

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

FERRAMENTA WEB PARA GERENCIAMENTO DE
PRODUÇÃO DE OBJETOS DE APRENDIZAGEM

JOSÉ CARLOS KNAUL

BLUMENAU
2009

2009/2-13

JOSÉ CARLOS KNAUL

**FERRAMENTA WEB PARA GERENCIAMENTO DE
PRODUÇÃO DE OBJETOS DE APRENDIZAGEM**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciência
da Computação — Bacharelado.

Prof. Adilson Vahldick - Orientador

**BLUMENAU
2009**

2009/2-13

FERRAMENTA WEB PARA GERENCIAMENTO DE PRODUÇÃO DE OBJETOS DE APRENDIZAGEM

Por

JOSÉ CARLOS KNAUL

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Adilson Vahldick, MSc, FURB

Membro: _____
Prof. Fabiane Barreto Vavassori Benitti, Doutora – FURB

Membro: _____
Prof. Everaldo Artur Grahl, MEng – FURB

Blumenau, 18 de novembro de 2009

Aos meus pais Nilo e Edla (*in memoriam*),
minha noiva Rafaela e amigos que apoiaram o
desenvolvimento deste trabalho.

AGRADECIMENTOS

A Deus por ter me mostrado a luz nos momentos escuros.

A minha família, especialmente minha irmã Milândia, que me trouxe para esta cidade para poder estudar.

A minha noiva Rafaela Tozzetti, pelo tempo, incentivo e compressão prestados nesta longa caminhada.

Ao meu orientador, Adilson Vahldick, por ter acreditado em mim e na conclusão deste trabalho.

A empresa Mult Soluções, pelo apoio, incentivo e oportunidades a mim dispensadas.

A todos aqueles que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Para que levar a vida tão a sério, se a vida é
uma alucinante aventura da qual jamais
sairemos vivos.

Bob Marley

RESUMO

Este trabalho apresenta a especificação e a construção de uma ferramenta web para o gerenciamento da produção de Objetos de Aprendizagem (OAs). Nesta ferramenta é possível planejar, acompanhar, revisar, distribuir e gerar OAs no formato SCORM, utilizando o envolvimento dos papéis de coordenador, professor conteudista, programador e *designer*. A ferramenta foi desenvolvida em Java utilizando como base o *framework Google Web Toolkit ExtJS* (GWT-Ext), e banco de dados Firebird 2.1.

Palavras-chave: Objetos de aprendizagem. *Design* instrucional. SCORM.

ABSTRACT

This paper presents the specification and construction of a web tool for managing the production of learning objects (LOs). In this tool you can plan, monitor, review, distribute and create LOs in SCORM format, using the involvement of the roles of coordinator, teacher content author, programmer and designer. The tool was developed using Java based of the framework Google Web Toolkit ExtJS (GWT-Ext), and database Firebird 2.1.

Key-words: Learning objects. Instructional design. SCORM.

LISTA DE ILUSTRAÇÕES

Figura 1 - Etapas do processo Sophia.....	13
Figura 2 - Etapas, atividades e papéis envolvidos no Processo Sophia.....	15
Figura 3 - Matriz de <i>design</i> instrucional.....	17
Figura 4 - SCORM como conjunto de especificações.....	21
Figura 5 - Funcionamento RPC.....	22
Figura 6 - Diagrama de implementação de um serviço RPC no GWT.....	23
Figura 7 - Tela de cadastro de OA no Processo OA.....	24
Figura 8 - Tela de cadastro de OA no GROA.....	25
Figura 9 - Diagrama de casos de uso.....	27
Quadro 1 - Caso de uso Cadastrar Usuário.....	28
Quadro 2 - Caso de uso Cadastrar Projeto.....	29
Quadro 3 - Caso de uso Preencher Matriz de Design Instrucional.....	30
Quadro 4 - Caso de uso Cadastrar Roteiro.....	31
Quadro 5 – Caso de uso Assumir Tarefa.....	31
Quadro 6 – Caso de uso Disponibilizar Material.....	32
Quadro 7 – Caso de uso Revisar Tarefas.....	32
Quadro 8 – Caso de uso Disponibilizar Unidade para Revisão.....	33
Quadro 9 – Caso de uso Validar Unidade de Aprendizagem.....	33
Quadro 10 – Caso de uso Verificar Revisões de Unidades.....	34
Quadro 11 – Caso de uso Gerar Arquivo SCORM.....	34
Quadro 12 – Caso de uso Baixar Arquivos do Projeto.....	34
Figura 10 – Diagrama de atividades.....	36
Figura 11 – Diagrama de pacotes da ferramenta.....	37
Figura 12 – Diagrama de classes do pacote cliente.....	38
Figura 13 – Diagrama de classes do pacote visual.....	41
Figura 14 - Diagrama de classe do pacote modelo.....	42
Figura 15 – Diagrama de classe do pacote servidor.....	42
Figura 16 – Diagrama de classe do pacote transferencia.....	43
Figura 17– Diagrama de estados da classe Atividade_Roteiro.....	43
Figura 18 – Diagrama de estados da classe Unidade.....	44

Figura 19 – Diagrama de entidade-relacionamento.....	45
Quadro 1 - Caso de uso Cadastrar Usuário.....	28
Quadro 2 - Caso de uso Cadastrar Projeto.....	29
Quadro 3 - Caso de uso Preencher Matriz de Design Instrucional.....	30
Quadro 4 - Caso de uso Cadastrar Roteiro.....	31
Quadro 5 – Caso de uso Assumir Tarefa	31
Quadro 6 – Caso de uso Disponibilizar Material.....	32
Quadro 7 – Caso de uso Revisar Tarefas.....	32
Quadro 8 – Caso de uso Disponibilizar Unidade para Revisão.....	33
Quadro 9 – Caso de uso Validar Unidade de Aprendizagem.....	33
Quadro 10 – Caso de uso Verificar Revisões de Unidades.....	34
Quadro 11 – Caso de uso Gerar Arquivo SCORM.....	34
Quadro 12 – Caso de uso Baixar Arquivos do Projeto	34
Quadro 13 – Classe GWTService.....	48
Quadro 14 – Classe IsSerializable Atividade.....	49
Quadro 15 – Interface assíncrona GWTServiceAsync	50
Quadro 16 – Chamada assíncrona do método Conexao pela classe MainEntryPoint..	50
Quadro 17 – Método para fazer a conexão com banco de dados na classe GWTServiceImpl	51
Quadro 18 – Classe ServletDownload.....	52
Quadro 19 – Classe ServletUpload.....	53
Quadro 20 – Método para geração do arquivo SCORM	53
Quadro 21 – Comparativo entre as ferramentas	66
Quadro 22 - Dicionário de Dados.....	74

LISTA DE SIGLAS

AJAX - *Asynchronous Javascript And XML*

API – *Application Programming Interface*

DER - Diagrama Entidade Relacionamento

DSC – Departamento de Sistemas e Computação

EA – *Enterprise Architect*

GROA - Gerenciador de Repositório de Objetos de Aprendizagem

GWT-Ext - *Google Web Toolkit ExtJS*

LTSC - *Learning Tecnology Standarts Committee*

OA – Objeto de Aprendizagem

RF – Requisito Funcional

RIVED - Rede Interativa Virtual de Educação

RNF – Requisito Não Funcional

RPC - *Remote Procedure Call*

SCORM - *Sharable Content Object Reference Model*

SB – *StoryBoard*

SGBD - Sistema de Gerenciamento de Banco de Dados

SQL - *Structured Query Language*

UML - *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS DO TRABALHO	14
1.2 ESTRUTURA DO TRABALHO	14
2 FUNDAMENTAÇÃO TEÓRICA.....	15
2.1 PROCESSO SOPHIA	15
2.2 DESIGN INSTRUCIONAL.....	16
2.2.1 Unidades de Aprendizagem	16
2.2.2 Matriz de <i>Design</i> Instrucional.....	17
2.2.3 Roteiros e <i>Storyboards</i>	18
2.2.4 Instrumentos de Avaliação.....	18
2.3 SCORM E CELINE	20
2.4 GWT-EXT.....	22
2.5 TRABALHOS CORRELATOS.....	23
2.5.1 Processo OA.....	23
2.5.2 GROA	25
3 DESENVOLVIMENTO.....	26
3.1 REQUISITOS PRINCIPAIS DO SOFTWARE DESENVOLVIDO	26
3.2 ESPECIFICAÇÃO	27
3.2.1 Diagrama de Casos de Uso	27
3.2.2 Diagrama de Atividades.....	35
3.2.3 Diagrama de Classes	36
3.2.4 Diagrama de Estados.....	43
3.2.5 Diagrama de Entidade-Relacionamento (DER).....	45
3.3 IMPLEMENTAÇÃO	46
3.3.1 Técnicas e ferramentas utilizadas.....	47
3.3.1.1 Implementação da ferramenta.....	47
3.3.2 Operacionalidade da implementação	54
3.4 RESULTADOS E DISCUSSÃO	66
4 CONCLUSÕES.....	67
4.1 EXTENSÕES	68
REFERÊNCIAS BIBLIOGRÁFICAS	69

1 INTRODUÇÃO

A utilização de novas tecnologias aliadas à necessidade de se aprender rapidamente vem tornando o ensino à distância uma ferramenta útil nos dias atuais (BETTIO; MARTINS, 2004). Para o final de 2010, espera-se que 83% das escolas brasileiras possuam computadores ligados a internet (SANTOS, 2008), estimulando cada vez mais a procura de mecanismos computacionais que possam auxiliar no processo de aprendizado eletrônico.

Nesta direção, Objetos de Aprendizagem¹ (OAs) são elementos de uma nova metodologia de ensino e aprendizagem baseado no uso do computador e da Internet (GUILLERMO; TAROUCO; ENDRES, 2005). É uma tecnologia recente que abre caminhos na educação à distância, e como material de apoio à aula presencial tradicional, trazendo inovações e soluções que podem beneficiar a todos os envolvidos no processo de ensino e aprendizagem.

Segundo o *Learning Technology Standarts Committee* (LTSC) (LTSC IEEE, 2007), OA é definido como uma entidade, digital ou não digital, que pode ser usada, re-usada ou referenciada durante o ensino com suporte tecnológico. Podem-se citar os OAs encontrados em *sites* como, por exemplo, a Rede Interativa Virtual de Educação (RIVED), que se apresentam na forma de conteúdos pedagógicos digitais e aplicativos destinados a estimular o raciocínio e a melhorar o aprendizado (RIVED, 2009).

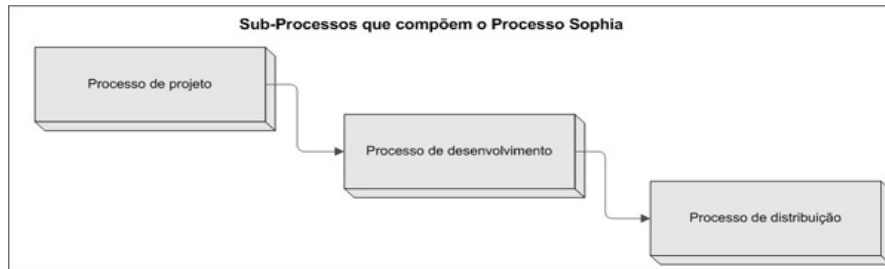
No entanto, segundo Kratz et al. (2007), para promover a reutilização de conteúdos de aprendizagem é necessário promover a sua normalização para que possam funcionar corretamente em qualquer sistema *e-learning*. A normalização promove a reutilização, a portabilidade dos conteúdos criados, a padronização dos processos de criação e a gestão dos conteúdos de aprendizagem.

Pessoa e Benitti (2008) propõem uma normalização no processo de produção de OA batizado de Processo Sophia, composto por três etapas: projeto, desenvolvimento e distribuição (Figura 1). Neste processo existe o envolvimento dos seguintes papéis: coordenador, professor conteudista², *designer*, programador, WebTutoria³ e alunos.

¹ Objetos de Aprendizagem do inglês *Learning Objects*.

² Professor responsável e especialista por fornecer os elementos necessários ao processo de ensino e aprendizagem, que são o conteúdo, os objetivos e a avaliação.

³ WebTutoria é uma equipe de apoio, formada por pessoal qualificado para responder dúvidas dos alunos sobre os conteúdos lecionados na disciplina.



Fonte: Pessoa e Benitti (2008).

Figura 1 - Etapas do processo Sophia

Na etapa de projeto é especificada a estrutura do OA, envolvendo definição de mídias e conteúdos, bem como o planejamento das atividades necessárias para seu desenvolvimento, alocando recursos e definindo cronograma (detalhando atividades, responsáveis e prazos).

A matriz de *design* instrucional é uma ferramenta que pode ser utilizada nesta fase de projeto de construção de OA (FILATRO, 2008). Através dela são especificadas as unidades de aprendizagem, objetivos, papéis, atividades, ferramentas, conteúdos e avaliação. A sua estrutura permite o rastreamento entre os objetivos, atividades e mecanismos de avaliação, facilitando o trabalho de validação.

Na etapa de desenvolvimento é produzido o OA, conforme definido na etapa de projeto, deixando-o pronto para disponibilizá-los aos alunos. É importante observar que esta etapa deve contemplar ao menos uma revisão de todo o projeto por parte do professor conteudista. Além disso, pode-se observar que o processo propõe que a produção do objeto envolva os papéis de programador e *designer*, visando imprimir qualidade técnica e de usabilidade ao objeto.

A etapa de distribuição contempla a distribuição do OA em repositórios, podendo utilizar padrões como SCORM⁴. O processo prevê ainda a avaliação do objeto pela WebTutoria e os alunos, visando obter *feedback* para melhoria contínua.

Considerando estes fatores, foi construída uma ferramenta web para o gerenciamento da produção de OA, concentrando-se: (i) na etapa de projeto; (ii) a atividade de revisão da etapa de desenvolvimento; (iii) a distribuição do objeto no repositório da etapa de distribuição, baseando-se no processo Sophia (PESSOA; BENITTI, 2008) utilizando a matriz de design instrucional (FILATRO, 2008) e gerando o objeto no formato SCORM.

⁴ *Sharable Content Object Reference Model* (SCORM) é uma coleção de padrões e especificações *e-learning* baseado na web.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi desenvolver uma ferramenta web para o gerenciamento da produção de OAs, baseando-se no processo Sophia contemplando a etapa de projeto, a atividade de revisão da etapa de desenvolvimento e a distribuição dos objetos da etapa de distribuição.

Os objetivos específicos do trabalho são:

- a) usar uma matriz de *design* instrucional como base para especificar objetos de aprendizagem;
- b) disponibilizar para o desenvolvedor a gravação dos objetos de aprendizagem no sistema;
- c) possibilitar ao professor conteudista a atividade de revisão do objeto de aprendizagem;
- d) rastrear e apresentar os elementos não atendidos após a revisão;
- e) manter e apresentar o histórico da atividade de revisão e desenvolvimento;
- f) permitir ao coordenador gerar o objeto no formato SCORM.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em três capítulos intitulados respectivamente como: fundamentação teórica, desenvolvimento e conclusões.

O capítulo 2 apresenta os aspectos teóricos estudados para o desenvolvimento da ferramenta. São abordados temas como *design* instrucional, unidades de aprendizagem, matriz de *design* instrucional e o Processo Sophia. Também são relacionados alguns trabalhos correlatos.

No capítulo 3 é descrito como foi realizado o desenvolvimento do trabalho, detalhando os requisitos da ferramenta, a especificação e a implementação. São apresentados os resultados encontrados com a finalização do trabalho.

Por fim, o capítulo 4 traz conclusões deste trabalho, bem como alguns aspectos que ficaram em aberto, servindo de sugestões para futuras extensões.

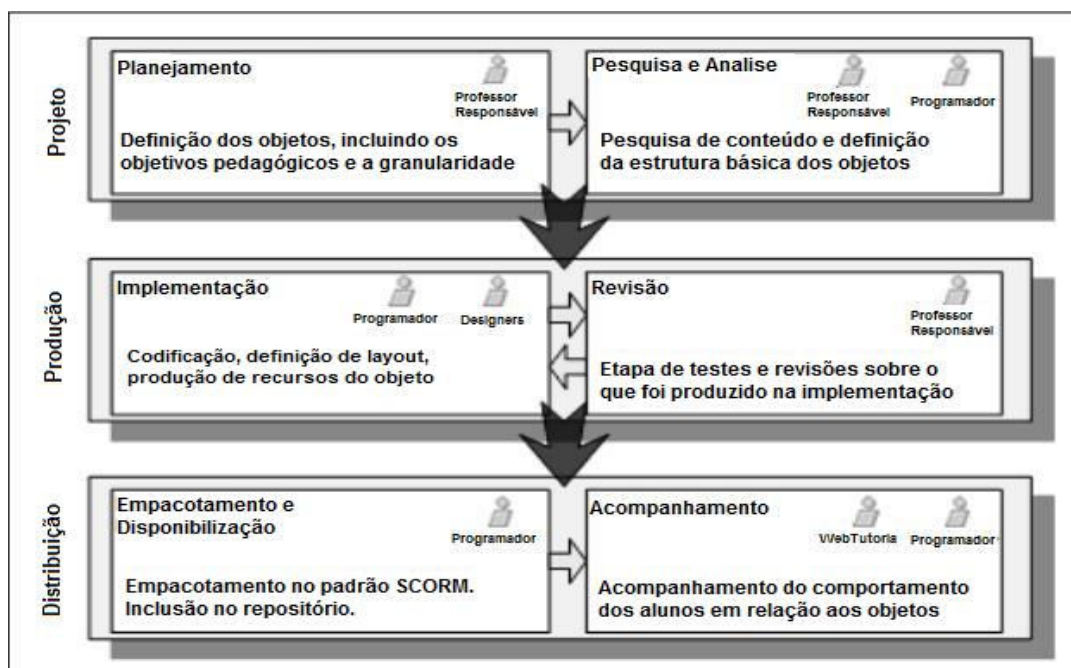
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os principais tópicos teóricos do trabalho. Na seção 2.1 é apresentado o Processo Sophia. Na seção 2.2 são apresentados os conceitos fundamentais de *Design Instrucional* e seus modelos de planejamento. Na seção 2.3 são apresentados os conceitos de SCORM. Na seção 2.4 é apresentado o *Google Web Toolkit ExtJS (GWT-Ext)* e seus meios de funcionamento. Por fim a seção 2.5 apresenta dois trabalhos correlatos ao trabalho proposto.

