

KATIA CRISTINA APARECIDA DAMACENO BORGES

**UMA ABORDAGEM BASEADA EM ARQUITETURA DE
SOFTWARE PARA CONSTRUÇÃO DE OBJETOS DE
APRENDIZAGEM**

**Dissertação apresentada à Universidade
Federal de Viçosa, como parte das
exigências do Programa de Pós-Graduação
em Ciência da Computação, para obtenção
do título de *Magister Scientiae*.**

VIÇOSA
MINAS GERAIS - BRASIL
2015

**Ficha catalográfica preparada pela Biblioteca Central da Universidade
Federal de Viçosa - Câmpus Viçosa**

T

B732a
2015
Borges, Katia Cristina Aparecida Damaceno, 1978-
Uma abordagem baseada em arquitetura de software para
construção de objetos de aprendizagem / Katia Cristina
Aparecida Damaceno Borges. – Viçosa, MG, 2015.
xiv, 88f. : il. (algumas color.) ; 29 cm.

Inclui anexos.

Inclui apêndices.

Orientador: José Luís Braga.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 84-88.

1. Arquitetura de Software. 2. Padrões de Software.
 3. Educação - Recursos de redes de computadores.
- I. Universidade Federal de Viçosa. Departamento de Informática.
Programa de Pós-graduação em Ciência da Computação.
II. Título.

CDD 22. ed. 005.1

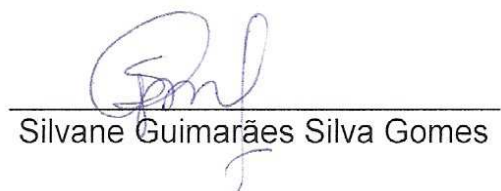
KATIA CRISTINA APARECIDA DAMACENO BORGES

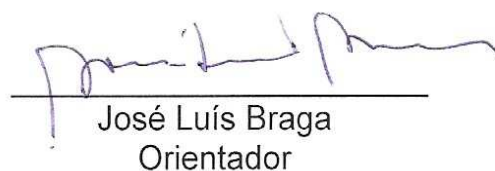
**UMA ABORDAGEM BASEADA EM ARQUITETURA DE
SOFTWARE PARA CONSTRUÇÃO DE
OBJETOS DE APRENDIZAGEM**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 04 de agosto de 2015.


Jugurta Lisboa Filho


Silvane Guimarães Silva Gomes


José Luís Braga
Orientador

*Dedico essa dissertação
A Deus*

Aos meus filhos João Pedro e Beatriz

Ao meu esposo Wander

*Ao meu orientador
José Luis*

*A minha co-orientadora
Iris Fabiana*

“Desistir... Eu já pensei seriamente nisso, mas nunca me levei a sério; é que tem mais chão nos meus olhos do que cansaço nas minhas pernas, mais esperança nos meus passos do que tristeza nos meus ombros, mais estrada no meu coração do que medo na minha cabeça”

(Cora Coralina)

AGRADECIMENTOS

Somente aqueles que nos são próximos sabem o quanto realmente custou à finalização de um trabalho. Não digo perto fisicamente, mas também à distância por conhecer tudo que significa a conclusão de mais esta etapa.

Neste sentido gostaria de contar aqui um pouco da história por traz deste trabalho: o conceito inicial do projeto surgiu a partir de um curso, modalidade à distância, ministrado pelo grupo de pesquisa INTERA da UFABC: Metodologia para Criação de OA. Na ocasião a professora Dr^a Juliana Cristina Braga concordou com minha matrícula no curso, pois o mesmo é destinado a professores vinculados a instituição. A partir daí a Dr^a Juliana sempre esteve presente e quando solicitada contribuía com o projeto.

Outro membro valioso que não mediu esforços em ajudar com diferentes rumos que o projeto seguiu foi a professora Dr^a Iris Fabiana de Barcelos Tronto, UFV campus de Rio Paranaíba, co-orientadora desse projeto. Mesmo durante o período de incerteza sobre a forma de padronizar a construção de OAs de diferentes tipos dentro de uma única Arquitetura de Software, ela não desistiu e chamou reforços para auxiliar-nos.

Um reforço imensurável foi o professor Dr. Eduardo Martins Guerra, INPE, este foi o visualizador da possibilidade de trabalharmos com a definição de Padrões para criação de OAs e assim, aplicarmos na construção da Arquitetura. Foi publicado o artigo: Padrões de Criação de OAs, no X Sugar Loaf Plop em 2014.

Por fim, mas não menos importante, está a figura do meu orientador, professor Dr. José Luis Braga. Este não foi apenas um grande pesquisador com visão ampliada do contexto ao qual estaria inserido este projeto, mas sim um pilar norteador de todo trajeto a ser percorrido a fim de alcançar o objetivo desta dissertação.

Assim, gostaria de agradecer a vocês: Dr^a Juliana, Dr^a Iris Fabiana, Dr. Eduardo e Dr. José Luis; por toda ajuda e apoio neste processo ao qual me conduziram.

Agradeço a Deus por toda força que deu a mim e minha família no decorrer deste período. Por nos manter unidos, apesar de toda dificuldade por nos enfrentada. Creio que sem a força vinda do Pai nada teria sido possível.

Ao meu ESPOSO, **Wander**, pela ajuda, carinho e atenção nos momentos mais difíceis pelos quais passei. Nosso amor se fortaleceu mais uma vez e vi o motivo real da nossa escolha anos atrás.

Aos meus FILHOS, **João Pedro** e **Beatriz**, vocês foram a luz que me conduziu e o local onde busquei forças para não desistir. Quando nos tornamos pais não sabemos ainda o quanto um filho será importante para o resto de nossa vida e hoje eu sei, vocês são o motivo mais importante para que eu continue a lutar.

A meus PAIS, **Eurípedes e Madalena**, por sua criação e orientação durante toda minha vida. Vocês são o motivo da minha vitória de todos os dias.

A todos os professores do Departamento de Informática (DPI-UFV), em especial, aos que durante minha estada em Viçosa foram meus professores. Se não fosse seus ensinamentos não chegaria onde estou hoje.

Aos funcionários do DPI, em especial ao **Altino**, que nunca mediu esforços para auxiliar em todos os assuntos relativos a problemas do mestrado.

Agradeço a todos os colegas e amigos que fiz no mestrado. Em especial gostaria de agradecer a vocês: **Cristiane Lana, Marques, Paola, Willian Reis, Marcos Vinícius**; por todos os momentos que passamos juntos, seja rindo ou sofrendo, mas principalmente unidos.

Agradeço ao Apoio Financeiro fornecido pela Coordenação de Aperfeiçoamento Pessoal de Nível Superior (CAPES), pois sem o mesmo seria impossível dedicar a esta pesquisa.

Enfim, agradeço a todos que direta ou indiretamente sempre torceram pelo meu sucesso. Seja com palavras, ações ou apenas um olhar compreensivo. O importante são as ações que fazemos e neste caso isto contou muito para alcançar o sucesso.

SUMÁRIO

LISTA DE FIGURAS.....	ix
LISTA DE TABELAS.....	xi
LISTA DE ACRÔNIMOS	xii
RESUMO	xiii
ABSTRACT.....	xiv
1 INTRODUÇÃO	1
1.1 O Problema.....	1
1.2 Justificativa	1
1.3 Hipótese	3
1.4 Objetivos	3
1.5 Organização do Texto	3
2 CONTEXTO	5
2.1 Objetos de Aprendizagem	5
2.1.1 Processo de construção de objetos de aprendizagem	6
2.2 Arquitetura de Software.....	8
2.2.1 Técnicas em Arquitetura de Software.....	9
2.2.1.1 Padrão Arquitetural	10
2.2.1.2 Modelo de Referência.....	15
2.2.1.3 Arquitetura de Referência	16
2.2.1.4 Padrões de Projeto	17
2.2.1.5 Idiomas	20
2.2.1.5 Linguagens de Descrição Arquitetural de Software.....	20
2.3 ZACHMAN FRAMEWORK.....	21
2.3.1 Métodos para Construção da Arquitetura utilizando Zachman Framework	25
2.4 TRABALHOS CORRELATOS.....	28

3	PADRÕES DESENVOLVIDOS PARA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM.....	29
3.1	Padrão Objeto de Aprendizagem Abstrato.....	29
3.2	Padrão Recuperação de Objetos de Aprendizagem.....	32
3.3	Padrão Objetos de Aprendizagem Composto.....	39
3.4	Padrão Metadados Essenciais para Objetos de Aprendizagem.....	40
4	ARQUITETURA DE APOIO À MODELAGEM CONCEITUAL DE OBJETOS DE APRENDIZAGEM.....	44
4.1	Nível Arquitetural "ESCOPO" Células A, B, C, D, E, F.....	44
4.1.1	Célula A - Dados (o quê).....	44
4.1.2	Célula B – Função (como).....	46
4.1.3	Célula C - Rede (Onde).....	46
4.1.4	Célula D - Pessoas (Quem).....	47
4.1.5	Célula E - Tempo (Quando).....	48
4.1.6	Célula F - Motivação (Por que).....	48
4.2	Nível Arquitetural "modelo de negócios" Células: G, H, I, J, K, L.....	48
4.2.1	Célula G - Dados (o quê) – Modelo Semântico.....	48
4.2.2	Célula H - Função(como) – Modelo de Processos da Organização.....	49
4.2.3	Célula I - Rede (Onde)- Sistema Logístico da Organização.....	49
4.2.4	Célula J - Pessoas (Quem) – Modelo de Workflow.....	50
4.2.5	Célula K - Tempo (Quando) – Cronograma principal.....	51
4.2.6	Célula L - Motivação (Por que) – Plano de Negócio.....	51
4.3	Nível Arquitetural "Modelo de Sistema"- Células M, N, O, P, Q.....	51
4.3.1	Célula M - Dados (o quê) Modelo Lógico de dados.....	51
4.3.2	Célula N – Função (como) – Arquitetura de Aplicações.....	52
4.3.3	Célula O - Rede (Onde) – Arquitetura Distribuída do Sistema.....	55
4.3.4	Célula P - Pessoas (Quem) – Arquitetura da Interface Humana.....	56
4.3.5	Célula Q - Tempo (Quando) – Estrutura do Processamento.....	57
4.3.6	Célula R - Motivação (Por que) – Regras do Negócio.....	58
5	ANÁLISE DE UM OBJETO DE APRENDIZAGEM COM RELAÇÃO À ARQUITETURA PROPOSTA.....	59
5.1	Aula Virtual.....	59
5.1.1	Nível Arquitetural "ESCOPO" Células A, B, C, D, E, F.....	59
5.1.2	Nível Arquitetural "modelo de negócios" Células: G, H, I, J, K, L.....	62
5.1.3	Nível Arquitetural "Modelo de Sistema"- Células M, N, O, P, Q.....	64

5.2	Discussão dos Resultados	66
6	CONCLUSÃO	68
	APÊNDICE A – GLOSSÁRIO DE TERMOS.....	71
	APÊNDICE B – MODELO DE DOCUMENTO DE SOLICITAÇÃO DE OBJETO DE APRENDIZAGEM.	80
	ANEXO A MODELO CONTEXTUALIZAÇÃO DE UM OBJETO DE APRENDIZAGEM	81
	ANEXO B – MODELO DE DOCUMENTO DE ELICITAÇÃO DE REQUISITOS OA	82
	ANEXO C – MODELO DOCUMENTO DE TESTES DE OA	83
	REFERÊNCIAS BIBLIOGRÁFICA	84

LISTA DE FIGURAS

Figura 1 Relacionamento entre os padrões para a composição de uma arquitetura de software.....	10
Figura 2 Sistema de Padrões POSASistema de Padrões POSA.....	11
Figura 3 Zachman Framework.....	24
Figura 4 Padrão Objeto de Aprendizagem Abstrato.....	30
Figura 5 Padrão Objeto de Aprendizagem Abstrato para Taxonomia.....	31
Figura 6 Padrão Metadado Obejto de Aprendizagem.....	33
Figura 7 OA "A basic property of integrals".....	35
Figura 8 Busca através dos Medados do OA " A basic property of integrals".....	36
Figura 9 Padrão Metadados Cestacore.....	38
Figura 10 Padrão Objeto de Aprendizagem Composto.....	39
Figura 11 Célula Dados (perspectiva planejador).....	45
Figura 12 Célula Função (perpectiva planejador).....	46
Figura 13 Business Location Diagram - Localização dos Órgãos Gestores da Educação no Brasil.....	47
Figura 14 Célula Pessoas (Perspectiva planejador).....	47
Figura 15 Célula Tempo (Perspectiva Planejador).....	48
Figura 16 Modelo Semântico.....	49
Figura 17 Órgãos Gestores da Educação por Estado.....	50
Figura 18 Organograma da Equipe de Criação de OAs.....	50
Figura 19 Padrão Objeto de Aprendizagem Abstrato.....	51
Figura 20 Padrão Metadados.....	52
Figura 21 Padrão Objetos de Aprendizagem Composto.....	52
Figura 22 Células Pessoas (perspectiva planejador).....	53
Figura 23 Diagrama de Atividades: Solicitação de Desenvolvimento de OA.....	53
Figura 24 Diagrama de Atividades Especificação do Contexto do OA.....	54

Figura 25	Diagrama de Atividades Elicitação de Requisitos do OA.....	54
Figura 26	Estilo Arquitetural em três camadas	55
Figura 27	Principais Atores e Casos de Uso do Processo de Criação do OA	56
Figura 28	Exemplo de Caso de Uso (UC13).....	57
Figura 29	Exemplo de Caso de Uso (UC10).....	57
Figura 30	Exemplo de Caso de Uso (UC17).....	57
Figura 31	Papéis da Equipe de Desenvolvimento do OA.	60
Figura 32	Extrato do artefato de contextualização da metodologia INTERA preenchido para o OA a ser desenvolvido.....	60
Figura 33	Extrato do artefato de Especificação de Requisitos da metodologia INTERA preenchido para o OA, Aula Virtual.....	61
Figura 34	Esboço das Telas do OA, Aula virtual	61
Figura 35	Modelo Semântico com base no OA, Aula Virtual	63
Figura 36	Organograma da Equipe de Criação de OAs.....	64
Figura 37	Estilo Arquitetural em 3 Camadas definido no OA aula virtual.....	65

LISTA DE TABELAS

Tabela 1 Características relativas a objetos de aprendizagem	6
Tabela 2 Categoria de Padrões de Projeto - GoF.....	17
Tabela 3 Sequência de Preenchimento das Células do Zachman Framework	26
Tabela 4 Artefatos Propostos por Pereira e Sousa (2004) para o Zachman	26
Tabela 5 Modelos usados no Enterprise Architecture para o add-in com Zachaman Framework.....	27
Tabela 6 Padrão Metadados DCMI utilizados no BIOE (2014).....	37
Tabela 7 Padrão Metadados Cestacore utilizados no CESTA	37
Tabela 8 Associação entre os metadados do Padrão Metadados Essenciais para OA, LOM e DCMI	41
Tabela 9 Comparação entre metadados usados em ROAs.....	43
Tabela 10 Artefatos produzidos nas Células do Zachman Framework	44
Tabela 11 Requisitos não Funcionais e Funcionais	58
Tabela 12 Implementação da Arquitetura Proposta pelo OA Aula virtual	66

LISTA DE ACRÔNIMOS

ADDIE	<i>Analysis, Design, Development, Implementation e Evaluation</i>
ADL	<i>Advanced Distributed Learning</i>
ADL	<i>Architecture Description Language</i>
BIOE	Banco Internacional de Objetos de Aprendizagem
CESTA	Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem
DCMI	<i>Dublin Core Metadata Initiative</i>
GoF	<i>Gang of Four</i>
IMS	<i>Instructional Management System</i>
INTERA	Inteligência, Tecnologias Educacionais e Recursos Acessíveis
LMS	<i>Learning Management Systems</i>
LOM	<i>Learning Object Metadata</i>
LP	Linguagem de Padrão
MEC	Ministério da Educação e Cultura
Moodle	<i>Modular Object-Oriented Dynamic Learning Environment</i>
MVC	<i>Model View Controller</i>
OA	Objeto de Aprendizagem
PAC	<i>Presentation-Abstraction-Control</i>
POSA	<i>Pattern-Oriented Software Architecture</i>
RIVED	Rede Interativa Virtual de Educação
ROA	Repositório de Objetos de Aprendizagem
TIC	Tecnologia da Informação e Comunicação
UML	<i>Unified Modeling Language</i>

RESUMO

BORGES, Katia Cristina Aparecida Damaceno, M.Sc., Universidade Federal de Viçosa, agosto de 2015. **Uma Abordagem baseada em Arquitetura de Software para Construção de Objetos de Aprendizagem.** Orientador: José Luis Braga; Co-orientadora: Iris Fabiana de Barcelos Tronto.

A inclusão da tecnologia dentro do contexto das salas de aulas é vista como uma ferramenta de grande utilidade. Uma das formas dessa inclusão é por meio dos Objetos de Aprendizagem, ou seja, qualquer recurso digital que possa ser reutilizado para apoio a transmissão do conhecimento (WILEY, 2001). O sucesso desses OAs deve-se à facilidade com que as novas gerações interagem com dispositivos eletrônicos, tais gerações são conhecidas como os nativos digitais. Prensky (2001) afirma que estes são os nascidos após a década de 80 e foram criados na cultura de jogos de computadores e, por isso, eles têm facilidades ao manipular ou interagir com as diversas fontes de informação apoiadas por tecnologia. Entretanto, a grande maioria dos docentes atuantes é composta por pessoas que nasceram antes da década de 80 e, portanto, são ditos imigrantes digitais (Prensky, 2001). Neste cenário surgem dificuldades em inserir e, principalmente, desenvolver OAs para o contexto de suas aulas. Uma dificuldade se dá devido a grande diversidade de materiais que podem ser ditos OAs e não existe um padrão para que os mesmo possam ser construídos. Além disso, uma equipe de criação de um OA é multidisciplinar, composta por pessoas da área pedagógica e tecnológica, dificultando desta forma a comunicação entre os membros de um mesmo projeto. Assim, o objetivo desta dissertação é propor uma arquitetura empresarial para descrever o cenário desde o escopo do negócio, modelo de negócio e modelo de sistema. Esta descrição foi amparada pelo Zachman Framework, um precursor em arquiteturas empresariais. Outra notável contribuição deste trabalho foi a escrita de padrões de criação de OAs.

ABSTRACT

BORGES, Katia Cristina Aparecida Damaceno, M.Sc., Universidade Federal de Viçosa, August 2015. **An Approach based on Software Architecture for Making Learning Objects.** Advisor: José Luis Braga; Co-advisor: Iris Fabiana de Barcelos Tronto.

The inclusion of the technology inside classrooms is seen as a tool of great usefulness. One way this is by inclusion of learning objects - LOs, in other words, any digital resource that can be reused for support the transmission of the knowledge (WILEY 2001). The success of those LOs is due to the easiness with that the new generations interact with electronic devices, such generations are known as digital natives. Prensky (2001) affirms that these are born after 1980s and they were created in the culture of games of computers and, therefore, they have facilities to manipulate or interact with the several leaning sources of information supported by technology. However, the vast majority of active teachers is composed by people that were born before of 1980s and, therefore, they are said immigrant digital (PRENSKY 2001). In this scenery there are difficulties appear in inserting and, mainly, to develop LOs for the context of their classrooms. A difficulty feels due to great diversity of materials that LOs can be said and it does not exist a pattern so that the same can be built. Besides, a team of creation of LO is multidisciplinary, composed by people of the pedagogic and technological field, thus hindering communication between members of a same project. Thus, the aim of this dissertation is to propose an enterprise architecture to describe the scenery from the scope of business, business model and system model. This description was supported by Zachman Framework, a precursor in enterprise architectures. Another notable contribution of this work was the writing of creating learning objects patterns.

1 INTRODUÇÃO

1.1 O Problema

A expansão provocada pela tecnologia da informação transformou o indivíduo de passivo a um sujeito ativo, produtor do seu próprio conhecimento. Através do uso da Internet o conhecimento não se atém mais ao formalismo tradicional, mas sim possibilita o desenvolvimento interativo entre o sujeito e o novo conhecimento.

Em pesquisa realizada em domicílios brasileiros, entre setembro de 2013 e fevereiro de 2014, pela CETIC.br (2014) a Internet foi encontrada em 43% dos lares brasileiros, ou seja, aproximadamente 27,2 milhões de domicílios, sendo que: 55% utilizam a Internet para realizar atividades ou pesquisas escolares direcionadas por instituições convencionais e 31% usam a Internet para estudar por conta própria, adquirindo um novo conhecimento por si mesmos.

A fim de atender esta crescente demanda de usuários, com diferentes formas de interação, é necessário ter uma abordagem que permita criar componentes de qualidade para disseminação de informação e de conhecimento.

Tais componentes são frequentemente denominados Objetos de Aprendizagem (OA), caracterizados como qualquer recurso digital que possa ser reutilizado para disseminação de informação e conhecimento (Wiley, 2000).

Evidenciados estes aspectos, o presente trabalho tem por principal finalidade propor uma abordagem constituída de um conjunto de padrões e de uma arquitetura para a melhoria do processo de construção de OAs. Para isto, a pesquisa descreve uma arquitetura de software empresarial, focando nas visões do dono, do planejador e lógica do sistema e ainda padrões para criação destes OAs.

1.2 Justificativa

Sabe-se que a educação ocorre tanto no setor formal mantido pelo governo e

entidades privadas de ensino, como no terceiro setor (como por exemplo as ONGs). Há uma grande disponibilização de cursos voltados para treinamentos e certificação de profissionais em diversas áreas. É necessário que os novos conhecimentos sejam atualizados constantemente, em decorrência dos avanços científicos e tecnológicos proporcionados pela própria educação.

A principal característica da educação é a necessidade de disseminar os conceitos e habilidades. Estes são descritos em conteúdos ou materiais didáticos. No início, o conhecimento era transmitido apenas pela fala, posteriormente foi inserida também a escrita. No início do século XX o conhecimento passou a ser disseminado através de rádios, depois televisores e finalmente houve uma conversão para uma nova mídia: os computadores.

Entretanto, uma das grandes dificuldades do ensino está no desenvolvimento de atividades de aprendizagem e dos recursos educacionais necessários para sua execução. Uma dificuldade é a convergência de uma equipe multidisciplinar, formada por especialistas de áreas distintas a fim de complementar as competências de forma sinérgica e colaborativa, para elaboração de materiais didáticos (QUADROS; MARTINS, 2005).

O processo de desenvolvimento dos OAs suscita enormes desafios aos pesquisadores e desenvolvedores de software. Talvez, um dos maiores desafios seja saber se um software ou OA é eficaz e possui os requisitos básicos de qualidade, sob o ponto de vista de sua aplicação.

Em especial, aspectos técnicos da Engenharia de Software, como a escolha de uma arquitetura adequada que atenda a domínios específicos de conhecimento, ainda não foram tratados extensivamente.

Além disso, devem-se estabelecer critérios a ser incorporada a formação de uma estrutura, estilo empresarial e de desenvolvimento destes objetos. Assim, entre outras questões pertinentes ao domínio, destacam-se: Quais seriam os primeiros passos a serem traçados? Qual documentação deve ser gerada?

Após a análise de alguns OAs existentes na bibliografia pesquisada, verificou-se a carência de uma padronização de estrutura e de métodos de construção para os diversos tipos de OAs. Foi observado ainda a não existência na literatura de uma arquitetura de software que descrevesse tais objetos de maneira generalizada.

1.3 Hipótese

A pergunta principal a que esta pesquisa se propõe responder é:

- Qual seria uma arquitetura de software para guiar implementações de objetos de aprendizagem para uso específico em Engenharia de Software?

A partir da compreensão do problema proporcionado pela pergunta principal acima, foi enunciada uma hipótese: *a implementação de OAs com uso de princípios delineados por uma arquitetura de software específica para esta, impacta na sua disseminação e uso.*

1.4 Objetivos

O objetivo principal deste trabalho é adaptar uma arquitetura de software que possa apoiar a construção de objetos de aprendizagem voltados ao domínio de conhecimento da Engenharia de Software.

A princípio este trabalho propunha especificamente:

1. Identificar as principais arquiteturas utilizadas na criação de objetos de aprendizagem;
2. Estabelecer uma relação entre as arquiteturas utilizadas e sua aplicabilidade em OAs específicos para a área de Engenharia de Software;
3. Desenvolver um OA utilizando a arquitetura proposta;
4. Avaliar a contribuição da arquitetura escolhida para verificação da hipótese levantada.

No entanto, no decorrer das pesquisas e com a complexidade relativa ao desenvolvimento do tema houve a necessidade de alterar o terceiro objetivo. Assim, optou-se pela verificação da funcionalidade da arquitetura junto a um OA existente na bibliografia, o Aula Virtual (Braga et al, 2013).

1.5 Organização do Texto

Esta dissertação está dividida em seis Capítulos e se organiza da seguinte forma: No Capítulo 1 é apresentada uma introdução ao problema a ser tratado, sua justificativa, seguida de hipótese e método de pesquisa. No Capítulo 2 é apresentada

uma revisão da literatura que constitui na fundamentação teórica da pesquisa e esta dividida em quatro Seções.

A partir dos subconceitos elencados no Capítulo 2, definem-se no Capítulo 3 os Padrões para Criação de OAs. Após os elementos descritos nos Capítulos 2 e 3, é apresentada a proposta da arquitetura para modelagem conceitual e lógica de criação de OAs no Capítulo 4.

No Capítulo 5 é realizada uma análise da arquitetura proposta em relação a um OA encontrado na literatura. Os resultados desta dissertação são sumarizados no Capítulo 6 e aponta as principais contribuições e trabalhos futuros.

2 CONTEXTO

2.1 Objetos de Aprendizagem

Na elaboração do projeto instrucional de um curso, ocorre a escolha, o desenvolvimento e a utilização de OAs. De forma que este coincida com os objetivos da aprendizagem e características dos aprendizes. Nesse sentido é necessário que os OAs atendam a características técnicas e pedagógicas (FERLIM, 2010). Durante este processo é possível saber se o objeto será apenas adaptado, reutilizado ou mesmo construído sem nenhum reaproveitamento.

Diferentes autores têm proposto diversas definições para o termo OA. Porém como forma de padronizar o termo o IEEE-LTSC 1484.12.1 LOM (2006) define este por “qualquer entidade digital ou não, que pode ser usada, reusada ou referenciada durante o ensino auxiliado por tecnologia”. De acordo com esta definição os OAs não são necessariamente digitais. Por exemplo: um livro impresso, uma apostila de apoio a determinado experimento, a lousa e outros similares são OAs que não estão ligados as TICs (Tecnologias da Informação e Comunicação). No entanto, para os critérios adotados nesta pesquisa, apenas os OAs desenvolvidos a partir das TICs são considerados como OAs.

Tarouco (2014) afirma que a principal diferença entre mídias, no contexto instrucional, entre: projetor, fitas de vídeo e os OAs digitais é que estes podem ser facilmente acessados via Internet e, assim, disseminados simultaneamente a uma quantidade ilimitada de pessoas.

Os OAs podem ser de diferentes tipos, como: vídeos, animações, simuladores, jogos, textos, apresentações, entre outros. Para estes deve haver apenas uma instrução ou a combinação de instrução e prática, ou ainda uma lógica e uma estrutura para gerar e/ou combinar com outros OAs, permitir ou não a avaliação dos estudantes com este objeto (WILEY, 2000). Baseado nas qualidades relativas a estes, Wiley (2000) propôs que os OAs reusáveis sejam descritos dentro de uma taxonomia, de acordo com a complexidade de cada um.

Braga et al (2012) afirmam que uma grande dificuldade enfrentada no desenvolvimento de um OA é produzi-lo reusável e com qualidade. Neste sentido a principal característica de um OA, a reusabilidade, estaria associada à granularidade. Mendes (2004)

propôs algumas características e elementos que compõem um OA estruturalmente e operacionalmente, descritas na Tabela 1.

Tabela 1 Características relativas à OAs

Característica	Conceito
Reusabilidade	Capacidade de reutilização do objeto, diversas vezes, em diferentes contextos de aprendizagem
Adaptabilidade	Adaptável em qualquer ambiente de ensino.
Granularidade	Pedaços menores do conteúdo, para facilitar o reuso.
Acessibilidade	Possibilidade de acesso via Internet para distribuir em diferentes locais.
Durabilidade	Independência de tecnologia, possibilidade de continuar sendo utilizado mesmo ocorrendo mudanças.
Interoperabilidade	Habilidade de operar em vários hardware, sistemas operacionais e navegadores.
Metadados	(dados sobre dados) fornece informações sobre o objeto, tais como: título, autor, data, assunto, etc.

