN-01

4.1.2 Requisitos Funcionais Hidra

A partir da Tabela 4.1 os seguintes requisitos funcionais da biblioteca Hidra foram criados.

Tabela 4.2: Requisitos Funcionais Hidra

ID	Requisito	RA Cambuci	Solução
RF-01	A biblioteca <i>Hidra</i> deve permitir a inclusão de ativos de software , levando em consideração a composição de um ativo por diferentes artefatos.	RA-AS[1]	Os ativos reusáveis de software são armazenados no repositório em forma de diretórios, por meio, da segunda forma de armazenamento RAS [?].
RF-02	A biblioteca <i>Hidra</i> deve fornecer mecanismos a fim de listar artefatos que compõem um ativo de software armazenado no repositório.	RA-AS[1]	Requisito implementado por meio dos métodos <i>Asset.getSolution()</i> e <i>Asset.setSolution()</i> .
RF-03	A biblioteca <i>Hidra</i> deve possuir uma estrutura padronizada de representação, comunicação e armazenamento de ativos de software.	RA-AS[2]	Foi adotado o padrão RAS atualmente em sua versão 2.2.

Continua na página seguinte

Tabela 4.2 – *Requisitos Funcionais Hydra*

ID	Requisito	RA Cambuci	Solução
RF-04	A biblioteca <i>Hydra</i> deve garantir que todo novo ativo de software seja validado e certificado de acordo com o padrão adotado.	RA-AS[2]	As regras especificadas no padrão RAS, expressas em forma de um XSD <i>NomeArquivoXSD.xsd</i> , são consultadas ao validar e certificar um Ativo antes de qualquer atualização ou inserção (método <i>Asset.validate()</i>).
RF-05	A biblioteca <i>Hydra</i> deve garantir que ativos de software, que não forem mais utilizados, possam ser removidos do repositório.	RA-AS[3]	Requisito implementado por meio do método <i>Repository.removeAsset(Asset asset)</i> .
RF-06	A biblioteca <i>Hydra</i> deve possibilitar a adição de informações para classificação de um ativo e também o contexto de sua utilização.	RA-AS[4]	Requisito implementado por meio dos métodos <i>Asset.getClassification()</i> e <i>Asset.setClassification()</i> .
RF-07	A biblioteca <i>Hydra</i> deve possibilitar a adição de informações sobre regras para instalação, personalização, e utilização do ativo.	extensão do requisito RA-AS[4] baseando-se no padrão RAS.	Requisito implementado por meio dos métodos <i>Asset.getUsage()</i> e <i>Asset.setUsage()</i> .
RF-08	A biblioteca <i>Hydra</i> deve possibilitar o registro de dependência entre ativos.	RA-AS[5]	Requisito implementado por meio dos métodos <i>Asset.getRelatedAssets()</i> e <i>Asset.setRelatedAsset()</i> .
RF-09	A biblioteca <i>Hydra</i> deve oferecer informações relevantes a todos os interessados, sobre mudanças que aconteçam no ativo de software: data de alteração, autor da alteração, o que foi alterado e descrição sobre a alteração.	RA-AS[6]	Requisito implementado por meio do método <i>Asset.getLog()</i>
RF-10	A biblioteca <i>Hydra</i> deve fornecer mecanismos a fim de listar ativos armazenados no repositório.	RA-AS[7]	Requisito implementado por meio do método <i>Repository.listAssets()</i>