2.1 PROCESSO SOPHIA

Segundo Pessoa e Benitti (2008), objetivando a padronização dos processos e a qualidade dos objetos de aprendizagem, foi desenvolvido o Processo Sophia.

O processo foi definido e dividido em etapas e atividades, onde os papéis estão distribuídos de acordo com as responsabilidades de cada membro da equipe. A Figura 2 destaca as etapas, atividades e papéis no processo.



Fonte: Pessoa e Benitti (2008).

Figura 2 - Etapas, atividades e papéis envolvidos no processo Sophia

Os papéis foram divididos em: (i) professores, responsáveis pelo conteúdo educativo dos objetos; (ii) programadores, responsáveis pela codificação, empacotamento e distribuição dos objetos; (iii) *designers*, responsáveis pelos *layouts* dos objetos; e (iv) WebTutoria, responsável pelo acompanhamento das dúvidas sobre o conteúdo abordado nos objetos.

2.2 DESIGN INSTRUCIONAL

Filatro (2008, p. 3) coloca o *design* instrucional como a ação intencional e sistemática de ensino que envolve o planejamento, o desenvolvimento, a aplicação de métodos, técnicas, atividades, materiais, eventos e produtos educacionais em situações didáticas específicas, a fim de promover, a partir dos princípios de aprendizagem e instrução conhecidos, a aprendizagem humana. Em outras palavras, define-se *design* instrucional como o processo (conjunto de atividades) de identificar um problema (uma necessidade) de aprendizagem e desenhar, implementar e avaliar uma solução para este problema.

Ainda segundo Filatro (2008, p. 9), *designer* instrucional é o responsável por projetar soluções para problemas educacionais específicos. Os campos de atuação dos *designers* instrucionais são ilimitados, visto que a aprendizagem permeia praticamente todas as atividades humanas, das brincadeiras de criança à certificação profissional, do treinamento motor à formação política, dos rituais religiosos às práticas esportivas. Contudo, deve-se restringir a ação do *designer* instrucional às iniciativas educacionais intencionais, ou seja, àquelas em que pessoas, grupos ou instituições empenham-se de forma deliberada e orientada para formar ou informar pessoas.

A seguir é apresentado o detalhamento do *design* instrucional, com o planejamento das unidades de aprendizagem, a utilização da matriz de *design* instrucional, o uso de roteiros e *storyboards* para a construção de OA e os tipos de instrumentos de avaliação.

2.2.1 Unidades de Aprendizagem

Segundo Filatro (2008, p. 43), unidade de aprendizagem é uma unidade atômica ou elementar que contém os elementos necessários ao processo de ensino/aprendizagem. Pode ser tão extensa quanto o currículo completo de um curso de graduação com quatro anos de

duração ou tão pequena como uma atividade de aprendizagem de 15 minutos.

No aprendizado eletrônico, o *design* de unidades de aprendizagem se dá com base nas seguintes premissas:

- a) uma unidade de aprendizagem visa um ou mais objetivos de aprendizagem (ou resultado esperado);
- a) para alcançar os objetivos, as pessoas assumem papéis no processo ensino/aprendizagem;
- b) cada papel desempenha uma ou mais atividades;
- c) as atividades seguem um fluxo, têm uma duração e são realizadas em um período de tempo determinado;
- d) as atividades são apoiadas por conteúdos e ferramentas;
- e) os conteúdos e ferramentas são organizados em um ambiente;
- f) a avaliação verifica se os objetivos da unidade de aprendizagem foram alcançados.

2.2.2 Matriz de *Design* Instrucional

Filatro (2008, p. 44) descreve que objetivos, papéis, atividades, conteúdos, ferramentas, ambientes e avaliação são elementos básicos do processo educacional e podem ser organizados em uma matriz que permite ter uma visão panorâmica de cada unidade de aprendizagem, como mostra a Figura 3.

	Unidades	Objetivos	Papéis	Atividades	Duração e período	Ferramentas	Conteúdos	Avaliação
1		o que se espera de cada unidade			Respectivamente, carga horária e distribuição no calendário			
2	unidades elementares do processo de ensino/aprendizagem		Quem faz o quê a fim de alcançar os objetivos				Objetos de aprendizagem, URLs e arquivos externos	
3						Serviços usados durante as atividades de aprendizagem e apoio		Mecanismos e critérios para verificar se os objetivos foram atingidos
4								
5								

Fonte: Filatro (2008, p. 45).

Figura 3 - Matriz de *design* instrucional

Por meio da matriz (figura 3), pode-se definir quais atividades serão necessárias para atingir os objetivos, elencar quais conteúdos e ferramentas serão precisos para a realização das atividades; estabelecer como se dará a avaliação do alcance dos objetivos; quais serão os níveis de interação entre os alunos e os conteúdos, as ferramentas, o educador e os outros alunos e que tipo de ambiente virtual será necessário para o desempenho das atividades.

Nela também organizam-se as atividades em um fluxo, que pode ser de livre exploração pelos alunos ou ser orientada pelos eventos instrucionais. A duração e o período das atividades são fatores que também podem ser definidos na elaboração da matriz.

É importante assinalar que a matriz pode ser usada como material de orientação da equipe de *design* e desenvolvimento instrucional, além de ser apresentada como um mapa do curso, na íntegra ou em versão simplificada, dependendo do público-alvo.

2.2.3 Roteiros e *Storyboards*

Segundo Filatro (2008, p. 57), criar um curso para aprendizado eletrônico não é como dar aula, proferir uma palestra ou ministrar um curso presencial. Isso porque não se trata de simplesmente definir um plano de curso, escolher os materiais e as estratégias e apresentar as aulas. O aprendizado eletrônico tem características midiáticas e, por isso mesmo, deve ser pensado com a lógica de produção de mídias.

Na criação de produtos multimídia comerciais, um roteiro escrito (*script*) ou um *Storyboard* (SB) são os meios tradicionais de especificar o conteúdo exato a ser produzido. No aprendizado eletrônico pode-se usar *scripts* ou SBs para especificar detalhadamente os conteúdos de um curso (na forma de textos, imagens e sons), as orientações de atividades propostas e os diálogos das personagens, determinado a sequência em que eles serão exibidos no produto final. Neste contexto, o *designer* instrucional utiliza documentos como *scripts* e SBs para descrever detalhadamente a estrutura e o fluxo da informação, os conteúdos e a interface do produto final.

2.2.4 Instrumentos de Avaliação

Segundo Filatro (2008, p. 130), a avaliação é usada para fazer o julgamento sobre a validade ou o sucesso de pessoas e coisas. É um processo orientado por objetivos e requer

atenção para os resultados e para os processos que conduziram a esses resultados.

Uma das funções essenciais da avaliação é verificar o conhecimento e as competências adquiridas pelos alunos no processo de ensino/aprendizagem. No aprendizado eletrônico, são vários os instrumentos para a avaliação da aprendizagem do aluno, que vão de enquetes rápidas e informais a espaços para armazenamento contínuo da produção individual, da aplicação automatizada de testes a recursos de avaliação cruzada em produções coletivas. A seguir são descritos os principais instrumentos de avaliação para o aprendizado eletrônico propostos por Filatro (2008, p. 132):

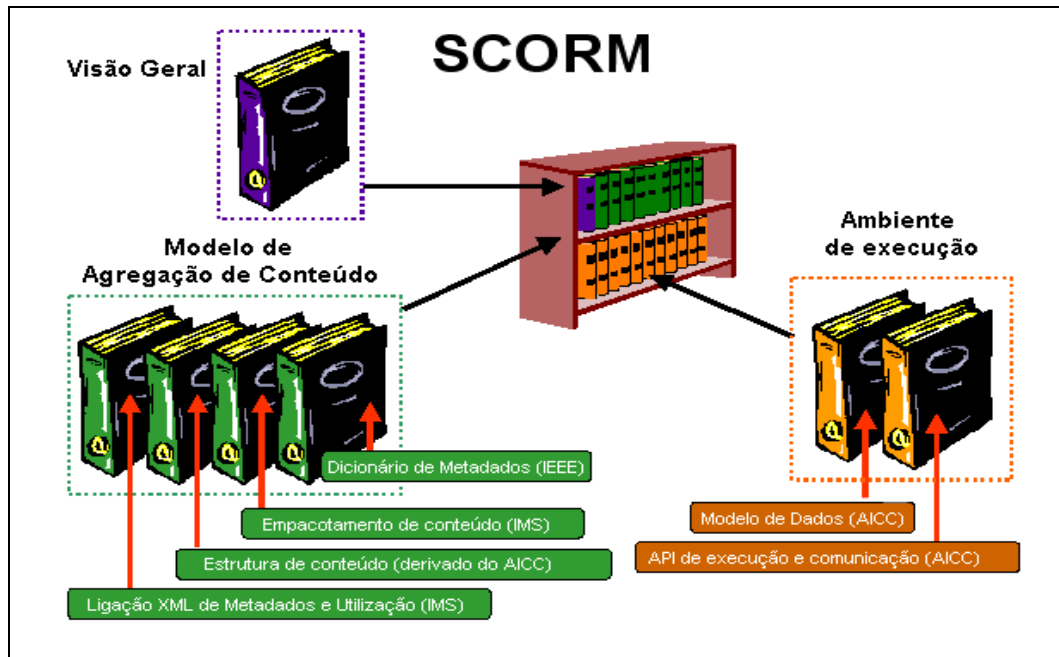
- a) testes de múltipla escolha: consistem em um enunciado com algumas alternativas de respostas, sendo uma delas claramente a resposta correta. Fáceis de administrar e corrigir eletronicamente, esses testes fornecem medidas objetivas de desempenho e permitem ampla cobertura dos conteúdos tratados;
- b) teste de verdadeiro/falso e sim/não: consistem em um enunciado com duas alternativas, sendo apenas uma delas a correta. São fáceis de administrar e corrigir eletronicamente e fornecem medidas objetivas de desempenho, mas favorecem o fator adivinhação – afinal, 50 por cento das respostas estão corretas e 50 por cento estão erradas;
- c) testes de associação ou correspondência: consistem em (1) uma lista de condições, (2) uma lista de respostas e (3) direções para relacionar as condições às respostas. São fáceis de administrar e corrigir eletronicamente, fornecem medidas objetivas de desempenho e permitem ampla cobertura dos conteúdos tratados;
- d) testes de preenchimento de lacunas: consistem em um enunciado que deve ser completado pelo aluno. Eles representam uma opção às questões abertas, minimizam a adivinhação quando comparados às questões de múltipla escolha e verdadeiro/falso e ainda fornecem medidas objetivas de desempenho;
- e) testes de arrastar-e-soltar: constituem em uma combinação dos testes de associação e de preenchimento de lacunas. Aqui, a ação do aluno consiste em arrastar com um *mouse* palavras ou expressões visíveis na tela para preencher lacunas de uma frase ou parágrafo;
- f) *puzzles*: jogo da memória, palavras cruzadas e caça-palavras são variações dos testes de associação e de preenchimento de lacunas que podem ser categorizados como *puzzles* – quebra-cabeças, desafios ou, conforme o dicionário Aurélio, “qualquer problema cuja solução exige trabalho de paciência”;
- g) questões dissertativas: consistem em perguntas abertas que tem como objetivo

avaliar competências de aplicação, análise, síntese e avaliação. Elas podem restringir-se a alguns parágrafos, exigindo respostas altamente focadas, ou dar oportunidade ao aluno de expressar pontos de vista em dissertações mais longas;

- h) questões de resolução de problemas: assim como as questões dissertativas, as questões de resolução de problemas avaliam a competência de aplicações, análise, síntese e avaliação e são relativamente fáceis de construir, se comparadas ao *design* de questões objetivas. No entanto, requer tempo e cuidado extra na correção, já que, na maioria dos casos há múltiplas respostas para solucionar o mesmo problema;
- i) rubricas: consistem na avaliação de aprendizagem complexas. Elas se baseiam em uma escala de múltiplos critérios que examinam de maneira mais descritiva e holística tanto os processos quanto os produtos da aprendizagem;
- j) portfólios: consistem em uma coleção de registros e artefatos acumulados que representam o que um aluno ou um grupo de alunos aprendeu ao longo do tempo;
- k) auto-avaliação: consiste em propor aos alunos, individualmente ou em grupos, testes objetivos e/ou questões dissertativa que devem ser analisados pelos próprios alunos;
- l) monitoramento automático: a maioria dos sistemas de gerenciamento da aprendizagem oferece, de forma não invasiva, dados sobre a frequência de conexão dos alunos e educadores, áreas do ambiente visitadas, mensagens de correio e fórum enviadas e rastreamento de percursos individuais de aprendizagem, entre outros.

2.3 SCORM E CELINE

Segundo Advanced Distributed Learning (2009), SCORM é um conjunto unificado de padrões e especificações para conteúdo, tecnologias e serviços para *e-learning*. Define um Modelo de Agregação de Conteúdo (*Content Aggregation Model*) e um Ambiente de Execução (*Run-Time Environment*) para objetos educacionais baseados na web. A Figura 4 ilustra o SCORM como conjunto de especificações.



Fonte: Advanced Distributed Learning (2009).

Figura 4 - SCORM como conjunto de especificações

Ainda segundo Advanced Distributed Learning (2009), o argumento para a utilização de SCORM no desenvolvimento de conteúdos para *e-learning* pode ser resumido no acrônimo “RAID”, ou seja, reusabilidade, acessibilidade, interoperabilidade e durabilidade. Um dos objetivos do SCORM é proporcionar a independência de plataforma na qual os OA são utilizados, assim como facilitar a migração de cursos entre diferentes ambientes de gerenciamento de aprendizagem que sejam compatíveis com este padrão.

Segundo Vahldick e Raabe (2008), os conteúdos desenvolvidos em conformidade com SCORM são independentes de contexto, ou seja, funcionará em situações variadas, seja inserido em um ambiente de gerenciamento de aprendizagem ou como parte de um curso *on-line* publicado diretamente na web ou ainda em cenário híbrido.

Vahldick (2008) desenvolveu um componente, usando a tecnologia Java para aplicações web, chamado CELINE. Os objetivos principais são permitir que sejam executados pacotes SCORM, assim como interferir durante a interação do estudante com esses pacotes. Esse componente, conectado a aplicação web, intermediará vários recursos necessários para atender a esses objetivos.

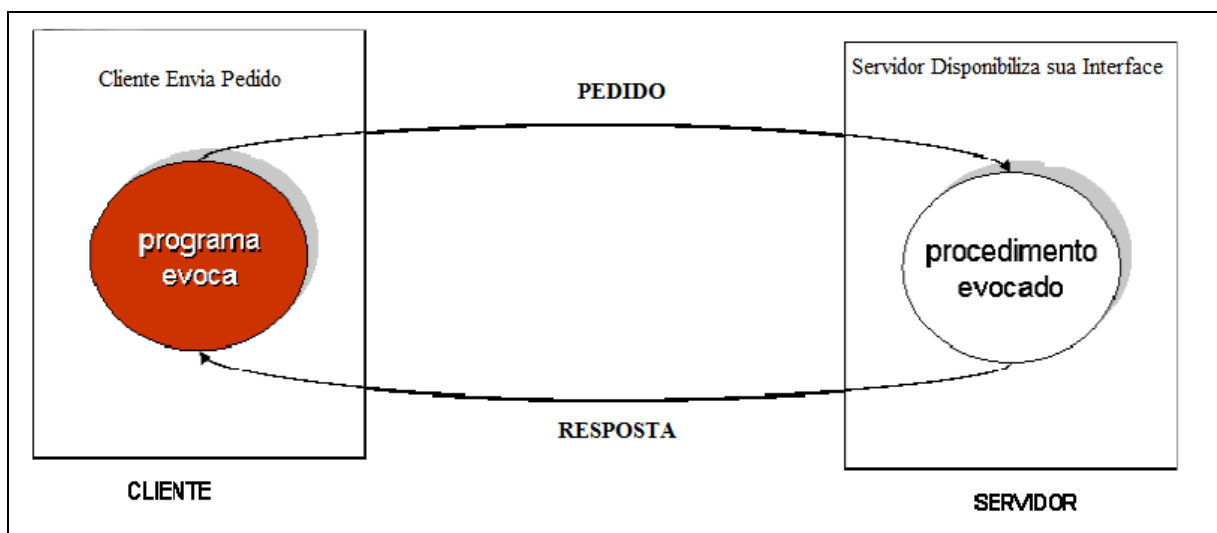
2.4 GWT-EXT

Google Web Toolkit (GWT) é um framework de desenvolvimento de aplicações web em Java utilizando AJAX que remove a maior parte do trabalho do desenvolvedor, pois não é necessário escrever nenhum código *JavaScript*. Toda a aplicação é escrita em Java, que na hora da implantação tem as partes que vão rodar no cliente traduzidas para *JavaScript* para que elas possam ser executadas nos navegadores disponíveis (ARAGÃO, 2008).

Ainda segundo Aragão (2008), ExtJS é uma biblioteca *JavaScript open source* para a construção de aplicações web ricas com interfaces gráficas sofisticadas e com alto grau de interatividade com o usuário.