Normalmente, os OAs são desenvolvidos para serem utilizados dentro de um ambiente educacional, às vezes, um LMS (*Learning Management Systems*) ou armazenados em um ROA (Repositório de OAs). Um LMS é um ambiente educacional apoiado por tecnologia que auxilia os educadores abrigando seus cursos. Já um ROA é um ambiente onde os OAs podem ser armazenados ou mesmo os links para estes são mantidos para consulta, visualização ou recuperação.

No entanto, para que os OAs possam ser reutilizados em diversos LMS e ROAs faz-se necessário que os mesmos tenham sido armazenados sob algum padrão de metadados. Existem diferentes padrões de metadados conhecidos como por exemplo, IMS -*Instructional Management Systems*- (2007), ARIADNE -*Alliance of Remote Instructional Authoring and Distribution Networks for Europe* (2013), ADL -*Advanced Distributed Learning* (2009), IEEE-LTSC LOM - *Learning Technology Standards Committee Learning Object Metadata*- (LOM, 2006), entre outros.

2.1.1 Processo de construção de objetos de aprendizagem

Segundo Wiley (2000) é o professor/construtor quem deve definir a teoria instrutiva com a qual o OA deve ser construído, esta seria um modelo único para variações e contextos diferentes ou um modelo independente dos componentes, no qual o instrutor pode escolher os métodos e as estratégias para suas finalidades. Tarouco (2014, pag20) afirma que é o desenvolvedor que deve decidir qual o tipo de OA é o mais adequado para determinado conteúdo.

Wiley (2000) afirma que um OA deve possuir três componentes para que seu desenvolvimento seja bem sucedido: uma teoria de projeto instrucional, uma taxonomia de OA e ainda informações que conectem a teoria instrucional à taxonomia, com informações sobre a metacognição proporcionada pelo OA. Wiley (2000) propôs uma taxonomia e classificou os OAs da seguinte forma:

- Fundamental – é o tipo mais simples, ou seja, um recurso digital individual. Projetado para ser reaproveitado em um maior número de contextos. Por exemplo: imagem, texto, etc. ;
- Combinado-fechado – composto por um número pequeno de recursos digitais combinados. Os OAs de combinação fechada normalmente têm finalidade única, isto é, fornecem instrução ou prática. Possuem lógica limitada e não devem ser reutilizados em diferentes contextos, como ocorre no fundamental. Exemplos: um mapa com legenda escrita descrevendo partes, filme digital que é combinação de vídeo e áudio;
- Combinado-aberto – a principal característica destes são o maior número de recursos digitais combinados, normalmente contendo objetos fundamentais e objetos de combinação fechada. Estes objetos envolvem teoria e prática combinadas. Um bom exemplo deste tipo são as páginas Web, pois seus componentes imagens, vídeos, textos e outras mídias de formato reutilizável são combinados no momento do pedido e geraram uma unidade instrucional completa;
- Gerador-apresentação – possui lógica e estrutura geradora para gerar ou combinar OAs de nível mais baixo (fundamental e combinado fechado) para criar apresentações para uso em referência, prática e testes. Objetos geradores possuem alto índice de reusabilidade em contextos semelhantes ao que foi projetado e relativamente baixo de reusabilidade em contextos diferentes do que foram projetados;
- Gerador-instrução – lógica e estrutura para combinar OAs de outras categorias, ainda é

possível avaliar a interação do aluno com material didático produzido. Objeto gerador de instrução tem alto grau de reusabilidade tanto em contextos diferentes, quanto em semelhantes ao que foram projetados.

No intuito de tentar aliar as características técnicas e as pedagógicas, foram propostos alguns processos de desenvolvimento para OAs. Dentre estas, Braga (2015) descreve:

- ADDIE (*Analysis, Design, Development, Implementation e Evolution*) composto por 5 etapas resumidas no seu acrônimo. Sua principal característica esta na evidenciação dos Objetivos educacionais.
- RIVED (Rede Interativa Virtual de Educação) composto por 6 etapas ou fases: **I.** Modelo de Design Pedagógico; **II.** Avaliação e disponibilização da fase 1; **III.** Revisão dos pareceres emitidos na fase 2; **IV.** Início da produção do OA; **V.** Desenvolvimento do material auxiliar par utilização do OA; **VI.** Disponibilização do OA para o público.
- SOPHIA além de um processo é um repositório que agrupa OA e é composto por três fases: 1. Projeto, 2. Desenvolvimento e 3. distribuição.
- INTERA (Inteligência, Tecnologias Educacionais e Recursos Acessíveis) esta metodologia ressalta o maior número e características de qualidade contida em um OA e melhor reuso. Devido a aliar os aspectos técnicos aos pedagógicos este foi o processo escolhido na descrição da arquitetura proposta.

2.2 Arquitetura de Software

O aumento da complexidade e tamanho dos softwares levaram engenheiros de software a buscarem por princípios de projeto, tais como modularização e o ocultamento da informação, a fim de produzir sistemas de informação com maior qualidade a um baixo custo (SHAW, GARLAN, 1994). Devido a isso, surgiu a disciplina de arquitetura de software.

Arquitetura de software possui diversas definições, Bass et al (2003), David Garlan e Mary Shaw (1994), The Open Group (2011 e 2013), Lankhorst (2012), entre outros, cada um aborda um aspecto diferente. De acordo com Bass et al (2003), existe uma dificuldade em padronizar a definição de arquitetura de software, para que esta seja universalmente aceita.

Ainda com intuito de padronizar as definições, surge a ISO/IEC 42010:2011, que

define Arquitetura de Software, como: “[...] *a organização fundamental de um sistema, expressa nos seus componentes, nos relacionamentos entre eles, com o ambiente e nos princípios que governam sua evolução.*” Então a arquitetura de um sistema pode expressar: elementos do sistema, princípios e padrões de organização ou desenho; princípios que governam a evolução do sistema.

Todas estas definições apenas indicam a preocupação em modelar de maneira abstrata, os elementos significantes, com o foco do arquiteto de software. De maneira a simplificar a complexidade do processo de desenvolvimento de software e reduzir o custo total do projeto. Logo, a arquitetura de software está situada entre duas partes distintas, o problema do negócio e a solução técnica, isto possibilita o alcance dos atributos de qualidade de ambas.

Existem pelo menos doze escolas que definem Arquitetura de software como disciplina (MALVEAU & MOWBRAY, 2000). Dentre outras podem ser citadas: arquitetura orientada a serviços (SOA); arquitetura empresarial orientado a serviços (SoEA); arquitetura de integração; arquitetura de software (SA) e o destaque para o Zachman Framework, um precursor em descrições de *Enterprise Architecture*. Criado em 1987 por Zachman (Zachman, 1987), este faz um relacionamento em formato de matriz entre as perspectivas dos vários atores envolvidos no processo (*Planner, Owner, Designer, Builder e Subcontractor*) nas linhas e um conjunto de seis questões elementares (colunas) (*What, How, Where, Who, When e Why*). Para cada célula existe uma descrição dos componentes e artefatos produzidos. Esta abordagem é bem generalista, porém exaustiva, com 30 pontos de vista, o que garante a caracterização detalhada da Arquitetura Empresarial de uma dada organização. Devido a esta generalidade é possível representar qualquer objeto complexo. Este foi o fato que contribuiu para a escolha deste framework, no presente trabalho e uma descrição mais detalhada do mesmo é feita na Seção 2.3.

2.2.1 Técnicas em Arquitetura de Software

Diferentes razões levam a construção ou alteração de uma arquitetura de software, entre estas as mais comuns são: requisitos de um novo sistema, deficiência na arquitetura atual, evolução tecnológica, etc.. Seja qual for o motivo, definir uma nova arquitetura é uma tarefa árdua e demanda extrema experiência.

Contudo, para que uma arquitetura tenha êxito é necessária à análise de projetos que obtiveram sucesso. Além disso, utilizar experiências anteriores de arquitetos e semelhanças existentes entre arquiteturas pode beneficiar no quesito reuso e adoção de estratégias previamente validadas (BASS et al., 2003). Por isso as estratégias bem sucedidas podem se tornar padrões referenciais para o projeto focado na arquitetura.

Bass et al. (2003) propõem que a padronização seja realizada por meio dos níveis de abstração existentes e para isso define os conceitos de Padrão/Estilo Arquitetural, Modelo de Referência e Arquitetura de Referência. O autor também apresenta o relacionamento entre estes para a especificação de uma arquitetura (Figura 1).

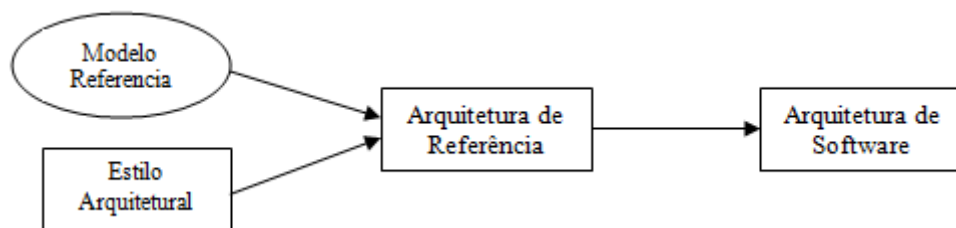


Figura 1 Relacionamento entre os padrões para a composição de uma arquitetura de software

Fonte: Bass et al (2003).

Ainda na tentativa de estabelecer referências para o projeto arquitetural mais detalhado, Buschmann (1996) apresenta a existência de padrões de projeto e idiomas. Os padrões de projeto buscam soluções para problemas de estruturas de sistemas orientadas a objeto e os idiomas são padrões de baixo nível, ou seja, soluções padronizadas para uma linguagem de programação em particular.

2.2.1.1 Padrão Arquitetural

Padrões foram utilizados primeiramente dentro do domínio da arquitetura para construção civil, por Alexander et al (1977), a fim de que pudessem ser representadas as soluções recorrentes encontradas dentro deste contexto. Então, um padrão pode ser definido como o processo de captura e apresentação do conhecimento de projeto na resolução de problemas de maneira estruturada, facilitando a comunicação dentro de um projeto, por especialistas e novatos (BUSCHMANN et al, 1996). Neste sentido os projetistas já familiarizados com certos padrões podem aplicá-los de maneira intuitiva em novos projetos sem a necessidade de redescobri-los (GAMMA et al., 1995).

Bass et al. (2003) abordam o conceito de padrões arquiteturais como sendo uma descrição dos elementos e tipos de relacionamentos, juntamente as restrições de como eles podem ser utilizados. Para estes autores, padrões arquiteturais podem ser vistos como restrições sobre a arquitetura, sobre seus elementos e seus padrões de interações. São essas restrições que definem o conjunto ou família de arquiteturas que a satisfazem.

Buschmann et al. (1996) propõem o POSA – *Pattern-Oriented Software Architecture* (Figura 2) como um sistema de padrões e afirmam que padrões arquiteturais fornecem esquemas de organização estrutural fundamentais para sistemas de software, pois apresentam as especificações de responsabilidades e incluem regras e diretrizes da organização do relacionamento entre eles. Neste sentido, os padrões arquiteturais do POSA se assemelham aos estilos arquiteturais descritos por Shaw and Garlan (1996).

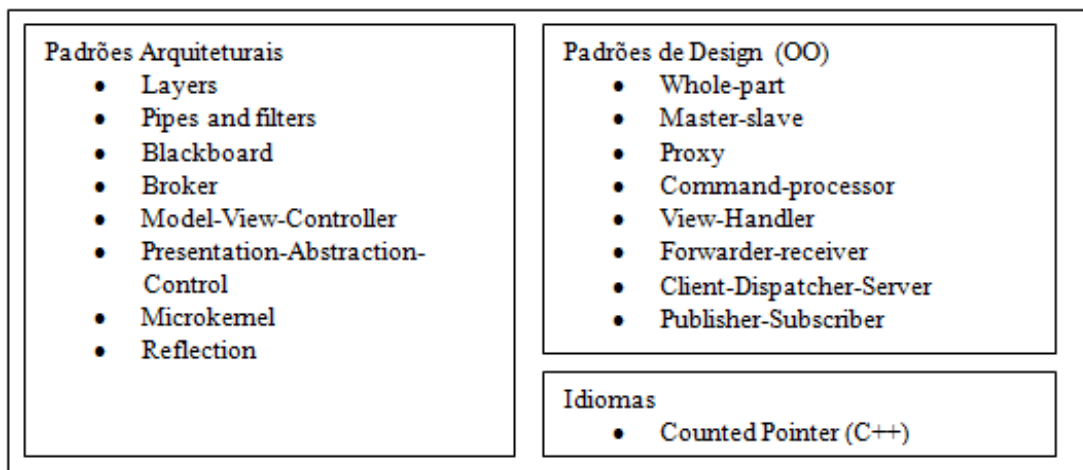


Figura 2 Sistema de Padrões POSA

Fonte: Buschmann et al (1996)

A existência de um padrão está, normalmente, condicionada a uma Linguagem de Padrão (LP), que possui um conjunto de informações instrutivas como (APPLETON et al, 1997):

- **Nome:** que é a própria identificação do padrão.
- **Contexto:** que traz a visão da situação onde o padrão deve ser empregado.
- **Problema:** é a expressão do problema que o padrão soluciona.
- **Solução:** apresenta a definição da resposta dada ao problema, pode ser descrita através de texto, diagramas e figuras. Ainda pode possuir as seguintes subseções:
 - **Estrutura:** informa a forma e organização estática do padrão.
 - **Participantes:** descreve cada um dos componentes.
 - **Dinâmica:** exhibe o comportamento dinâmico do padrão.
 - **Implementação:** mostra detalhes de implementação.

- **Variantes:** discutem possíveis variações e especializações da solução.
- **Forças:** descrevem os requisitos característicos ao problema nos quais a solução deve atender; as restrições aplicáveis a solução e as propriedades desejáveis.
- **Exemplos:** ilustram uma ou mais aplicações do padrão.
- **Consequências:** indicam a situação do sistema após a aplicação do padrão, seus benefícios e malefícios.
- **Racional:** explicação das regras ou passos do padrão que define como ele trata as influências contrárias, definidas como Forças, a fim de alcançar o seu objetivo.
- **Usos Conhecidos:** descrevem alguns lugares onde o padrão já é aplicado com sucesso.
- **Padrões Relacionados:** identificam, caso existam, outros padrões que são importantes para o padrão descrito.

Padrões arquiteturais possuem em suas especificações três itens que são comuns em suas propriedades:

1. Estabelecem um vocabulário comum ao projeto – tipos de componentes e conectores tais como, base de dados, filtros, clientes, servidores, etc.
2. Fornecem restrições de topologia e com isso as composições permitidas desses elementos.
3. Definem uma interpretação semântica baseada na composição dos elementos do projeto, juntamente as restrições impostas pelas regras de configuração.

Dentre os padrões arquiteturais ou estilos, os mais comuns são: (BUSCHMANN et al, 1996):

- ***Pipes & Filters*** (canais e filtros): este estilo é largamente empregado para scripts Unix e aplicações de processamento de sinal. Neste, os processos são encapsulados em *filters* e os dados são transmitidos pelos *pipes* localizados entre os *filters*, assim a saída de um componente é a entrada do outro, sucessivamente. Eventualmente, pode haver bifurcações (*forks*) e o processamento ocorrer em paralelo (Sommerville 2011).
- ***Layers Architecture*** (Arquitetura em camadas): o problema é decomposto em grupos de subtarefas, assim cada grupo apresenta um nível particular de abstração e necessita conhecer e interagir, apenas com a camada (n – 1), desta estrutura. Foi muito utilizado no modelo OSI da ISO para comunicação de redes de computadores (SOMMERVILLE 2011).

- **Blackboard** (Quadro negro ou Arquitetura de Repositório) uma coleção de problemas, que quando decompostos em subproblemas abrangem muitas áreas/campos de conhecimento. Assim, é necessário que subsistemas especialistas, independentes, resolvam determinada parte da tarefa completa, a fim de que todos trabalhem juntos para resolver o problema central. O trabalho ocorre através da memória compartilhada, porém não existe comunicação entre os subsistemas (Sommerville 2011).
- **Model-View-Controller** (Modelo-Visão-Controlador - MVC) é amplamente utilizado em aplicações Web (FOWLER, 2011), pois promove a organização do sistema em três componentes: *Modelo*, *Visão* e *Controlador*. Este padrão remove o acoplamento entre a Visão e o Modelo por estabelecer um protocolo de subscrição-notificação entre os dois. Sempre que há uma alteração no Modelo, este notifica quais Visões devem sofrer alteração, ou seja, as que dependem dele. O Controlador define a forma de apresentação da interface do usuário, baseada nas interações com este (Sommerville 2011).
- **Presentation-Abstraction-Control** (PAC): define uma estrutura hierárquica para componentes cooperativos. Cada aspecto específico da funcionalidade da aplicação é de responsabilidade de um agente, e é composto por três componentes: apresentação, abstração e controle. Desta forma, separa os aspectos de interação homem-máquina dos agentes de seu núcleo funcional e sua comunicação com outros agentes.
- **Broker**: é utilizado para estruturar sistemas distribuídos, pois fornece a separação entre componentes que interagem através de chamadas remotas de serviços. Assim, este padrão promove o desacoplamento da implementação dos objetos que interagem por invocações de serviços remotos. O *broker* é responsável por coordenar a comunicação, encaminhando as solicitações e transmitindo os resultados ou exceções.
- **Microkernel**: se aplica a sistemas de software que devem estar aptos a adaptação a mudanças nos requisitos do sistema. Promove a separação do núcleo funcional mínimo das funcionalidades estendidas e das partes específicas do cliente e também serve como soquete ao qual estas extensões podem ser conectadas e coordenadas.
- **Reflection**: permite que estruturas e comportamentos de um sistema sejam

mudados dinamicamente. Suporta a modificação de aspectos funcionais, tais como estruturas de tipos e mecanismos de chamada de função. Este padrão divide a aplicação em duas partes: um metanível que possui informações sobre as propriedades selecionadas do sistema que torna o software consciente disso e um nível-base que inclui a lógica da aplicação. A implementação é feita sobre o metanível. Uma interface é especificada para manipulação do metanível e protocolo dos metaobjetos.

Mesmo que o padrão defina uma solução para um problema particular, estes problemas estão envolvidos em diferentes domínios. Desta forma, um sistema não pode ser estruturado de acordo com um único padrão particular. A variedade de requisitos faz com que os sistemas utilizem vários padrões a fim de atingir um metassistema (BUSCHAMANN, 1996).

A escolha de um padrão depende do tipo de sistema e seus requisitos não funcionais. Desta forma, Buschmann et al (1996) classificou os padrões de acordo com o auxílio ao qual estes poderiam trazer ao tipo de sistema:

- ***From Mud to Structure*** (da desordem à estrutura): padrões que permitem a decomposição adequada de um sistema de tarefas em subtarefas que cooperam entre si. Ex.: *Pipes & Filters, Layers, Blackboard*.
- **Sistemas Distribuídos**: padrões que fornecem infraestrutura adequada a sistemas que possuem componentes localizados em diferentes processadores ou em vários subsistemas e componentes. Ex.: *Broker, Microkernel e Pipes & Filters*.
- **Sistemas interativos**: padrões que permitem estruturar sistemas com interface homem-máquina. Ex.: *Model-View-Controller, Presentation-Abstraction-Control*.
- **Sistemas Adaptáveis**: padrões que oferecem infraestruturas para apoio a decomposição adequada de subsistemas e componentes complexos em partes cooperativas. Ex.: *Microkernel, Reflection*.
- **Decomposição Estrutural**: padrões que definem a decomposição adequada de subsistemas e componentes complexos para fornecer um serviço complexo. Ex.: *Whole-Part*.
- **Organização do Trabalho**: padrões que fornecem uma definição de como os componentes colaboram entre si para fornecer um serviço complexo. Ex.: *Master-Slave*.

- **Controle de Acesso:** padrões que mantêm e controlam o acesso a serviços ou componentes. Ex.: *Proxy*.
- **Gerenciamento:** padrões para tratar de coleções homogêneas de objetos, serviços e componentes em sua plenitude. Ex.: *Command Processor, View Handler*.
- **Comunicação:** padrões que auxiliam a organização da comunicação entre componentes. Ex.: *Forwarder-Receiver, Client-Dispatcher-Server, Publisher-Subscriber*.
- **Tratamento de Recursos:** padrões que ajudam a gerenciar componentes e objetos compartilhados.

2.2.1.2 Modelo de Referência

Um modelo de referência é caracterizado pela abstração da realidade, este pode ser expresso pelo formalismo de modelagem, conforme os objetivos de um usuário (VERNADAT, 1996; LANKHORST, 2012).

“Um modelo de referência é um framework abstrato para entendimento dos relacionamentos significantes entre as entidades de algum ambiente. Ele habilita o desenvolvimento de arquiteturas específicas usando padrões consistentes ou especificações suportando aquele ambiente. Um modelo de referência consiste de um conjunto mínimo de conceitos unificados, axiomas e relacionamentos com um domínio de um problema particular, e é independente de padrões específicos, tecnologias, implementações, ou outro detalhe concreto” (MACKENZIE et al., 2006).

Segundo Bass et al. (2003), um modelo de referência é uma divisão de funcionalidades juntamente ao fluxo de dados entre as partes. Assim, ele se caracteriza pela decomposição padrão de um problema conhecido em partes que cooperam entre si. Os modelos de referência são oriundos de domínios maduros e descrevem em termo geral como as partes se inter-relacionam para alcançar o propósito coletivo.

Lankhorst (2012) afirmam que dentro de um domínio específico, o modelo de referência fornece um vocabulário e um entendimento comum sobre os elementos deste com objetivo de comunicar algo aos leitores do modelo, ou entre pessoas que participam da

construção do modelo.

O modelo de referência de um domínio é o resultado do processo de amadurecimento da solução em relação à necessidade de representações mais abstratas que caracterizam o domínio.

Ao utilizar um modelo de referência não há uma representação formal para sua composição. Assim, blocos e linhas podem ser utilizados, seguidos por uma descrição textual sobre a funcionalidade de cada elemento apresentado. O OASIS (Mackenzie et al. 2006) utiliza os mapas conceituais para representar o modelo.

2.2.1.3 Arquitetura de Referência

Segundo Taylor, Medvidovic & Dashofy (2009) *“uma arquitetura de referência é um conjunto de decisões de design principal que são simultaneamente aplicáveis a múltiplos sistemas correlatos, tipicamente dentro do domínio de uma aplicação, com pontos de variação definidos explicitamente”*.

A arquitetura de referência consiste em componentes de software e nos relacionamentos entre eles que implementam funcionalidades definidas no modelo de referência e o fluxo de dados entre eles, num dado domínio de aplicação. Essa é aplicada a um domínio particular.

A generalização de um ou vários sistemas finais de um domínio comum é a arquitetura de referência. Assim, uma arquitetura de referência define uma infraestrutura comum e as interfaces de componentes que serão incluídas no sistema final, isto facilita a derivação e extensão de novas arquiteturas de software para classes de sistemas comuns.

Para Bass et al (2003) a principal finalidade da especificação de uma arquitetura de referência é criar um ambiente de desenvolvimento utilizado para construir aplicações de forma a proporcionar agilidade e melhor desempenho, qualidade e reusabilidade.

Há possibilidade de ter diversas arquiteturas de referência baseadas em um mesmo modelo de referência. Isto porque podem existir diversas maneiras de transformar um modelo em uma arquitetura, além de que alguns modelos são tão genéricos que podem ser aplicados a domínios de problemas diferentes. Da mesma forma, pode haver diversas arquiteturas de software, uma para cada arquitetura de referência.

2.2.1.4 Padrões de Projeto

Padrões de projeto (*design patterns*) definem soluções para problemas de desenvolvimento de software, identificam o particionamento de classes e instâncias, bem como, suas regras, colaborações e a distribuição das responsabilidades. Um padrão de projeto tem foco num problema particular do projeto orientado a objeto. Cada padrão descreve onde ele deve ser aplicado, no entanto, este considera outras restrições de projeto, consequências e custos de seu uso (GAMMA et al., 1995). Também é possível encontrar dicas sobre a implementação e exemplos.

Gamma et al. (1995) organizaram um catálogo de padrões de projeto o qual possui uma maior organização e estrutura exibindo as relações entre eles. Estes são conhecidos na comunidade como GoF (Gang of Four), isto porque o livro foi escrito por quatro autores e passou a ser uma referência em padrões de projeto. Os GoFs são genéricos e podem ser aplicados em qualquer projeto orientado a objeto. Os padrões GoF são classificados em dois critérios: Escopo e Propósito. O escopo define se o padrão atua sobre uma classe ou objetos da classe. Já o propósito reflete o que o padrão faz. A Tabela 2 exhibe os padrões GoF.

Tabela 2 Categoria de Padrões de Projeto - GoF

Fonte: Gamma et al (1995)

		Propósito		
		De criação	Estrutural	Comportamental
Escopo	Classe	Factory Method	Adapter	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Padrões de Criação:

Esses padrões abstraem e/ou adiam o processo de criação dos objetos. Ajudam a tornar um sistema independente de como seus objetos são criados, compostos e representados.

- Abstract factory: permite criar famílias de objetos relacionados ou dependentes a partir de uma interface comum, sem especificar suas classes concretas.
- Builder: permite separar a construção de um objeto complexo de sua representação. Então, o mesmo processo de construção pode criar diferentes representações.
- Factory method: permite que as classes deleguem às subclasses a instanciação.
- Prototype: permite a criação de objetos usando uma instancia de prototipação e assim cria cópias a partir de um modelo original.
- Singleton: garante que determinada classe terá apenas uma instância e fornece um ponto global de acesso para essa classe.

Padrões Estruturais:

Estes padrões têm foco na forma como classes e objetos são compostos para formar estruturas maiores. Os de classes fazem uso de herança para compor interfaces ou implementações, já os de objeto descrevem maneiras de compor objetos para obter novas funcionalidades.

- Adapter: permite que classes com interfaces incompatíveis possam interagir, pois converte a interface de uma classe em outra interface.
- Bridge: remove o acoplamento de uma abstração (interface) de sua implementação para que ambos possam variar independentes.
- Composite: permite que os componentes tratem objetos individuais e composição de objetos.
- Decorator: vincula responsabilidades adicionais a um objeto em tempo de execução.
- Facade: disponibiliza uma interface simplificada e unificada para um conjunto de interfaces em um subsistema.
- Flyweight: usa compartilhamento de pequenos recursos para economizar espaço, assim suporta um grande número de objetos de pequena granularidade de maneira eficiente.
- Proxy: redireciona o acesso aos objetos e toma o lugar de outro objeto controlando o acesso a ele.

Padrões de Comportamento:

Estes padrões preocupam com os algoritmos e atribuições de responsabilidades entre os objetos. Além de descrever padrões de objetos e classes, também descrevem padrões de comunicação entre os objetos.

- Chain of Responsibility: evita acoplar as solicitações do emissor ao receptor por permitir que mais de um objeto trate a solicitação. Restringe o objeto receptor e repassa requisições, junto a restrição para o objeto que a manipula.
- Command: faz o encapsulamento da requisição e a transforma em objetos, permite assim parametrizar clientes com diferentes requisições, filas ou log de requisições e suporta operações que não podem ser desfeitas.
- Interpreter: a partir de uma linguagem de programação define uma representação para sua gramática juntamente a um interpretador que a use para interpretar as sentenças na linguagem.
- Iterator: fornece a maneira de acessar os elementos de um objeto agregado sequencialmente e sem expor sua representação.
- Mediator: define um objeto que encapsula o comportamento coletivo em um objeto mediador separado dos demais objetos. O mediador funciona como intermediário entre os objetos do grupo e evita que se referenciem uns aos outros explicitamente.
- Memento: sem violar o encapsulamento, permite armazenar o estado interno de um objeto, para que quando e se necessário, este possa ser restaurado.
- Observer: define uma dependência um para muitos entre objetos de modo que suas dependências sejam notificadas e atualizadas automaticamente quando um objeto troca de estado.
- State: permite a um objeto fazer alteração de seu comportamento quando o seu estado interno muda.
- Strategy: define uma família de algoritmos, os encapsula e os tornam intercambiáveis. Este padrão permite a variação do algoritmo, independente do cliente que o utiliza.
- Template Method: define um esqueleto de um algoritmo em operação, posterga alguns passos para as subclasses sem mudar a estrutura do algoritmo.
- Visitor: representa uma operação a ser executada nos elementos de uma estrutura de objetos, isto sem que, a interface de programação dessas classes seja afetada.