Continua na página seguinte

Tabela 4.2 – *Requisitos Funcionais Hydra*

ID	Requisito	RA Cambuci	Solução
RF-11	A biblioteca <i>Hydra</i> deve fornecer mecanismos a fim de recuperar um ativo armazenado no repositório (download).	RA-AS[7]	Requisito implementado por meio do método <i>Repository.retrieveAsset()</i>
RF-12	A biblioteca <i>Hydra</i> deve fornecer mecanismos a fim de buscar ativos armazenados no repositório.	RA-AS[7]	Requisito não implementado na versão atual da biblioteca <i>Hydra</i> (Trabalho Futuro).
RF-13	A biblioteca <i>Hydra</i> deve fornecer mecanismos que garantem a atomicidade, consistência e isolamento de transações de controle de ativos de software	RA-AS[17]	A biblioteca provê recursos para que as transações sejam controladas considerando os aspectos citados: os recursos da API jGit e a camada de serviços. Mas essa implementação deverá ser realizada diretamente no repositório.
RF-14	A biblioteca <i>Hydra</i> deve fornecer mecanismos que permitem a persistência de diferentes tipos de ativos.	RAS[1]	O padrão RAS, adotado na biblioteca <i>Hydra</i> para a implementação dos requisitos relacionados ao Ativo de Software, permite a persistência de diferentes tipos de ativos, desde que a estrutura de cada ativo seja descrita em sua solução (métodos <i>Asset.getSolution()</i> e <i>Asset.setSolution()</i>).

Continua na página seguinte

Tabela 4.2 – *Requisitos Funcionais Hydra*

ID	Requisito	RA Cambuci	Solução
RF-15	A biblioteca <i>Hydra</i> deve fornecer mecanismos que permitem a persistência ativos implementados em diferentes linguagens de programação.	RAS[2]	O padrão RAS, adotado na biblioteca <i>Hydra</i> para a implementação dos requisitos relacionados ao Ativo de Software, permite a persistência de ativos implementados em diferentes linguagens de programação, desde que as regras para instalação, personalização, e utilização de cada ativo seja descrita na especificação de seu uso (métodos <i>Asset.getUsage()</i> e <i>Asset.setUsage()</i>).
RF-16	A biblioteca <i>Hydra</i> deve fornecer uma camada de serviços (Webservice) com informações sobre suas características e direções normativas de uso, por meio de descrições padronizadas seguindo o padrão DNS para descoberta de serviços.	RAS[5]	Requisito não implementado na versão atual da biblioteca <i>Hydra</i> (Trabalho Futuro).
RF-17	A biblioteca <i>Hydra</i> deve viabilizar o desenvolvimento de um repositório de ativos de software com uma camada webservice com descrições semânticas, permitindo assim sua classificação nos repositórios de serviços.	RAS[6]	Requisito não implementado na versão atual da biblioteca <i>Hydra</i> (Trabalho Futuro).
RF-18	A biblioteca <i>Hydra</i> deve viabilizar o desenvolvimento de repositório de ativos de software que tenham à disposição informações e documentos relacionados às suas características de qualidade.	RAS[7]	Requisito não implementado na versão atual da biblioteca <i>Hydra</i> (Trabalho Futuro).
RF-19	A arquitetura de referência deve prover mecanismos para a captura, monitoramento, registro e sinalização do não cumprimento de requisitos de qualidade estabelecidos entre serviços provedores e serviços clientes.	RAS[8]	Requisito não implementado na versão atual da biblioteca <i>Hydra</i> (Trabalho Futuro).

4.1.3 Requisitos Não Funcionais Hidra

A partir da Tabela 4.1 os requisitos não funcionais da biblioteca Hidra foram criados.

Tabela 4.3: Requisitos Não-Funcionais Hidra

ID	Requisito	RA Cambuci	Solução
RN-01	A biblioteca <i>Hidra</i> deve fornecer mecanismos para que repositórios de ativos de software aceitem múltiplas fontes de origem de ativos, por meio de serviços web que possam ser publicados, localizados e utilizados de maneira uniforme.	RA-AS[9], RAS[3], RAS[4], RAS[10]	Requisito é atendido por meio da camada de serviço provida pela biblioteca, que segue o padrão REST, que permitirá ao repositório desenvolvido, tendo como base a <i>Hidra</i> , fácil acesso e integração a múltiplas ferramentas.
RN-02	A biblioteca <i>Hidra</i> deve fornecer mecanismos de versionamento aos ativos de software.	RA-AS[10]	Requisito é atendido por meio da camada de persistência provida pela biblioteca, que utiliza a API jGit para manipulação das operações de Gerenciamento de Configuração sobre um repositório Git, e permitirá que o repositório armazene múltiplas versões de um mesmo ativo.
RN-03	A biblioteca <i>Hidra</i> deve oferecer mecanismos para gerenciamento da configuração de ativos de software.	RA-AS[11]	Requisito é atendido por meio da camada de persistência provida pela biblioteca, que utiliza a API jGit para manipulação das operações de Gerenciamento de Configuração sobre um repositório Git, e permitirá que o repositório gerencie configurações de ativos de software.
RN-04	A biblioteca <i>Hidra</i> permitir que repositórios garantam que seus ativos de software não sofram alterações não autorizadas.	RA-AS[16], RAS[4]	Requisito é atendido por meio do controle de usuários da camada de persistência provida pela biblioteca, que utiliza a API jGit para manipulação das operações de Gerenciamento de Configuração sobre um repositório Git. Na versão inicial, a biblioteca utiliza um usuário padrão informado no seu arquivo de propriedades (hidra.properties).