O framework GWT-Ext consiste na integração das tecnologias GWT e ExtJS, proporcionando uma poderosa biblioteca de *widgets* que provê ricos componentes de interface. Surge como um meio de se criar componentes tão ricos como apresentados pelo ExtJS usando apenas a linguagem Java como proposto pelo GWT.

Segundo Lufér (2008), o GWT-Ext trabalha com a tecnologia *Remote Procedure Call (RPC)*, que é definido como um protocolo para execução remota de *procedures* em computadores ligados em rede, como um modelo de integração Cliente/Servidor. Na Figura 5 pode-se visualizar o funcionamento RPC.

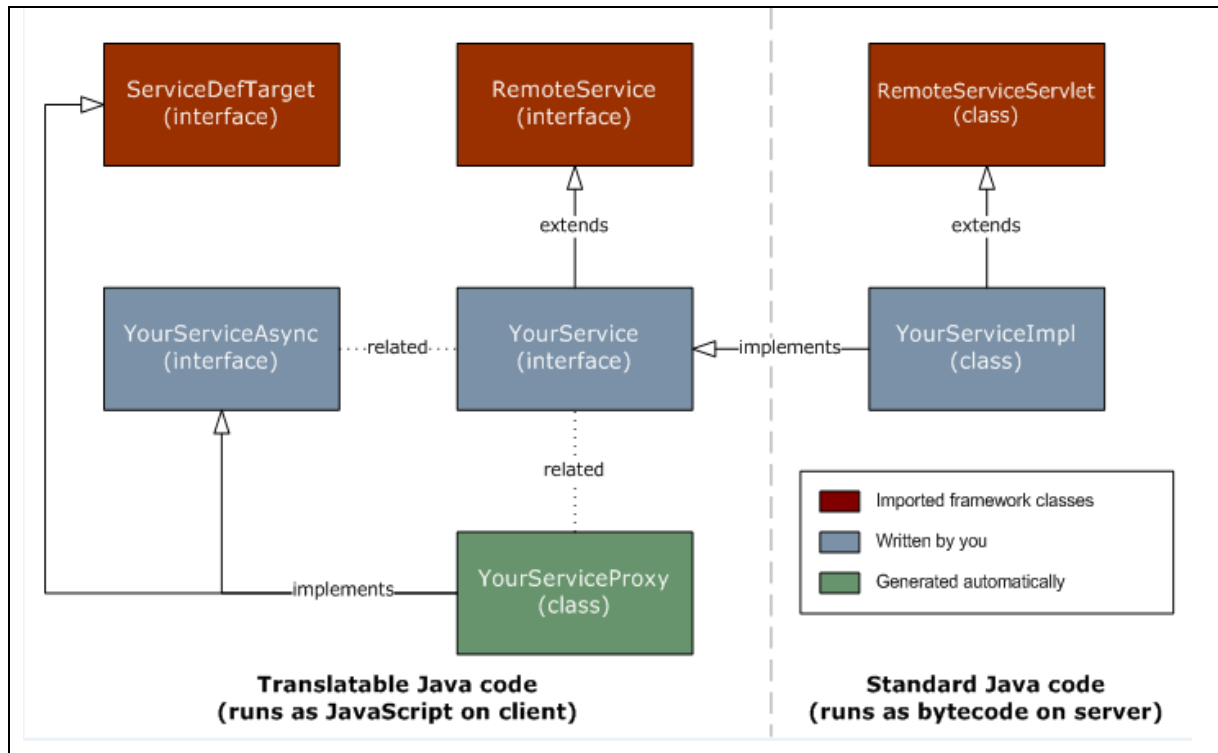


Fonte: Lufér (2008).

Figura 5 - Funcionamento RPC

A aplicação que executa no cliente (navegador) faz uma invocação remota a um serviço implementado no servidor que pode então utilizar toda a expressividade de *Application Programming Interface (APIs)* do Java para retornar os dados para o cliente.

Para criar um serviço a ser invocado pelo cliente, precisa-se inicialmente definir uma *interface* síncrona e uma *interface* assíncrona de serviço. A *interface* assíncrona é necessária porque a invocação ao serviço utilizando AJAX acontece de forma assíncrona. No servidor é necessário definir uma classe que estenda o tipo *servlets* para a *interface* real do serviço, possuindo os métodos que retornarão valor para o cliente. A Figura 6 representa o diagrama de implementação de um serviço RPC no GWT.



Fonte: Lufér (2007).

Figura 6 - Diagrama de implementação de um serviço RPC no GWT

2.5 TRABALHOS CORRELATOS

Existem alguns projetos que desempenham papel semelhante ao proposto neste trabalho. A seguir são apresentados trabalhos correlatos ao tema proposto, como o Processo OA e o GROA.

2.5.1 Processo OA

O Processo OA (Figura 7) é uma ferramenta web, desenvolvida em PHP e ExtJs,

gerenciadora de objetos de aprendizagem (PESSOA; BENITTI, 2008) utilizando como base para o seu desenvolvimento o Processo Sophia. Nesta ferramenta podem-se cadastrar novos objetos de aprendizagem (em diversos formatos), elencar papéis de produção aos construtores do OA (conteudista, coordenador, produtor e revisor) e fazer a avaliação dos mesmos. Também pode-se definir atividades aos papéis referenciando ao seu determinado assunto.

Fonte: Pessoa e Benitti (2008).

Figura 7 - Tela de cadastro de OA no processo OA

As principais características do processo OA são:

- a) utilização simples e objetiva;
- b) interface web;
- c) inclusão dinâmica dos OAs em diversos formatos;
- d) facilidade na estruturação dos OAs;
- e) definição dos papéis aos usuários por OA;
- f) acompanhamento da produção dos OAs;
- g) acompanhamento do histórico das avaliações dos OAs;
- h) fácil customização utilizando recursos de clicar e arrastar.

2.5.2 GROA

O GROA é um Gerenciador de Repositório de Objetos de Aprendizagem (GROA) (Figura 8) e disponibiliza recursos de criação de repositórios capazes de armazenamento, gerenciamento, indexação e estruturação de objetos de aprendizagem, e capazes de operar com serviços da web (MIRANDA, 2004). Este sistema utiliza banco de dados orientado a objetos, integrado a um servidor web. As funcionalidades oferecidas pelo GROA são: cadastro de OA, edição de OA já cadastrados no sistema, anotações (comentários) sobre estes objetos e a busca de OAs.

Fonte: Miranda (2004).

Figura 8 - Tela de cadastro de OA no GROA

Suas principais características são:

- a) interface com a web para fácil acesso a partir de qualquer navegador;
- b) distribuição dos OAs em diversos formatos;
- c) classificação e localização dos OAs através de tópicos (assuntos);
- d) facilidade na recuperação dos OAs;
- e) geração de relatórios com históricos de distribuição, localização e recuperação dos OAs.

3 DESENVOLVIMENTO

Neste capítulo é apresentado o desenvolvimento da ferramenta, com base nos conceitos apresentados no capítulo anterior. São descritos os requisitos, os diagramas da *Unified Modeling Language* (UML) utilizados na especificação da ferramenta, a implementação e os resultados obtidos.

3.1 REQUISITOS PRINCIPAIS DO SOFTWARE DESENVOLVIDO

A partir dos estudos preliminares de algumas ferramentas citadas e analisando os resultados esperados para a construção da ferramenta, identificou-se os seguintes requisitos:

- a) permitir o cadastro de usuários vinculando a seus respectivos papéis (Requisito Funcional - RF);
- b) permitir o cadastro de projetos vinculando a seus respectivos professores conteudistas (RF);
- c) permitir o preenchimento da matriz de *design* instrucional (RF);
- d) permitir a definição dos roteiros (*script*) para cada unidade de aprendizagem (RF);
- e) permitir adicionar tarefas aos programadores e designers de acordo com os roteiros das unidades de aprendizagem (RF);
- f) permitir disponibilizar OA para revisão (RF);
- g) permitir validar OAs de acordo com os objetivos das unidades de aprendizagem (RF);
- h) gerar a partir do OA da unidade um arquivo no formato SCORM (RF);
- i) ser implementado na linguagem de programação Java (Requisito Não Funcional - RNF);
- j) utilizar o banco de dados Firebird 2.1 (RNF);
- k) ser implementado utilizando o ambiente de programação Netbeans (RNF);
- l) Apache TomCat deverá ser usado para a execução do aplicativo (RNF).

3.2 ESPECIFICAÇÃO

A ferramenta foi especificada utilizando a ferramenta Enterprise Architect, com base nos conceitos de orientação a objetos e nos diagramas da UML, gerando como resultado os diagramas de casos de uso, atividades, classes e estados.

3.2.1 Diagrama de Casos de Uso

Foram identificados os seguintes papéis: (i) coordenador, responsável pelos cadastramentos da ferramenta, criação e coordenação dos projetos. Também é responsável pelo empacotamento e distribuição dos OAs via geração do arquivo no formato SCORM; (ii) professor conteudista, responsável pelo conteúdo educativo dos OAs, acompanhando-os pela matriz de *design* instrucional; (iii) designer e programador, responsáveis pela criação e desenvolvimento dos OAs.

O diagrama de casos de uso é ilustrado na Figura 9 e apresenta as ações que os usuários podem executar ao utilizarem a ferramenta, e são detalhados nos quadros 1 a 12.

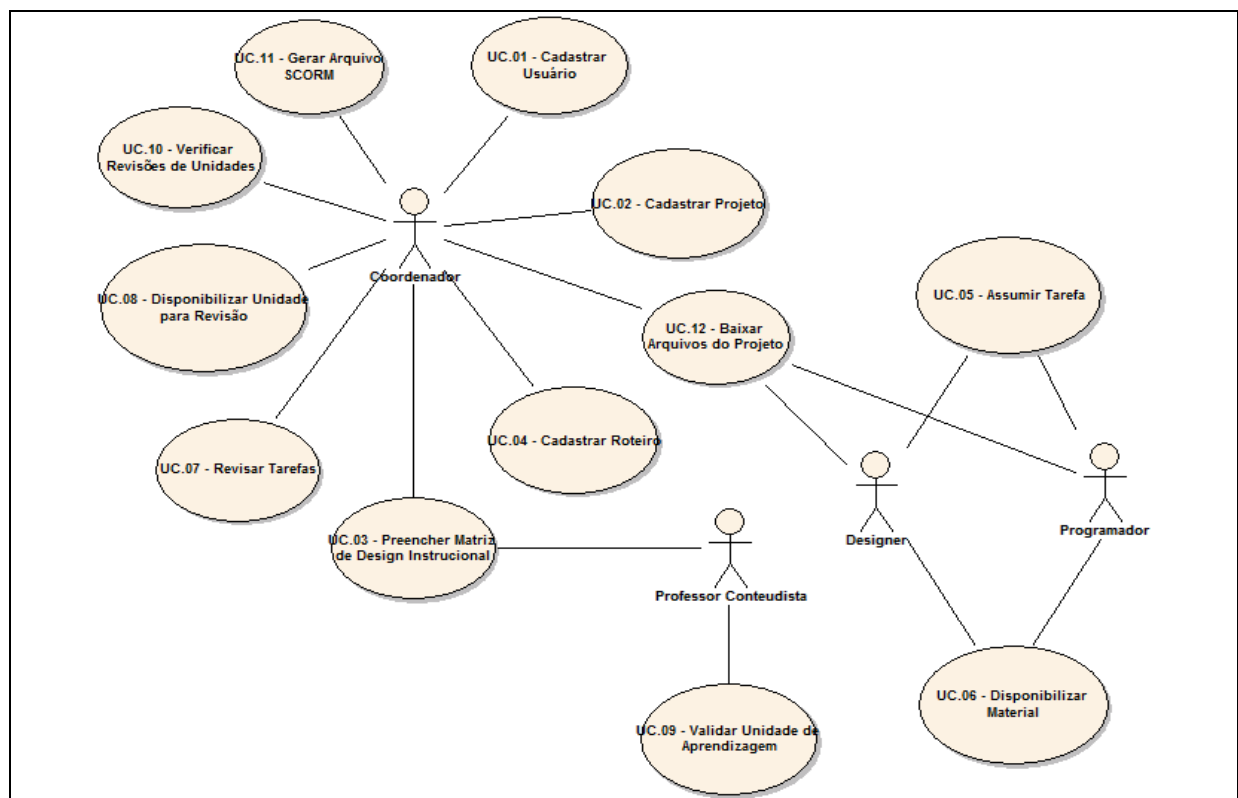


Figura 9 - Diagrama de casos de uso

Caso de Uso	UC.01 - Cadastrar Usuário
Descrição	Permite ao coordenador incluir, alterar e excluir usuários do sistema. Além disso, permite a consulta dos usuários cadastrados.
Ator	Coordenador
Pré-condição	O coordenador precisa estar logado no sistema.
Cenário Principal	<p>1) O coordenador deseja cadastrar um usuário.</p> <p>2) O coordenador informa nome, nick, senha, confirmação de senha e email.</p> <p>3) O coordenador elenca papéis ao usuário: coordenador, professor conteudista, designer ou programador.</p> <p>4) O coordenador finaliza o cadastro clicando no botão "Salvar".</p> <p>5) O sistema inclui o usuário.</p>
Alternativo 1	<p>No passo 2, o coordenador pode editar os dados do usuário selecionado.</p> <p>2.1) O sistema apresentará um tela de edição para alterar as informações do usuário.</p> <p>2.2) O coordenador preenche os campos e confirma.</p> <p>2.3) O sistema salva as informações.</p>
Alternativo 2	<p>No passo 2, o coordenador pode excluir o usuário selecionado.</p> <p>2.1) O coordenador exclui o usuário selecionado.</p> <p>2.2) O sistema salva as informações.</p>
Exceção 1	<p>No passo 2.1, na alternativa de editar um usuário.</p> <p>2.1.1) Caso não tenha nenhum usuário selecionado, o sistema apresenta mensagem "Necessário selecionar um usuário para poder editá-lo".</p>
Exceção 2	<p>No passo 2.1, na alternativa de excluir um usuário.</p> <p>2.1.1) Caso não tenha nenhum usuário selecionado, o sistema apresenta mensagem "Necessário selecionar um usuário para poder excluí-lo".</p>
Exceção 3	<p>No passo 4, na alternativa de salvar novo usuário.</p> <p>4.1) Caso já exista um usuário cadastrado no sistema com o mesmo nick, sistema apresenta mensagem "Já existe um usuário cadastrado com este nick".</p>
Pós-condição	Um usuário foi incluído, alterado ou excluído.

Quadro 1 - Caso de uso Cadastrar Usuário

Caso de Uso	UC.02 - Cadastrar Projeto
Descrição	Permite ao coordenador incluir, alterar e excluir projetos. Além disso, permite a consulta de projetos já cadastrados.
Ator	Coordenador
Pré-condição	O coordenador precisa estar logado no sistema.
Cenário Principal	<p>1) O coordenador deseja cadastrar um projeto.</p> <p>2) O coordenador informa nome, assunto, data início, prazo de entrega para produção, professor conteudista responsável e observações.</p> <p>3) O coordenador finaliza o cadastro clicando no botão "Salvar".</p> <p>4) O sistema inclui o projeto.</p>
Alternativo 1	<p>No passo 2, o coordenador pode editar os dados do projeto selecionado.</p> <p>2.1) O sistema apresentará um tela de edição para alterar as informações do projeto.</p> <p>2.2) O coordenador preenche os campos e confirma.</p> <p>2.3) O sistema salva as informações.</p>
Alternativo 2	<p>No passo 2, o coordenador pode excluir o projeto selecionado.</p> <p>2.1) O coordenador exclui o projeto.</p> <p>2.2) O sistema salva as informações.</p>
Exceção 1	<p>No passo 2.1, na alternativa de editar um usuário.</p> <p>2.1.1) Caso não tenha nenhum usuário selecionado, o sistema apresenta mensagem "Necessário selecionar um usuário para poder editá-lo"</p>
Exceção 2	<p>No passo 2.1, na alternativa de excluir um usuário.</p> <p>2.1.1) Caso não tenha nenhum usuário selecionado, o sistema apresenta mensagem "Necessário selecionar um usuário para poder excluí-lo"</p>
Exceção 3	<p>No passo 4, na alternativa de salvar novo projeto.</p> <p>4.1) Caso a data de prazo de entrega da produção for menor que a data de início de projeto o sistema apresenta mensagem "Data de entrega de produção deve ser maior do que a data de início"</p>
Pós-condição	Um projeto foi incluído, alterado ou excluído. O projeto foi associado com professor conteudista.

Quadro 2 - Caso de uso Cadastrar Projeto

Caso de Uso	UC.03 – Preencher Matriz de Design Instrucional
Descrição	Permite ao usuário incluir, alterar, excluir e consultar unidades de aprendizagem na matriz de design instrucional.
Ator	Coordenador e professor conteudista
Pré-condição	O coordenador ou conteudista precisa estar logado no sistema. Necessário ter um projeto cadastrado anteriormente no sistema.
Cenário Principal	<ol style="list-style-type: none"> 1) O usuário deseja incluir uma unidade de aprendizagem. 2) O usuário informa um nome, descrição, duração, objetivos, atividades, conteúdos e as formas de avaliação (instrumentos de avaliação e descrição). 3) O usuário finaliza o cadastro clicando no botão "Salvar". 4) O sistema grava a unidade de aprendizagem na matriz.
Alternativo 1	<p>No passo 2, o usuário pode editar a unidade de aprendizagem na matriz instrucional.</p> <ol style="list-style-type: none"> 2.1) O sistema apresentará um tela de edição para alterar as informações da unidade de aprendizagem. 2.2) O usuário preenche os campos e confirma. 2.3) O sistema salva as informações.
Alternativo 2	<p>No passo 2, o usuário pode excluir a unidade de aprendizagem selecionada.</p> <ol style="list-style-type: none"> 2.1) O usuário exclui a unidade de aprendizagem. 2.2) O sistema salva as informações.
Pós-condição	Uma unidade foi incluída, alterado ou excluída. A unidade foi associada a um projeto. O status foi atribuída como “Aberta”.