2.2.1.5 Idiomas

Idiomas são padrões de baixo nível específicos a uma linguagem de programação. Assim, buscam descrever como solucionar um problema específico de implementação em determinada linguagem de programação, tal como o gerenciamento de memória em C++. Os idiomas também podem visar à implementação concreta de um padrão de projeto específico, por exemplo, como implementar o padrão *Adapter* em Java. Não há uma separação clara entre padrões de projeto e idiomas (BUSCHMANN et al, 1996).

Idiomas podem fornecer soluções para o uso efetivo das características de uma linguagem de programação. Com eles, estas características são catalogadas e padronizadas de maneira a permitir um estilo de programação consistente e comum.

Para definir um idioma é feita a tríade descrita a seguir:

- **Nome:** uma identificação única para o idioma. Este deve ser expressivo a fim de garantir sua rápida localização.
- **Problema:** descreve o problema que será resolvido pelo idioma. Deve permitir a contextualização completa do problema.
- **Solução:** informa exatamente os passos a serem seguidos para alcançar a solução do problema. Pode vir acompanhada de exemplo de uso que demonstram claramente como a solução pode ser alcançada.

Um idioma pode abranger variadas áreas e facilitar a comunicação entre programadores, assim aceleraram o desenvolvimento e facilitam as atividades de manutenção. Desta forma, existem idiomas para resolver a forma como os loops são usados, o formato de nomes, formatação de código fonte, gerenciamento de memória, e assim por diante. A qualidade e facilidade de acesso de um idioma influencia diretamente a eficiência do mesmo.

2.2.1.5 Linguagens de Descrição Arquitetural de Software

Normalmente, arquiteturas de software não possuem uma representação formal e estas são representadas através de caixas e linhas, em que a natureza dos componentes,

propriedades, semântica das conexões e comportamento do sistema são pobremente definidos (Bass et al, 2003).

Uma Linguagem de descrição de arquitetura (*Architecture Description Language – ADL*) é um meio formal ou semiformal de especificação de arquiteturas de um sistema de software, de modo a permitir a definição dos componentes, como se comportam padrões e mecanismos para interações entre os componentes (SHAW and GARLAN, 1996). Ainda deve possuir, quando possível, uma notação gráfica amigável.

Ao longo do tempo, diversas ADLs têm sido propostas, embora nenhuma tenha se tornado um padrão de fato, todas têm objetivo de expressar de maneira suficiente a arquitetura (LANKHORST, 2012). Alguns exemplos de ADLs encontradas na literatura são: ACME (Garlan, 1997), Aesop (Garlan, and Ockerbloom., 1994), C2 (Medvidovic, 1996), Darwin (Magee, 1995), DSSA ADAGE (Coglianese e Szymanski, 1993) EAADL (Oh et al, 07), Rapide (Luckham, 1996), SADL (Moriconi and. Qian and Riemenschneider 1997), UniCon (Shaw et al, 1995), Wright (Allen and Garlan, 1997).

As ADLs fornecem uma direção para representar Arquiteturas de Software. No entanto, se o grau de complexidade da arquitetura de software não for tão elevado, o uso de uma ADL pode ser desnecessário. Porém ADLs semiformais, mais gráficas podem ser úteis, pois fornecem elementos gráficos padrões para representação da arquitetura.

A UML (Unified Modeling Language) 2.0 tem sido usada no mapeamento de conceitos orientados a objetos de arquiteturas de software. Devido a familiaridade desta junto aos desenvolvedores esta é uma alternativa para mapear requisitos arquiteturais de software (PANDEY, 2010).

2.3 ZACHMAN FRAMEWORK

O The Open Group (2011), define framework de arquitetura (Architectural Framework) como uma estrutura conceitual ou conjunto de estruturas, para desenvolvimento, implementação e manutenção de arquiteturas. Esses frameworks facilitam o desenvolvimento incremental de arquiteturas e descrevem convenções e práticas de arquiteturas em comunidades e domínios de aplicação (ISO, 2011). Os frameworks de arquitetura são encontrados em diversos contextos, como em processos que facilitam a comunicação e interoperação entre projetos e/ou organizações nas descrições de arquiteturas; no

desenvolvimento de ferramentas e métodos para modelagem de arquiteturas (ISO, 2011).

Contudo, com a evolução e crescente complexidade dos sistemas de informação surge a necessidade de documentar as Enterprise Architecture – EA (arquiteturas empresariais), estas promovem um alinhamento entre o negócio e a tecnologia. Assim, Zachman (1987) propôs um framework para dar suporte ao desenvolvimento destas. Neste são considerados: pessoas, processos e tecnologias que devem incorporar uma arquitetura empresarial, representadas por perspectivas através das quais se podem observar uma empresa, ou seja, visões organizacionais.

Moshiri e Hill (2011) destacam que o Zachman framework é um sistema para organizar artefatos de arquitetura e que o termo que melhor o define é taxonomia. Para Zachman (2008) seu framework não é nem metodologia e nem processo, mas sim estabelece definições e não as transformações.

O Zachman Framework (Figura 3) apresenta seis diferentes perspectivas, representadas pelas linhas, em qualquer projeto de desenvolvimento de sistemas. Cada uma dessas perspectivas oferece recursos e restrições na arquitetura do SI, a união destas fornece uma descrição arquitetural completa. Estas perspectivas são (ROCHA, 2010):

- **Escopo:** (Contextual) Perspectiva planejador – fornece a descrição geral da organização existente, o posicionamento externo e interno e o que a diferencia das demais organizações do mesmo setor. Permite a identificação da viabilidade financeira, restrições e escopo.
- **Modelo de negócio:** (conceitual) Perspectiva do dono do negócio - refere-se a perspectiva dos envolvidos no negócio e não dos acionistas. Nesta visão todo o negócio é envolvido, tanto no que se refere ao entendimento, quanto a como efetivamente funciona.
- **Modelo do Sistema** (lógico) Perspectiva do analista de negócio/projetista – descreve meios para se conciliar os interesses descritos no modelo de negócio e no modelo tecnológico, a fim de produzir representações funcionais e especificações técnicas. É independente de qualquer tecnologia a ser adotada.
- **Modelo tecnológico** (físico): Perspectiva do analista do sistema – trata do endereçamento das restrições descritas no Modelo do Sistema para a tecnologia a ser aplicada (bases de dados, linguagens de programação, sistemas operativos), Ferramentas Case, compiladores e a organização do trabalho para cumprir os prazos.

- **Componentes:** Perspectiva do programador/subcontratado - nesta perspectiva é traduzido o modelo tecnológico para linguagem de programação. É definido quais partes e como serão construídas. O subcontratado deve desenvolver exatamente o que foi descrito.
- **Sistema Funcional:** Perspectiva do operador – visão do sistema terminado, representado pela interface e funcionalidade do produto final.

Para garantir que todos os envolvidos na solução de um problema tenham um ponto de convergência entre as diferentes perspectivas que são as dimensões. O Zachman Framework fornece seis pontos de convergência, são eles (ROCHA, 2010):

- *What* (O que é feito?) – Dados – descrevem a composição do material, em sistemas de software são os dados. Contudo, estes dados precisam estar relacionados com outros dados para fazer sentido no contexto.
- *How* (Como isso funciona?) – Função – descreve todas as funções ou transformações do produto. Na maioria das vezes esta coluna descreve o modelo entrada-processo-saída.
- *Where* (Onde os elementos estão localizados?) – Rede – descreve a geometria ou conectividade do produto.
- *Who* (Quem faz o trabalho?) – Pessoas – descrevem as pessoas envolvidas, manuais, instruções de operação ou modelos utilizados para realização de tarefas. Esta coluna trata da alocação do trabalho e da estrutura de autoridade e responsabilidade da organização.
- *When* (Quando as coisas acontecem?) – Tempo – descreve o ciclo de vida, calendário e o cronograma para controlar as atividades.
- *Why* (Por quê?) – Motivação – descreve as metas, planos e regras que prescrevem políticas e fins que norteiam a organização, ou seja a motivação da organização.

Todos os itens descritos nas 36 células do Zachman Framework referem-se a uma arquitetura, modelo, representação ou descrição do que a organização tem que documentar. As células são autônomas entre si, o que possibilita que sejam descritas e modeladas de maneira independente. Entretanto, a configuração das células em determinada linha garante que uma visão/perspectiva seja garantida. Bem como as células de uma coluna relacionam entre si, garantindo que o objetivo esteja sobre o mesmo tipo de elementos.

The Zachman Framework for Enterprise Architecture™

The Enterprise Ontology™



Figura 3 Zachman Framework

Fonte: The Zachman Framework™: The Official Concise Definition (ZACHMAN, 2009)

Zachman não propôs nenhum padrão de modelagem do seu framework e com isto, ao longo do tempo, diversos autores propuseram formas para que o mesmo fosse modelado, como, por exemplo, Gerber; Van Der Merwe; Bayes (2013) propõem que a UML seja considerada para dar suporte a modelagem das células do Zachman Framework.

Diversos ajustes têm sido realizados no Zachman Framework, particularmente no nível de grafismos e uso de uma linguagem mais precisa. Porém os conceitos essenciais permaneceram inalterados (ZACHMAN, 2008).

Para o desenvolvimento deste trabalho o framework de Zachman será adotado, principalmente por ainda ser um framework atualizado e sua capacidade de descrever diversas perspectivas em detalhes. A versão utilizada é a de 2003, a fim de garantir consistência entre os métodos e as ferramentas utilizadas.

2.3.1 Métodos para Construção da Arquitetura utilizando Zachman Framework

Devido ao número potencial de artefatos envolvidos numa arquitetura descrita com o Zachman Framework, alguns desafios podem ser encontrados. Neste sentido este tópico tem sido alvo de diversos estudos e propostas, para que fosse garantido o alinhamento entre os dados, aplicações e tecnologia através da construção de uma série de matrizes de relacionamento.

Pereira e Sousa (2004) sugerem um método que cobre as três primeiras perspectivas do Zachman Framework. Os autores propõem a definição dos diversos artefatos de cada célula e a sequência de seu preenchimento. A Tabela 3 descreve a sequência de preenchimento das células. O conteúdo de cada célula por identificação (letra), sequência de preenchimento (número da fase) e precedências (identificação de células que devem ser anteriormente preenchidas). Assim, na linha “Escopo” não há pré-requisito para seu preenchimento e esta corresponde a fase 1, portanto, não existe sequência para seu preenchimento, o que não ocorre nas demais células.

Tabela 3 Sequência de Preenchimento das Células do Zachman Framework

	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>When</i>	<i>Why</i>
Scope (Planner)	A,1	B,1	C,1	D,1	E,1	F,1
Enterprise Model (Owner)	G,2,A	H,3, (B + G)	I,4,(C+H)	J,5, (D+I)	K,4, (E+H)	L, 4, (F+H)
System Model (Designer)	M, 3, G	N, 4, H	O, 5, N	P, 6, (J+N)	Q, 5, N	R, 5, (L+N)
	Data	Function	Network	People	time	motivation

Com relação aos artefatos propostos para preenchimento das células da Tabela 3, Pereira e Sousa (2004) sugerem que estes sejam preenchidos de acordo com a Tabela 4.

Tabela 4 Artefatos Propostos por Pereira e Sousa (2004) para o Zachman

	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>When</i>	<i>Why</i>
Scope (Planner)	List of things important for the business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations important to the Business	List of Events Significant to the Business	List of Business Goals/Strategies
	-List	-Hierarchical list or tree	-Hierarchical list or tree	-List	-List	-Indented list
Enterprise Model (Owner)	Semantic Model	The Business Process Model	The Business Logistics System	Work Flow Model	Master Schedule	Business Plan
	-Entities Diagram -Entities Dictionary	-Entities vs. Process Matrix -Processes Dictionary -Activity Diagram	-Functional Decomposition	-Organization Chart -Process vs. Organization Matrix	-Business Execution Plan	-Business Table
System Model (Designer)	Logical Data Model	Application Architecture	The Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Roles
	-Class Diagram	-System vs. Processes Matrix -System vs. Entities Matrix -System Dictionary	-Systems Diagram	-Systems vs. Roles Matrix	-State Diagram	-Systems vs. Business Roles Matrix.
	Data	Function	Network	People	time	motivation

Embora a principal crítica quanto a adoção do Zachman Framework seja o excesso de documentação e demora no processo de construção, a adoção de ferramentas que dão suporte a modelagem de software contribui para que esta desvantagem seja sanada. Estas ferramentas permitem: a validação dos artefatos usados; edição cooperativa sincronizada; a publicação,

fato que ajuda com a evolução do negócio e adaptações externas; suportam a geração de código a partir dos modelos; torna-se a base de conhecimento sobre a organização.

Dentre as ferramentas que dão suporte a modelização de arquiteturas, destacam-se as seguintes:

- ARIS (Ids Scheer, 2014);
- Enterprise Architecture (Sparx, 2014);
- EVA -Enterprise Value Architect- Netmodeler (Promis, 2014);
- Casewise (Caseware, 2014);
- IBM Rational System Architect (IBM, 2014).

Considerando o custo e funcionalidade da utilização, neste trabalho optou-se pela utilização da ferramenta Enterprise Architect versão 10.1, versão trial. Os Artefatos por ela produzidos podem ser visualizados na Tabela 5.

Tabela 5 Modelos usados no Enterprise Architecture para o add-in com Zachaman Framework

Fonte: (Sparks, 2008)

	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>When</i>	<i>Why</i>
SCOPE (Planner)	List of things important for the business	List of Processes the Business Performs	List of Locations in which the Business Operates	List of Organizations important to the Business	List of Events Significant to the Business	List of Business Goals/Strategies
	-Business Data (2)	-High level business Process (2)	-Business Location (2)	-Organization Chart (2)	-Business Events (2)	-Business Motivation (2)
ENTERPRISE MODEL (Owner)	Semantic Model	The Business Process Model	The Business Logistics System	Work Flow Model	Master Schedule	Business Plan
	-Data Map(2) -Add-in Generated Process Map (2)	-Process Analysis (2)	-Business Logistics (2)	-BPMN (2)	-Event Schedule (2)	-Strategy Map (2) -Mind Mapping (3)
SYSTEM MODEL (Designer)	Logical Data Model	Application Architecture	The Distributed System Architecture	Human Interface Architecture	Processing Structure	Business Roles
	-Class (PIM) (1)	-Activity (1)	-Data Distribution Architecture (2)	-Use Case (1)	-State Transition (1)	-Business Rule Model (2) -Requirements (3).
TECHNOLOGY MODEL (Builder)	Physical Data Model	System Design	Technology/Architecture	Presentation Architecture	Control Structure	Rule Designs
	Physical Data Model (2)	Class (PSM) (1) Component (1)	Deployment (1)	User Interface (3)	Interaction (1) Communication (1)	Rule design (2)
DETAILED REPRESENTATIONS (Subcontractor)	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Specification
	Data definition (2) (Enterprise Architect DDL Generation)	(Enterprise Architect Code Generation)	Network Architecture (2)	Security Architecture (2)	Timing (1)	Rule Specification (2)
	Data	Function	Network	People	time	motivation

Diagramas UML (1); UML *Profile* para Zachaman Framework (2); Extensões Enterprise Architect (3)

2.4 TRABALHOS CORRELATOS

Esta Seção descreve os trabalhos relacionados ao desenvolvimento de OAs, especificamente os relacionados a arquitetura de software e a padrões para desenvolvimento dos OAs.

A pesquisa realizada por Talarico, et al (2006) aponta a catalogação de práticas relacionadas aos aspectos cognitivos dos OAs elaborados, neste foi apresentada a linguagem de padrões Cog Learn. A principal contribuição deste trabalho encontra-se no relacionamento de um conjunto de padrões pedagógicos que abordam o planejamento e a sequência de cursos baseados em práticas presenciais e padrões IHC. O foco dado é a estruturação do conhecimento e contextualização, como a organização dos conteúdos para facilitar a apresentação de novos conceitos aos alunos.

Em Mohan (2005) o foco principal é a definição de padrões para o requisito de avaliação do conteúdo pelo aluno. Neste são verificados os aspectos relacionados a forma como o conteúdo apresentado foi assimilado pelo aluno e como o conhecimento foi realmente absorvido.

A pesquisa realizada por Zimmermann (2006) enfatiza a utilização de mecanismos que facilitem o registro de utilização do recurso digital. No entanto, não visa a facilitação da criação dos objetos de aprendizagem.

Castillo e Ayala (2008) propõem uma arquitetura para OAs usados em dispositivos móveis, *m-learning*. Os autores trabalham com a adaptação de OAs para dispositivos móveis, com base em banco de dados de OAs.

Uma arquitetura de referência para ambientes de aprendizagem, *e-learning*, foi descrita por Fioravanti, Nakagawa e Barbosa, (2010). A finalidade dessa arquitetura era contribuir para evolução de ambientes educacionais, bem como para o desenvolvimento desses, respeitando os critérios de qualidade e reuso dos mesmos.

Embora existam inúmeras pesquisas relacionadas a OAs, durante a revisão bibliográfica, não foram localizados trabalhos que descrevessem uma arquitetura de software para um conjunto geral de OAs (vantagem de generalidade e reusabilidade). Então, diante da heterogeneidade oriunda dos vários tipos de OAs e suas possíveis implementações, uma solução encontrada foi a criação de padrões de desenvolvimento de OAs. Tais padrões auxiliam na descrição da camada lógica da arquitetura proposta.

3 PADRÕES DESENVOLVIDOS PARA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM

Este capítulo descreve os padrões utilizados no nível lógico do Zachmann Framework para a descrição da arquitetura de criação de OAs. Os mesmos foram publicados na X Conferência Sugar Loaf Plop de 2014 (Borges et al, 2014). Assim, para estabelecer uma base de comparação entre cenários de aplicação e abrangência, os padrões abordados nas Seções subsequentes são descritos de acordo com o seguinte formato proposto por Alexander et al (1977):

- **Nome:** a própria identificação do padrão;
- **Contexto:** visão da situação onde o padrão deve ser empregado.
- **Problema:** é a expressão do problema que o padrão soluciona.
- **Forças:** descrevem as forças que direcionam o padrão para suas possíveis soluções.
- **Solução:** definição da resposta dada ao problema
- **Conseqüências:** indicam a situação do sistema após a aplicação do padrão.
- **Padrões Relacionados:** identificam, caso existam, outros padrões que são importantes para o padrão descrito.
- **Usos Conhecidos:** descrevem alguns lugares onde o padrão já é aplicado com sucesso.

3.1 Padrão Objeto de Aprendizagem Abstrato

Contexto

Os ambientes virtuais de aprendizagem geram a necessidade de utilização de novas tecnologias no cotidiano de professores e alunos, na adoção de OAs. Assim, vários fatores devem ser considerados num OA, como sua capacidade de reuso, flexibilidade, facilidade de atualização, customização, interoperabilidade e o aumento do conhecimento.

Desta forma, os OAs devem ser construídos e utilizados em qualquer formato, dentro desses ambientes virtuais de aprendizagem, tais como: textos, trechos de áudio ou vídeos em diversos formatos; apresentações (*Power point*); jogos; questionários; *applets* Java; *Macromedia Flash*, etc. Apesar de serem diferentes entre si, em alguns pontos desse sistema

eles precisam ser tratados da mesma maneira. Um exemplo é na montagem de um curso, no qual são inseridos diversos OAs distintos.

Problema

Como representar OAs de tipos diferentes de maneira que possam ser tratados de forma comum?

Forças

Os OAs de tipos diferentes possuem informações e características diferentes.

Mesmo OAs de um mesmo tipo, podem possuir diferentes formatos, como vídeo.

Em alguns pontos do sistema os OAs precisam ser tratados da mesma forma, independente de seu formato.

Solução

Criar uma abstração para representar os OAs e fazer com que os tipos mais específicos de objetos estendam essa abstração.

O padrão OA Abstrato abstrai o tipo do OA a ser desenvolvido e fornece a este OA uma estrutura reutilizável de seus componentes. A Figura 4 apresenta um diagrama de classes com a estrutura do padrão OA Abstrato.

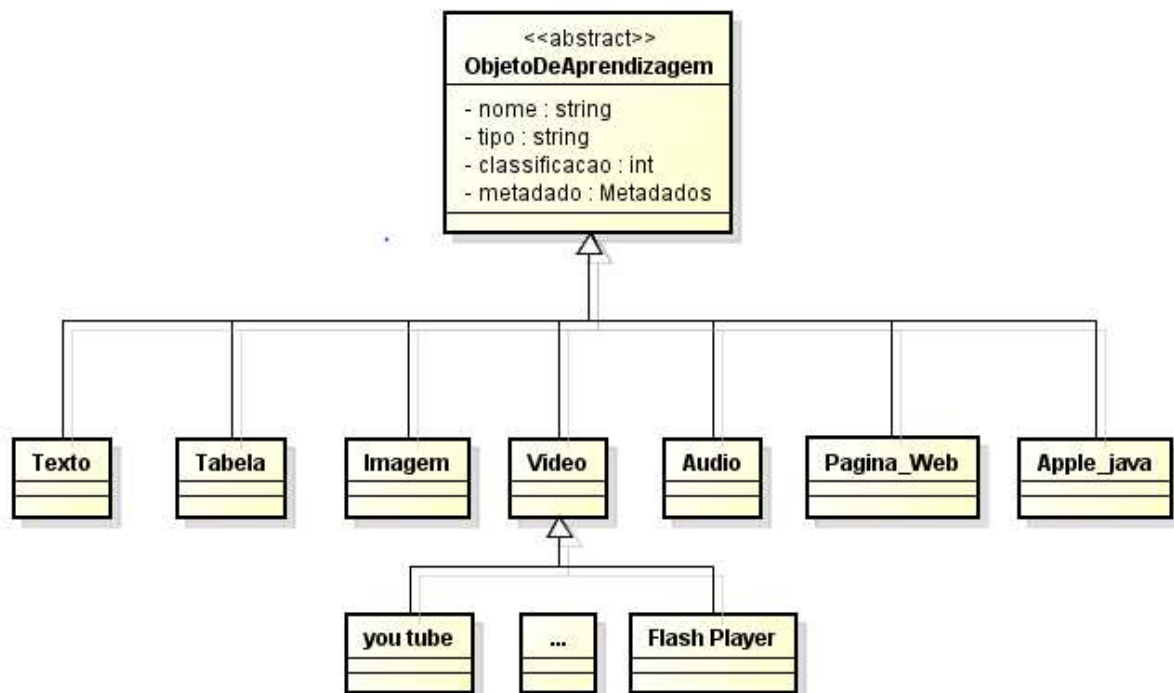


Figura 4 Padrão Objeto de Aprendizagem Abstrato

Este padrão pode ser adaptado de forma a classificar os OAs segundo a Taxonomia de Wiley (2000), apresentada na Seção 2, na qual são classificados de acordo com as características de qualidade com as que cada um se diferencia. Esta classificação pode ser útil,

pois distingue um OA baseado na complexidade que cada um possui, isto facilita a flexibilidade e reusabilidade (Figura 5).

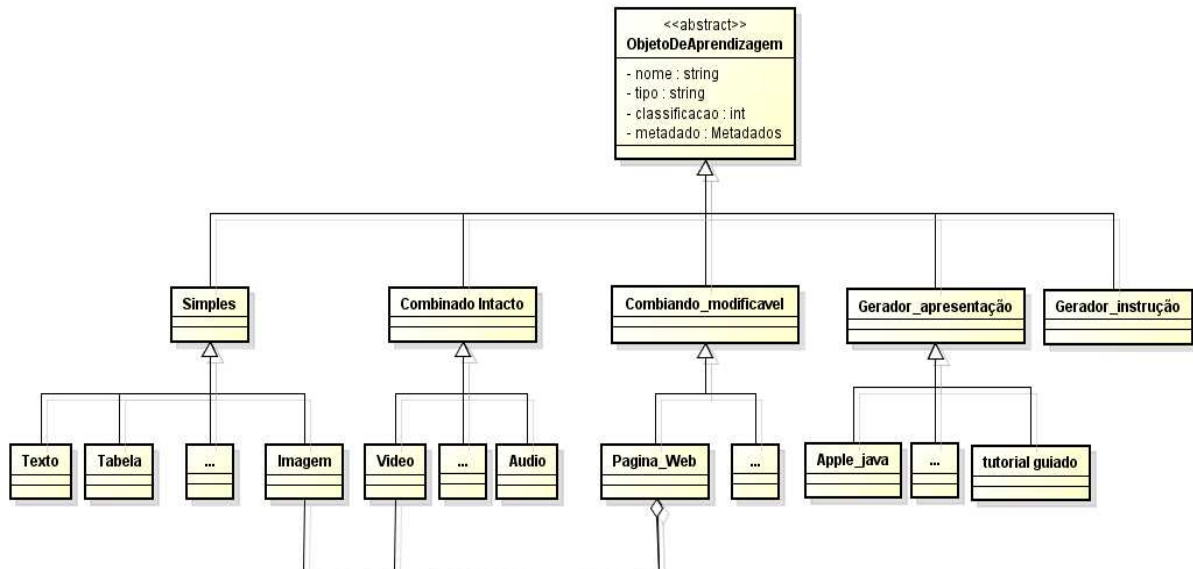


Figura 5 Padrão Objeto de Aprendizagem Abstrato para Taxonomia (Wiley, 2000)

Consequências

- (+) A partir da abstração é possível definir informações e comportamentos comuns a todos os OAs.
- (+) É possível estender o sistema para a adição de novos tipos de OAs.
- (+) Garante o princípio de reuso do OA.
- (-) Pode sobrecarregar a solução em sistemas que utilizam um único tipo de OA.

Padrões Relacionados

O padrão Entity, descrito em Domain Driven Design (Evans, 2004) é empregado, pois ele configura de maneira simplificada a distinção de cada objeto. Desta forma, entidades serão objetos definidos por sua identidade e não por seus atributos.

Usos Conhecidos

Ferramentas de autoria de OAs são amplamente utilizadas no desenvolvimento de novos OAs. Seu uso é justificado pela simplicidade do processo de geração destes, possibilitando até mesmo aos menos experientes em programação a criação de diversos tipos de OAs. Dentre estas, a ferramenta proposta por Souza (2010) "m-aula", cuja proposta é o desenvolvimento de OAs para dispositivos móveis, utiliza o padrão *Objeto de Aprendizagem Abstrato* no desenvolvimento de novos OAs. Outro uso encontrado na literatura, foi no MOODLE (2014) (*Modular Object-Oriented Dynamic Learning Environment*) um LMS, executado num ambiente virtual. Este programa permite a criação e execução de cursos

online, páginas de disciplinas, grupos de trabalho e comunidades de aprendizagem, estando disponível em mais de 75 línguas diferentes e mais de 200 países. Esta plataforma permite a organização de cursos e páginas de Internet, disponibilizando materiais estáticos (páginas de textos) e dinâmicos (atividades, como: fórum, lição, questionário, livro eletrônico, etc.). A classe hotpot dentro do LMS MOODLE executa exatamente o papel do Padrão *Objeto de Aprendizagem Abstrato* como se fosse a classe OA abstrata e a partir dela é possível desenvolver os questionários interativos.

3.2 Padrão Recuperação de Objetos de Aprendizagem

Contexto

Existem vários repositórios de OAs, nos quais são armazenados diversos tipos de OA, de diversas fontes. No entanto, não basta simplesmente armazená-los, é necessário fazer uso do que foi guardado. Isto resguarda os princípios básicos dos OAs: reusabilidade, acessibilidade, interoperabilidade e durabilidade.

Para isto é necessária a possibilidade de recuperação do OA armazenado num repositório. Este necessita seguir um padrão de armazenamento pré-determinado, ou seja, identificadores dos OAs.

Assim, através de dados armazenados sobre os OAs, é possível reconhecer e determinar o tipo exato destes, retornando ao buscador um OA relacionado ao termo pesquisado, o que favorece o reuso destes.

Problema

Como buscar os OAs adequados segundo as necessidades de ensino?

Forças

Buscar um OA requer conhecimento sobre quais metadados o descrevem.

Existem vários padrões para dados de OAs.

Solução

Associe um conjunto de metadados que caracterizem o OA independente do seu tipo para permitir que a busca seja feita através deles.

Os metadados fornecem ao OA uma espécie de rótulo de identificação. Este facilitará a recuperação de OAs em banco de dados, principalmente quando se tratar de objetos não textuais e assim não podem ser indexados por sistemas automáticos, como vídeos ou áudio.