Continua na página seguinte

Tabela 4.3 – *Requisitos Não-funcionais Hydra*

ID	Requisito	RA Cambuci	Solução
RN-05	A biblioteca <i>Hydra</i> deve ser extensível de modo a viabilizar o desenvolvimento de repositório de ativos de software escalável, capaz de evoluir de maneira incremental, por meio da composição de novas funcionalidades disponíveis na forma de serviços.	RAS[9]	Requisito é atendido por meio dos padrões adotados para a implementação da biblioteca: i) Padrão RAS para representação e manipulação de Ativos de Software Reusáveis; ii) Divisão dos recursos providos em camadas (jGit para persistência, Hydra para regras de negócio, HydraService para fornecimento de serviços; iii) Padrões de Projeto (tanto padrões GRASP quanto padrões GoF) adotados na implementação, como por exemplo, Singleton, Facade, Strategy, Especialista na Informação).

4.2 Modelo de Especificação de Ativos Reutilizáveis

4.2.1 Considerações Iniciais

No âmbito da Engenharia de software, a utilização de ativos de softwares reutilizáveis são de grande interesse, visto que a pratica de sua utilização, fornece vantagens e benefícios muito importantes ao desenvolvimento de softwares, como melhoria de qualidade, aumento de produtividade e redução de custos. A biblioteca desenvolvida neste trabalho, possui relação com o modelo de especificação de ativos reutilizáveis (RAS OMG-2005), pois contempla o desenvolvimento de repositórios de ativos de software que estejam de acordo com as orientações do padrão citado. Essa seção explora os principais conceitos que abragem as orientações especificadas no modelo RAS.

4.2.2 Definições e Especificações de Ativos de Software

Um ativo de software segundo a definição da OMG (Object Management Group - 2005), é algo que visa a solução de algum problema em um determinado contexto, possui pontos de variabilidades e regras de como seu reuso pode ser realizado. A figura a seguir ilustra a composição de um ativo de software.

elementos fundamentais. O perfil de uma ativo não altera a definição semântica de um elemento descrito no núcleo. O núcleo RAS não é instanciável, assim como uma classe abstrata, portanto um ativo deve possuir um perfil particular, esse perfil pode estender o núcleo RAS, ou estender um outro perfil como mostrado na figura a seguir.