Quadro 3 - Caso de uso Preencher Matriz de Design Instrucional

Caso de Uso	UC.04 - Cadastrar Roteiro
Descrição	Permite ao coordenador inserir, alterar, excluir ou consultar roteiros para uma unidade de aprendizagem.
Ator	Coordenador
Pré-condição	O coordenador precisa estar logado no sistema. É necessário ter unidades cadastradas no projeto.
Cenário Principal	<ol style="list-style-type: none"> 1) O coordenador deseja cadastrar um roteiro. 2) O coordenador seleciona a unidade de aprendizagem. 3) O coordenador digita o roteiro (script). 4) O coordenador inclui as tarefas informando título, descrição, data entrega e responsável (programador ou designer). 5) O coordenador finaliza o cadastro clicando no botão "Salvar". 6) O sistema inclui o roteiro com suas tarefas.
Alternativo 1	<p>No passo 2, o coordenador pode editar os dados do roteiro selecionado.</p> <ol style="list-style-type: none"> 2.1) O sistema apresentará um tela de edição para alterar as informações do roteiro. 2.2) O coordenador preenche os campos e confirma. 2.3) O sistema salva as informações.
Alternativo 2	<p>No passo 2, o coordenador pode excluir o roteiro selecionado.</p> <ol style="list-style-type: none"> 2.1) O coordenador exclui o roteiro. 2.2) O sistema salva as informações.
Exceção 1	<p>No passo 4, na alternativa de incluir as atividades do roteiro.</p> <ol style="list-style-type: none"> 4.1) Caso não for preenchido os campos título, data entrega e responsável, sistema apresenta mensagem "Necessário preencher os campos Título, Data entrega e Responsável "
Pós-condição	<p>Um roteiro foi incluído, alterado ou excluído. Uma ou mais tarefas foram incluídas, alteradas ou excluídas. Para cada tarefa foi associado um programador ou designer e atribuído o status de "Aberta".</p>

Quadro 4 - Caso de uso Cadastrar Roteiro

Caso de Uso	UC.05 - Assumir Tarefa
Descrição	Permite ao designer ou programador consultar todas as tarefas disponíveis a serem assumidas e assumir uma delas.
Ator	Programador ou designer
Pré-condição	O designer ou programador precisa estar logado no sistema. Deve existir alguma tarefa com o status de "Aberta" ou "Com Problemas" e ter o usuário logado como responsável da tarefa.
Cenário Principal	<ol style="list-style-type: none"> 1. O usuário visualiza todas as tarefas com o status "Aberta" ou "Com Problemas". 2. O usuário seleciona uma tarefa e assume. 3. O sistema muda o status da tarefa para "Em Andamento". 4) O sistema inclui uma ocorrência guardando a data e o usuário.
Pós-condição	<p>Uma tarefa foi assumida pelo programador ou designer. O status da tarefa foi atribuído como "Em Andamento".</p>

Quadro 5 – Caso de uso Assumir Tarefa

Caso de Uso	UC.06 - Disponibilizar Material
Descrição	Permite ao usuário (designer/programador) fazer o <i>upload</i> de um material criado para determinada tarefa e colocar o status da tarefa como “Pronta para Revisão”.
Ator	Programador ou Designer
Pré-condição	O programador ou designer precisa estar logado no sistema. Precisa existir pelo menos uma tarefa com o status “Em Andamento”.
Cenário Principal	<ol style="list-style-type: none"> 1) O usuário deseja disponibilizar o material de uma tarefa; 2) O usuário seleciona a tarefa. 3) O usuário digita uma observação e escolhe o arquivo para <i>upload</i>. 4) O usuário faz o <i>upload</i> do material. 5) O sistema coloca a tarefa com o status "Pronta para Revisão". 6) O sistema inclui uma ocorrência para a tarefa colocando a data e o usuário que disponibilizou o material. 7) O sistema inclui o material.
Pós-condição	Foi realizado o <i>Upload</i> do material. Foi atribuído o status “Concluída” para a tarefa. Uma ocorrência foi incluída e associada à tarefa.

Quadro 6 – Caso de uso Disponibilizar Material

Caso de Uso	UC.07 - Revisar Tarefas
Descrição	Permite ao coordenador revisar as tarefas com o status "Pronta para Revisão".
Ator	Coordenador
Pré-condição	O coordenador precisa estar logado no sistema. Precisa existir pelo menos uma tarefa com o status “Pronta para Revisão”.
Cenário Principal	<ol style="list-style-type: none"> 1. O coordenador visualiza todas as tarefas com o status “Pronta para revisão”. 2. O coordenador seleciona uma tarefa. 3. O coordenador baixa o arquivo. 4. O coordenador marca a tarefa como concluída. 5) O sistema inclui uma ocorrência para a tarefa guardando a data e usuário que à revisou. 6. O sistema salva as informações.
Alternativo 1	No passo 4, quando o coordenador informar que a tarefa está com problemas. <ol style="list-style-type: none"> 4.1) O coordenador digita uma observação. 4.2) O sistema inclui uma ocorrência para a unidade guardando a data e usuário que validou. 4.3) O sistema salva as informações.
Pós-condição	A tarefa foi colocada como “Concluída” ou “Com Problemas”. Uma ocorrência foi incluída e associada à tarefa. Se todas as tarefas da unidade estão com o status “Concluídas” então é atribuído o status de “Concluída” para a unidade.

Quadro 7 – Caso de uso Revisar Tarefas

Caso de Uso	UC.08 - Disponibilizar Unidade para Revisão
Descrição	Permite ao coordenador liberar unidades de aprendizagem para revisão pelo professor conteudista. Além disso, permite ao coordenador disponibilizar um OA que representa a unidade.
Ator	Coordenador
Pré-condição	O coordenador precisa estar logado no sistema. Precisa existir pelo menos uma unidade com o status de "Concluída".
Cenário Principal	<ol style="list-style-type: none"> 1) O coordenador visualiza todas as unidades de aprendizagem que tenha todas as tarefas com status "Concluída". 2) O coordenador seleciona uma unidade. 3) O coordenador informa o nome do arquivo raiz. 4) O coordenador disponibiliza um arquivo que representa a unidade. 5) O sistema marca a unidade como "Pronta para Revisão". 6) O sistema inclui uma ocorrência para esta unidade guardando a data e usuário que disponibilizou o material. 7) O sistema grava as informações.
Pós-condição	Foi disponibilizado um material para uma unidade. Foi incluída uma ocorrência e associada à unidade.

Quadro 8 – Caso de uso Disponibilizar Unidade para Revisão

Caso de Uso	UC.09 - Validar Unidade de Aprendizagem
Descrição	Permite ao professor conteudista fazer o download do material previamente disponibilizado e validar a unidade de aprendizagem de acordo com os objetivos.
Ator	Professor conteudista
Pré-condição	O professor conteudista precisa estar logado no sistema. Deve existir uma unidade com o status "Pronta para Revisão".
Cenário Principal	<ol style="list-style-type: none"> 1) O conteudista visualiza todas as unidades com o status "Pronta para Revisão". 2) O conteudista seleciona uma unidade. 3) O conteudista baixa o arquivo da unidade. 4) O conteudista informa quais objetivos da unidade foram atendidos. 5) O conteudista finaliza a validação da unidade clicando no botão "Salvar". 6) O sistema inclui uma ocorrência para esta unidade guardando a data e usuário que validou. 7) O sistema marca a unidade como "Concluída".
Alternativo 1	<p>No passo 4, quando o conteudista informa os objetivos e nem todos foram atendidos.</p> <ol style="list-style-type: none"> 4.1) O sistema apresentará um tela para o conteudista informar uma observação para cada objetivo não atendido. 4.2) O sistema inclui uma ocorrência para a unidade guardando a data e usuário que à validou. 4.3) O sistema marca a unidade como "Com problemas"
Pós-condição	A unidade foi associada como "Concluída" ou "Com Problemas". Foi incluída uma ocorrência e associada a unidade.

Quadro 9 – Caso de uso Validar Unidade de Aprendizagem

Caso de Uso	UC.10 - Verificar Revisões de Unidade
Descrição	Permite ao coordenador adicionar novas tarefas as unidades com problemas.
Ator	Coordenador
Pré-condição	O coordenador precisa estar logado no sistema. É necessário ter unidades com o status "Com Problemas".
Cenário Principal	<ol style="list-style-type: none"> 1. O coordenador visualiza todas as unidades com o status "Com problemas". 2. O coordenador seleciona a unidade. 3. O coordenador adiciona novas tarefas (título, descrição, data de entrega e responsável). 4. O sistema marca a unidade como "Em andamento".
Pós-condição	Serão adicionadas uma ou mais tarefas.

Quadro 10 – Caso de uso Verificar Revisões de Unidades

Caso de Uso	UC.11 - Gerar Arquivo SCORM
Descrição	Permite ao coordenador gerar do OA um arquivo no formato SCORM.
Ator	Coordenador
Pré-condição	O coordenador precisa estar logado no sistema. É necessário ter unidades que estejam com o status "Concluída".
Cenário Principal	<ol style="list-style-type: none"> 1) O coordenador deseja gerar um arquivo no formato SCORM. 2) O coordenador seleciona as unidades concluídas. 3) O sistema gera o arquivo no formato SCORM.
Pós-condição	Será gerado um arquivo no formato SCORM para a unidade selecionada.

Quadro 11 – Caso de uso Gerar Arquivo SCORM

Caso de Uso	UC.12 - Baixar Arquivos do Projeto
Descrição	Permite aos usuários (coordenador/programador/designer) fazerem <i>downloads</i> dos materiais das tarefas concluídas.
Ator	Coordenador, programador e designer
Pré-condição	O usuário precisa estar logado no sistema. Deve existir alguma tarefa com o status de "Concluída".
Cenário Principal	<ol style="list-style-type: none"> 1) O usuário deseja fazer o <i>download</i> de um material. 2) O usuário seleciona a tarefa que deseja baixar. 3) O sistema faz o <i>download</i> do arquivo referente a tarefa selecionada.

Quadro 12 – Caso de uso Baixar Arquivos do Projeto

3.2.2 Diagrama de Atividades

Esta seção apresenta o diagrama de atividades cujas funcionalidades estão descritas nos casos de uso apresentadas na seção 3.2.1 (Diagramas de casos de uso).

A Figura 10 apresenta o processo de gerenciamento de produção de objetos de aprendizagem. Primeiramente o coordenador cadastra o projeto, para que em seguida o professor conteudista possa definir as unidades de aprendizagem deste projeto. Após definidas as unidades de aprendizagem, o coordenador cria roteiros para as unidades e estabelece as tarefas a serem realizadas para cada membro da equipe (programador ou designer). Neste momento, se já existir alguma tarefa do mesmo projeto e que esteja concluída, o usuário poderá baixá-la. Quando um membro conclui sua tarefa, ele irá disponibilizar o material criado. Cabe então ao coordenador revisar esse material. Se todas as tarefas da unidade atenderam o solicitado, o coordenador irá disponibilizar o material já desenvolvido. O professor conteudista irá validar esta unidade de aprendizagem e se ela atender o solicitado o coordenador irá gerar um arquivo no formato SCORM desta unidade.

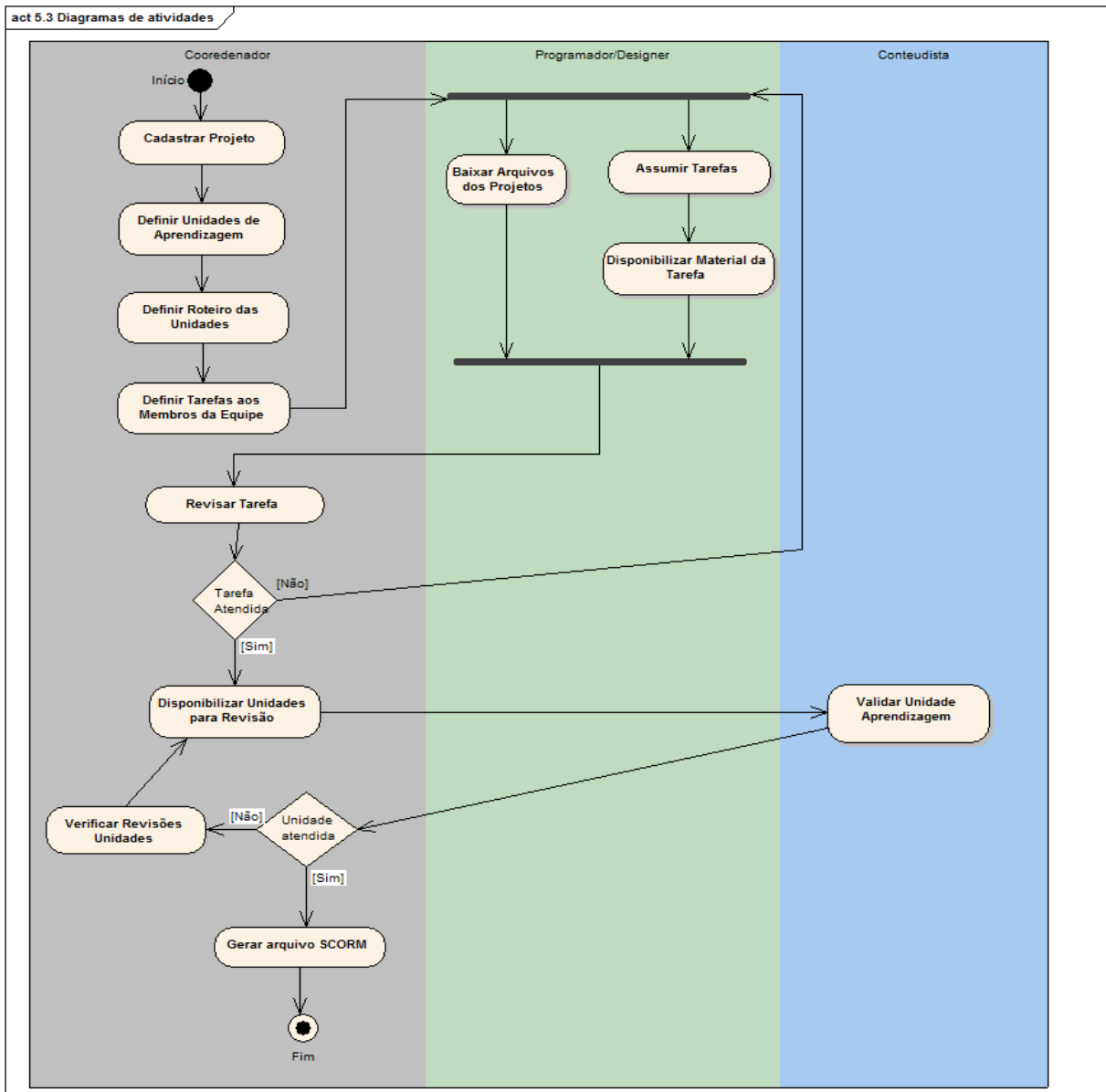


Figura 10 – Diagrama de atividades

3.2.3 Diagrama de Classes

Para contemplar as atividades descritas e para uma melhor divisão das atribuições, a ferramenta foi dividida em cinco pacotes, que interagem através de ligações lógicas conforme é apresentado no diagrama de pacotes na Figura 11. É importante ressaltar que os diagramas a seguir estão num formato reduzido, suprimindo alguns atributos e/ou métodos para um melhor entendimento, apresentando somente as operações e atributos fundamentais ao entendimento do problema. A seguir são explanados cada um dos pacotes e suas responsabilidades na ferramenta.