Logo, para a escolha de metadados é necessário a adoção de padrões. Estes garantem a reutilização dos OAs e facilidade de uso dos mesmos. Normalmente, apresentam os aspectos gerais do objeto, ou seja, revelam aspectos técnicos para desenvolvedores de OA e pedagógicos para professores e educadores.

Neste sentido, o padrão de metadados de OAs (Figura 6) fornece a estrutura necessária para criação e recuperação de um OA armazenado num ROA.

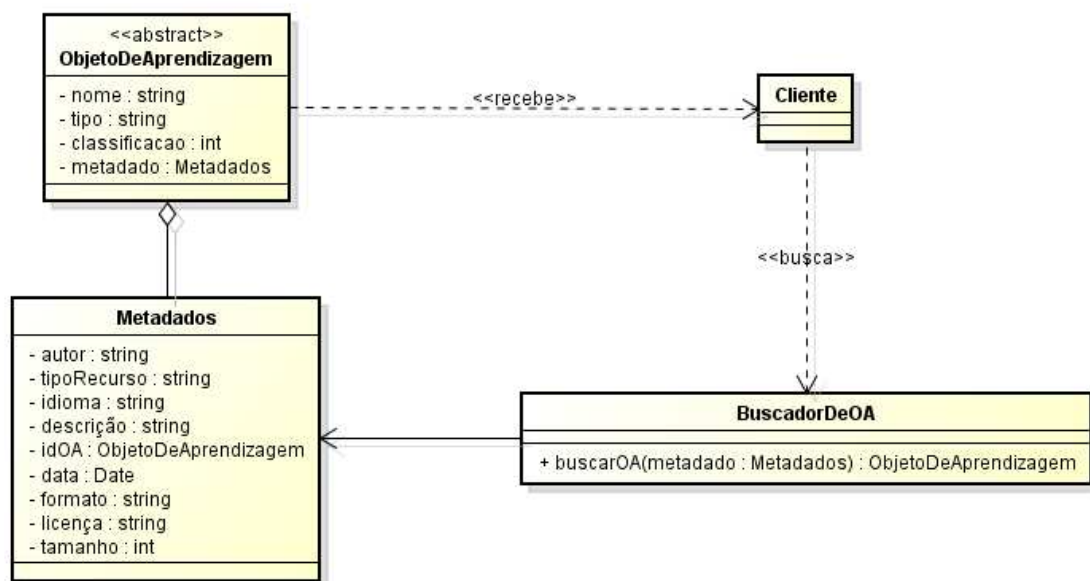


Figura 6 Padrão Metadado Objeto de Aprendizagem

Consequências

- (+) Os metadados favorecem a busca de OAs armazenados.
- (+) A utilização de metadados pode garantir a interoperabilidade.
- (-) O uso excessivo de categorias pode dificultar ou até mesmo ocasionar o não preenchimento das mesmas e desta forma, gerar inconsistência na base de dados.

Padrões Relacionados

O padrão *Value Object* (Evans, 2004) é empregado e permite que o conceito de metadados possa ser trabalhado, não se preocupando com a identidade do objeto.

Os diferentes tipos de OAs necessitam serem recuperados, independente de seus formatos, isto é possível através da utilização do padrão descrito na Seção 3.1 *Padrão Objeto de Aprendizagem Abstrato*.

Usos Conhecidos




Diversos repositórios de OAs utilizam Metadados de OAs, pois através da especificação dos metadados dos OAs é possível fazer a busca e recuperação destes OAs nos ROAs. Estes favorecem o reuso dos OAs criados em diversos locais e disponibilizados via *Web* para o público em geral. Abaixo seguem exemplos de alguns ROAs que utilizam o padrão *Recuperação de OAs*, destacando recursos de armazenamento e recuperação através de metadados:

O Banco Internacional de Objetos Educacionais (BIOE) (2014) é um ROA criado pelo Ministério da Educação (MEC) em parceria com o Ministério da Ciência e Tecnologia (MCT), Rede Latino-americana de portais Educacionais (RELPE), Organização dos Estados Ibero-americanos (OEI) e algumas universidades brasileiras. O BIOE é responsável por armazenar e disponibilizar OAs de acesso livre, em diferentes formatos, à comunidade educacional brasileira e internacional. O padrão de metadados utilizado, para o armazenamento e posterior recuperação, é o DCMI (2008) cujos campos utilizados estão descritos na Tabela 6, os campos em negrito são de preenchimento obrigatório no momento do armazenamento do OA, os demais são opcionais.

A Figura 7, mostra a utilização dos metadados no OA “*A basic property of integrals*” armazenado no BIOE. A busca (Figura 8) por este OA foi realizada com uso dos metadados armazenados quando o OA foi disponibilizado no ROA.

Tabela 6 Padrão Metadados DCMI utilizados no BIOE (2014)

Rótulo	Atributo
Título do recurso	dc.title
Título alternativo do recurso	dc.title.alternative
Autor	dc.contributor.author
Fonte do recurso	dc.publisher
Endereço eletrônico	dc.source
Nível de ensino	dc.audience.educationlevel
Componente curricular	dc.description.tableofcontents
Tema	dc.subject.category
Modalidade	dc.subject.category2
Data de publicação	dc.date.created
Tipo do recurso Formato	dc.type dc.format
Idioma	dc.language
Descrição do recurso	dc.description.abstract
Pré-requisito do recurso	dc.description

 educação infantil	 ensino fundamental	 ensino médio	 educação profissional
--	---	---	--


Classificação	Estadísticas	Registro completo
★★★★★		
Campo Dublin Core	Valor	
dc.audience.educationlevel	Educação Superior	
dc.audience.mediator	Universidade Federal de São Carlos (UFSCAR)	
dc.contributor.author	Rosemond, Paul	
dc.date.accessioned	2008-09-20T01:01:35Z	
dc.date.available	2008-09-20T01:01:35Z	
dc.date.created	2008	
dc.date.issued	2008-09-20T01:01:35Z	
dc.description2	To show a basic property of integrals	
dc.description3	This demonstration needs the "MathematicaPlayer.exe" to run. Found in http://objetoseducacionais2.mec.gov.br/handle/mec/4737	
dc.description	Calculus, Integrals	
dc.description.abstract	As you move b along the x axis, you divide the interval into two parts; the total area is the sum of the areas on the left and right	
dc.description.tableofcontents	Componente Curricular::Educação Superior::Ciências Exatas e da Terra::Matemática	
dc.identifier.uri	http://objetoseducacionais2.mec.gov.br/handle/mec/5585	
dc.language	en	
dc.location.country	us	
dc.publisher	Wolfram Demonstration Project	
dc.rights.license	Demonstration freeware using Mathematica Player	
dc.rightsholder	The Wolfram Demonstrations Project & Contributors	
dc.source	http://demonstrations.wolfram.com/ABasicPropertyOfIntegrals/	
dc.subject.keyword	Integrals	
dc.subject.keyword	Área	
dc.subject.keyword	graphic	
dc.title	A basic property of integrals	
dc.type	Animação/simulação	

Figura 7 OA "A basic property of integrals"
Fonte BIOE (2014)

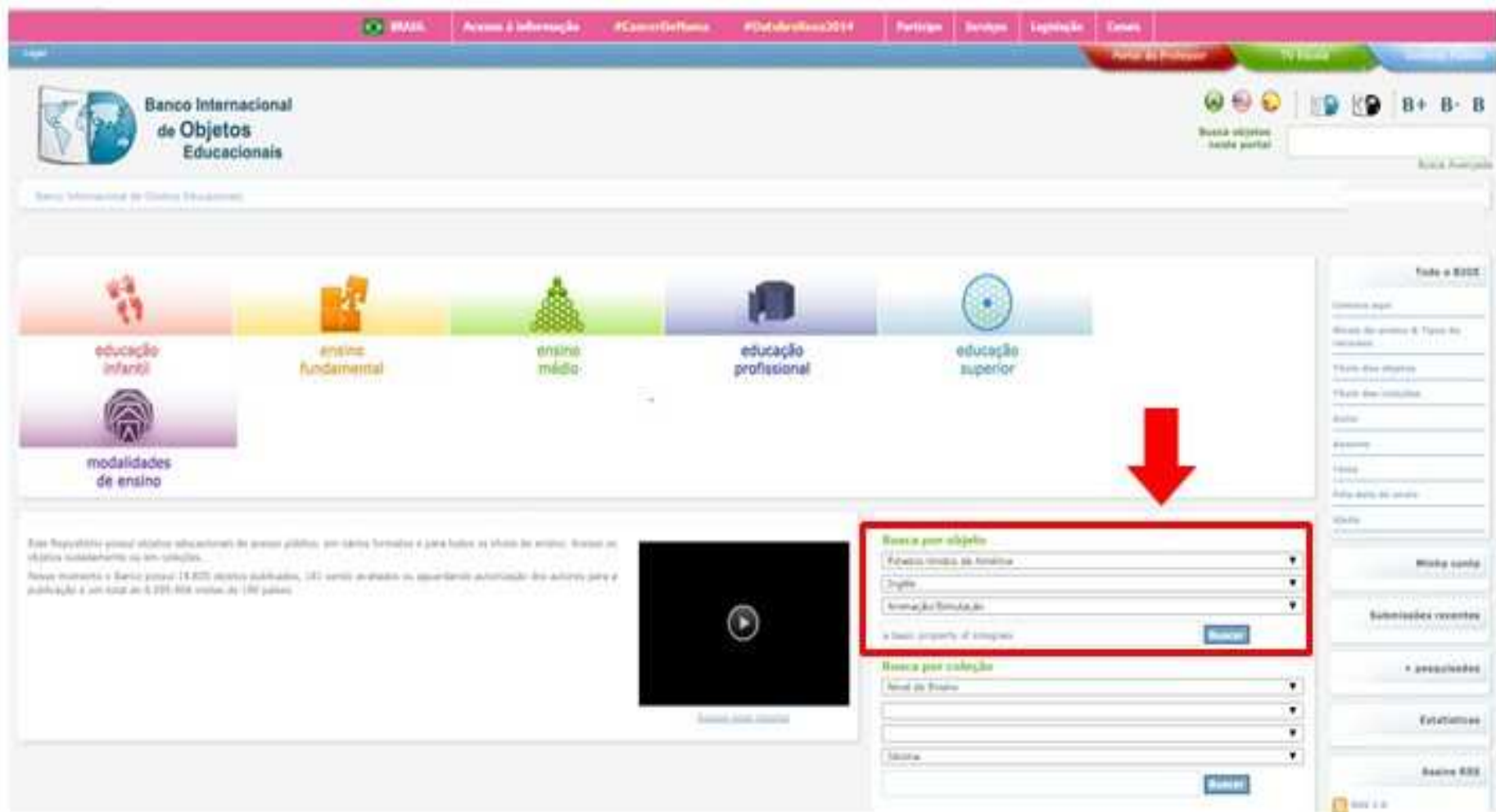


Figura 8 Busca através dos Medados do OA " A basic property of integrals" no BIOE (2014)

continuação Tabela 6 Padrão Metadados DCMI utilizados no BIOE (2014)

Rótulo	Atributo
Objetivo	dc.description2
Publicação associada	dc.relation.isversionof
Arquivo Inicial para execução	dc.realtion.requires
Direito autoral	dc.rightsholder
Licença	dc.rights.license
Tamanho	dc.format.extent
País	dc.location.country
Palavras-chave	dc.subject.keyword
Submetida por	dc.audience.mediator
Data da Submissão	dc.submitted dc.dateaccessioned dc.date.avaiable
	dc.date.issued
	dc.indentifier.uri
	dc.creator.identifier
Inserido por	

A catalogação de OAs no ambiente CESTA (Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem) foi baseada na especificação de metadados da norma IEEE 1484.12.1 (LOM, 2006) a qual define nove categorias para descrição do OA. Dessas categorias, apenas cinco foram utilizadas no projeto CESTA, intitulado de metadados Cestacore, descritas na Tabela 7. Estes identificadores são usados tanto no armazenamento, quanto na recuperação dos dados.

Tabela 7 Padrão Metadados Cestacore utilizados no CESTA
Fonte CESTA (2014)

Categoria	Identificador	Descrição
Geral	Identificador	Identificação única do objeto
	Título	Nome dado ao Objeto
	Idioma	Idioma utilizado no objeto
	Descrição	Descrição textual do conteúdo
	Palavras chave	Palavras-chave descrevendo os tópicos do objeto
Ciclo de Vida	Versão	A versão/edição do objeto
	Status	Estado atual do objeto (rascunho, revisado, editado, indisponível)
	Tipo de contribuição	Autor, editor, desconhecido, iniciador, designer gráfico, técnico, provedor de conteúdo, roteirista, designer instrucional, especialista em conteúdo
	Entidades que contribuíram	Pessoas e/ou organizações que contribuíram na evolução do objeto
	Data	Data da contribuição
Técnica	Formato	Formato de todos os componentes do objeto
	Tamanho	Tamanho em bytes do objeto
	Localização	URL (Universal Resource Locator) do objeto
	Tipo de tecnologia	Sistema operacional e navegador

continuação Tabela 7 Padrão Metadados Cestacore utilizados no CESTA
Fonte CESTA (2014)

Categoria	Identificador	Descrição
	Nome da tecnologia	PC-DOS, MS-Windows, Mac-OS, Multi-OS, Unix ou nenhum sistema operacional e navegador ou nenhum navegador
	Duração	Tempo de duração (usado em sons, vídeos, animações).
Educacional	Tipo de interatividade	Modo dominante de aprendizagem (ativa, expositiva, mista)
	Recurso de Aprendizagem	Tipo específico do objeto (exercício, simulação, questionário, diagrama, figura, gráfico, índice, slide, tabela, teste, experiência, texto, problema, auto avaliação, palestra);
	Nível de interatividade	Grau de interatividade (muito baixo, baixo, médio, alto, muito alto)
	Usuário final esperado	Tipo de usuário para o qual foi desenvolvido o objeto (professor, autor, aluno, gerenciador)
	Ambiente de utilização	Escola, universidade, treinamento, outros.
	Faixa etária	Idade do usuário final esperado
	Descrição	Comentários sobre como esse objeto deve ser usado
Direitos	Custo	Se utilização do objeto requer pagamento (sim ou não)
	Direito autoral	Se a restrições de direito autoral para uso do Objeto (sim ou não)
	Condições de uso	Comentários sobre as condições de uso do objeto

A Figura 9, permite a visualização da busca através de metadados no ROA CESTA.

Figura 9 Padrão Metadados Cestacore
Fonte CESTA (2014)

3.3 Padrão Objetos de Aprendizagem Composto

Contexto

Ministrar disciplinas em ambientes virtuais de aprendizagem requer que o professor faça a junção de diversos tipos de OA, tais como: apresentações tipo Power Point, textos, vídeos, jogos educativos, entre outros. Esta junção pode ser chamada de curso e os OAs inseridos possibilitam um melhor aproveitamento destes, em sistemas de gerenciamento de aprendizagem (LMS).

Problema

Como definir sequências reutilizáveis de OAs existentes?

Forças

- Na maioria das vezes o professor ao desenvolver uma disciplina/curso desconhece o conceito de OA.
- Muitos LMS não foram desenvolvidos tendo por base os conceitos de OA.
- A elaboração de um curso não deve ser complexa ou demandar tempo em demasiado.

Solução

Desenvolva um OA que seja composto por diversos OA desenvolvidos segundo o critério do padrão de OA abstrato.

Ao se desenvolver um curso com base no conceito de OA abstrato, este poderá ser reaproveitado em todo ou partes do mesmo, já que os OA simples fazem parte do contexto do mesmo. Desta forma, o próprio professor pode em outro momento reaproveitar o conteúdo do curso seja ele em parte ou como um todo. (Figura 10).

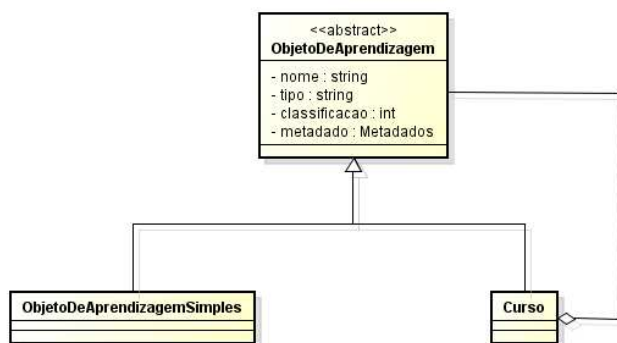


Figura 10 Padrão Objeto de Aprendizagem Composto

Consequências

- (+) A definição de uma estrutura de composição de OA facilitará a criação de cursos.
- (+) Permite que os OAs sejam criados de forma mais granular.
- (-) Nem todos os LMS podem suportar a criação de cursos no formato de OA.

Padrões Relacionados

Objetos de Aprendizagem Composto permite a criação de diferentes OAs, tendo em vista seu relacionamento com o padrão proposto na Seção 3.1, padrão *Objeto de Aprendizagem Abstrato*, e este possuir seu próprio conjunto de metadados.

Ainda, o padrão *Objetos de Aprendizagem Composto* utiliza o padrão *Composite* (Gamma et al, 1995), o que permite que OAs individuais e compostos (cursos), possam ser tratados de maneira uniforme.

Usos Conhecidos

Na versão 1.9.3, do MOODLE (2014), existe um pacote chamado *course*, este associa-se internamente com outros pacotes, como: *question*, *lesson*, *jornal*, *quiz*, etc, e executa a composição destes da forma como descrita pelo padrão *Objetos de Aprendizagem Composto*.

3.4 Padrão Metadados Essenciais para Objetos de Aprendizagem

Contexto

O Reuso é uma característica essencial em um OA. Quando estes são organizados por uma classificação de metadados e armazenados em um repositório, esta característica é mantida.

Por definição metadados são informações sobre um objeto, seja ele físico ou digital. Permitem a descrição das principais características, as mais relevantes, para que seja realizada a catalogação do OA em repositórios para posterior pesquisa e recuperação desses objetos para reuso.

Existem diversos padrões de metadados para OAs, entre estes se destacam: o DCMI (*Dublin Core Metadata Initiative*) com 15 elementos (DCMI, 2008), LOM (2006) com 78 elementos, ISO 19788 (*International Standards Organization*) (ISO, 2006).

Problema

Qual a quantidade necessária de características dos OA para que seja possível o armazenamento e recuperação destes nos Repositórios de OAs?

Forças

- Repositórios de OAs necessitam seguir um padrão de metadados para que seja realizado o armazenamento e recuperação de OAs.
- A principal característica de um OA é a reutilização do mesmo. Então, é necessário padronizar seu armazenamento para uma busca posterior.
- Dentre os padrões para dados de OAs alguns possuem vários campos relacionados ao OA.

Solução

Identifique quais metadados são essenciais à caracterização de um OA qualquer, permitindo assim o armazenamento e recuperação do mesmo sem inconsistência.

O padrão Metadados Essenciais para Objetos de Aprendizagem fornece uma quantidade, dita essencial, para as características dos OAs. Este conjunto de metadados foi baseado nos padrões mais utilizados que são o LOM (2006) e o DCMI (2008). A lista de metadados essenciais pode ser visualizada na Tabela 9. Do padrão DCMI (2008), apenas os metadados DC.Coverage, DC.Type e DC. OtherContributor não foram selecionados.

Tabela 8 Associação entre os metadados do Padrão Metadados Essenciais para OA, LOM e DCMI

Categoria	Rotulo	LOM	DCMI
Geral			
	Identificador	1.1.2 General.Identifier.Entry	DC.Identifier
	Título do recurso	1.2 General Title	DC.title
	Idioma	1.3 General. Language	DC.language
	Descrição	1.4 General Description	DC.description.abstract
	Palavras chave	1.5 General.Keyword	DC.subject.keyword
Ciclo de Vida			
	Entidades que contribuíram	2.3.2 Life Cycle. Entity	DC.contributor.author
	Data	2.3.3 LifeCycle.Contribute.Date	DC.Date
Técnica			
	Formato	4.1 Technical. Format	DC.format
	Localização	7.1 Relation.Kind	DC.source
Educacional			
	Usuário final esperado	9. Classification	DC.audience.educationlevel
	Faixa etária	5.6 Educational Context	DC.subject.category2
Direitos			
	Direito autoral	6.3 Rights.Description	DC.rights

Consequências

(+) Uma quantidade menor de metadados a serem preenchidos evitam que existam campos não preenchidos referentes aos dados a serem inseridos no ROA.

(+) Agilidade no momento de inserção de um objeto num ROA.

(-) Possível perda de informações relevantes sobre o OA.

Padrões Relacionados:

O padrão Metadados essenciais para Objetos de Aprendizagem permite que diferentes OAs sejam armazenados baseado nos principais metadados, este padrão está fortemente relacionado com os padrões propostos nas Seções 3.1 e 3.2, padrões Objeto de Aprendizagem Abstrato e Recuperação de OA e este possui seu próprio conjunto de metadados.

Usos Conhecidos

A catalogação de OAs no ambiente CESTA (Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem) foi baseada na especificação de metadados da norma IEEE 1484.12.1-LOM (LOM, 2006) a qual define nove categorias para descrição do OA. Dessas categorias, apenas cinco foram utilizadas no projeto CESTA, intitulado de metadados Cestacore, descritas na Tabela 9. Estes identificadores são utilizados tanto no armazenamento, quanto na recuperação dos dados.

Os metadados utilizados pelo repositório Merlot é composto por vinte e cinco elementos, entre os quais catorze elementos pertencem ao padrão LOM (2006) e seis são próprios do Merlot. Estes metadados são apresentados na Tabela 9, nesta apenas os de preenchimento obrigatório foram colocados pois, sem os mesmos não será possível cadastrar um OA no repositório.

Na Tabela 9 também são colocados os metadados correspondentes utilizados no BIOE.

Tabela 9 Comparação entre metadados usados em ROAs

Categoria	Cesta	BIOE	Merlot
Geral	Identificador	Data da Submissão (dc.submitted dc.dateaccessioned dc.date.avaliabile dc.date.issued dc.indentifier.uri dc.creator.identifier)	Identificador
	Título	Título do recurso (dc.title)	Title 1.2 General Title
	-	Título alternativo do recurso	-
			Mirror Site URL
			Display Image URL
	Idioma	Idioma (dc.language)	Language 1.3 General. Language
			Language of material 1.3 General. Language
	Descrição	Descrição do recurso (dc.description.abstract)	Description 1.4 General Description
		Pré-requisito do recurso (dc.description)	
		Objetivo (dc.description)	
	Palavras chave	Palavras-chave dc.subject.keyword	Keywords
		País dc.location.country	

4 ARQUITETURA DE APOIO À MODELAGEM CONCEITUAL DE OBJETOS DE APRENDIZAGEM

O foco principal deste trabalho é o desenvolvimento de uma arquitetura de referência para apoiar a criação de OAs, de forma, a contribuir para reusabilidade dos OAs. Esta arquitetura teve como base o Zachman Framework, que foi utilizado apenas até a terceira linha, ou seja, a visão da lógica do negócio. O preenchimento das respectivas células seguiu o método proposto por Pereira e Souza (2004).

A construção da arquitetura foi realizada através da ferramenta Enterprise Architect, versão trial, a qual suporta o Zachman Framework, além da modelagem em UML 2.1 e padrões de criação de OAs expostos no Capítulo III. A Tabela 10 resume a lista de artefatos desenvolvidos para modelagem desta arquitetura, os mesmos são apresentados nas seções a seguir.

Tabela 10 Artefatos produzidos nas Células do Zachman Framework

	<i>O quê</i>	<i>Como</i>	<i>Onde</i>	<i>Quem</i>	<i>Quando</i>	<i>Por que</i>
Escopo (planejador)	Lista de coisas importantes para o negocio	Lista de processos de negocio realizados	Lista de locais em que operam os processos	Lista de organizações importantes para o negocio	Lista de eventos significativos para o negocio	Lista de metas/estratégias do negocio
	-diagrama de dados do negocio	-diagrama de alto nível de processos	-diagrama de localização do negocio	-diagrama de caso de uso	-diagrama de eventos do negocio	-lista de motivação
Enterprise Model (dono)	Modelo Semântico	Modelo do processo negocio	Sistema de Logistica do negocio	Modelo de fluxo de trabalho	Calendário principal	Planejamento negocio
	-diagrama de entidade	-----	-decomposição funcional	-Organograma da organização	-----	-----
System Model (arquiteto)	Modelo lógico de dados	Arquitetura Aplicação	Arquitetura Sistema distribuído	Arquitetura Interface Humana	Estrutura processamento	Regras de negocio
	-padrões de projeto	-diagrama de atividade	- diagrama da arquitetura distribuída	- diagrama de caso de uso	-----	-Requisitos funcionais e não funcionais.
	Dados	Função	Rede	Pessoas	Tempo	Motivação

4.1 Nível Arquitetural "ESCOPO" Células A, B, C, D, E, F

4.1.1 Célula A - Dados (o quê)

Esta célula capta detalhes importantes para a organização. Para isto esta célula foi descrita com uso do diagrama de dados de negócio.

A Figura 11 mostra o pacote de ativos e recursos humanos necessários para o domínio em questão. Para visualizar o conteúdo e especificações é necessário abrir os pacotes.

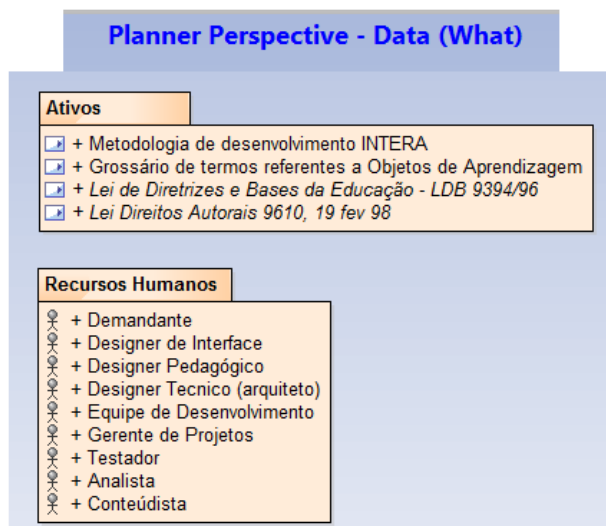


Figura 11 Célula Dados (perspectiva planejador)

Ativos:

- Metodologia de Desenvolvimento INTERA: um Framework de processos para desenvolvimento de qualquer tipo de conteúdo digital utilizado para a aprendizagem (Braga, Pimentel e Dotta, 2013);
- Glossário de Termos Referentes a OAs (Apêndice A);
- Lei de Diretrizes e Bases da Educação – LDB 9394/96;
- Lei de Direitos Autorais 9610/98.

Recursos Humanos Relevantes baseados na Metodologia INTERA:

- Demandante: responsável por solicitar o OA a ser desenvolvido;
- Gerente de Projeto: responsável por planejar e gerenciar o projeto de criação do OA. Mantém a comunicação entre a equipe, acompanha o cronograma, escopo e custo. Distribui e gerencia as atividades da equipe;
- Analista: responsável por elicitar os requisitos e análise do contexto dos requisitos, também elabora o planejamento de qualidade e dos testes do OA;
- Conteudista: responsável por elaborar ou reutilizar as situações didáticas do conteúdo, mantém o OA dentro dos objetivos pedagógicos no qual foi concebido;
- Designer de Interface: responsável por projetar os componentes da Interface do OA, a fim de potencializar o entendimento do conteúdo;

- Designer Pedagógico: responsável por realizar o planejamento pedagógico e avaliação pedagógica do OA;
- Designer Técnico (Arquiteto): responsável por escolher a tecnologia para desenvolvimento do OA, baseado nos requisitos e contexto;
- Equipe de desenvolvimento: responsável por desenvolver ou produzir o OA;
- Testador: responsável por realizar diversos testes nos OAs para garantir a qualidade.

4.1.2 Célula B – Função (como)

Com intuito de capturar a lista de processos realizados pela organização a Enterprise a célula foi descrita com uso do diagrama de alto nível dos processos de negocio.

A Figura 12 mostra o processo de construção de um OA, baseado na metodologia INTERA e adaptado a esta arquitetura.