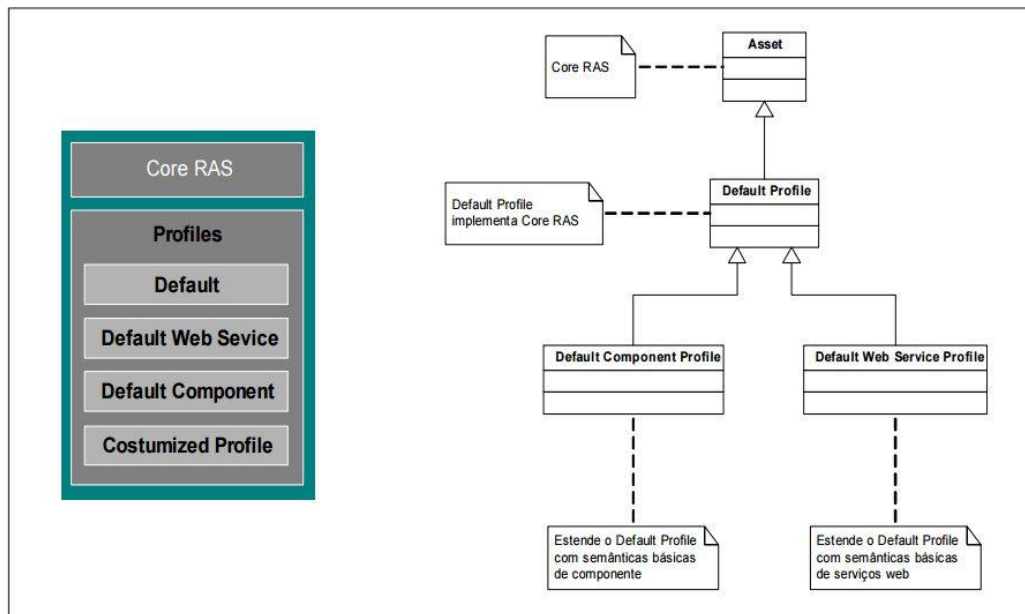


Figura 4.2: Estrutura de um Ativo de Software [1]

Um ativo de software é composto pela sua Classificação (*Classification*), Solução (*Solution*), Utilização (*Usage*), e pelo seu Perfil (*Profile*). Caso esteja relacionado a outros ativos também possuirá (Related Assets). A descrição de cada atributo é informada a seguir:

- Classificação (*Classification*): : um conjunto de descritores que são usados para fazer a classificação do ativo, como descrever os contextos nos quais a reutilização do ativo é relevante.
- Solução (*Solution*): contém as descrições de um ou mais artefatos que fazem parte do ativo.
- Utilização (*Soltuion*): contém as regras para a instalação, customização e uso do ativo, ou seja, possui as instruções correspondentes a cada artefato que o ativo possui e um contexto de referência.
- Ativos Relacionados (*Related Assets*): condiz com as descrições dos relacionamentos existentes entre os ativos.
- Perfil (*Profile*): representa o perfil do ativo.

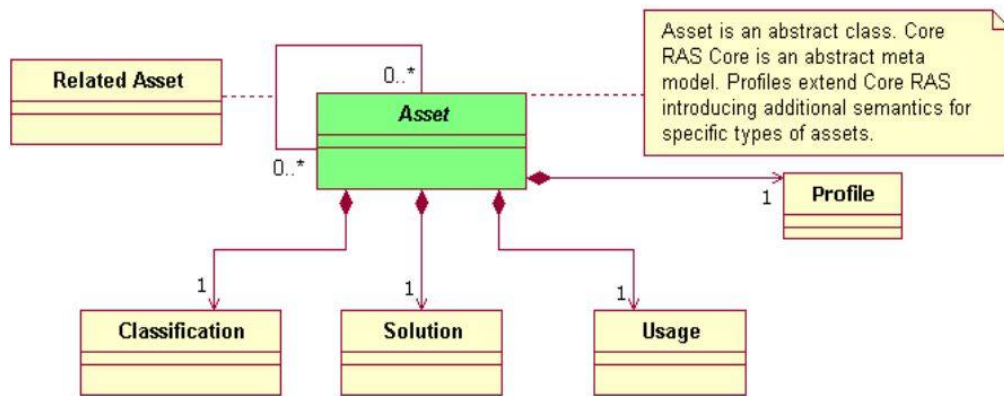


Figura 4.3: Estrutura de um ativo de Software (OMG, 2005)

4.2.4 Considerações Finais

Nesta seção apresentamos as principais definições e orientações do modelo RAS utilizado como fonte de informação para desenvolvimento deste trabalho. Um dos aspectos abordados pela biblioteca Hidra é a validação, aceitação e persistência, em repositórios de ativos de software, condizentes com o padrão RAS descrito nesta seção.

Capítulo 5

Conclusão

Escreva a conclusao aqui.

Referências Bibliográficas

- [1] Henrique Rocha de Faria and Reginaldo Arakaki. Um Modelo de Processo de Apoio ao Desenvolvimento de Software Baseado em Componentes, Orientado a Qualidade, e Centrado em um Repositório, 2005.
- [2] Márcio Osshiro and Maria Istela Cagnin. Reutilização De Modelagem De Negócios Baseada Em Visões, 2014.