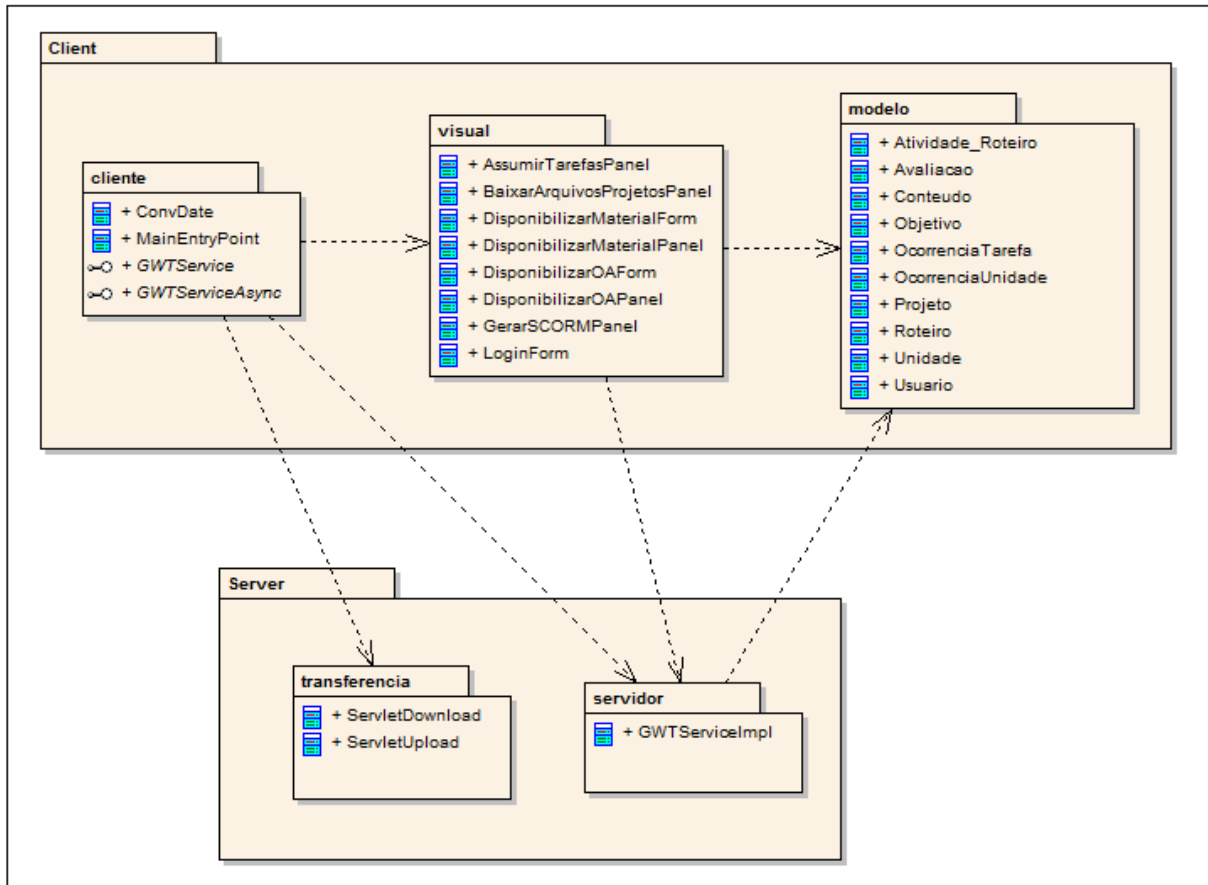


Figura 11 – Diagrama de pacotes da ferramenta

O pacote `cliente` é apresentado na figura 12 e representa as classes de inicialização da ferramenta na parte do cliente (navegador) e classes para invocação remota dos serviços no servidor. A aplicação que executa no cliente faz uma invocação a um serviço implementado no servidor através de *Servlet*, que pode então se utilizar de toda a expressividade e APIs do Java para retornar os dados para o cliente, seja o acesso por arquivos ou banco de dados. As classes que compõem o pacote `cliente` são:

- `MainEntryPoint`: classe que possui o método genérico “*onModuleLoad*” de uma aplicação GWT, chamado quando a aplicação começa a executar no cliente (navegador). Esta classe é responsável também pela criação da tela do *login* e por todos os casos de uso da ferramenta;
- `ConvDate`: classe responsável para fazer as conversões de datas e horas utilizadas nos casos de uso criado pela classe `MainEntryPoint`;
- `GWTService`: classe para definir inicialmente a *interface* do serviço a ser invocado pelo cliente. Essa *interface* deve obrigatoriamente realizar a *interface* de marcação `com.google.gwt.user.client.rpc.RemoteService` e os seus métodos são automaticamente invocáveis remotamente. Essa interface deve ser declarada dentro do

pacote Client da aplicação;

- d) `GWTServiceAsync`: *interface* assíncrona para a execução do serviço no cliente. A definição dessa *interface* assíncrona é necessária porque a invocação ao serviço utilizando AJAX acontece de forma assíncrona, para o cliente, a aplicação não para de executar, a chamada ao serviço acontece em *background*.

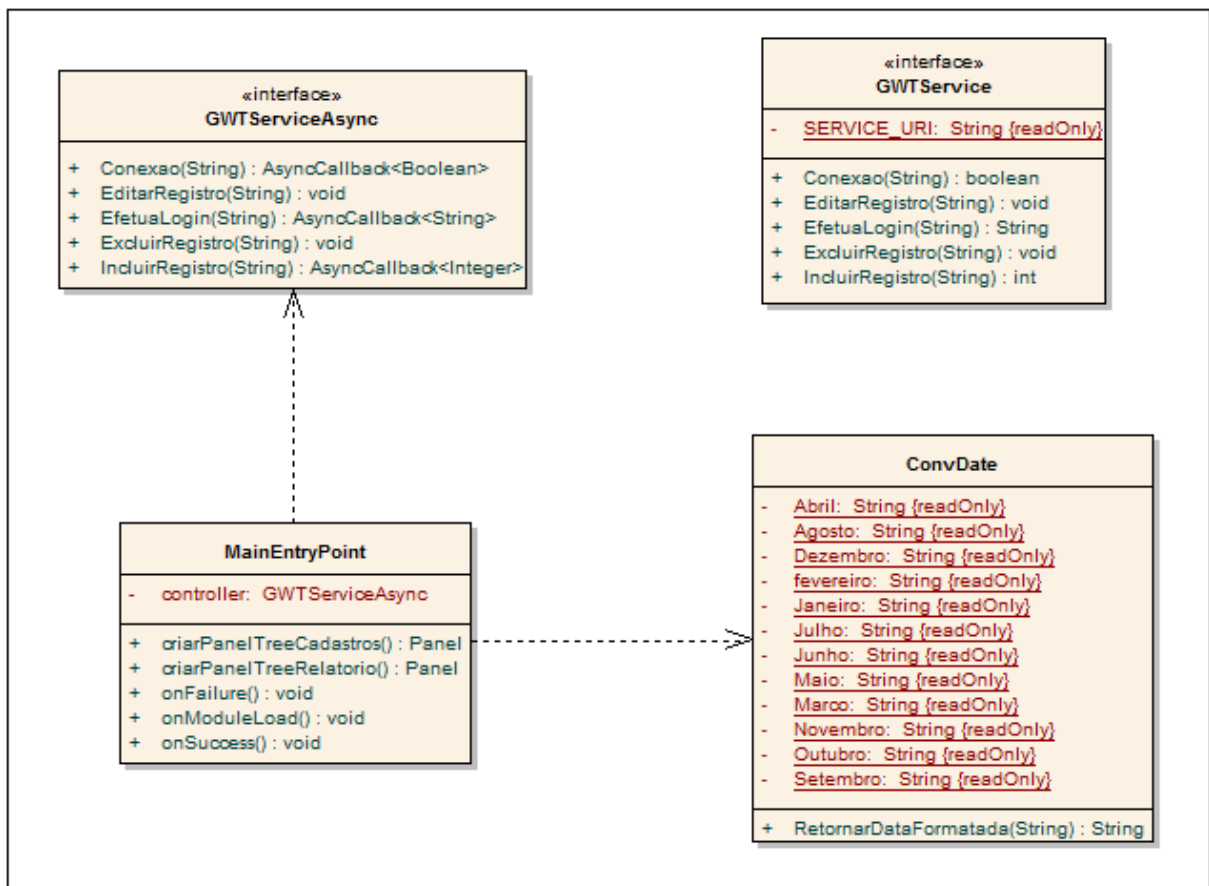


Figura 12 – Diagrama de classes do pacote cliente

As classes do pacote `visual` são apresentadas na Figura 13 e representam a camada de apresentação da ferramenta. São responsáveis pela *interface* com o usuário, gerando as telas no navegador. Além disso, essas classes também são responsáveis pelo controle e execução dos casos de uso. O pacote `visual` é composto pelas seguintes classes:

- `LoginForm`: classe que contém a apresentação da tela de *login*, juntamente com os métodos para a validação dos campos de usuário e senha;
- `UsuariosPanel`: classe que contém a apresentação da tela de cadastro de usuários, possibilitando selecionar um usuário para consulta ou exclusão (UC.01);
- `UsuariosForm`: classe que contém a apresentação da tela de edição de usuário, possibilitando ao usuário incluir e alterar os dados de usuário no banco de dados (UC.01);

- d) `ProjetosPanel`: classe que contém a apresentação da tela cadastro de projetos, possibilitando selecionar um projeto para consulta ou exclusão (UC.02);
- e) `ProjetosForm`: classe que contém a apresentação da tela de edição de projetos, possibilitando ao usuário incluir e alterar os dados de projetos no banco de dados (UC.02);
- f) `MatrizPanelWindow`: classe que contém a apresentação da tela matriz de *design* instrucional, possibilitando selecionar uma unidade para consulta ou exclusão (UC.03). A partir desta tela também é possível chegar ao cadastro de roteiros;
- g) `MatrizForm`: classe que contém a apresentação da tela de edição da matriz de *design* instrucional, possibilitando ao usuário incluir e alterar os dados da matriz no banco de dados (UC.03);
- h) `RoteirosForm`: classe que contém a apresentação da tela de cadastro de roteiros vindos da tela de matriz de *design* instrucional, possibilitando ao usuário incluir e alterar os dados do roteiro no banco de dados (UC.04);
- i) `RoteirosFormRevisoes`: classe que contém a apresentação da tela de cadastro de roteiros vindos da tela de revisões, possibilitando ao usuário incluir e alterar os dados do roteiro no banco de dados (UC.04);
- j) `AssumirTarefasPanel`: classe que contém a apresentação da tela de assumir tarefas, possibilitando ao usuário assumir determinada tarefa e consultar as ocorrências das tarefas no banco de dados (UC.05);
- k) `DisponibilizarMaterialPanel`: classe que contém a apresentação da tela disponibilizar material, possibilitando ao usuário a consulta das tarefas que estão em andamento e a seleção de determinada tarefa para poder disponibilizar o material (UC.06);
- l) `DisponibilizarMaterialForm`: classe que contém a apresentação da tela para seleção do material da tarefa à ser disposto no servidor via *servlets* (UC.06);
- m) `RevisarTarefasPanel`: classe que contém a apresentação da tela revisão de tarefas, possibilitando ao usuário fazer o *download* do arquivo referente a determinada tarefa e a consulta das ocorrências das tarefas no banco de dados (UC.07);
- n) `RevisarTarefasForm`: classe que contém a apresentação da tela trâmite tarefa, possibilitando ao usuário dar um parecer à tarefa, colocando ela como concluída ou com problemas (UC.07);
- o) `DisponibilizarOAPanel`: classe que contém a apresentação da tela para

- disponibilizar as unidades para revisão, possibilitando ao usuário a consulta das unidades que estão em andamento e a seleção de determinada unidade para poder disponibilizar seu OA que a representa (UC.08);
- p) `DisponibilizarOAForm`: classe que contém a apresentação da tela para seleção do material da unidade à ser disposto no servidor via *servlets* (UC.08);
 - q) `ValidarUnidadePanel`: classe que contém a apresentação da tela validar unidade de aprendizagem, possibilitando ao usuário fazer o *download* do arquivo referente a determinada unidade e a consulta das ocorrências das unidades no banco de dados (UC.09);
 - r) `VerificarRevisoesPanel`: classe que contém a apresentação da tela verificar revisões das unidades, possibilitando ao usuário a seleção de determinada unidade para poder revisá-la. A partir desta tela também é possível chegar ao cadastro de roteiros (UC.10);
 - s) `GerarSCORMPanel`: classe que contém a apresentação da tela gerar arquivo SCORM, possibilitando ao usuário a seleção de determinada unidade para fazer a geração do arquivo que à representa no formato SCORM (UC.11);
 - t) `BaixarArquivosProjetosPanel`: classe que contém a apresentação da tela baixar arquivos projetos, possibilitando ao usuário a seleção de determinada tarefa para fazer o *download* do arquivo que à representa (UC.12);
 - u) `WaitingBar`: classe que contém a apresentação da tela de *wait*, ou seja, esta classe será invocada quando a ferramenta estiver fazendo um *download* ou *upload* de algum arquivo, sinalizando o progresso do mesmo.

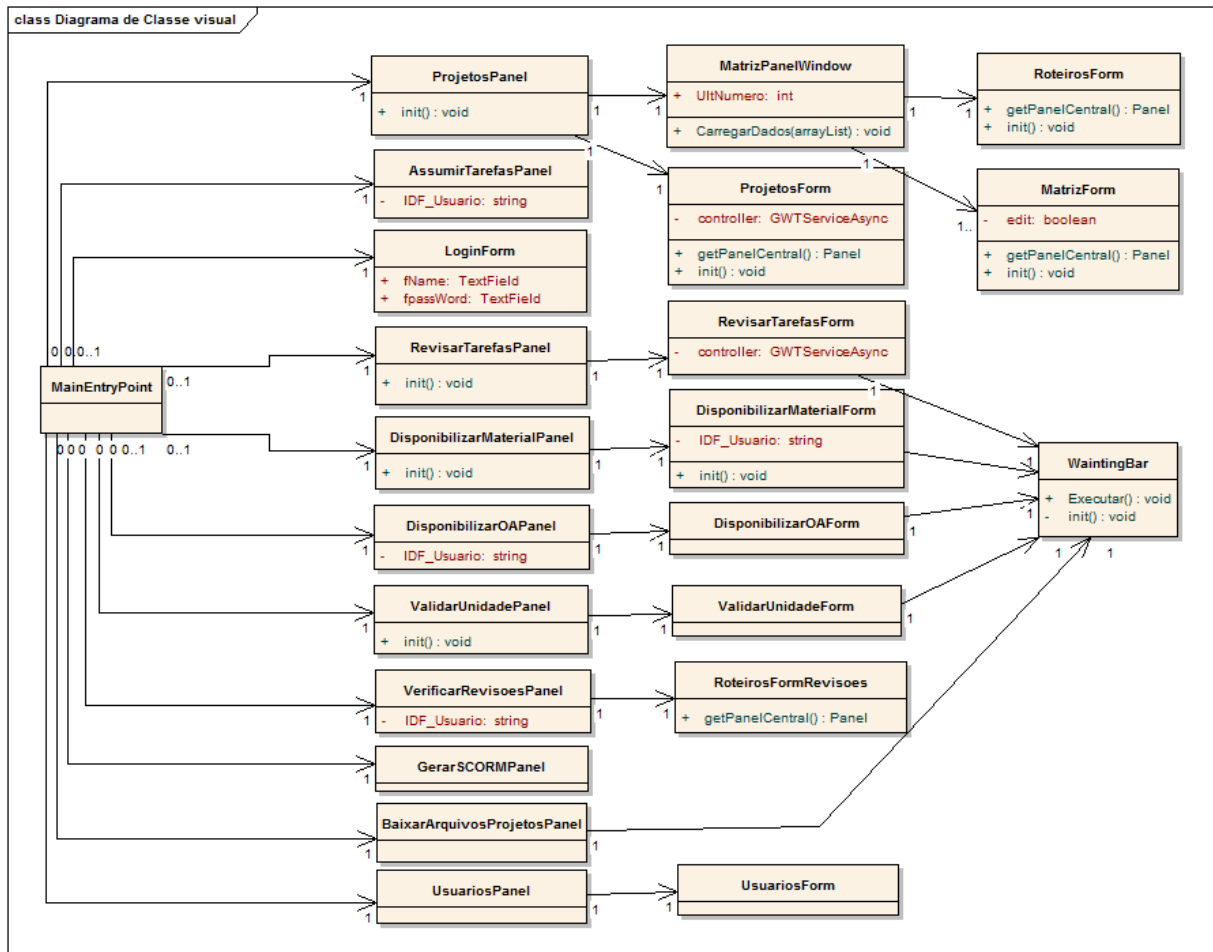


Figura 13 – Diagrama de classes do pacote `visual`

As classes do pacote `modelo` são apresentadas na Figura 14 e representam a camada de modelo. Estas classes são baseadas em um mapeamento das tabelas do banco de dados. Todas as classes realizam a *interface* `Serializable` e são instanciadas a partir das tabelas e repassadas aos objetos do pacote `visual` para preenchimento dos campos das telas. A partir dessas classes os dados são mapeados e gravados nessas tabelas.

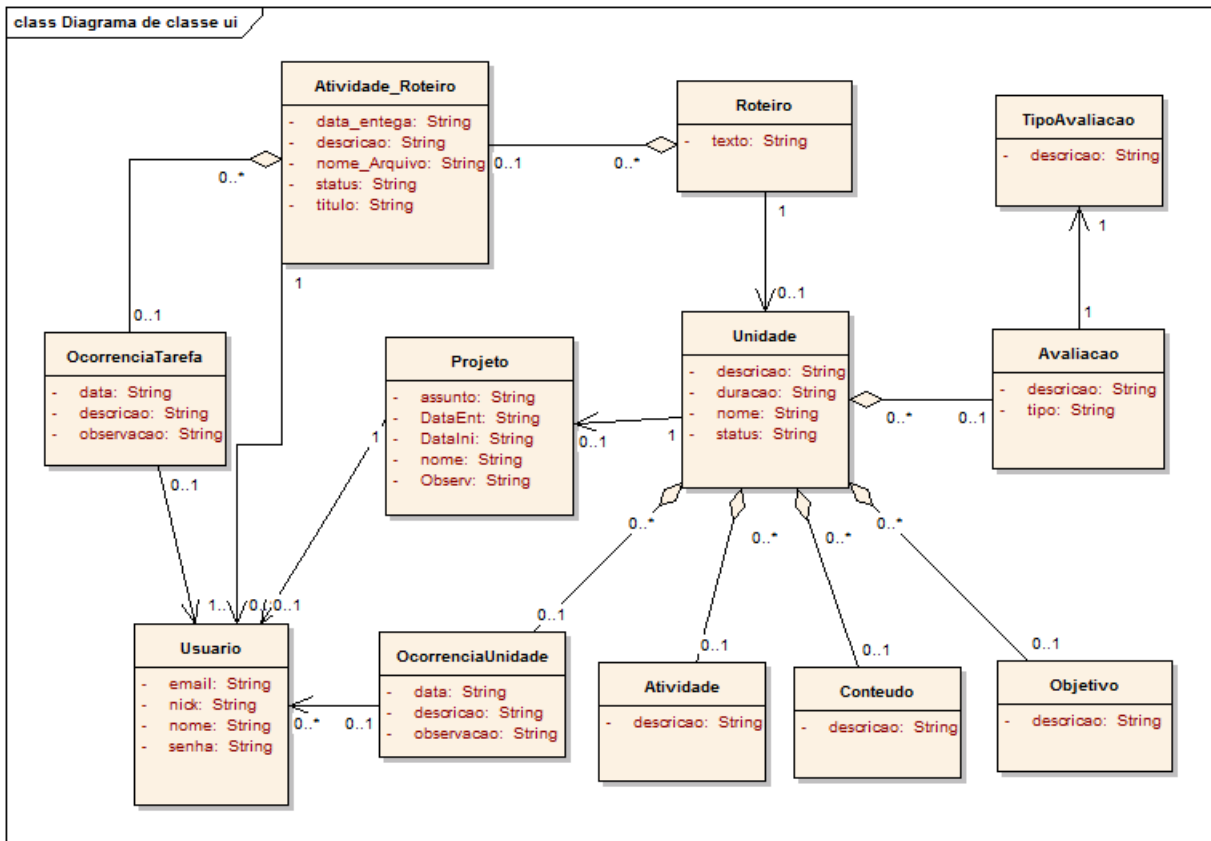


Figura 14 - Diagrama de classe do pacote modelo

A classe do pacote `servidor` é representada na Figura 15, neste pacote contém a classe `GWTServiceImpl` que realiza a *interface* `RemoteServiceServlet`. Essa classe possui o método que é responsável para buscar a instância da conexão do acesso ao banco de dados e também métodos que passam como parâmetro uma *Structured Query Language* (SQL) que serão usados para consulta, inserção, exclusão e alteração dos dados nas tabelas. Possui também o método responsável pela geração do arquivo no formato SCORM.