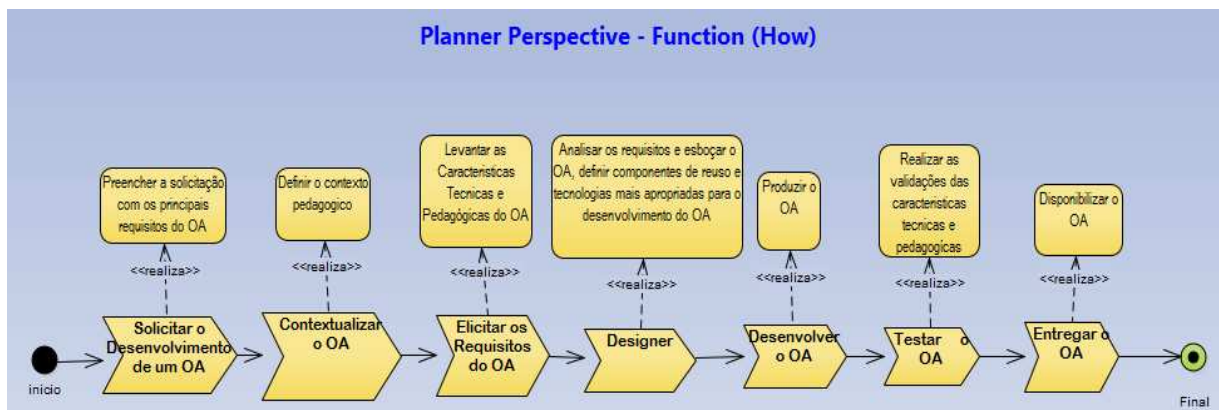


Figura 12 Célula Função (perspectiva planejador)

4.1.3 Célula C - Rede (Onde)

Esta célula pretende capturar os locais relevantes na organização. Assim, o diagrama de localização do negocio, ou seja, um diagrama semelhante a um diagrama de blocos foi construído.

Neste caso, os blocos separam informações referentes aos contatos dos principais órgãos governamentais gestores da educação no Brasil. A Figura 13 separa os órgãos em

pequenos blocos, a fim de facilitar o acesso aos contatos de cada um. Na arquitetura para acessar os dados do contato, basta clicar em um dos blocos.

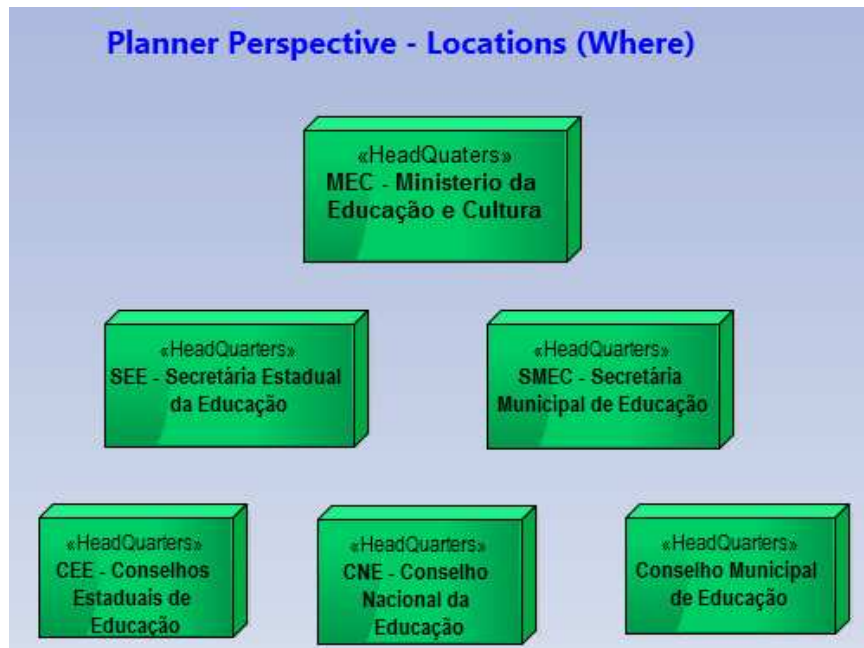


Figura 13 Business Location Diagram - Localização dos Órgãos Gestores da Educação no Brasil

4.1.4 Célula D - Pessoas (Quem)

Esta célula destina-se a capturar pessoas ou outras organizações que relacionam com o domínio de OA. Nesta foi utilizado uma variante do diagrama de Caso de Uso, o diagrama da organização (Figura14).

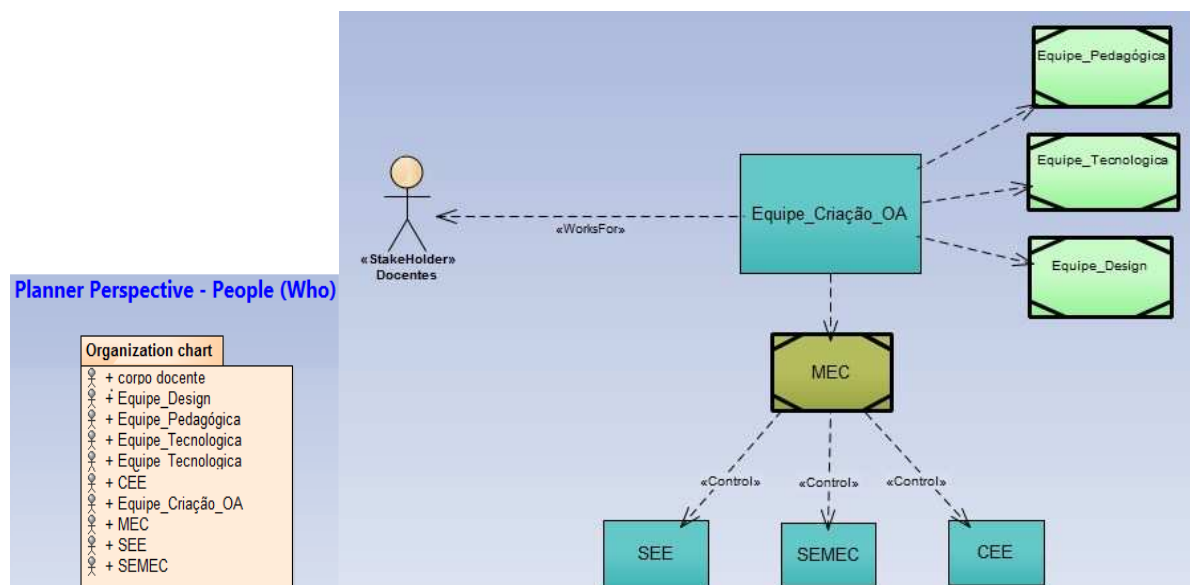


Figura 14 Célula Pessoas (Perspectiva planejador)

4.1.5 Célula E - Tempo (Quando)

Esta célula capta os eventos mais relevantes para a organização, desta forma, o diagrama de eventos do negocio foi desenvolvido. A Figura 15 apresenta os principais eventos relacionados à criação de OAs.

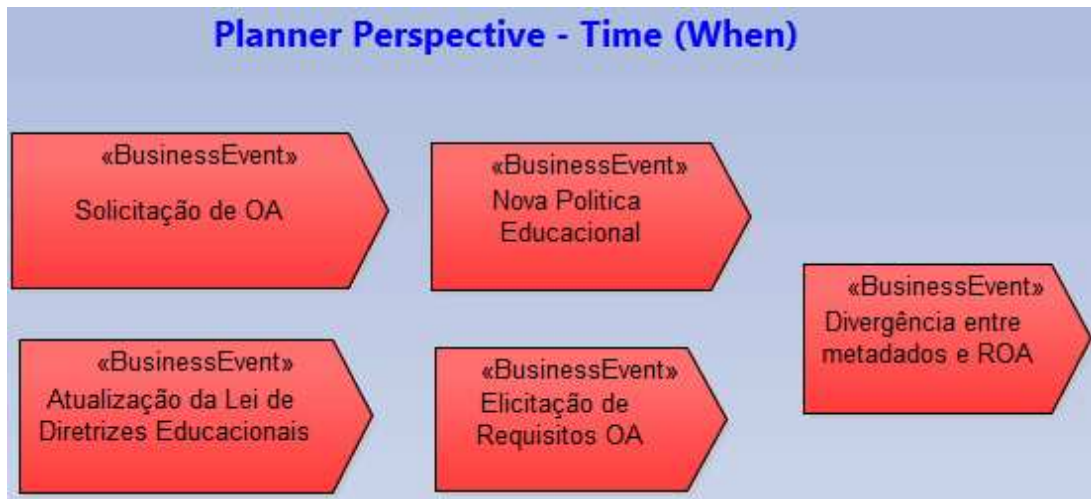


Figura 15 Célula Tempo (Perspectiva Planejador)

Embora todos os eventos listados sejam importantes, a ocorrência de algum deles pode gerar a inserção de outros.

4.1.6 Célula F - Motivação (Por que)

A motivação básica da criação desta arquitetura é o crescimento tecnológico ocorrido nos últimos tempos, bem como, a regulamentação do ensino pela lei 9394/96 (BRASIL, 1996).

4.2 Nível Arquitetural "modelo de negócios" Células: G, H, I, J, K, L

4.2.1 Célula G - Dados (o quê) – Modelo Semântico

As informações semânticas foram representadas através do Diagrama de Entidade Relacionamento. Na Figura 16 pode ser observado o modelo semântico, onde é possível visualizar as principais entidades informacionais e seus relacionamentos. A finalidade do uso,

bem como, o OA se especializa. No entanto, essa especialização é disjunta e os mesmos apenas podem assumir uma entidade por vez.

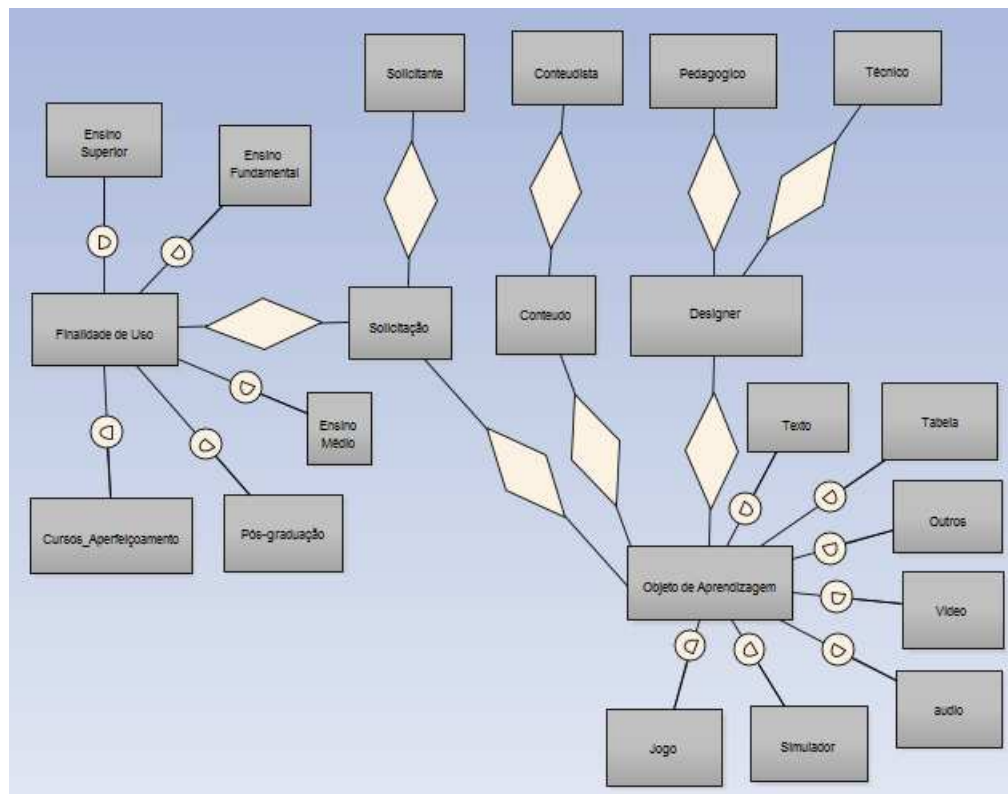


Figura 16 Modelo Semântico

O diagrama foi construído baseado nos textos disponíveis na célula A, no nível arquitetural “escopo”, desta arquitetura.

4.2.2 Célula H - Função(como) – Modelo de Processos da Organização

Tendo em vista que a arquitetura descrita nesta célula é muito semelhante à célula B, da perspectiva anterior, não foi necessário descrevê-la aqui.

4.2.3 Célula I - Rede (Onde)- Sistema Logístico da Organização

Aqui é apresentado o sistema logístico do negócio, ou seja, a localização dos órgãos relacionados a gestão da educação por estado. A Figura 17 mostra a rede de órgãos gestores da educação Estadual no Brasil. Para acessar os dados sobre cada órgão é necessário a visualização da arquitetura criada para a célula C, onde é possível ver a descrição de cada sigla e seu endereço web.

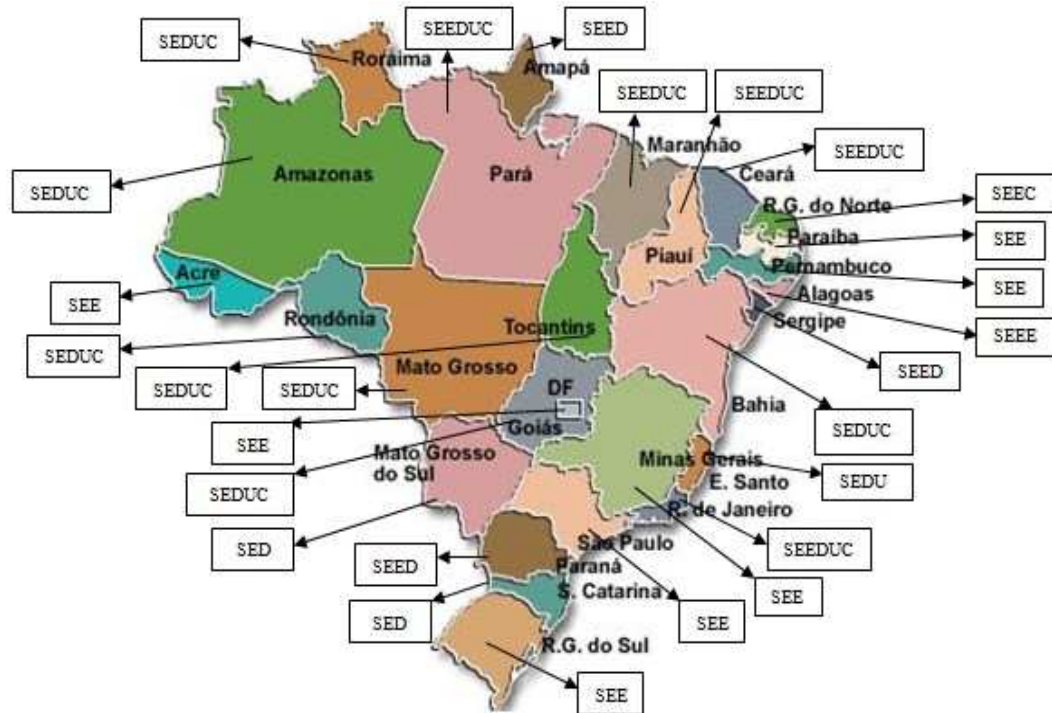


Figura 17 Órgãos Gestores da Educação por Estado

4.2.4 Célula J - Pessoas (Quem) – Modelo de Workflow

No intuito de capturar a informação referente a esta célula a Sparx sugere *Work flow Model*. Porém, esta representação está mais relacionada com a função do que com as pessoas envolvidas no processo. Desta forma, optou-se por utilizar a representação do organograma da organização. A Figura 18 representa a visão geral desta célula.

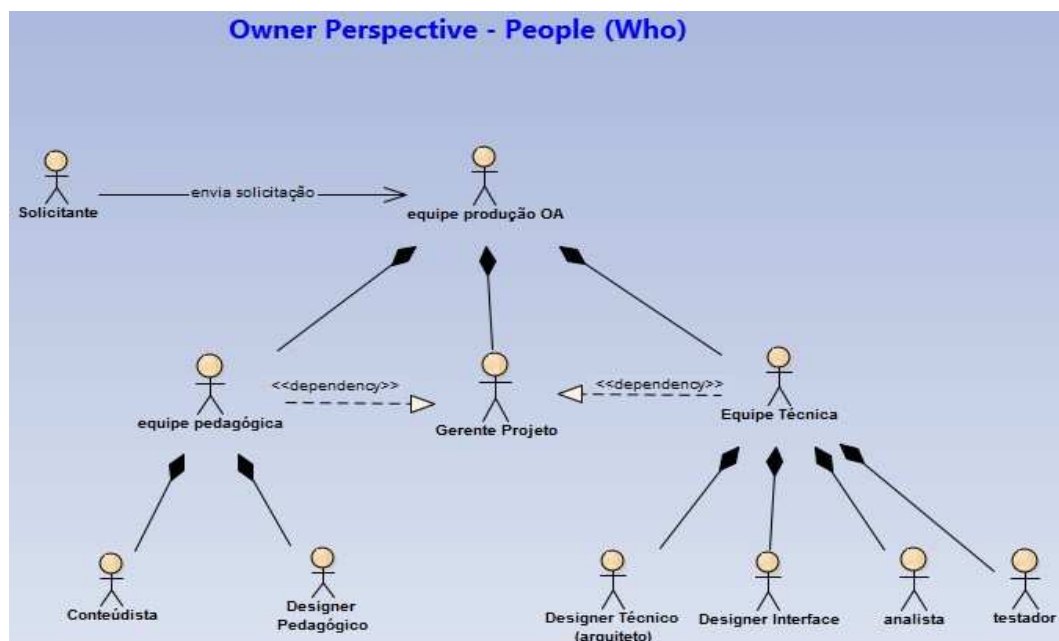


Figura 18 Organograma da Equipe de Criação de OAs

4.2.5 Célula K - Tempo (Quando) – Cronograma principal

Devido ao fato do “Tempo” aqui citado assemelhar com o que foi descrito na célula E da perspectiva anterior, seria redundante reescrevê-lo.

4.2.6 Célula L - Motivação (Por que) – Plano de Negócio

Conforme foi apresentada na perspectiva anterior na célula F, a “motivação” ainda continua sendo a mesma e, portanto, não cabe descrevê-la novamente.

4.3 Nível Arquitetural "Modelo de Sistema"- Células M, N, O, P, Q

4.3.1 Célula M - Dados (o quê) Modelo Lógico de dados

Como os OAs podem se especializar em diferentes tipos e a implementação destes diferencia entre si, optou-se por descrever os dados por meio de padrões de desenvolvimento de software, os mesmos foram apresentados de maneira detalhada no Capítulo III (Figuras 19, 20 e 21).

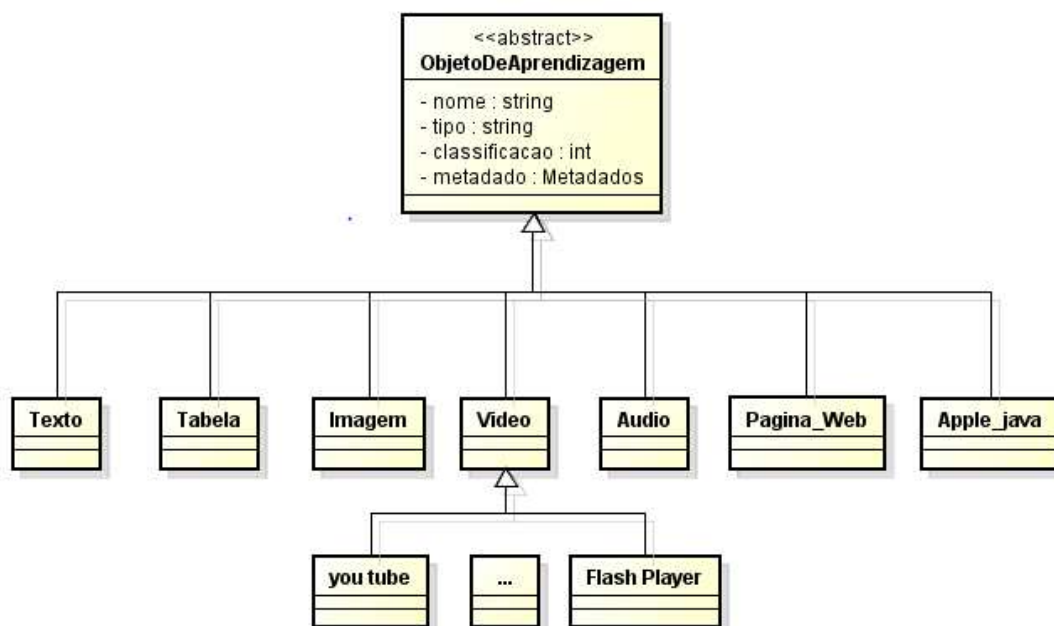


Figura 19 Padrão Objeto de Aprendizagem Abstrato

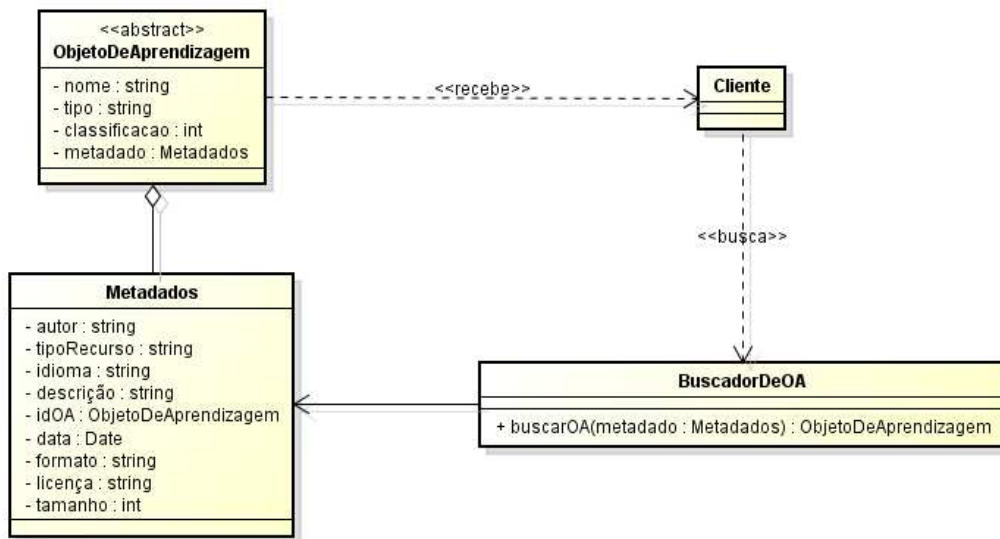


Figura 20 Padrão Metadados

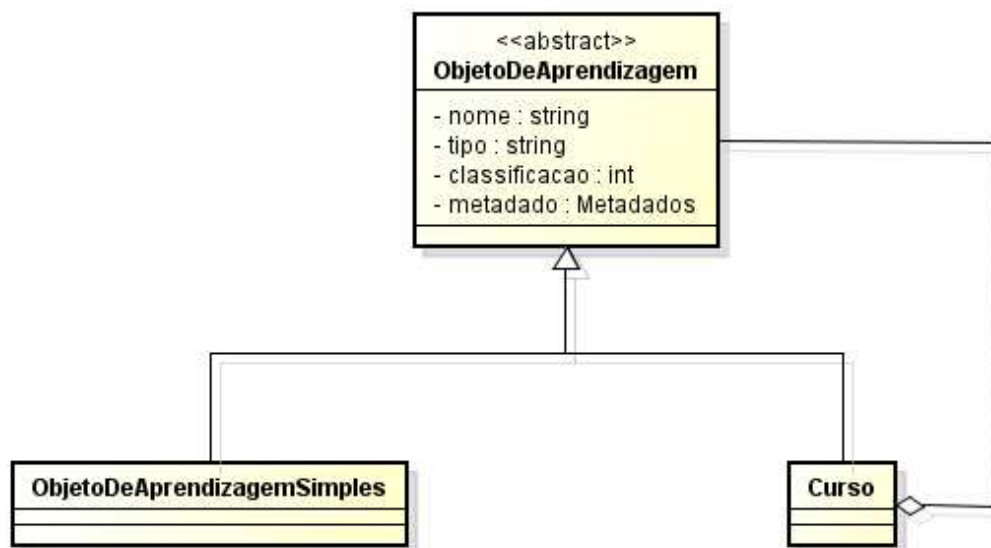


Figura 21 Padrão OAs Composto

4.3.2 Célula N – Função (como) – Arquitetura de Aplicações

O diagrama de atividade descreve as principais funções do sistema, na Figura 22.

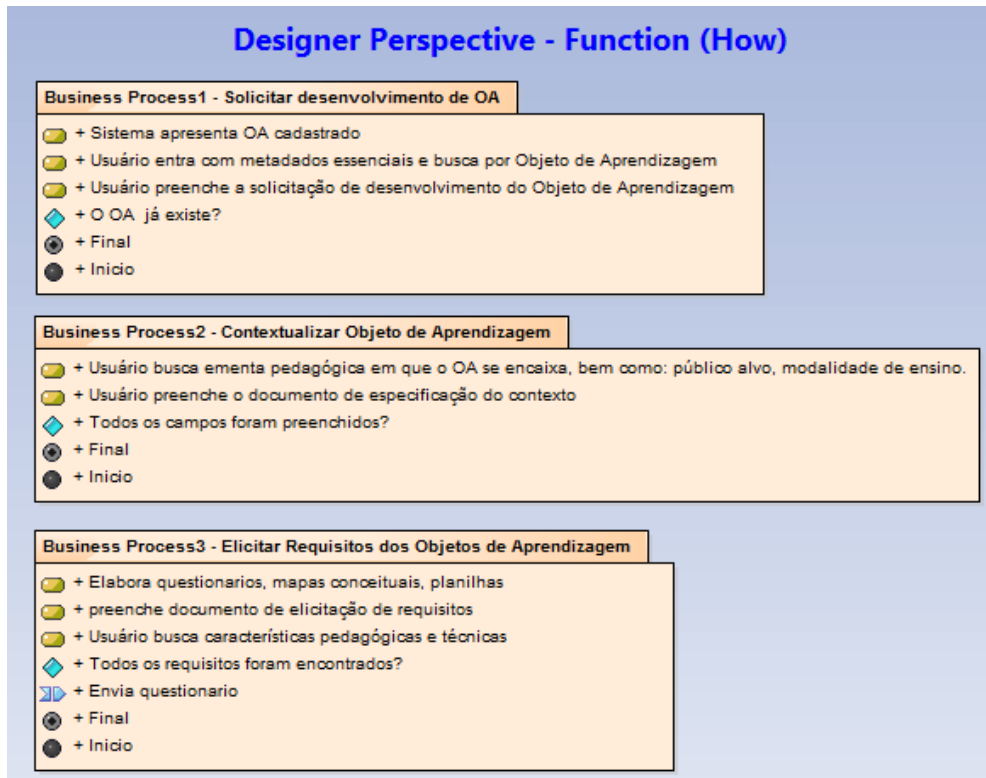


Figura 22 Células Pessoas (perspectiva planejador)

A Figura 23 apresenta o diagrama de atividades para solicitação de desenvolvimento de um OA (Apêndice B).

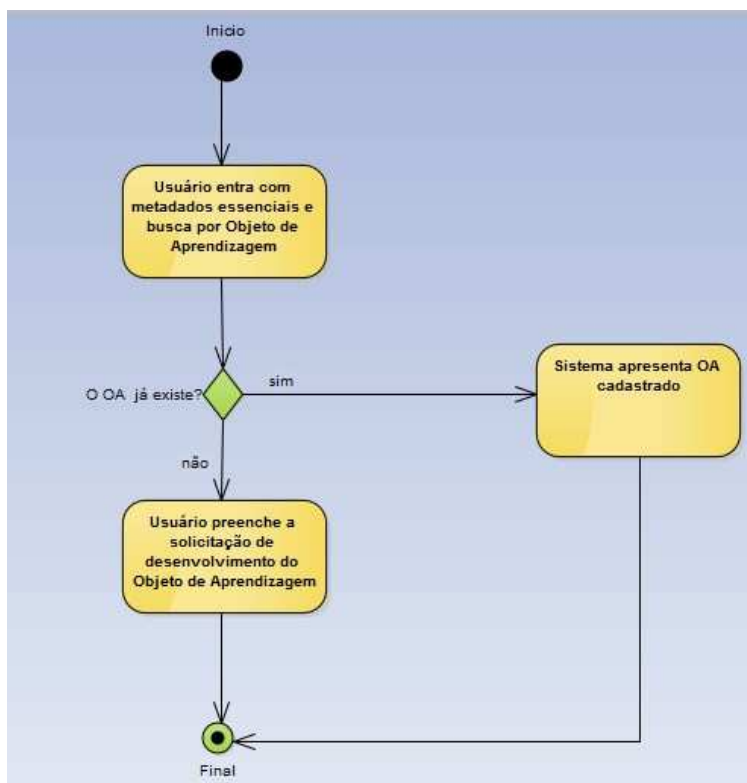


Figura 23 Diagrama de Atividades: Solicitação de Desenvolvimento de OA

O diagrama de atividades apresentado pela Figura 24 descreve a especificação do contexto do OA a ser produzido.

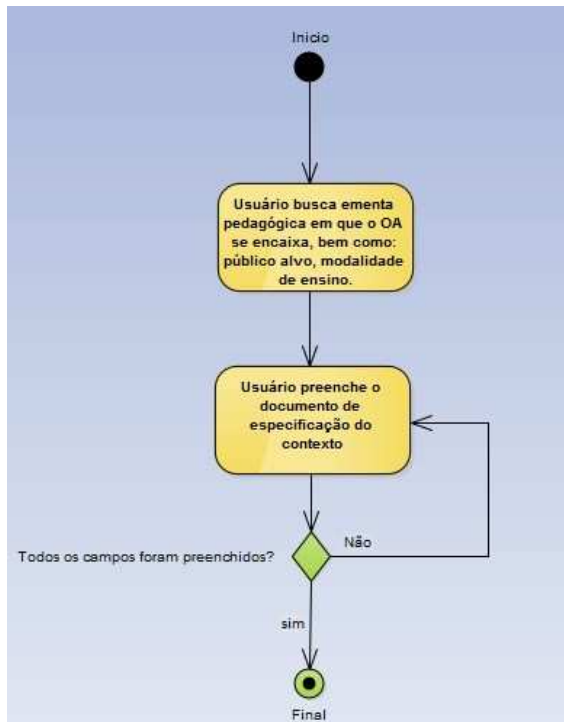


Figura 24 Diagrama de Atividades Especificação do Contexto do OA

A Figura 25 exibe o diagrama de atividades para o preenchimento dos requisitos do OA a ser produzido, é gerado um artefato semelhante ao Anexo B.

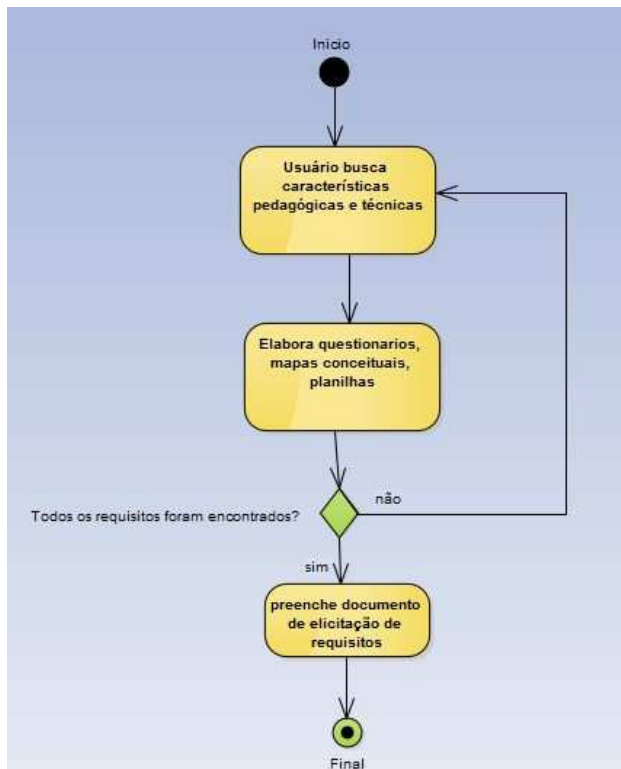


Figura 25 Diagrama de Atividades Elicitação de Requisitos do OA

4.3.3 Célula O - Rede (Onde) – Arquitetura Distribuída do Sistema

A representação da Arquitetura Distribuída do Sistema foi descrita com o uso do digrama de dados da arquitetura distribuída, ou seja, um diagrama de instalação UML o qual permite uma representação independente de plataforma para qual o sistema de informação será instalado.

Optou-se pelo modelo em três camadas, o qual é composto pela camada de apresentação onde o sistema pode ser acessado, preferencialmente, pela Internet; a camada do negócio composta pelo servidor da aplicação e a camada de dados onde os OAs produzidos são armazenados. Este sistema em camadas permite a interoperabilidade necessária do sistema, facilidade de manutenção e instalação futura. A Figura 26 apresenta este estilo.

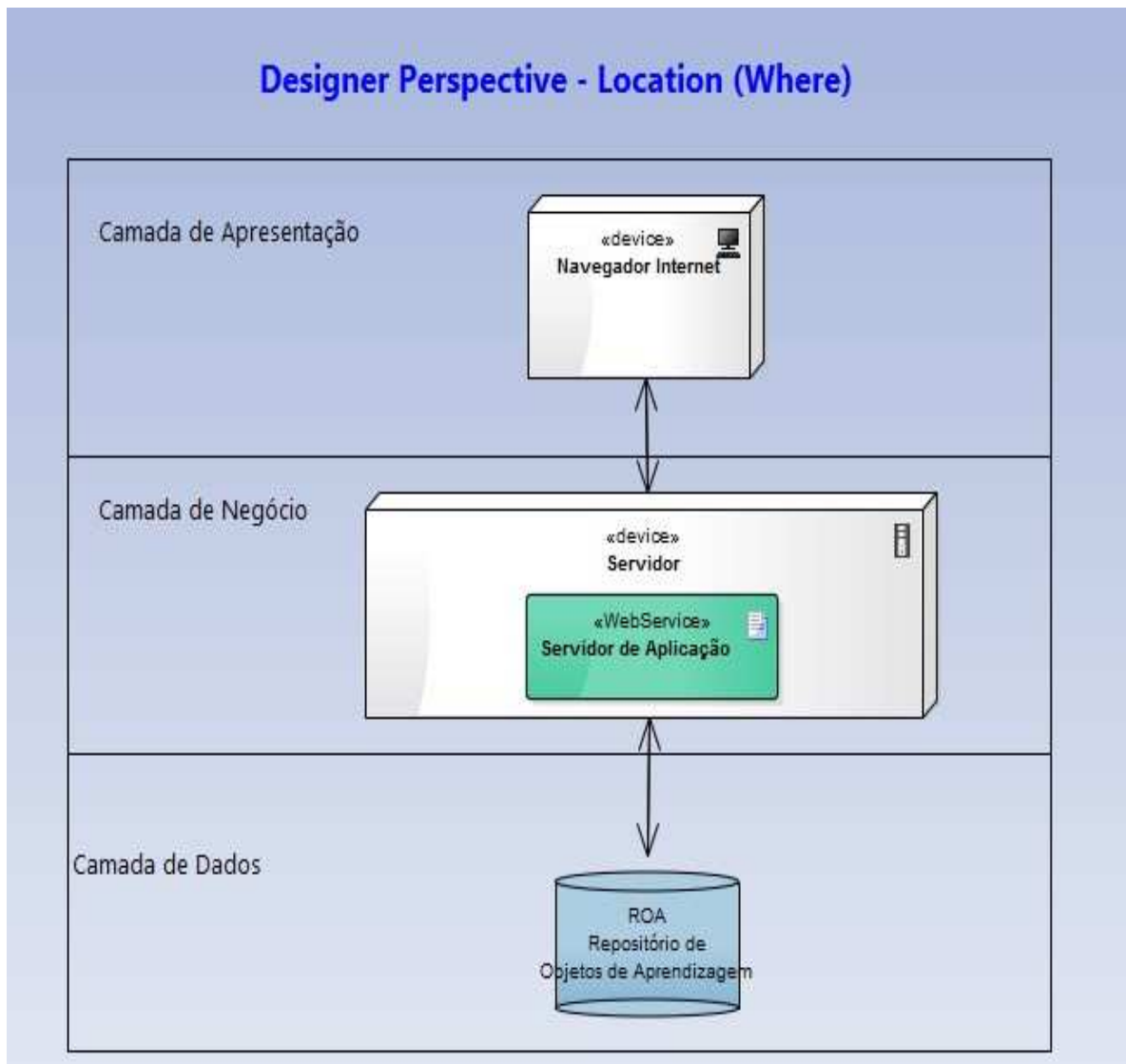


Figura 26 Estilo Arquitetural em três camadas

4.3.4 Célula P - Pessoas (Quem) – Arquitetura da Interface Humana

Nesta célula foi utilizado o diagrama de Caso de Uso para seu preenchimento. Conforme foi visto na Célula J os principais atores e casos de uso do processo podem ser visualizados na Figura 27.

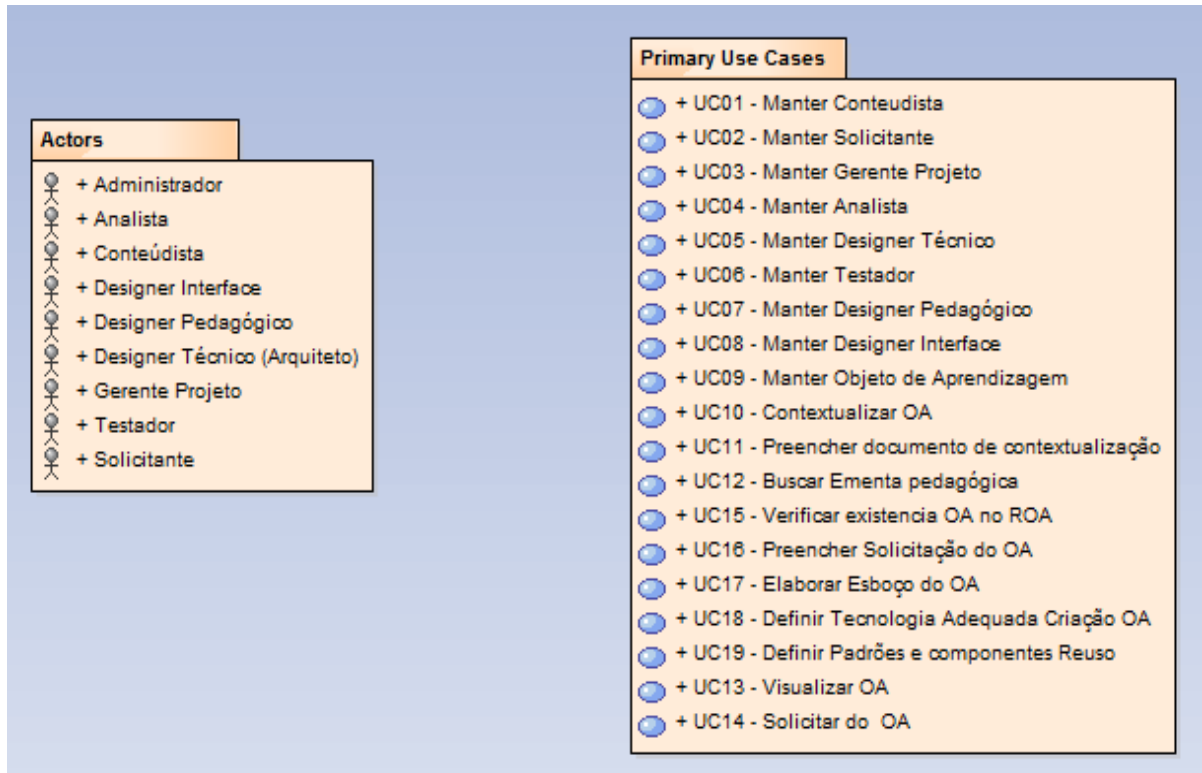


Figura 27 Principais Atores e Casos de Uso do Processo de Criação do OA

A nomenclatura dos casos de uso foi constituída por um identificador UC e o número correspondente do Caso de Uso, a fim de melhorar a identificação dos mesmos. Os Casos de Uso “manter” UC01 a UC 09, referem ao conhecido CRUD: editar, inserir, remover, buscar. Ao ser clicado sobre um Caso de Uso da Figura acima é aberta uma nova janela com detalhamento referente ao caso de uso, conforme é exemplificado pelas Figuras 28, 29 e 30. Na Figura 28 o usuário preenche o Apêndice B e na Figura 29 o artefato gerado é semelhante ao Anexo A.

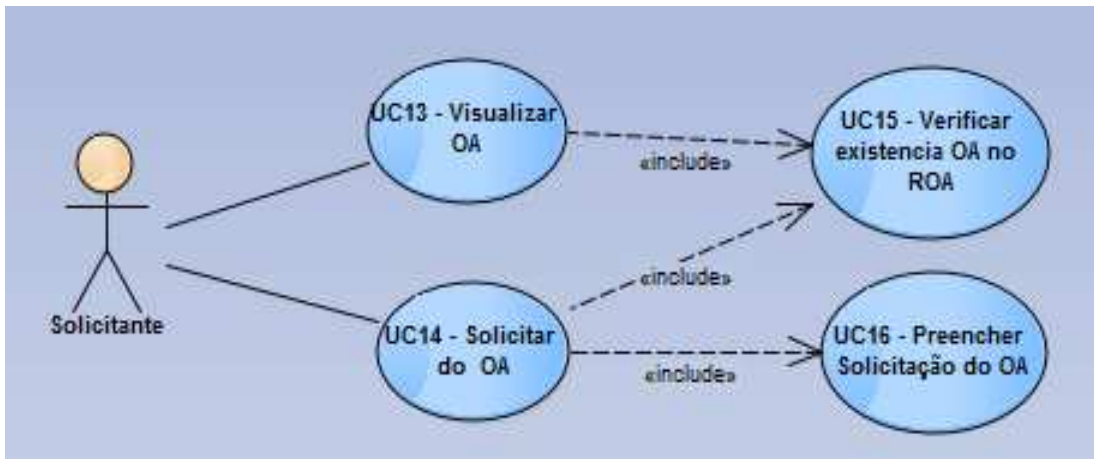


Figura 28 Exemplo de Caso de Uso (UC13)

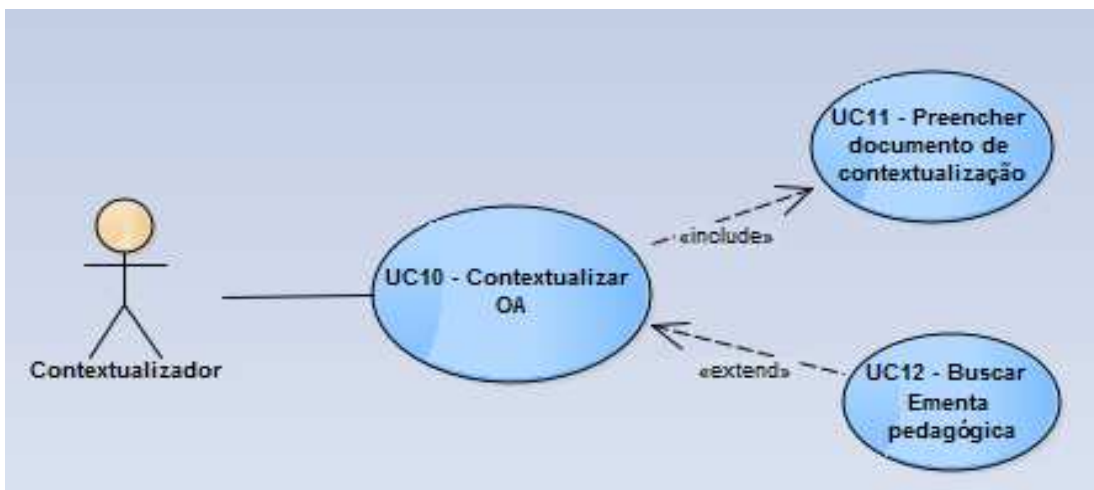


Figura 29 Exemplo de Caso de Uso (UC10)

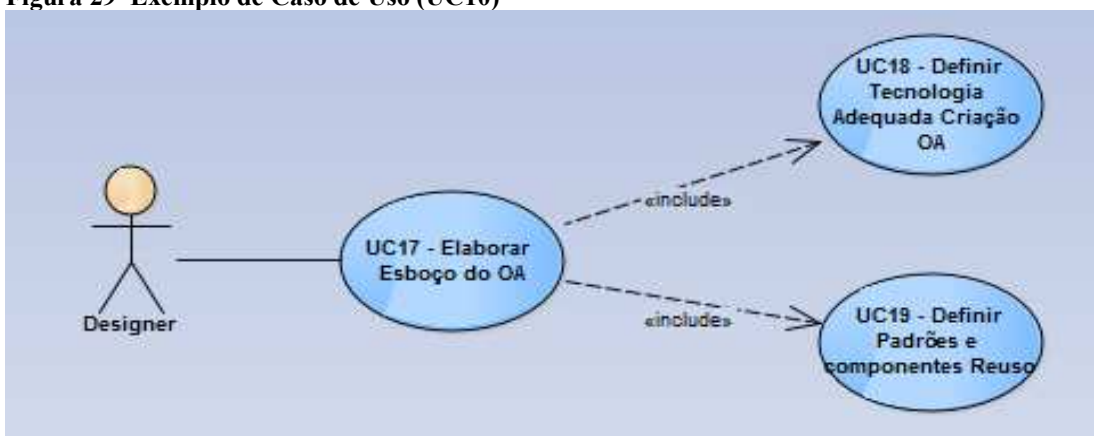


Figura 30 Exemplo de Caso de Uso (UC17)

4.3.5 Célula Q - Tempo (Quando) – Estrutura do Processamento

Devido não haver diferenças consideráveis em relação às perspectivas anteriores, não existe a necessidade de mencionar novamente.

4.3.6 Célula R - Motivação (Por que) – Regras do Negócio

Esta célula descreve os requisitos básicos para um Sistema de Informação de Criação de OAs. Na Tabela 11 são listados os requisitos não funcionais – RNF e os requisitos funcionais – RF, com seus respectivos códigos numéricos.

Tabela 11 Requisitos não Funcionais e Funcionais

Código	Requisitos
RNF01	Permitir a criação de OAs que possam ser reaproveitados em diferentes contextos de aprendizagem.
RNF02	Permitir que os OAs sejam adaptáveis a qualquer ambiente de ensino.
RNF03	Permitir a criação de OAs inseridos num padrão de metadados.
RNF04	Permitir que os OAs sejam disponibilizados via Internet.
RNF05	Produzir OAs que funcione independente de plataforma.
RNF06	Permitir que os OAs produzidos possam operar em diferentes hardwares, sistemas operacionais e browsers.
RNF07	Produzir OAs que forneçam o objetivo da aprendizagem e o feedback.
RNF08	Garantir que os OAs produzidos apresentem resultados dentro do esperado.
RNF09	Garantir que os OAs produzidos não apresentem defeitos técnicos ou pedagógicos.
RNF10	Produzir OAs fáceis de serem instalados.
RNF11	Produzir OAs que sejam fáceis de serem utilizados.
RNF12	Manter o mesmo método de desenvolvimento de OAs independente se via Web ou off-line.
RF01	Permitir a gestão dos analistas.
RF02	Permitir a gestão dos conteudistas.
RF03	Permitir a gestão dos designers pedagógicos.
RF04	Permitir a gestão dos designers Interface.
RF05	Permitir a gestão dos designers técnicos (arquiteto).
RF06	Permitir a gestão dos testadores.
RF07	Permitir a gestão do gerente projeto.
RF08	Manter os dados dos OAs produzidos.
RF09	Exportar os OAs produzidos.
RF10	Permitir gestão de solicitações de OAs.
RF11	Permitir a gestão de versões dos OAs.
RF12	Permitir a busca de OAs produzidos.
RF13	Permitir a gestão de contextualizações de OAs.

5 ANÁLISE DE UM OBJETO DE APRENDIZAGEM COM RELAÇÃO À ARQUITETURA PROPOSTA

No Capítulo anterior foi introduzida a arquitetura para desenvolvimento de OAs voltada para descrição de uma arquitetura empresarial e assim contribuir para o reuso de OAs. Nessa concepção foram identificados os requisitos e elaborada uma arquitetura que solucionasse problemas deste domínio de aplicação.

Resta agora apresentar a aplicação da arquitetura proposta de modo a se comprovarem os benefícios de sua utilização. Neste sentido, a Seção 5.1 apresenta uma análise da arquitetura em pauta, no desenvolvimento de um OA, do tipo Aula Virtual, proposto por Braga et al (2013). Alguns dos requisitos da arquitetura proposta foram implementados de forma total, parcial ou não atendeu ao solicitado (veja Tabela 11 ao final do capítulo).

5.1 Aula Virtual

Este OA foi desenvolvido em formato de site da *Web* para atender a área de Estudo da Língua Estrangeira (Inglês), principalmente na expressão linguística (oral e escrita). Assim, o mesmo tem como foco atender crianças com dificuldade de aprendizagem no auxílio do aprimoramento de leitura e escrita, raciocínio lógico e ordenação de espaço-tempo. Para sua utilização, o usuário deve possuir conhecimentos prévios em alfabetização e navegação básica na *Web*. Por se tratar de um OA que relaciona objetos a palavras e identificação de cores, através da reusabilidade, o mesmo pode ser utilizado dentro da disciplina de Artes, relacionando Cores e Formas.

5.1.1 Nível Arquitetural "ESCOPO" Células A, B, C, D, E, F

Célula A - Dados (o quê)

O OA implementa parcialmente os requisitos da célula A, pois no seu desenvolvimento foi utilizado o ativo metodologia INTERA. No entanto, não são descritos os demais ativos (Glossário de Termos Referentes a OAs; Lei de Diretrizes e Bases da Educação – LDB 9394/96; Lei de Direitos Autorais 9610/98). Com relação aos recursos humanos relevantes baseados na metodologia INTERA todos os requisitos foram atendidos, conforme Figura 31.

Membro	Papeis
Professor 1	Demandante e Conteudista, Testador
Professor 2	Design Pedagógico, Gerente de Projetos, Desing Técnico, Analista, Testador
Aluno 1	Equipe de desenvolvimento (programador), Testador, Designer de Interface

Figura 31 Papéis da Equipe de Desenvolvimento do OA.

Fonte: Braga et al 2013

Célula B – Função (como)

Para o completo preenchimento desta célula, um passo importante deve ser executado, a solicitação do desenvolvimento de um OA, no caso do Aula Virtual, não foi encontrado. Contudo, os demais passos foram executados e, assim, o OA atende parcialmente os requisitos da arquitetura.

O contexto e cenário do OA foram definidos pela professora demandante, responsável pela descrição do mesmo, conforme Figura 32.

1. Caracterização do O.A.	
1.1 Tipo do O.A.:	Site da web
1.2 Objetivos pedagógicos que se deseja atingir:	O aluno ao final deverá ter atingido o aprimoramento da leitura e escrita, raciocínio lógico, ordenação de tempo e espaço.
1.3 Área de conhecimento:	Língua Estrangeira
1.4 Disciplina principal:	Língua Inglesa
1.5 Ementa em que o OA se encaixa:	Expressão linguística (oral e escrita).
1.6 Tópicos dentro da ementa:	Identificação de cores e objetos
1.7 Descreva brevemente o O.A.:	Este AO contém tabela com palavras, cores e objetos para compreensão melhor identificação.
1.8 Público Alvo:	Crianças de 4 à 7 anos
1.9 Conhecimento prévio do público alvo:	Conhecimentos básicos de leitura e escrita.
1.10 Grau de Acessibilidade:	O OA poderá ser acessado por dispositivos móveis e computadores.
1.11 Fluência tecnológica:	Navegação básica na Web.
1.12 Problema atual:	Crianças com dificuldades em aprendizagem, identificação de cores.
1.13 Solução esperada:	Espera-se que a criança consiga relacionar a palavra ao objeto, aprendendo assim o seu significado na L. Inglesa.

Figura 32 Extrato do artefato de contextualização da metodologia INTERA preenchido para o OA a ser desenvolvido

Fonte: Braga et al (2013).

No passo de Elicitação de requisitos do OA a professora demandante preencheu o artefato de Especificação de requisitos, conforme Figura 33.

Artefato: Especificação de Requisitos
<p>Objetivos deste documento: Listar todos os requisitos do Objeto de Aprendizagem a ser desenvolvido.</p>
<p>Requisitos didático-pedagógicos: Deseja se basear em alguma teoria de aprendizagem que você conhece? Em caso</p>
<p>positivo cite qual: Eu gostaria de me basear no aprendizado através de jogos, fazendo com que o aluno se sensibilize para a construção de seu conhecimento com oportunidades prazerosas para o desenvolvimento de suas cognições. Existe algum material didático (livro, sites, apostilas) que você gostaria de aproveitar para ajudar na elaboração do conteúdo do seu OA? Em caso positivo cite qual ou quais e indique a localização de cada um deles. Ou então envie os arquivos zipados juntamente com este documento. Não. Existe algum OA que você gostaria de indicar aqui para servir de base para a elaboração do seu OA? http://penta3.ufrgs.br/CESTA/jogos/abruaxaria/bruxaria.swf Indique com um X se deseja que o seu OA contenha outros tipos de OAs</p> <ul style="list-style-type: none"> • apresentação do conteúdo em forma de texto ou html () • atividades () • “feedback” das atividades (x) • Outros () <p>Coloque aqui outras informações que achar importante. Este OA me chamou a atenção por ser muito bem elaborado para uso em várias disciplinas.</p>
<p>Requisitos de Funcionalidade: Que tipo de OA você gostaria que fosse desenvolvido? Indique com o X</p> <ul style="list-style-type: none"> • Imagem (x) • Vídeo () • Texto (x) • Animação (x) • Software de simulação () • Software do tipo jogo (x) • Curso on-line () <p>Quais as funcionalidades você gostaria que esse OA tivesse? Eu gostaria que este OA fosse muito parecido com esse que dei como exemplo, a partir de uma história divertida, o autor comunica-se com o leitor (fala), orientando como prosseguir para alcançar o objetivo.</p>

Figura 33 Extrato do artefato de Especificação de Requisitos da metodologia INTERA preenchido para o OA, Aula Virtual

Fonte: Braga et al (2013).

No passo Design, a partir dos requisitos foi definido o esboço do OA, Figura 34.



Figura 34 Esboço das Telas do OA, Aula virtual

Fonte Braga et al (2013).

Para o passo de desenvolvimento do OA, os autores escolheram a ferramenta de autoria XERTE específica para criação de aulas online. De acordo com Braga et al (2013), a

escolha se deu por ser uma ferramenta gratuita e de fácil manuseio, por um aluno de graduação.

Os testes do OA não haviam sido executados pela equipe.

A disponibilização do OA, seu manual de utilização e instalação ocorreu através do Tidia-ae, um ambiente virtual de aprendizagem utilizado na UFABC.

Célula C - Rede (Onde)

O OA não atende aos requisitos solicitados, pois não há descrição para as informações referentes aos contatos dos principais órgãos governamentais, gestores da educação no Brasil.

Célula D - Pessoas (Quem)

O OA atende a este requisito porém, segundo Braga et al (2013), devido às condições escassas de recursos financeiros, vários papéis são executados pela mesma pessoa. Desta forma, o corpo docente exerce o papel de equipe design, pedagógica e tecnológica, já a equipe de criação do OA, além de um docente, é composta por um aluno de graduação.

Célula E - Tempo (Quando)

Os eventos mais relevantes para a organização, neste caso, foram contextualização e elicitação de requisitos. Então, o OA atende parcialmente aos requisitos especificados, na arquitetura.

Célula F - Motivação (Por que)

A motivação descrita para criação deste OA foi a validação da Metodologia INTERA. Assim, não atende a motivação indicada na proposta da arquitetura.

5.1.2 Nível Arquitetural "modelo de negócios" Células: G, H, I, J, K, L

Célula G - Dados (o que) – Modelo Semântico

O OA, Aula Virtual, não atende ao modelo proposto na arquitetura, visto que, não há descrição, no mesmo para o modelo de dados adotado. No entanto, foi possível identificar

algumas entidades sugeridas nesta arquitetura. Assim, as mesmas foram marcadas no diagrama entidade relacionamento (Figura 35) como possíveis relações existentes.