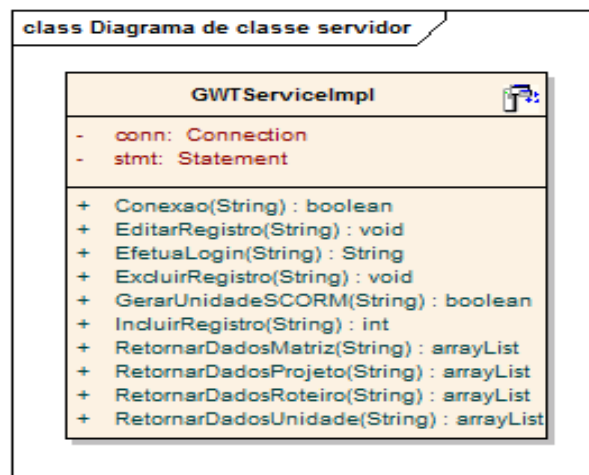


Figura 15 – Diagrama de classe do pacote servidor

As classes do pacote `transferencia` são apresentadas na figura 16 e representam a camada de transferência de arquivos no sistema. Realizam a *interface* `HttpServlet` e servem

para fazer *downloads* (ServletDownload) e *uploads* (ServletUpload) de arquivos no servidor.

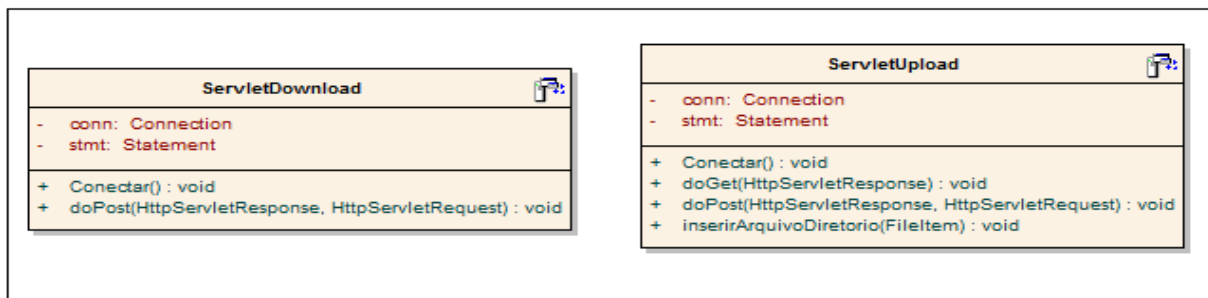


Figura 16 – Diagrama de classe do pacote transferencia

3.2.4 Diagrama de Estados

O diagrama de estados apresenta os possíveis estados de um objeto e as transações responsáveis pelas suas mudanças de estado (SAMPAIO, 2002). Foram elencados as classes `Atividade_Roteiro` e `Unidade` para apresentação do diagrama de estados, pois elas possuem alguns estados e funcionalidades essenciais no funcionamento da ferramenta.

A classe `Atividade_Roteiro` representa as tarefas que o programador e designer precisam desempenhar na construção de um OA. A Figura 17 apresenta seus estados possíveis.

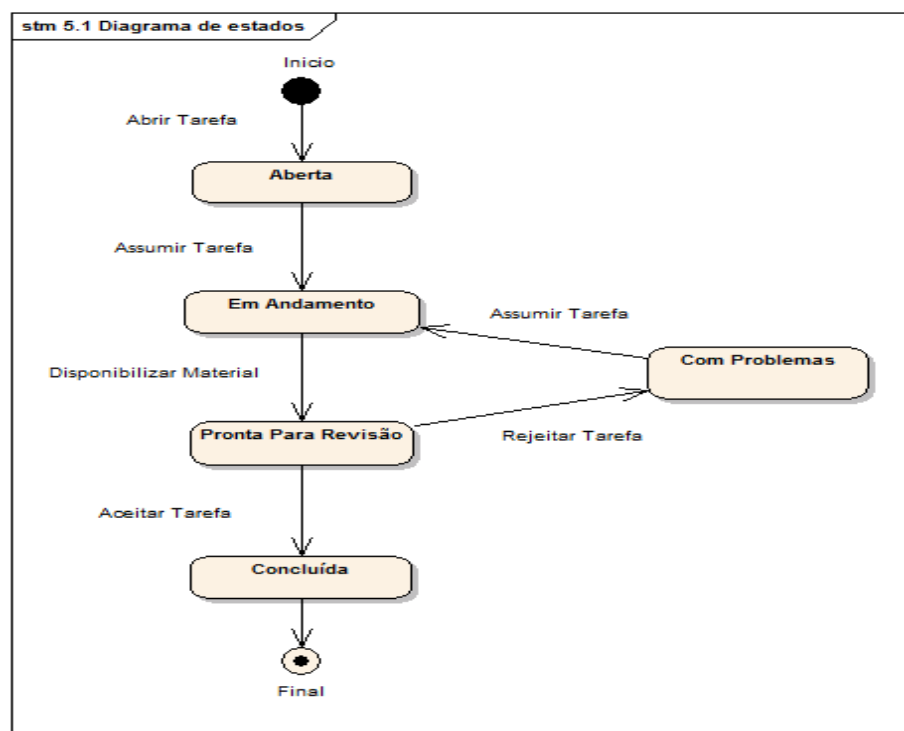


Figura 17– Diagrama de estados da classe `Atividade_Roteiro`

A seguir é feita uma explicação do relacionamento dos estados possíveis com os casos de uso disponíveis na ferramenta para a classe *Atividade_Roteiro*.

- a) *Aberta*: é atribuído esse *status* quando a tarefa é criada (UC.04);
- b) *Em Andamento*: é atribuído esse *status* quando o programador ou designer responsável assumir a tarefa (UC.05);
- c) *Pronta Para Revisão*: é atribuído esse *status* quando o programador ou designer disponibilizar o material referente à tarefa para revisão (UC.06);
- d) *Com Problemas*: é atribuído esse *status* quando o coordenador rejeita o material disposto para a tarefa, ou seja, quando o material não atender o solicitado (UC.07);
- e) *Concluída*: é atribuído esse *status* quando o coordenador aceita o material disposto para a tarefa, ou seja, quando o material atender o solicitado (UC.07).

A classe *Unidade* representa as unidades de aprendizagem na matriz de *design* instrucional. A Figura 18 apresenta seus estados possíveis.

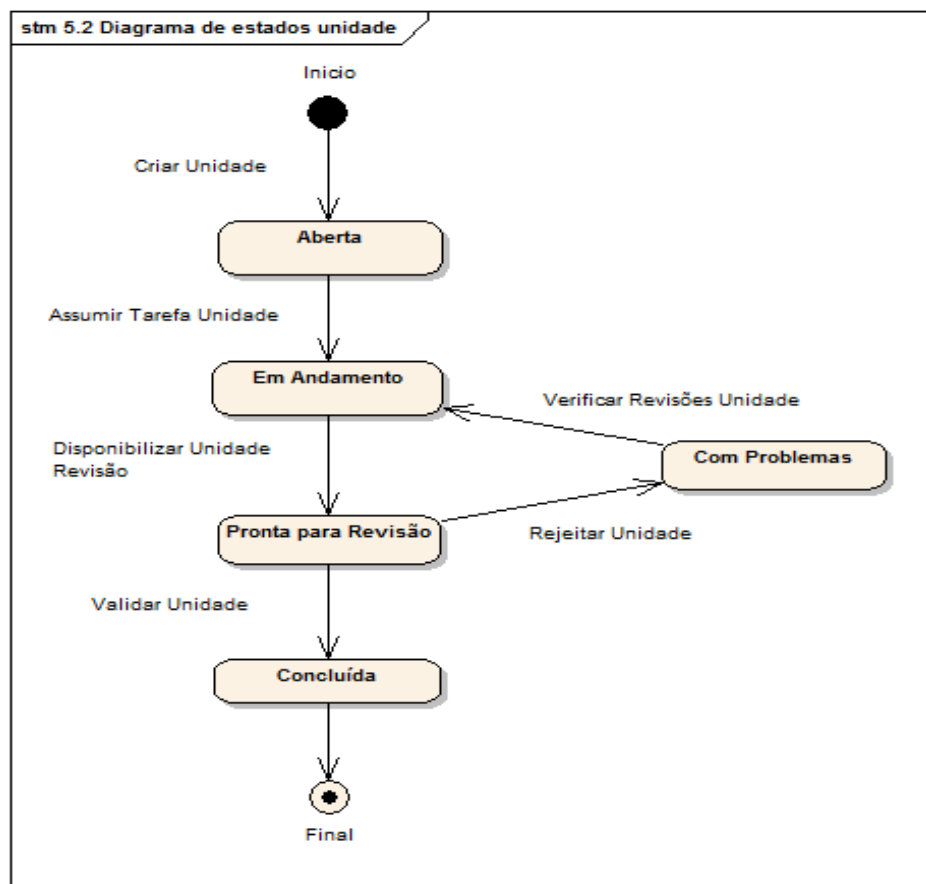


Figura 18 – Diagrama de estados da classe *Unidade*

A seguir é feita uma explicação do relacionamento dos estados possíveis com os casos de uso disponíveis na ferramenta para a classe *Unidade*.

- a) *Aberta*: é atribuído esse *status* quando a unidade de aprendizagem é criada (UC.03);

- b) Em Andamento: é atribuído esse *status* quando um programador ou designer assumir uma tarefa desta unidade que esta sob sua responsabilidade (UC.04);
- c) Pronta Para Revisão: é atribuído esse *status* quando o coordenador disponibilizar a unidade para revisão (UC.08);
- d) Com Problemas: é atribuído esse *status* quando o professor conteudista rejeita a unidade de aprendizagem (UC.09);
- e) Concluída: é atribuído esse *status* quando o professor conteudista aceita a unidade de aprendizagem (UC.09);

3.2.5 Diagrama de Entidade-Relacionamento (DER)

A Figura 19 apresenta o diagrama de entidade-relacionamento do banco de dados, que foi modelado utilizando a ferramenta DBDesigner 4.0.

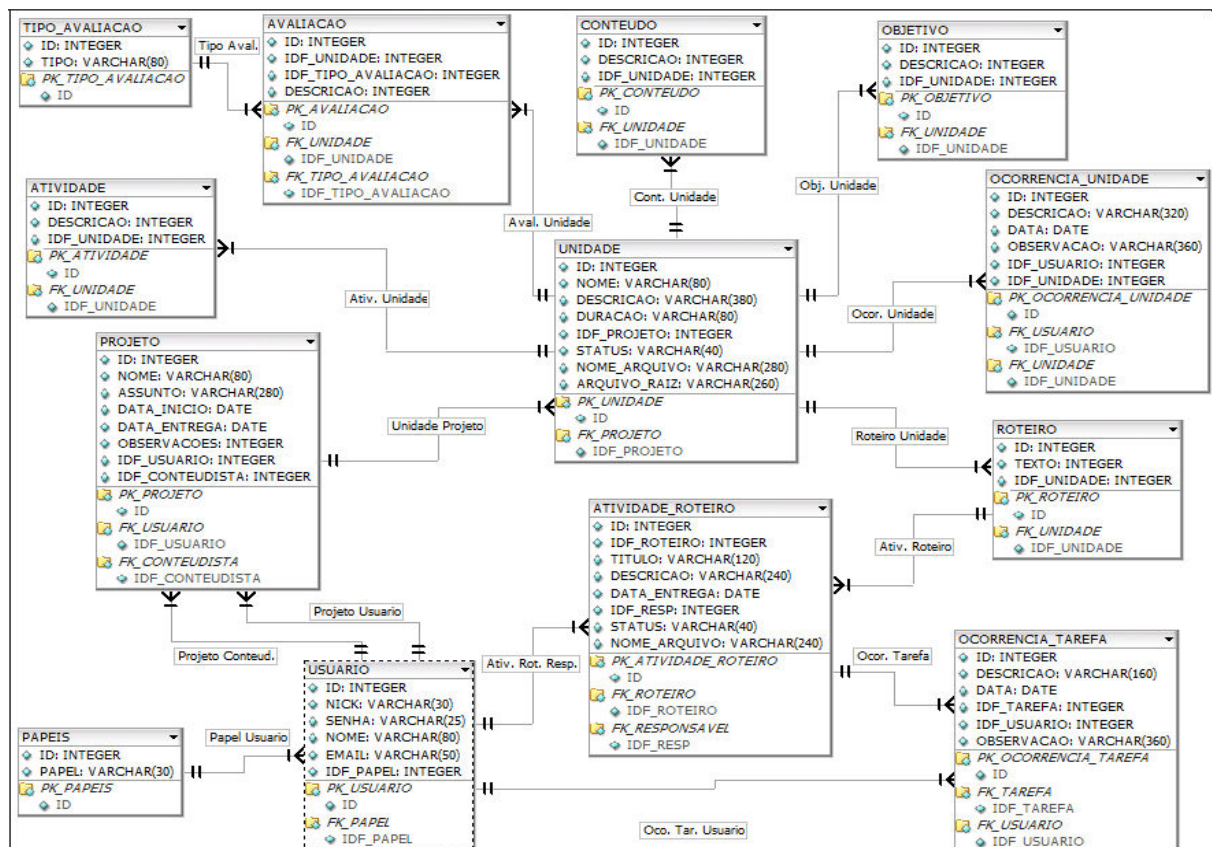


Figura 19 – Diagrama de entidade-relacionamento

As siglas “PK” e “FK” utilizados no DER significam respectivamente *Primary Key* e *Foreign Key*. Sob o ponto de vista de um banco de dados relacional, referem-se a tuplas (conjuntos) de um ou mais campos, cujos valores, considerando a combinação de valores de

todos os campos da tupla, nunca se repetem e que podem ser usados como um índice para os demais campos da tabela do banco de dados (SAMPAIO, 2002). A seguir é feita uma breve descrição de cada tabela:

- a) USUARIO: armazena as informações do cadastro de usuários;
- b) PAPEIS: armazena os tipos de papéis possíveis para o sistema: coordenador, professor conteudista, programador e *designer*;
- c) PROJETO: armazena as informações do cadastro de projetos;
- d) UNIDADE: armazena as informações do preenchimento da matriz de *design* instrucional;
- e) OBJETIVO: armazena as informações referentes aos objetivos cadastrados no preenchimento da matriz de *design* instrucional;
- f) CONTEUDO: armazena as informações referentes aos conteúdos cadastrados no preenchimento da matriz de *design* instrucional;
- g) ATIVIDADE: armazena as informações referentes às atividades cadastradas no preenchimento da matriz de *design* instrucional;
- h) AVALIACAO: armazena as informações referentes às avaliações cadastradas no preenchimento da matriz de *design* instrucional;
- i) TIPO_AVALIACAO: armazena os tipos de instrumentos de avaliação;
- j) ROTEIRO: armazena as informações do cadastro de roteiros;
- k) ATIVIDADE_ROTEIRO: armazena as informações referentes às tarefas cadastradas no cadastro de roteiros;
- l) OCORRENCIA_TAREFA: armazena as ocorrências das tarefas. Toda vez que algum usuário altera o status da tarefa, é incluída uma nova ocorrência;
- m) OCORRENCIA_UNIDADE: armazena as ocorrências das unidades. Toda vez que algum usuário altera o status da unidade, é incluída uma nova ocorrência.

3.3 IMPLEMENTAÇÃO

Com base na especificação, nas próximas seções são apresentadas as técnicas e ferramentas utilizadas, detalhamento de trechos de código fonte mais relevantes e a operacionalidade da implementação da ferramenta.

3.3.1 Técnicas e ferramentas utilizadas

A ferramenta foi implementada empregando o paradigma de orientação a objetos na linguagem de programação Java, utilizando-se o ambiente de desenvolvimento Netbeans 6.7, juntamente com o framework de desenvolvimento de aplicações web *Google Web Toolkit ExtJS* (GWT-Ext). Para a geração do arquivo no formato SCORM foi utilizado o *framework* CELINE e para o armazenamento dos dados e informações relevantes, foi utilizado o Sistema de Gerenciamento de Banco de Dados (SGBD) Firebird 2.1.