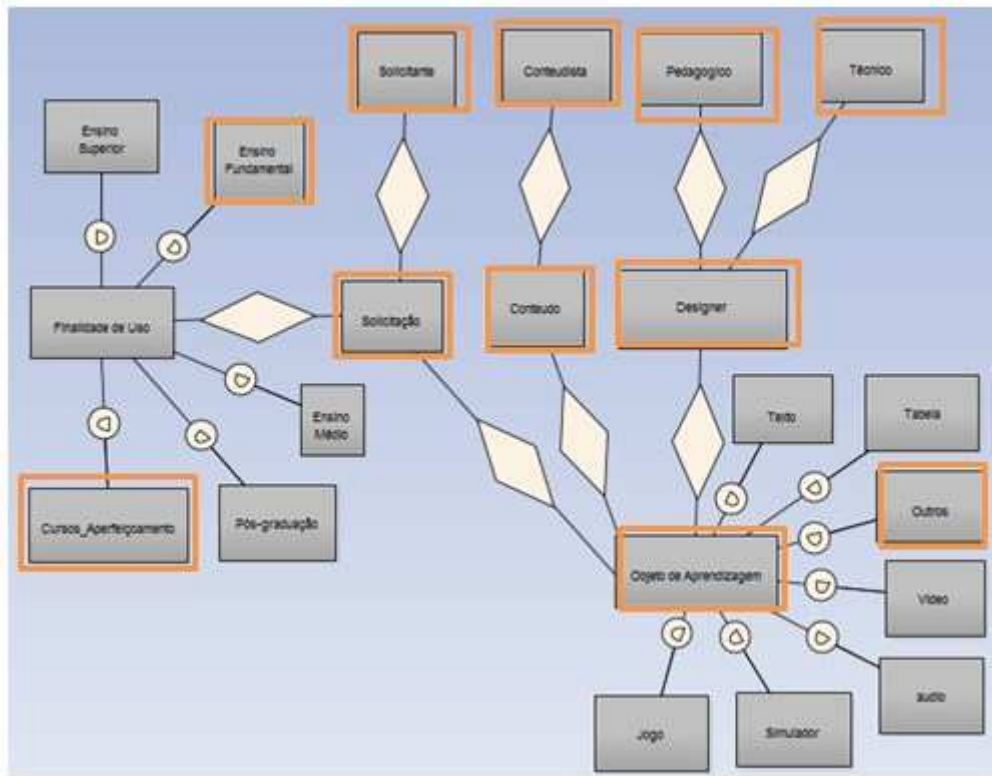


Figura 35 Modelo Semântico com base no OA, Aula Virtual

CÉLULA H - FUNÇÃO(COMO) – MODELO DE PROCESSOS DA ORGANIZAÇÃO

Não há necessidade de descrever esta célula, pois a mesma é similar ao que foi descrito no nível escopo e o OA, Aula Virtual, atendeu parcialmente ao requisito.

CÉLULA I - REDE (ONDE)- SISTEMA LOGÍSTICO DA ORGANIZAÇÃO

Como se trata de uma organização localizada no estado de São Paulo e o OA esta disponibilizado no Tidia-ae vinculado à UFABC, supõe-se que este esteja diretamente vinculado às normas do MEC e a Secretária Estadual da Educação – SEE de São Paulo. E neste caso, o OA atende a proposta da arquitetura.

CÉLULA J - PESSOAS (QUEM) – MODELO DE WORKFLOW

Na Figura 36 é descrito o organograma de criação do OA, Aula Virtual, que foi baseado na descrição dos autores e configurado dentro do modelo proposto pela arquitetura. Então, o OA analisado atende a este requisito.

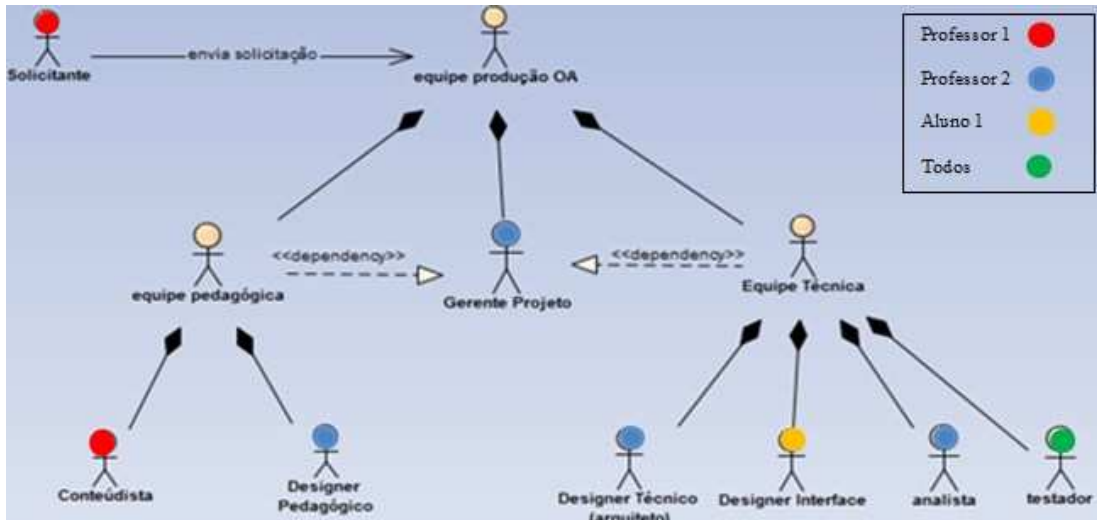


Figura 36 Organograma da Equipe de Criação de OAs

Célula K - Tempo (Quando) – Cronograma principal

A célula “Tempo” assemelha-se com o que foi descrito na célula E, da perspectiva anterior, considera que o OA atende parcialmente este requisito.

Célula L - Motivação (Por que) – Plano de Negócio

Conforme foi apresentada na perspectiva anterior na célula F, a “motivação” ainda continua a mesma e, portanto, o OA não a atende.

5.1.3 Nível Arquitetural "Modelo de Sistema"- Células M, N, O, P, Q

Célula M - Dados (o quê) Modelo Lógico de dados

Como consta na descrição da aula virtual, esta foi desenvolvida com uso da ferramenta de autoria Xerte. Neste caso, não foi disponibilizado nenhum documento relativo ao diagrama de classes utilizado para documentação do OA. Logo, o OA não atendeu ao proposto pela arquitetura.

Célula N – Função (como) – Arquitetura de Aplicações

Não foi bem explicado no artigo a sequência lógica para elaboração do OA, já que a metodologia INTERA permite ir e vir sem necessidade de uma sequência. O artigo descreve como primeiro processo a contextualização do OA, seguido pelo levantamento de requisitos de um OA, Design, Desenvolvimento, testes, avaliação e gestão de projetos.

De acordo com a arquitetura proposta, o OA a ser desenvolvido passa pela verificação de sua existência no ROA. O OA, Aula Virtual, não descreve esta etapa e não atende ao proposto pela arquitetura.

Célula O - Rede (Onde) – Arquitetura Distribuída do Sistema

A disponibilização do OA, Aula Virtual, é na Internet e o mesmo segue o modelo de três camadas: camada de apresentação onde o sistema pode ser acessado, preferencialmente pela Internet; a camada do negócio composta pelo servidor da aplicação, o Tidia-ae e a camada de dados, onde os OAs produzidos são armazenados. Desta forma é alcançada a interoperabilidade necessária do sistema, facilidade de manutenção e instalação futura. A Figura 37 apresenta o estilo adotado e fica claro que o OA atende ao proposto pela arquitetura.

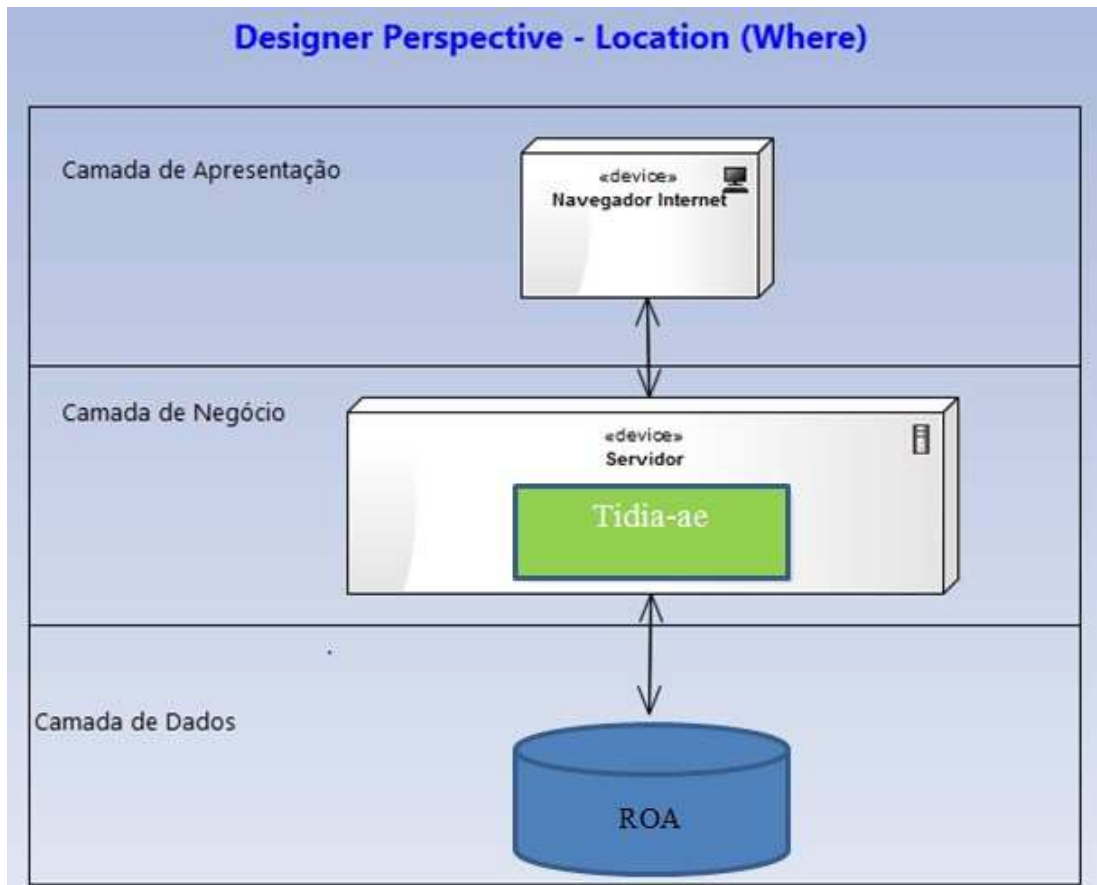


Figura 37 Estilo Arquitetural em 3 Camadas definido no OA aula virtual

Célula P - Pessoas (Quem) – Arquitetura da Interface Humana

No OA, Aula Virtual, os principais autores são definidos no início do processo de construção do mesmo. A descrição é bem sucinta e já foi inserida no nível anterior. Não foi detectada a existência de um diagrama de caso de uso referente às funções determinadas a

execução do processo de desenvolvimento. Então, o OA não atende ao descrito nesta arquitetura.

Célula Q - Tempo (Quando) – Estrutura do Processamento

Não foi acrescido nenhuma diferença em relação as perspectivas anteriores, portanto, o OA atende parcialmente ao especificado na arquitetura.

Célula R - Motivação (Por que) – Regras do Negócio

Na descrição do OA, Aula Virtual, não há especificação dos requisitos não funcionais e funcionais, referentes ao mesmo. Neste sentido o OA, não atende ao proposto na arquitetura.

5.2 Discussão dos Resultados

Neste Capítulo foi realizada uma análise comparativa entre o que foi proposto pela arquitetura desenvolvida (Capítulo IV) e o implementado no OA, Aula Virtual, (BRAGA et al, 2013). Destarte, para cada uma das células do Zachman Framework foram coletados os artefatos referentes ao proposto pela arquitetura e documentados pelo autor do OA.

A Tabela 11 apresenta um resumo da comparação e as respectivas informações acerca da implementação completa (atende), parcial se atendeu a apenas alguns dos artefatos ou que não foi realizada (não atende)

Tabela 12 Implementação da Arquitetura Proposta pelo OA Aula virtual

	Dados	Função	Rede	Pessoas	Tempo	Motivação
Escopo	Parcial	Parcial	Não Atende	Atende	Parcial	Não Atende
Modelo de Negócio	Não Atende	Parcial	Atende	Atende	Parcial	Não Atende
Modelo de Sistema	Não Atende	Não Atende	Atende	Não Atende	Parcial	Não Atende

No nível Arquitetural “Escopo”, o OA em estudo atendeu apenas 18% ,ou seja, apenas a célula referente as pessoas envolvidas no processo foi realizado, em sua totalidade. Se o OA

tivesse seguido o que foi proposto pela arquitetura, sua documentação constaria de artefatos relevantes ao processo de criação do OA. Tais artefatos ficariam disponíveis não somente para a criação deste AO, mas também pela independência proporcionada pela arquitetura estes poderiam ser reutilizados na criação de outros tipos de OAs.

No nível arquitetural, “Modelo de Negócios”, o percentual de similaridade sobe para, aproximadamente, 33%. Ainda é visível o não comprometimento do OA Aula Virtual, criado com a documentação para sua possível reutilização. A célula modelo semântico, que descreve a forma com a qual o OA deve fazer seus relacionamentos, não foi encontrada na documentação pesquisada. Este fato causa a produção de um OA de qualidade inferior.

Dentro do nível arquitetural, “Modelo de Sistema”, houve uma alta taxa de não atendimento, ou seja, o OA não atendeu a 80% do que foi proposto pela arquitetura. Esta discrepância deve-se ao fato do OA não ter seguido os padrões propostos no Capítulo III, tendo em vista que a ferramenta de autoria Xerte não fornece detalhes das classes e relações estabelecidas.

As relações estabelecidas neste Capítulo, implementação do OA e a proposta da arquitetura, denotam o empobrecimento, em termos de documentação, proporcionado na criação deste, sem uma arquitetura que forneça os artefatos a serem documentados.

6 CONCLUSÃO

No decorrer desta pesquisa verificou-se que apesar de existir uma grande demanda para o desenvolvimento de OAs, estes, em sua grande maioria, são desenvolvidos de maneira inadequada e custosa. Provavelmente, a principal causa seja a falta de uma Arquitetura de Software voltada para este domínio de aplicação. O estudo realizado sobre OAs mostrou que estes são desenvolvidos de acordo com tipos específicos de OA e em função disto, são construídos com base em heurísticas das equipes de desenvolvimento. Como resultado ocorre um engessamento do processo produtivo de software e aumento do custo global desses OAs. Outro ponto considerado por esta pesquisa foi a estrutura empresarial a ser montada para o desenvolvimento de OAs, que na sua maioria também são estruturados heurísticamente, com aumento do custo efetivo do processo. Este custo elevado, movido pelo modelo de desenvolvimento de OAs, não pode mais ser suportado e, para isso, demandam soluções.

Assim, a característica-chave para reverter este caso é a estruturação lógica da forma como um OA é desenvolvido, bem como, a montagem do departamento/empresas de desenvolvimento de OA. Para tal, faz-se necessário a especificação de um padrão estrutural de software para a criação de OAs que mantenha os requisitos desse domínio e, ao mesmo tempo, promova a redução de custos, aumento da reusabilidade e ganho de qualidade nos OAs produzidos. Dentro desse contexto é relevante uma Arquitetura de Software.

A fim de estabelecer uma lógica estrutural que promovesse a redução de custos e um ganho de qualidade na produção de OAs, esta dissertação propôs a estruturação de um departamento de criação de OA fosse norteada por uma Arquitetura de Software Empresarial, voltada para os requisitos desse domínio. Após a verificação dos requisitos necessários para desenvolver OAs foram escritos padrões de criação de OAs, a partir destes padrões foi descrita uma arquitetura empresarial para o desenvolvimento de OAs.

A construção dessa arquitetura empresarial contou com o benefício do Zachman Framework, suportado pela versão trial da ferramenta Enterprise Architect. Diversas vantagens se destacaram nesta abordagem, dentre estas: a possibilidade de utilizar uma linguagem comum a desenvolvedores, a UML; ainda permitir um referencial comum tanto aos recursos humanos associados ao negócio quanto à tecnologia.

Devido a grande variedade de artefatos permitidos pelo Zachman Framework, neste trabalho, optou-se por utilizar um método híbrido, as propostas de Pereira & Sousa (2004) e da Sparx Systems (Sparx, 2014), para construir a Arquitetura Empresarial proposta. Na

prática a sequência proposta foi ligeiramente alterada, visto que o foco seria a modelagem de uma nova organização, sentiu-se a necessidade de priorizar as células associadas ao negócio, em particular a primeira linha do Zachman Framework. Também se fez necessário alterar alguns dos artefatos produzidos pela ferramenta. Outra questão foi a descrição de uma arquitetura geral para qualquer tipo de OA a ser criado e para isto a terceira linha foi descrita parcialmente com a utilização dos padrões de criação de OA do Capítulo III.

Durante o processo de escrita dos padrões de software para o desenvolvimento de OAs ficou claro que sem a devida universalização dos tipos de OAs, seria impossível descrever uma arquitetura que atenda a qualquer tipo de OA a ser implementado. Outro fato que foi constatado neste trabalho é o fato de que a utilização de um framework para a criação de uma Arquitetura Empresarial e conseqüentemente, para o planejamento estratégico de um departamento de desenvolvimento de OA ter um potencial elevado ao alinhar os objetivos do negócio.

Dessa forma, a proposta arquitetural descrita pelo Capítulo IV é uma indicação para a construção de sistemas de baixo custo, factíveis aos prazos e de qualidade, ou seja, a arquitetura é condizente com o objetivo principal desse trabalho. Observando o domínio dos OAs não foi visto a necessidade de criar uma arquitetura específica a OAs voltados para o domínio de Engenharia de Software, como era a proposta inicial deste. Isto fez com que a arquitetura atendesse a uma maior amplitude de contextos.

Fazia parte do escopo do projeto desta dissertação o projeto e desenvolvimento de um OA que utilizasse a arquitetura proposta, também seria avaliada a contribuição dessa arquitetura para validação da hipótese dessa dissertação. Porém, devido a complexidade dos estudos aqui apresentados quanto da implementação, foi decidido utilizar um OA presente na literatura, Aula Virtual (BRAGA et al, 2013).

Na análise do OA Aula Virtual foi constatado que, devido ao fato de o mesmo seguir os passos da metodologia INTERA, este atendeu a diversas células da arquitetura aqui descrita. No entanto, a metodologia não contempla os requisitos referentes à documentação do objeto criado e assim, não cobre diversas visões essenciais para o reuso dos artefatos por ela criado.

Em termos de linhas de trabalhos futuros foram identificadas algumas oportunidades e interesses, descritos a seguir, vinculados a arquitetura proposta:

- **Avaliar usabilidade:** realizar testes junto a usuários finais para verificação do grau de satisfação ao utilizar a arquitetura proposta e assim, mensurar o quão esta

pode auxiliar o processo de criação de um OA ;

- **Incorporação de ontologias:** algumas ontologias tem sido desenvolvidas para facilitar a disseminação de OAs e estas podem ser incorporadas a arquitetura proposta;
- **Linguagem completa de criação de OA:** identificar outros padrões utilizados na criação de OAs e após incorporá-la aos padrões já existentes.

APÊNDICE A – GLOSSÁRIO DE TERMOS

OA Complexo Um OA que pode ter sub-objetos (p.ex., um módulo ou um curso).

Curso Uma coleção de módulos.

Repositórios federados são os repositórios hospedados por outras instituições que você pode acessar remotamente (por exemplo, MERLOT ou aqueles que utilizam o protocolo SRU (Procurar e recuperar por URL)).

Pesquisa Federada Pesquisa de repositórios federados (hospedados por outras instituições que você pode acessar remotamente).

Metadados de Colheita Recuperar metadados a partir de OAs em outro repositório.

Repositório do OA (LOR (ROA)) Uma biblioteca de OAs autônomos (LOs) a partir da qual os usuários acessam os LOs (ou seja, módulos, tópicos ou cursos).

OA (LO) Um recurso ou coleção de objetos auxiliar ao processo de aprendizado (p.ex., um arquivo, um tópico, um curso etc.).

Metadados Uma definição ou descrição de dados, ou os “dados sobre os dados”. O Repositório de Aprendizagem aceita apenas marcas de metadados criados com o uso de IEEE LOM MAP (Metadata Application Profile) (LOM, 2006). Todas as pesquisas usam os campos de metadados IEEE LOM.

Módulo Uma coleção de tópicos e/ou outros módulos.

OER (Open Educational Resources) Os materiais didáticos digitais que estão disponíveis gratuitamente online por meio de licenças abertas.

Unidade organizacional Um departamento, semestre, oferta de curso ou outro item da estrutura organizacional do Ambiente de Aprendizagem.

Permissões Configurações que controlam quais funcionalidades os usuários podem acessar no Repositório de Aprendizagem, conforme atribuídas em Funções e Permissões. É diferente das permissões de confiança (descritas adiante).

QTI Package (especificação Question and Test Interoperability) um forma de especificação padrão usado nos materiais de avaliação. Os pacotes de especificação, por exemplo, itens como perguntas, avaliações e resultados, permitindo que o conteúdo seja produzido e utilizado em sistemas diferentes.

Funções Uma função atribuída conforme definida pelas inscrições do usuário. A função do usuário define suas permissões.

SRU (Search and Retrieve by URL) um protocolo padrão para pesquisas na Internet.

Taxon Um elemento individual em um caminho taxon.

Caminho Taxon Um esquema de classificação hierárquica no qual cada nível seguinte detalha a definição do nível superior.

Tópicos Associações de arquivo simples (por exemplo, um arquivo PowerPoint, HTML, multimídia, um URL etc.). Os tópicos não têm itens secundários e não podem ser descendentes de um curso. Os tópicos podem conter arquivos, mas não os módulos e cursos.

Confiança A relação entre um LOR e uma unidade organizacional. Uma confiança contém três elementos: um LOR, uma unidade organizacional e as permissões de confiança.

Permissões de confiança As funções e permissões atribuídas a cada LOR confiável.

Adobe Photoshop - é o editor de imagens líder no mercado, desenvolvido pela empresa Adobe Systems.

Aplicativos - são programas instalados no computador que têm uma finalidade específica, por exemplo: para editar textos, há o Word, Writer, entre outros; para edição de imagens, há o PaintBrush, CorelDraw, PhotoShop etc.; para controle de estoque, financeiro, cadastro de alunos, existe o Excel, Cal, Access etc.

Arquivo compactado - também conhecido como arquivo zipado, é um arquivo que reúne vários outros arquivos com economia de espaço. Geralmente, um arquivo compactado tem terminação ".ZIP" ou ".RAR".

Arquivos digitais - arquivo em formato eletrônico; somente visualizado por meio de um equipamento específico, como um computador ou um aparelho de DVD, conforme o tipo de arquivo que se pretende visualizar e/ou editar.

Assíncrono – é um adjetivo que, dentro do âmbito da Informática na Educação, refere-se ao tipo de comunicação em que as pessoas interagem em horários e locais diferentes. Exemplos de ferramentas de comunicação assíncrona: e-mail, fóruns e listas de discussão. Quando as pessoas conversam ao mesmo tempo ocorre a comunicação síncrona. Exemplos de ferramentas de comunicação síncrona: Bate Papo e MSN.

Áudio-conferência – conferência por meio digital realizada sem o recurso visual.

Autonomia - capacidade de realizar tarefas de modo independente, sem que alguém tenha que dizer o que, quando e como fazer.

Autoria – condição de autor, daquele que escreve.

Banco de dados - local virtual onde estão agrupados, de forma organizada, dados/registros/informações sobre um mesmo tema/assunto.

Blog - é como um site. Contudo, por sua facilidade de atualização e por ser gratuito, é muito utilizado por pessoas físicas, as quais fazem dele um diário online ou um veículo para se expressar sobre determinada área ou assunto. Blogs contém textos, imagens, links, indicações de outros blogs etc. Também empresas possuem blogs, o objetivo, no entanto, é se aproximar do cliente e formar mais um canal de comunicação. Blogs também podem ter uma aplicação pedagógica, para tanto sua construção e atualizações devem ter fins educacionais.

BrOffice - pacote de aplicativos com licenciamento livre (ao contrário do Pacote Microsoft Office), disponível para diversas plataformas: Windows, Mac OS X, Linux, Solaris.

Busca avançada - é uma pesquisa mais refinada que realiza combinação entre diferentes campos em um formulário com o objetivo de encontrar temas específicos.

Capturar (imagens) - introduzir dados (de imagem) no computador. Copiar uma imagem e depois salvá-la em um arquivo é o mesmo que capturar uma imagem.

Colaboração - ainda que muitos autores defendam que colaboração e cooperação são sinônimos, colaboração teria mais a ver com contribuição, ou seja, cada um faz a sua parte.

Censo Demográfico - é uma pesquisa realizada com a população do país com o fim de recolher informações sobre a mesma. No Brasil o Censo é desenvolvido e aplicado pelo IBGE - Instituto Brasileiro de Geografia e Estatística.

Chat - ferramenta que permite conversa entre duas ou mais pessoas, por meio do computador, em tempo real. Esta conversa acontece por meio de palavras digitadas que aparecem na tela da(s) pessoa(s) com quem está se falando.

Contextualizar - é aproximar um fato à realidade dos sujeitos envolvidos. Pedagogicamente, contextualizar significa abordar um assunto com elementos conhecidos, ou seja, trabalhar com a realidade do aluno, com aquilo que ele conhece.

Cooperação - ainda que muitos autores defendam que colaboração e cooperação são sinônimos, a cooperação prevê maior interação entre as partes, resultando em um produto/serviço mais elaborado. Neste sentido, a cooperação envolveria colaboração, mas a colaboração não envolveria cooperação.

Debian - o projeto Debian, é uma organização voluntária de desenvolvedores Linux. Conhecido principalmente por seu sistema de gestão de pacotes e repositórios.

Design - é a parte visual (estética) de um recurso.

Designer - aquele que faz o design.

Diagrama - representação visual de conceitos; esquema.

Dispositivos móveis - é um equipamento que pode funcionar como um computador de bolso habitualmente equipado com uma pequena tela e um teclado. Os mais utilizados são: Smartphone, PDA, Telefones Celulares, Televisão portátil e demais aparelhos para acessar a Internet.

Download - a tradução é descarregar ou baixar. Em outras palavras, significa salvar/arquivar no computador dados retirados da rede de computadores ou de um software.

Driver – programa que faz a comunicação entre o componente (Hardware) e o Sistema Operacional em uso no computador.

DVD – abreviatura de Digital Vídeo Disc ou Digital Versatile Disc, cujas traduções são Disco Digital de Vídeo ou Disco Digital Versátil. Possui maior capacidade de armazenagem que o CD (Compact Disc / Disco Compacto).

Elemento gráfico – representação visual de um elemento, como um círculo, um retângulo etc.

Encarte - nesse contexto trata-se de anúncio publicitário que vem encartado (anexado) em jornais ou revistas

Estabilizador - é um equipamento que ajuda a proteger seu computador das oscilações de tensão que podem ocorrer na rede elétrica. Como o próprio nome diz, ele "estabiliza" a corrente elétrica evitando danos em seu computador. No caso de uma queda repentina de energia o estabilizador não mantém seu computador ligado, podendo ocorrer a perda de dados caso você esteja com algum trabalho em andamento. Vários dispositivos são ligados nele, como: CPU, monitor, caixa de som e impressora. O estabilizador é ligado diretamente na rede elétrica, por isso antes de ligar veja se ele está na voltagem (110v ou 220v) correta.

Exclusão digital – condição em que as pessoas não dominam os conhecimentos necessários para entender e acompanhar as evoluções tecnológicas, ficando, por isso, muitas vezes à margem da sociedade. A Exclusão digital, por essa razão, pode contribuir para a manutenção e ampliação das desigualdades.

Extensão – são as letras que ficam após o nome do arquivo e identificam o tipo de programa em que ele foi criado.

Ferramentas de autoria - são recursos amigáveis para que pessoas com pouco conhecimento ou não programadores, possam desenvolver com rapidez, amigabilidade e independente de tempo, lugar ou situação física, um determinado conteúdo ou programa.

Formatos de entrada e saída de vídeo – formato é a organização de dados em um arquivo, sendo identificado por uma extensão. Os formatos de entrada e saída de vídeo possuem extensões de arquivos de vídeo.

Folder - é um impresso publicitário com dobras.

GIF – sigla para *Graphic Interchange Format (Formato de Intercâmbio Gráfico)*. Trata-se de um arquivo de imagem bastante utilizado na Internet devido ao seu tamanho compacto. No entanto, possui paleta limitada de cores. O termo pode ainda referir-se ao "Gif animado", que são as animações formadas por várias imagens GIF compactadas em uma só. Este tipo de gif é bastante utilizado para enfeitar páginas pessoais, blogs etc.

Gnome - ambiente gráfico que privilegia aspectos como a usabilidade, acessibilidade e internacionalização dos sistemas operacionais que o utilizam como interface. Menos intuitivo que o **KDE**, porém mais adaptável.

GNU/Linux - sistema operacional, responsável pela interface homem/máquina. Mantido por comunidades internacionais, atualmente interdisciplinares, de profissionais ligados à informática, educação, design gráfico entre outros. Esse sistema foi concebido por Linus Trovalds em 1991 e desde então vêm sendo aperfeiçoado.

Gráfico - representação visual de dados, geralmente numéricos.

Hardware - é a parte física do computador, são as peças e equipamentos que fazem o computador funcionar.

Hipertexto – documento que contém links, os quais remetem a outras informações ou elementos, como páginas na Internet, recursos, aplicativos, vídeos, imagens, sons.

HTML - é a linguagem utilizada para construir páginas web.

Humanoide – é o ser ou objeto no qual são colocadas características humanas. O estilo humanoide é bastante utilizado nos desenhos animados, quadrinhos e filmes de ficção científica.

Imagem 3D - imagem que apresenta três dimensões ou que é criada com o objetivo de proporcionar a ilusão de que têm três dimensões (altura, largura e profundidade). As imagens 3D tornam-se, portanto, mais semelhantes aos objetos originais.

Imagens Vetoriais – baseadas em vetores matemáticos, são imagens geradas a partir de descrições geométricas de suas formas. Os programas de imagens vetoriais mais conhecidos são o Adobe Illustrator e o Corel Draw.

Inteligência coletiva – conhecimento construído a partir de relações sociais estabelecidas no ciberespaço, com auxílio das tecnologias da informação e comunicação da web, reconhecendo

o enriquecimento mútuo das pessoas. O saber está na sociedade, mas ninguém sabe tudo e todos sabem algo.

Interativo – conceito associado às tecnologias de informação e comunicação (ver "Tecnologias de Informação e Comunicação"), no qual o usuário interage com o conteúdo em tempo real.

Interface - têm diferentes significados, mas na informática geralmente se refere ao layout de uma página, na qual os diferentes elementos permitem que o usuário interaja com os aplicativos, recursos, ferramentas disponibilizados.

Interoperabilidade - capacidade de interação e troca de informações, entre dois ou mais sistemas (diversas tecnologias de informação e comunicação), ajustadas a procedimentos definidos, buscando alcançar os objetivos esperados.

Intuitivo – uma ferramenta intuitiva é aquela em que, observando os ícones e a disposição dos materiais, compreende-se facilmente sua utilização.

KDE - ambiente gráfico responsável por fornecer uma interface organizada e consistente facilitando a interação entre o usuário e o computador. Seu ponto forte é a barra de tarefas, que pode ser utilizada por qualquer usuário de forma intuitiva.

Kubuntu – distribuição Linux do projeto Ubuntu com interface gráfica KDE. Essa interface é mais simples e permite que qualquer pessoa possa utilizá-la sem maiores dificuldades.

Layout - esboço ou rascunho. Projeto ou planejamento. Representa ideias iniciais acerca da distribuição de determinados elementos em uma página (digital ou não), como textos e imagens.

Linguagem de programação – método que indica ao computador um número indeterminado de operações que ele deve executar, em outras palavras, é um método para passar instruções para o computador.

Link - é uma palavra ou imagem que remete a outra informação ou elemento. São os links que tornam um documento hipertextual.

Logo - linguagem de programação simples, voltada para crianças. Idealizada por Wally Feurzeig e Seymour Papert em 1967, faz parte de jogos como SuperLogo e MegaLogo, ou ainda o KTurtle.

Mapa conceitual – esquema mental estruturado sobre determinado assunto no qual se estabelecem pontos de ligação entre termos e/ou expressões.

Microsoft Office - pacote de aplicativos da empresa Microsoft que é semelhante ao pacote do BrOffice. A última versão, o Office 2007, possui os seguintes programas: Word (editor de

texto), Excel (editor de planilhas), PowerPoint (apresentações), Access (banco de dados), Outlook (visualizador de e-mail e organizador pessoal), InfoPath (formulários de dados em XML), Publisher (editoração e diagramação eletrônica), e SharePoint Workspace (ferramenta que permite o uso colaborativo de documentos).

Material didático - todo material que serve de apoio/recurso para o processo de ensino e aprendizagem. O êxito no uso do material dependerá da intencionalidade e articulação com a prática pedagógica.

Mídia – meio de comunicação que permite difusão da informação, tal como o vídeo, o computador, o áudio, o livro, entre outros.

Multimídia – é a combinação de diferentes meios de comunicação, como áudio, vídeo, texto, imagem, animação.

Navegar - percorrer páginas na web.

Navegador web – também conhecido simplesmente como Navegador ou ainda como web browser ou somente browser. Trata-se de um programa que tem por fim exibir ao usuário um determinado conteúdo da web (página da web).

Online - A tradução seria "estar em linha". Significa estar ligado à rede ou a outro(s) computador(es).

Página HTML - é um conjunto de comandos e textos que determinam a apresentação dos itens que devem constar na página web (Ver "HTML").

Palavra-chave - é uma palavra que resume o significado de um documento ou texto. É bastante utilizada em sistemas de busca.

Pendrive - é um dispositivo para armazenamento, uma memória, também conhecido como "chaveiro de memória" ou "disco removível".

Planilha - também conhecida como Planilha Eletrônica é um aplicativo no qual tabelas são utilizadas com o fim de realizar cálculos ou apresentar dados. Exemplos de Planilhas são o Calc e o Excel.

Podcast – arquivo de áudio digital publicado na Internet. Pode ser em formato mp3 ou AAC.

Portal – site na Internet que funciona como centralizador e distribuidor de conteúdo para outros sites ou subsites. Exemplos: lojas online; serviços de “Bank-Line” e portais que unem diferentes ferramentas, como Google + Google Docs + Google Agenda.

Postar – sinônimo de escrever (uma mensagem, resposta) em ambientes e/ou ferramentas virtuais.

Prática pedagógica – atividade pedagógica planejada e colocada em ação.

Print screen – trata-se de uma tecla do computador. Usando-se o Sistema Operacional Windows, a pressão sobre a tecla permite capturar (copiar) a imagem que aparece na tela.

Projeto Político Pedagógico – documento que expressa a reflexão da comunidade escolar, portanto, participativo. Visa atender às diretrizes do sistema nacional de Educação, mas também responder às questões que permeiam docentes, discentes e os pais dos alunos. Trata-se de um instrumento teórico-metodológico que reflete a identidade, a realidade e a ação educativa da escola.

Processamento - em âmbito tecnológico, processamento significa transformar os dados fornecidos (números, caracteres, imagens, etc.) em informação a fim de gerar um resultado.

Projeto interdisciplinar - projeto educacional no qual determinado conteúdo ou tema é trabalhado de forma integrada por diferentes disciplinas

Projeto político pedagógico – documento que expressa a reflexão da comunidade escolar, portanto, participativo. Visa atender às diretrizes do Sistema Nacional de Educação, mas também responder às questões que permeiam docentes, discentes e os pais dos alunos. Trata-se de um instrumento teórico-metodológico que reflete a identidade, a realidade e a ação educativa da escola.

Projektor multimídia - também conhecido como Data Show, é um aparelho que amplia imagens (da tela do computador ou de outro dispositivo) em uma parede ou telão.

Quadro branco – é uma tela/espaco branco que pode receber intervenções, sejam elas de caráter textual, desenho ou imagem. Pode ou não ser permitido que todos que estão tendo acesso ao quadro branco também possam editá-lo, isso dependerá dos objetivos do professor.

Quiz – conjunto de perguntas e respostas sobre um assunto geral ou específico.

Recursos informáticos – refere-se aos materiais relacionados à informática, tanto hardware quanto software.

Recursos tecnológicos – no contexto da Informática na Educação é sinônimo de recursos informáticos. Refere-se aos materiais relacionados à informática, tanto hardware quanto software.

Rede - computadores conectados. Exemplos: Computadores em uma empresa podem estar ligados em rede e trocar informações, dados, e-mails.

Repositórios de OAs - são bancos de dados sobre OAs.

Reutilização - é a possibilidade de empregar materiais (digitais ou não digitais), por várias vezes, desempenhando a mesma função ou não, em contextos diferenciados, economizando tempo e possibilitando melhores resultados.

Scanner – trata-se de um periférico de entrada, ou seja, de um acessório que pode ser conectado ao computador. Sua função é digitalizar. Em outras palavras, ele copia imagens, fotos e textos e os transforma em arquivos de imagem, que podem ser salvos no computador.

Simulação - é a recriação de fatos do mundo real. No âmbito da educação tem a intenção de proporcionar ao aluno situações em que ele possa testar diferentes ações, construindo hipóteses. OAs em formato de simulação, por exemplo, podem recriar situações cotidianas, como trabalhar em uma empresa, administrar uma cidade, socorrer feridos, etc.

Sistema de busca - é uma ferramenta de pesquisa para auxiliar o usuário a encontrar determinado material/conteúdo. O sistema pode solicitar a digitação de termo(s) para a busca, como uma ou mais palavras-chave (busca simples) ou combinar diferentes campos (busca avançada). Exemplos de sites que trabalham com sistema de busca: Google e Alta Vista.

Sistema Operacional - é o gerenciador dos recursos instalados no computador e o programa que oferece uma interface entre o computador e o usuário. O Linux e o Windows são exemplos de sistemas operacionais.

Site – conjunto de páginas web. Exemplo: Site do MEC (www.mec.gov.br). A página inicial é uma página web, que dá acesso a várias outras páginas web.

Software – neste contexto significa programa de computador. Ou seja, uma sequência de instruções as quais são interpretadas e executadas por um processador.

Software livre – é o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição. A forma usual de um software ser distribuído livremente é sendo acompanhado por uma licença de software livre (como a GPL ou a BSD) e com a disponibilização do seu código-fonte.

Stop-motion – técnica de animação bastante utilizada em desenhos animados no cinema.

Tecnologias da Informação e Comunicação (TIC's) – Conjunto de recursos tecnológicos, geralmente utilizados de forma integrada. Hoje em dia as TIC's são utilizadas em quase todas as áreas, inclusive na educação, contribuindo para a efetivação de processos de ensino e aprendizagem e possibilitando a modalidade a distância. Exemplos de TIC's: Internet, webcam, telefone celular, rádio.

Tutorial – manual, eletrônico ou não, que indica como manusear algum tipo de recurso.

Upload – exportar um arquivo eletrônico para a web. Quando se anexa um arquivo a um e-mail, faz-se o upload do arquivo.

Web rádio – é o rádio via Internet ou rádio online, o qual pode transmitir programas gravados ou ao vivo.

APÊNDICE B – MODELO DE DOCUMENTO DE SOLICITAÇÃO DE OBJETO DE APRENDIZAGEM.

Solicitação do Objeto de Aprendizagem

Data da solicitação: ____/____/____

Professor Solicitante: _____

Título

Sugerido: _____

Área de Concentração: _____

Ano/Série: _____

Descrição da
Atividade: _____

O que o aplicativo proporciona aos alunos: _____

Tipo de Atividade: () interativa () sem interação

Aplicativos relacionados:

Material Didático Relacionado:

Nome: _____

Autor:

Volume: _____ Série: _____

Descrição da Atividade _____

ANEXO A MODELO CONTEXTUALIZAÇÃO DE UM OBJETO DE APRENDIZAGEM

Nome do Autor:
Data:

Contextualização do O.A.

1. Caracterização do O.A.	
1.1 Tipo do O.A.:	
1.2 Objetivos pedagógicos que se deseja atingir:	
1.3 Área de conhecimento:	
1.4 Disciplina principal:	
1.5 Ementa em que o OA se encaixa:	
1.6 Tópicos dentro da ementa:	
1.7 Descreva brevemente o O.A.:	
1.8 Público Alvo:	
1.9 Conhecimento prévio do público alvo:	
1.10 Grau de Acessibilidade:	
1.11 Fluência tecnológica:	
1.12 Problema atual:	
1.13 Como o OA pode contribuir para solução do problema:	
2. Reusabilidade do OA	
2.1 Disciplinas que o OA também poderá ser utilizado:	
2.2 Tópicos dentro das disciplinas:	
2.3 Componentes do OA:	
2.4 Problema pedagógico que o O.A. pode solucionar:	
2.5. Como o O.A. pode contribuir para a solução do problema pedagógico:	
3. Cenário de uso do OA	
3.1 Modalidade:	
3.2 Descrição do cenário:	

ANEXO B – MODELO DE DOCUMENTO DE ELICITAÇÃO DE REQUISITOS OA

Documento de Elicitação de Requisitos

Descrição:

Este documento foi gerado a partir do documento de contextualização do sistema. Os requisitos descritos neste documento estão descritos nas seguintes categorias: Didático-Pedagógico, funcionalidade, Interface Usuário, Disponibilidade, Acessibilidade, Acesso, Desempenho, Portabilidade e Licenciamento.

Objetivo:

A finalidade deste documento é especificar os requisitos a serem desenvolvidos no *(OBJETO DE APRENDIZAGEM A SER DESENVOLVIDO)*, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para realização dos testes e homologação do sistema.

Item	Categoria	Descrição
01	Didático-Pedagógico	.
...
02	Funcionalidade	
...
03	Interface com Usuário	.
...
04	Disponibilidade	
05	Acessibilidade	
06	Acesso	
07	Desempenho	
08	Portabilidade	
09	Licenciamento	

ANEXO C – MODELO DOCUMENTO DE TESTES DE OA

TESTES			
NOME DO OA:			
item	O que está sendo avaliado?	Quais os critérios da avaliação?	Resultado*
1	Funcionalidade	Adequação: apresenta as funções que se propõe?	
2		Acurácia: Gera resultados precisos dentro do esperado?	
3		Interoperabilidade: exporta dados para outros sistemas?	
4		Segurança – acesso: possui perfil de acesso diferente? (visão do professor, visão do aluno, etc.)	
5		Segurança – privacidade: o objeto especifica uma política de privacidade e segurança dos dados fornecidos pelos usuários;	
6		Autoria: o responsável pelo objeto está claramente identificado?	
7		Contato: existe um meio de verificar a legitimidade do objeto, como endereço e-mail, através do qual se possa estabelecer contato para mais informações?	
8	Confiabilidade	Maturidade: foi detectado algum tipo de defeito?	
9	Manutenibilidade:	Modificabilidade: É fácil de modificar?	
		O código fonte esta disponível para modificações?	
10	Portabilidade	Adaptabilidade a sistemas operacionais (SO): funciona em vários SO?	
11		Adaptabilidade a ambientes virtuais de aprendizagem: funciona em vários ambientes de EAD?	
12		Facilidade na instalação: é fácil de instalar?	
13	Usabilidade	Inteligibilidade (interface intuitiva): é fácil de aprender a utilizar?	
14		Operacionabilidade: é fácil de memorizar? (não esquece os comandos facilmente)	
15		Ajuda: o sistema de ajuda é suficiente para o entendimento?	
16		Erros: as mensagens de erros são claras? Está claro quem tem a responsabilidade pela precisão da informação apresentada?	
17		Design de apresentação: o design do objeto facilita o aprendizado?	
18	Acessibilidade	Acessibilidade de dispositivos: pode ser acessado por diferentes navegadores e dispositivos?	
19		Acessibilidade de pessoas: possui versão adaptada para diferentes tipos de usuários (deficientes visuais, etc.). Narração para cegos, letras grandes para idosos, é acessível a leitores de tela?	
20	Confiabilidade pedagógica	Atualidade: o conteúdo está atualizado?	
21		Corretude: há erros de digitação, grafia e gramática e outras inconsistências?	
22		Motivação: você acredita que o OA é interessante e desafiador para o aluno?	
23		Percepção do objetivo: o aluno consegue perceber facilmente o objetivo do aprendizado?	
24		Feedback: a ferramenta fornece feedback suficiente para o aprendizado do aluno?	
25		Adequado ao propósito: o conteúdo do objeto está de acordo com o propósito pedagógico?	
26		Linguagem: o conteúdo é escrito em estilo de linguagem clara e consistente que está de acordo com o público-alvo?	
27		Coerência: o objeto apresenta conteúdo contextualizado e coerente com os objetivos pedagógicos específicos da área e do nível de ensino?	
28		Alto grau de interatividade: o objeto oferece grau de interatividade alto para o aluno, permitindo que ele interfira bastante na resolução do problema?	
29		Adequação ao tempo: o objeto apresenta uma carga de conteúdo didaticamente adequada para o tempo previsto no uso do objeto?	

REFERÊNCIAS BIBLIOGRÁFICA

- Alexander, C. et al. (1977). *A Pattern Language*. Oxford University Press, N.Y.
- Allen, R. & Garlan, D. (1997) A formal basis for architectural connection. *ACM Transactions on Software Engineering and Methodology*.
- Appleton, B. (1997). *Patterns and Software: Essential Concepts and Terminology*. Disponível em: <<http://www.sci.brooklyn.cuny.edu/~sklar/teaching/s08/cis20.2/papers/appleton-patterns-intro.pdf>> Acessado em: 30 Mai 2014.
- ADL. Sharable Content Object Reference Model (SCORM) 2004 4rd Edition. 2009. Disponível em: <http://www.adl.net>. Acesso em: 25 Nov 2013.
- ARIADNE: Alliance of Remote Instructional Authoring and Distribution Network for Europe. Disponível em: <<http://www.ariadne-eu.org/index.php>>. Acesso em: 12 jan. 2008.
- Bass, L.; Clements, P; Kazman, R.. (2003). *Software Architecture in Practice*. 2;ed
- BIOE Banco Internacional de Objetos Educacionais (2014) disponível em <<http://objetoseducacionais2.mec.gov.br/retrievefile/manual>>
- Borges, K.C.A.D.; Tronto, I.F.B; Braga, J.L.; Guerra, E.M. (2014) Padrões para criação de objetos de aprendizagem. X Sugar Loaf Plop. Disponível em: <http://hillside.net/sugarloafplop/papers/7.1.pdf>. Acessado em: 10 Jan 2015.
- Braga, J.; Dotta, S. C.; Pimentel, E.; Stransky, B. (2012). Desafios para o desenvolvimento de objetos de aprendizagem reutilizáveis e de qualidade. *Desafie-Workshop de desafios da computação aplicados à Educação*. Curitiba.
- Braga, J. C. et al. (2013) Validando a metodologia INTERA no Desenvolvimento de um objeto de aprendizagem do tipo aula virtual. In: *Anais do X Congresso Brasileiro de Ensino Superior a Distância*. Belém, UFPA/Unired. Disponível em: <http://www.aedi.ufpa.br/esud/trabalhos/poster/A>. Acessado em: 20 Mar 2015.
- Braga, J. C. (Org.); et al. (2014) *Objetos de Aprendizagem Volume I - Introdução e Fundamentos*. 1. ed. Santo André: Editora da UFABC.
- Braga, J. C. (2015) *Objetos de Aprendizagem Volume II - Metodologia de Desenvolvimento*. 1. ed. Santo André: Editora da UFABC .
- Brasil. (1996) Secretaria de Educação Fundamental. *Parâmetros curriculares nacionais : introdução aos parâmetros curriculares nacionais / Secretaria de Educação Fundamental*. – Brasília : MEC/SEF, 1997. 126p. Disponível em: <portal.mec.gov.br/seb/arquivos/pdf/livro01.pdf> Acessado em: 10 nov.2014.
- Brasil. (1998) Secretaria de Educação Fundamental. *Parâmetros curriculares nacionais : terceiro e quarto ciclos do ensino fundamental: introdução aos parâmetros curriculares nacionais / Secretaria de Educação Fundamental*. – Brasília : MEC/SEF, 1998. 174 p. Disponível em: <<http://portal.mec.gov.br>>. Acessado em 10 nov.2014.

- Brasil. (2000) Secretaria de Educação Fundamental. Parâmetros curriculares nacionais : Ensino Medio: introdução aos parâmetros curriculares nacionais / Secretaria de Educação Fundamental. – Brasília : MEC/SEF, 2000. Disponível em: < <http://portal.mec.gov.br>>. Acessado em 10 nov.2014
- Buschmann, F., et. al. (1996) *Pattern-oriented Software Architecture: On Patterns and Pattern Language*. Vol. 1. John Wiley & Sons.
- Casewise, (2014). Disponível em:<<http://www.casewise.com/products/enterprise-architecture-frameworks>> Acessado em: 12 set 2014.
- Castillo, S. & Ayala, G., (2008). ARMOLEO: Architecture for Mobile Learning Objects, Electronics, Communications and Computers. CONIELECOMP 2008, 18th International Conference on, vol., no., pp.53,58, 3-5 March 2008 doi: 10.1109/CONIELECOMP.2008.12
- CESTA. Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem. (2008). Disponível em < <http://www.cinted.ufrgs.br/CESTA/CESTACore.html>>. Acessado em: 20 Jun 2014.
- Cetic (2014) Disponível em < <http://www.cetic.br/pesquisa/domicilios/indicadores/>>/ Acessado em: 10 Jan 2015.
- Coglianese, L. and Szymanski, R. (1993) DSSA-ADAGE: An Environment for Architecture-based Avionics development. In Proceedings of AGARD'93.
- DCMI Dublin Core metadata element set, version 1.1: reference description. Disponível em: < <http://dublincore.org/documents/dces/>> Acesso em: 6 jun.2014.
- Evans, E. (2004), *Domain Driven Design*. Reading. MA: Addison-Wesley.
- Ferlin, J., Kemezinski, A., Murakami, E., Hounsell, M. S. (2010) “Metadados Essenciais: Uma metodologia para Catalogação de Objetos de Aprendizagem no Repositório Digital ROAI”. In: Anais do XVI Workshop Sobre Informática na Escola WIE 2010. Belo Horizonte–MG.
- Fowler, M. (2011). *Patterns of enterprise application architecture*. Addison-Wesley Longman Publishing Co., Inc.
- Fioravanti, M.L.; Nakagawa, E.Y.; Barbosa, E.F. (2010) EducAR: Uma Arquitetura de Referência para Ambientes Educacionais. Anais SBIE, Disponível em:< <http://www.br-ie.org/pub/index.php/sbie/article/view/1489>> Acessado em: 11 Ago.2014.
- Gamma, E., et al. (1995), *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading. MA: Addison-Wesley
- Garlan, D.; SHAW, M. (1994). *An introduction to software architecture*.
- Garlan, D., Allen, R and Ockerbloom, J. (1994) Exploiting style in architectural design environments. In Proc. Of SIGSOFT'94: The second ACM SIGSOFT Symposium on the Foundations of Software Engineering, Pages 170-185

- Garlan, D., Monroe, R., & Wile, D. (1997). ACME: An architecture description language. In Proceedings of CASCON'97.
- Gerber, A.; Van Der Merwe, A.; Bayes, K. (2013). An investigation into UML case tool support for the Zachman Framework. In: Enterprise Systems Conference (ES), IEEE, 2013. p. 1-9.
- ISO. ISO/IEC 42010 (2011) Systems and software engineering - Recommended practice for architectural description of software-intensive systems.
- IMS GLC (2007) "IMS Content Packaging Specification Primer", http://www.imsglobal.org/content/packaging/cpv1p2pd2/imscp_primerv1p2pd2.html.
- Lankhorst, M. (2012). Enterprise Architecture at Work: Modeling, Communication and Analysis. 3rd. ed. [S.l.]: Springer. 364 p.
- LOM - International Organization for Standardization. (2006) Working Draft(WD2) for ISO/IEC 19788- 2 – Metadata for Learning Resources – Part 2: data elements. Genève.
- Luckham, D. C., et al. (1995). Specifications and analysis of system architecture using Rapide. IEEE Transactions on Software Engineering, 21(4):336-355.
- Mackenzie, C., et al. (2006). Reference Model for Service Oriented Architecture 1.0. OASIS Standans. Disponível em: <<http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>>. Acessado em: 18 set. 2014.
- Magee, J., et al. (1995). Specifying distributed software architectures. In Proc. Of the Fifth European Software Engineering Conference, ESEC'95.
- Malveau, R., & Mowbray, T. J. (2003). Software Architect Bootcamp. Prentice Hall Professional Technical Reference.
- Medvidovic, N., Oreizy, P., Robbins, J. E. & Taylor, R. N. (1996) Using object-oriented typing to support architectural design in the C2 style. In SIGSOFT'96: Proceedings of the 4Th ACM symposium on the Foundations of Software Engineering.
- Mendes, R.M.; Souza, V.I.; Caregnato, S.E. (2004). A propriedade intelectual na elaboração de objetos de aprendizagem. Encontro Nacional de Ciência da Informação, 5. Editora da Universidade Federal da Bahia Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/548/000502901.pdf?sequence=1>>. Acesso em: NOV 2014.
- MOODLE "Modular Object-Oriented Dynamic Learning Environment", (2014) disponível em: <www.moodle.org> . Acessado em: 15 Jun 2014.
- Mohan, P. (2005). Learning object repositories. In Informing Science and IT Education Joint Conference
- Moriconi, M., Qian, X. & Riemenschneider, R. (1995). Correct architecture refinement. IEEE Transactions on Software Engineering, Special Issue on Software Architecture, 21(4):356-372.

- Moshiri, S. & Hill, R. (2011) Cloud computing for enterprise architectures. In: . [S.l.]: Springer. cap. Enterprise Architecture Fundamentals, p. 343.
- Pandey, R. K. (2010). ,Architectural Description Languages (ADLs) vs UML: A Review,&rdquo, *SIGSOFT Software Eng. Notes*, vol. 35, pp. 1-5.
- Pereira, C. M. & Sousa, P. (2004). A method to define an Enterprise Architecture using the Zachman Framework. In: Proceedings of the 2004 ACM symposium on Applied computing. p. 1366-1371.
- Premsky, M.(2001). Digital Natives Digital Immigrants. In: PRENSKY, Marc. On the Horizon. NCB University Press, Vol. 9 No. 5.
- Oh, Y., et al. (2007). Extended architecture analysis description language for software product line approach in embedded systems. In Formal Methods and Models for Codesign. MEMOCODE 2007. 5th IEEE/ACM International Conference on (pp. 87-88). IEEE.
- Quadros, T.; Martins, J.S.B. (2005). A prática interdisciplinar em programas de educação a distância num cenário de novas tecnologia da informação e comunicação. In: XVI Simpósio Brasileiro de Informática na educação. Juiz de Fora, Mg, Brasil, (SBIE).
- Rocha, A. & Santos, P. (2010). Introdução ao Framework de Zachman. Apontamentos, Universidade Fernando Pessoa, Porto, Portugal.
- Sommerville, I. .(2011). Engenharia de Software.
- Shaw, M., et al. (1995). Abstractions for software architecture and tools to support them. *IEEE Transactions on Software Engineering*. 21(4):314-335.
- Shaw, M. and Garlan, M. (1996). *Software Architecture: Perspectives on an Emerging Discipline*. Prentice-Hall.
- SCHEER. A. *Architecture of Integrated Information Systems Foundations of Enterprise Modelling*, Gennany, 1992.
- Souza, W. E. de. (2010). m-AULA: ferramenta de autoria de objetos de aprendizagem para dispositivos móveis. Dissertação (Mestrado em Informática) – Pontifícia Universidade Católica de Minas Gerais-PUC Minas, Belo Horizonte
- Talarico Neto, A.; et al. (2006). Cog-learn: uma linguagem de padrões para e -learning. *Revista de Informática na Educação*, Rio de Janeiro, 13 (3). p. 33-50.
- Tarouco, L.M.R. (2014). *Objetos de Aprendizagem Teoria e Prática/ Organizadores Liane Margarida Rockenbach Tarouco, Barbara Gorziza Ávila, Edson Felix dos Santos e Marta Rosecler Bez, Valeria Costa*. Porto Alegre: Evangraf, 2014. 504 páginas: il. CINTED/UFRGS.
- Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2009). *Software architecture: foundations, theory, and practice*. Wiley Publishing.
- The Open Group. (2011).. *TOGAF Version 9.1*. [S.l.]: Van Haren Publishing, 654 p.

- The Open Group. (2013) ArchiMate 2.0 Translation Glossary: English – Brazilian Portuguese.
- Vernadat, F (1996). Enterprise modeling and integration: principles and applications. London, Chapman & Hall.
- Wiley, D.A. (2000) Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy. In: WILEY, D.A. (Ed.). The Instructional Use of Learning Objects: Online Version,. Disponível em: [HTTP://reusability.org/read/chapters/wiley.doc](http://reusability.org/read/chapters/wiley.doc)
- Zachman, J. (1987). A Framework for Information Systems Architecture. IBM Systems Journal, Vol. 26, N° 3, pp. 276-292.
- Zachman, J. (2009). The Zachman Framework™: The Official Concise Definition.. Disponível em: <https://www.zachman.com/about-the-zachman-framework> Acessado em: 10 ago. 2014.
- Zimmermann, B., et al. (2006). Patterns for Tailoring E-learning Materials to Make Them Suited for Changed Requirements. VikingPlop, Hlsingør, Dãnamark.