3.3.1.1 Implementação da ferramenta

O GWT-Ext trabalha com invocação remota de serviços, ou seja, a aplicação *JavaScript* executada no cliente faz uma invocação a um serviço implementado no servidor para retornar os dados para o cliente via acesso a arquivo ou *Java Data Base Connectivity* (JDBC).

Para criar um serviço que vai ser invocado, precisa-se inicialmente definir a *interface* do serviço. Essa *interface* deve obrigatoriamente estender a *interface* `com.google.gwt.user.client.rpc.RemoteService` e os seus métodos são automaticamente definidos como métodos “invocáveis” remotamente. A classe que definiu essa *interface* foi a `GWTService` do pacote `cliente` que está representada no Quadro 13.


```

package org.TCC_NOVO.client;
@RemoteServiceRelativePath("controller")
public interface GWTService extends RemoteService{

    public static final String SERVICE_URI = "/controller";

    public static class Util {
        public static GWTServiceAsync getInstance() {
            GWTServiceAsync instance = (GWTServiceAsync) GWT.create(GWTService.class);
            ServiceDefTarget target = (ServiceDefTarget) instance;
            target.setServiceEntryPoint(GWT.getModuleBaseURL() + SERVICE_URI);
            return instance;
        }
    }

    public boolean Conexao(String odbc);
    public String EfetuaLogin(String odbc);
    public ArrayList<Usuario> RetornarDadosUsuario(String Query);
    public int IncluirRegistro(String Query);
    public void ExcluirRegistro(String Query);
    public void EditarRegistro(String Query);
    public ArrayList<Projeto> RetornarDadosProjeto(String Query);
    public ArrayList<Unidade> RetornarDadosMatriz(String Query);
    public ArrayList<Roteiro> RetornarDadosRoteiro(String Query);
    public ArrayList<Unidade> RetornarDadosUnidade(String Query);
    public ArrayList<TipoAvaliacao> RetornarDadosTipoAvaliacao(String Query);
    public void GravarDadosUnidade(Unidade unid);
    public void AlterarDadosUnidade(Unidade unid);
    public void ExcluirDadosUnidade(Unidade unid);
    public void GravarDadosRoteiro(Roteiro rote);

    ...
}

```

Quadro 13 – Classe GWTService

O GWT-Ext consegue enviar diretamente objetos comuns do Java, como todos os tipos primitivos, String e Date, assim como arrays desses tipos. As classes definidas pelo desenvolvedor necessitam ser enviadas por invocações remotas e devem realizar a *interface* com.google.gwt.user.client.rpc.IsSerializable e ter como atributos apenas outros objetos que também implementem essa *interface* ou os objetos comuns do Java. O Quadro 14 exemplifica a classe Atividade realizando essa *interface*.

```

package org.TCC_NOVO.client.ui;

import com.google.gwt.user.client.rpc.IsSerializable;
import java.io.Serializable;

public class Atividade implements Serializable, IsSerializable{
    int ID = 0;
    String descricao = "";

    public Atividade(int ID, boolean alt, String desc) {
        this.ID = ID;
        this.descricao = desc;
    }

    public Atividade() { }

    public void setDescricao(String descricao) {
        this.descricao = descricao; }

    public void setID(int ID) {
        this.ID = ID; }

    public String getDescricao() {
        return descricao; }

    public int getID() {
        return ID; }
}

```

Quadro 14 – Classe IsSerializable Atividade

Com a *interface* base do serviço e o objeto que ele retorna definidos corretamente, precisa-se definir uma “*interface* assíncrona” para a execução do serviço no cliente. A definição dessa *interface* é necessária porque a invocação ao serviço utilizando-se AJAX acontece de forma assíncrona. Para o cliente, a aplicação não pára de executar, pois a chamada ao serviço acontece em *background*.

A *interface* assíncrona do serviço tem as declarações dos métodos praticamente iguais as da *interface* serviço mostrada no Quadro 14, mas os métodos de serviço agora devem retornar “void” e o último parâmetro deve ser um objeto do tipo `com.google.gwt.user.client.rpc.AsyncCallback`, que é o objeto que vai funcionar como um “listener” do evento de execução do serviço. `AsyncCallback` é uma *interface* que define dois métodos, “`onSucess()`” que é chamado quando o método executa corretamente e tem como parâmetro o resultado da invocação do serviço e “`onFailure()`” que é executado quando a chamada ao serviço não acontece de forma correta e recebe como parâmetro um objeto do tipo `Throwable` com informações sobre o erro acontecido. O nome da *interface*

assíncrona deve ser o mesmo nome da *interface* real do serviço, adicionando o sufixo “Async” e as duas devem estar dentro do mesmo pacote. O Quadro 15 representa a classe `GWTServiceAsync` que define a *interface* assíncrona da ferramenta.

```
package org.TCC_NOVO.client;

import org.TCC_NOVO.client.ui.OcorrenciaTarefa;
import org.TCC_NOVO.client.ui.OcorrenciaUnidade;
import org.TCC_NOVO.client.ui.Projeto;
import org.TCC_NOVO.client.ui.Roteiro;
import org.TCC_NOVO.client.ui.TipoAvaliacao;
import org.TCC_NOVO.client.ui.Unidade;
import org.TCC_NOVO.client.ui.Usuario;
import org.TCC_NOVO.client.ui.Objetivo;
import org.TCC_NOVO.client.ui.Atividade_Roteiro;
import com.google.gwt.user.client.rpc.AsyncCallback;
import java.util.ArrayList;

public interface GWTServiceAsync {
    public void Conexao(String odbc, AsyncCallback<Boolean> callback);
    public void EfetuaLogin(String Query, AsyncCallback<String> callback);
    public void IncluirRegistro(String Query, AsyncCallback<Integer> callback);
    public void ExcluirRegistro(String Query, AsyncCallback callback);
    public void EditarRegistro(String Query, AsyncCallback callback);
    public void RetornarDadosUsuario(String Query, AsyncCallback<ArrayList<Usuario>> callback);
    public void RetornarDadosProjeto(String Query, AsyncCallback<ArrayList<Projeto>> callback);
    public void RetornarDadosMatriz(String Query, AsyncCallback<ArrayList<Unidade>> callback);
    public void RetornarDadosUnidade(String Query, AsyncCallback<ArrayList<Unidade>> callback);
    :
}

```

Quadro 15 – Interface assíncrona `GWTServiceAsync`

Todas as aplicações GWT-Ext iniciam com uma classe que implementa a *interface* `EntryPoint`. A classe `MainEntryPoint` dessa ferramenta implementa essa *interface*. Nessa classe existe o método que chama a conexão com o banco de dados e o método `AsyncCallback` de retorno, exemplificado a chamada assíncrona da ferramenta conforme ilustrado no Quadro 16.

```
final AsyncCallback callbackConexao = new AsyncCallback() {
    @Override
    public void onSuccess(Object result) {
        //MessageBox.alert("Conectado com Sucesso!");
    }
    @Override
    public void onFailure(Throwable caught) {
        MessageBox.alert("Error", caught.getMessage());
    }
};

//método chamado para fazer a conexao com o banco de dados no servidor..
controller.Conexao(odbc, callbackConexao);

```

Quadro 16 – Chamada assíncrona do método `Conexao` pela classe `MainEntryPoint`

A classe `GWTServiceImpl` é descendente de `RemoteServiceServlet` e implementada no pacote `servidor`. Possui um método para conexão com o banco de dados e métodos de consulta, inclusão ou alteração de dados no banco. O Quadro 17 ilustra o método de conexão com o banco de dados na classe `GWTServiceImpl`.

```

public class GWTServiceImpl extends RemoteServiceServlet implements GWTService {

    private Connection conn;
    private Statement stmt;

    //Método para Estabelecer a Conexão com o Banco de Dados...
    @Override
    public boolean Conexao(String odbc) {
        boolean bConect = false;
        try {
            Class.forName("org.firebirdsql.jdbc.FBDriver");
            conn = DriverManager.getConnection(
                odbc,
                "sysdba",
                "masterkey");

            stmt = conn.createStatement();
            bConect = true;

        } catch (Exception e) {
            System.out.println("Não foi possível conectar ao banco: " + e.getMessage());
        }

        return bConect;
    }
    ...
}

```

Quadro 17 – Método para fazer a conexão com banco de dados na classe `GWTServiceImpl`

Para a manipulação dos arquivos nos processos de *download* e *upload*, desenvolveu-se as classes `ServletDownload` e `ServletUpload` respectivamente. Ambas descendem de `HttpServlet`. No Quadro 18 pode-se visualizar a classe `ServletDownload` e no Quadro 19 a classe `ServletUpload`.

```

package org.TCC_NOVO.server.Transferencia;
import ...
public class ServletDownload extends HttpServlet{
    public void doPost(HttpServletRequest request, HttpServletResponse res)
        throws ServletException, IOException {
        ResultSet rs = null;
        String fileName = "";
        boolean isMultiPart = FileUpload.isMultipartContent(request);
        if (isMultiPart) {
            Conectar();
            FileItemFactory factory = new DiskFileItemFactory();
            ServletFileUpload upload = new ServletFileUpload(factory);
            try {
                List items = upload.parseRequest(request);
                Iterator iter = items.iterator();
                while (iter.hasNext()) {
                    FileItem item = (FileItem) iter.next();
                    if (!item.isFormField()) {
                        File file = new File(fileName);
                        if (file.exists() && file.isFile()) {
                            res.setContentType("application/octet-stream");
                            res.setContentLength((int)file.length());
                            res.setHeader("Content-disposition", "attachment;filename="+file.getName());
                            BufferedInputStream in = new BufferedInputStream(new FileInputStream(file));
                            BufferedOutputStream out = new BufferedOutputStream(res.getOutputStream());
                            byte b[] = new byte[1024]; //tamanho máximo permitido, podendo ser alterado
                            while (in.read(b) >= 0) {
                                out.write(b);
                            }
                            in.close();
                            out.close();
                        } else {
                            throw new ServletException("Arquivo inválido ou corrompido.");
                        }
                    }
                }
            } catch (FileUploadException ex) { ex.printStackTrace();}
            catch (Exception ex) { ex.printStackTrace();}
        }
    }
}

```

Quadro 18 – Classe ServletDownload

```

package org.TCC_NOVO.server.Transferencia;

public class ServletUpload extends HttpServlet{
    @Override
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        boolean isMultiPart = FileUpload.isMultipartContent(request);
        if (isMultiPart) {
            Conectar();
            FileItemFactory factory = new DiskFileItemFactory();
            ServletFileUpload upload = new ServletFileUpload(factory);
            try {
                List items = upload.parseRequest(request);
                Iterator iter = items.iterator();
                while (iter.hasNext()) {
                    FileItem item = (FileItem) iter.next();
                    if (!item.isFormField()) {
                        if (item.getName().length() > 0) {
                            this.inserirArquivoDiretorio(item);
                        }
                    }
                }
            }
            catch (FileUploadException ex) {
                ex.printStackTrace();
            }
            catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
}

```

Quadro 19 – Classe ServletUpload

Para a geração do arquivo no formato SCORM foi desenvolvido o método GerarUnidadeSCORM da classe GWTServicImpl conforme ilustrado no Quadro 20.

```

@Override
public boolean GerarUnidadeSCORM(String arquivo, String nomeProj,
    String nomeUnid, String nomeArqRaiz){
    // Cria o diretório caso ele não exista
    File diretorio = new File("c:/Uploads/SCORM");
    if (!diretorio.exists())
        diretorio.mkdir();

    ContentPackageBuilder content = new ContentPackageBuilder("c:/Uploads/SCORM");
    content.add(new File(arquivo), nomeUnid, nomeArqRaiz);
    try{
        content.build(nomeProj, diretorio+"//"+nomeProj+".zip");
        return true;
    }catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

Quadro 20 – Método para geração do arquivo SCORM

3.3.2 Operacionalidade da implementação

Nessa seção são apresentadas todas as funcionalidades da ferramenta através de um estudo de caso. O estudo de caso apresentado mostrará todos os passos necessários para a construção do material necessário para lecionar a ferramenta *PowerPoint* como uma unidade de aprendizagem do projeto *Microsoft Office*. Serão utilizados os usuários jckr (coordenador), fabi (designer), maria (professor conteudista) e joao (programador).

A primeira tela do sistema é utilizada para autenticação do usuário, como pode ser observado na Figura 20.

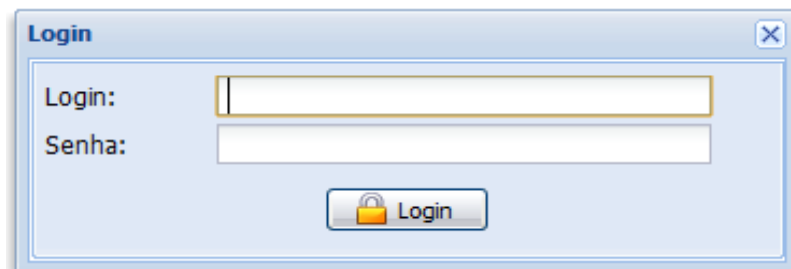


Figura 20 – Tela de autenticação de usuário

A tela principal da ferramenta é exibida após a autenticação (Figura 21).

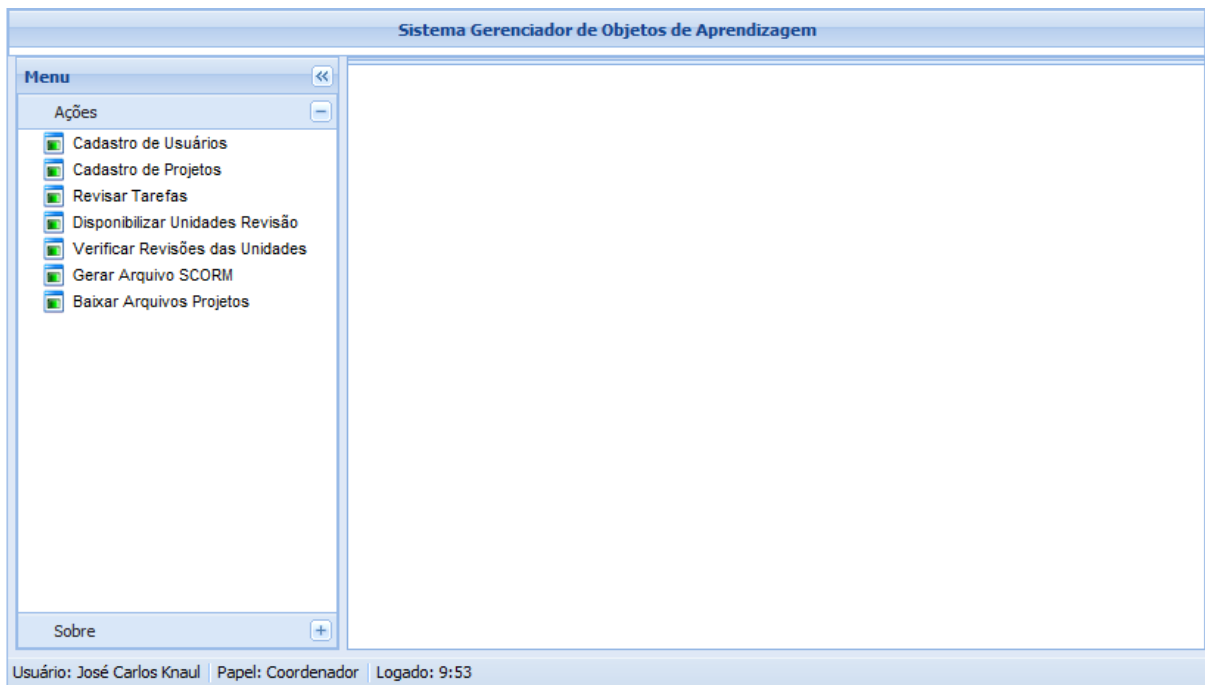


Figura 21 – Tela principal da ferramenta

No canto inferior esquerdo são apresentados o nome e o papel do usuário logado. Acima, no lado esquerdo estão as opções do *menu*, cada uma associada a um caso de uso. Essas opções são exibidas de acordo com o nível de permissão do usuário, que é estabelecido

através do seu papel. Quando clicado em qualquer opção do *menu*, sua tela principal será exibida no espaço em branco à direita.

A Figura 22 apresenta o cadastro de usuários. Pode-se observar que a tela de cadastro de usuário possui uma grade com todos os usuários já cadastrados e três botões que seguem como padrão na maioria das telas:

- a) **Incluir**: quando clicado será aberta uma tela para fazer a inclusão de registros;
- b) **Editar**: após selecionar uma linha da grade, será aberta uma nova tela para fazer a alteração do registro. A Figura 23 apresenta a tela para edição dos dados do usuário;
- c) **Remover**: após selecionar uma linha da grade, o sistema irá abrir uma tela para confirmação da exclusão, e em seguida excluirá o registro.

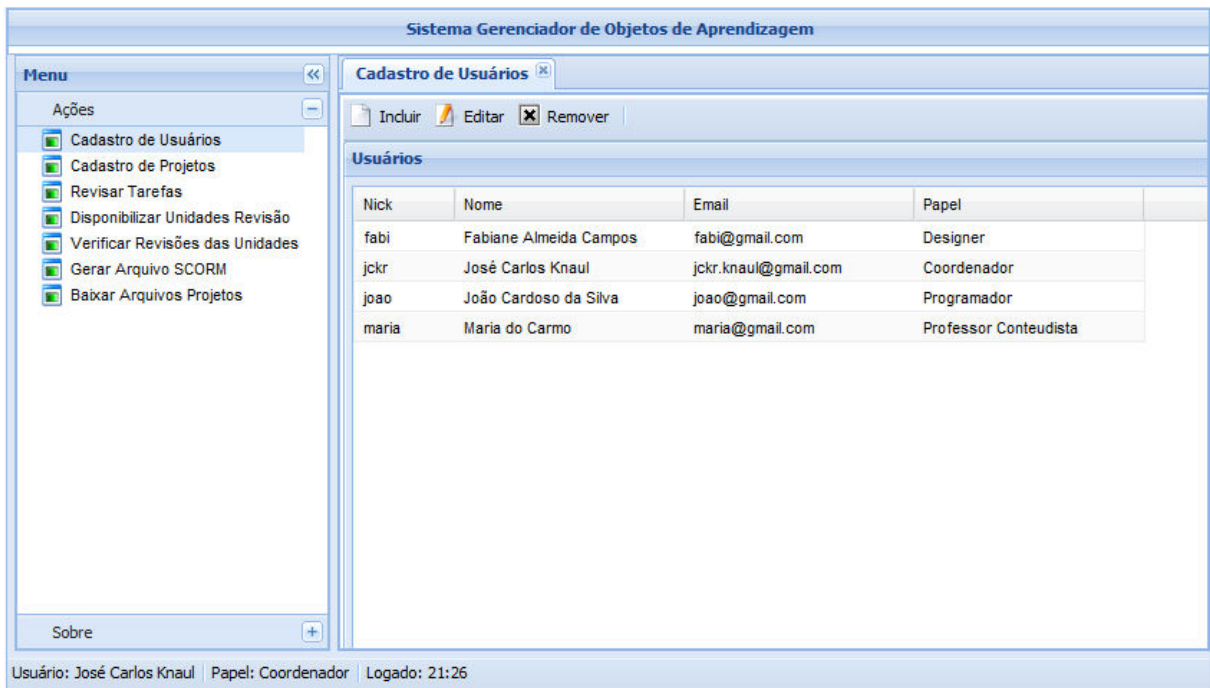


Figura 22 – Cadastro de usuários

The dialog box is titled 'Cadastro de Usuários' and contains the following fields and controls:

- Nick: jckr
- Senha: masked with four dots
- Conf. Senha: masked with four dots
- Nome: José Carlos Knaul
- Email: jckr.knaul@gmail.com
- Papel: Coordenador (dropdown menu)
- Buttons: Salvar (with a green checkmark icon) and Cancelar (with a red X icon)

Figura 23 – Tela para digitar os dados do usuário

A Figura 24 demonstra a tela para inclusão do projeto que é preenchida pelo usuário coordenador (jckr). O coordenador informa um nome, assunto, professor conteudista responsável, data início, data de entrega e observações.

Cadastro de Projetos

Nome: Curso Microsoft Office

Assunto: Desenvolver material necessário para leccionação do curso Microsoft Office.

Conteudista Resp.: maria

Data Inicio: 01/11/09

Data Entrega: 30/11/09

Observações:

Microsoft Office

O curso será lecionado no mês de dezembro e seguirá a seguinte sequência:

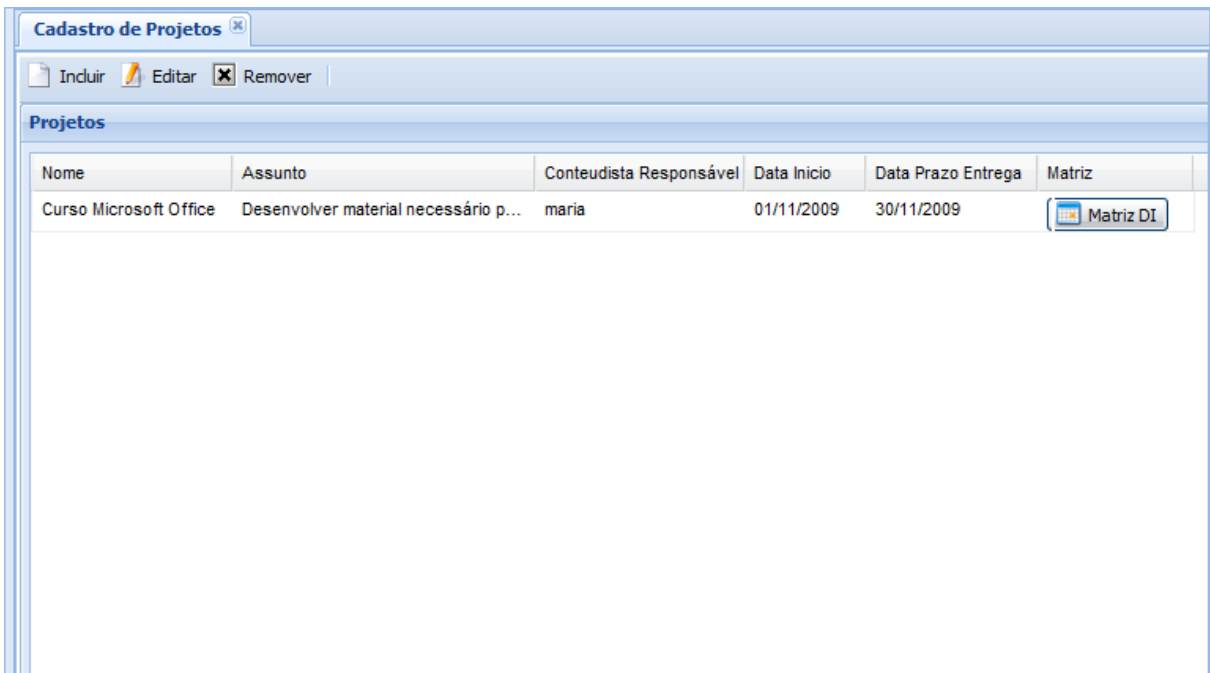
- 1- PowerPoint;
- 2- Word;
- 3- Excel;
- 4- Outlook.

Att. José Carlos Knaul
Coordenador

Salvar Cancelar

Figura 24 - Tela para digitar os dados do Projeto

Na grade do cadastro de projetos (Figura 25) pode-se observar uma coluna com botões (Matriz DI). Esses botões permitem o acesso ao cadastro de unidades de aprendizagem do projeto. Na Figura 26, é demonstrada a tela de cadastro das unidades de aprendizagem, exemplificando a ferramenta *Microsoft PowerPoint* como uma unidade de aprendizagem do curso *Microsoft Office*. Nesta tela são informados os objetivos, atividades, conteúdos e avaliações para a leccionação da unidade *PowerPoint*.



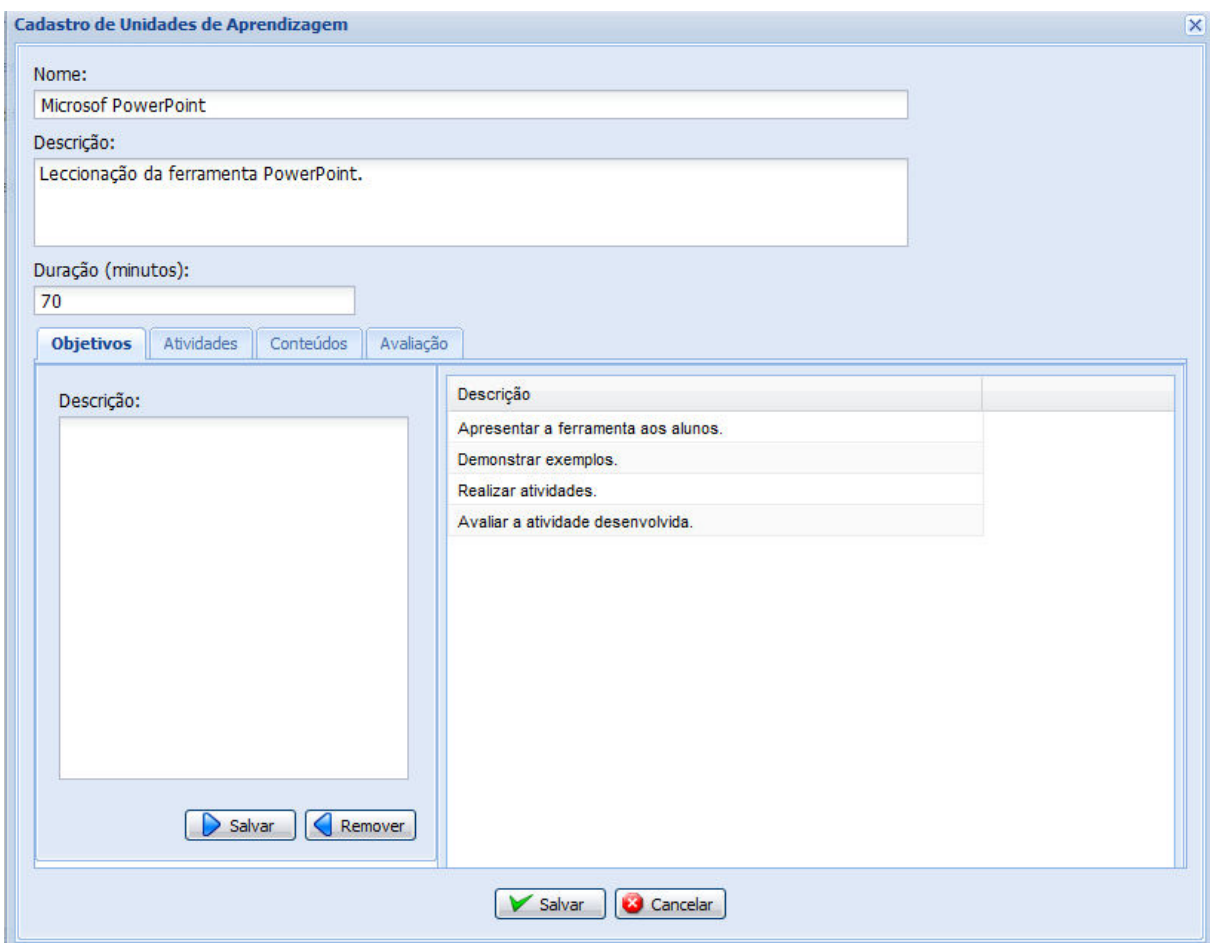
Cadastro de Projetos

Incluir Editar Remover

Projetos

Nome	Assunto	Conteudista Responsável	Data Inicio	Data Prazo Entrega	Matriz
Curso Microsoft Office	Desenvolver material necessário p...	maria	01/11/2009	30/11/2009	Matriz DI

Figura 25 – Grade dos Projetos



Cadastro de Unidades de Aprendizagem

Nome: Microsof PowerPoint

Descrição: Leccionação da ferramenta PowerPoint.

Duração (minutos): 70

Objetivos | Atividades | Conteúdos | Avaliação

Descrição:

Descrição
Apresentar a ferramenta aos alunos.
Demonstrar exemplos.
Realizar atividades.
Avaliar a atividade desenvolvida.

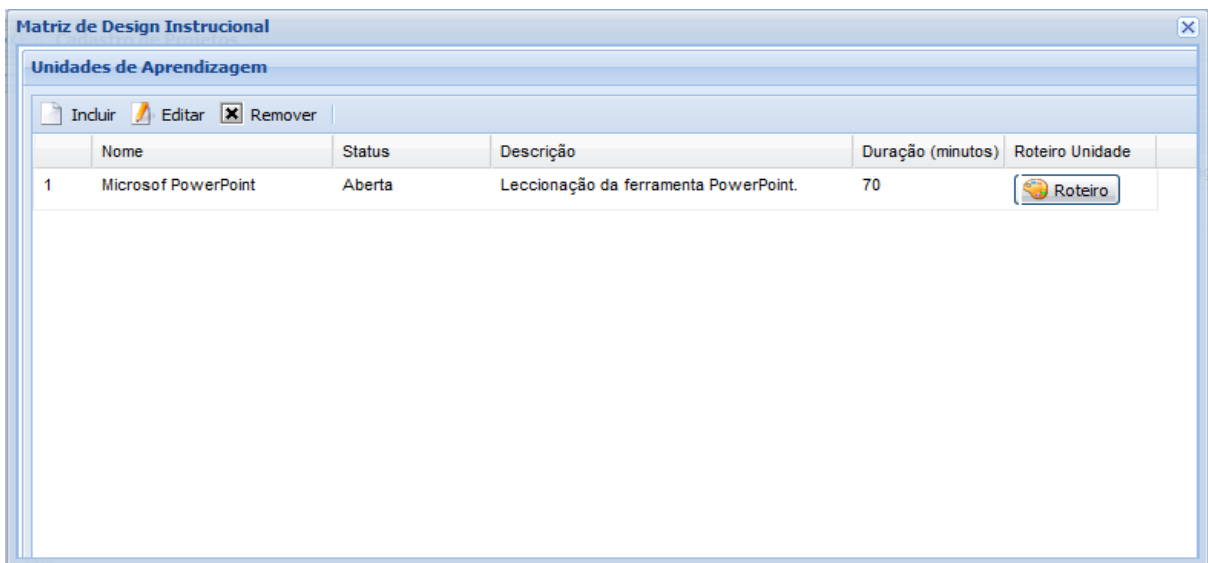
Salvar Remove

Salvar Cancelar

Figura 26 – Tela de Cadastro das Unidades de Aprendizagem

Na grade das unidades de aprendizagem (Figura 27) pode-se observar que ela possui uma coluna com botões de roteiro. Quando clicado nestes botões, será aberta uma tela com o

respectivo cadastro de roteiro da unidade de aprendizagem conforme ilustrado na Figura 28.



The screenshot shows a window titled "Matriz de Design Instrucional" with a sub-header "Unidades de Aprendizagem". It contains a table with the following data:


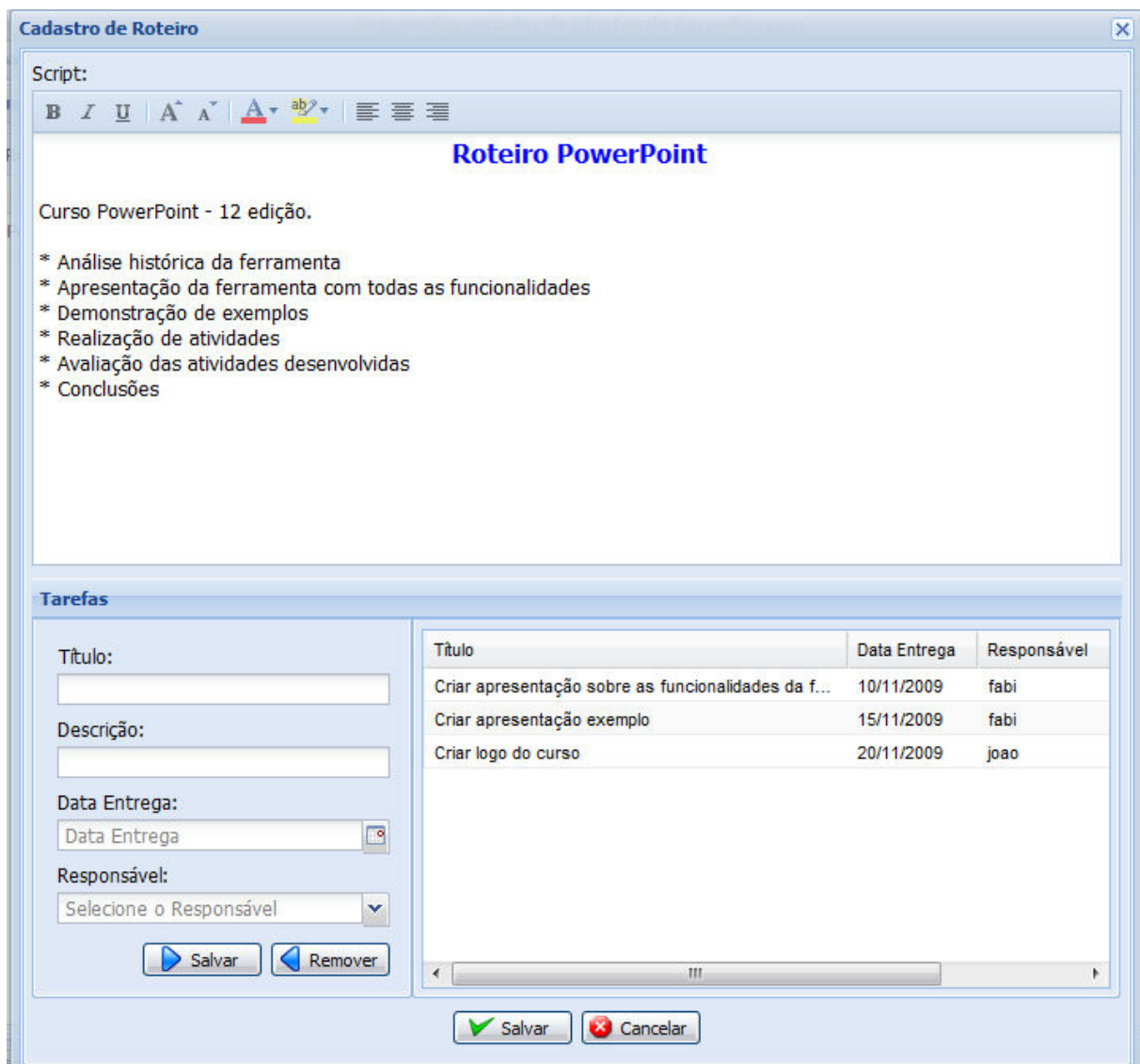
	Nome	Status	Descrição	Duração (minutos)	Roteiro Unidade
1	Microsof PowerPoint	Aberta	Leccionação da ferramenta PowerPoint.	70	 Roteiro

Figura 27 – Grade das Unidades de Aprendizagem



The screenshot shows a window titled "Cadastro de Roteiro". It features a "Script:" section with a rich text editor containing the following text:

Roteiro PowerPoint

Curso PowerPoint - 12 edição.

- * Análise histórica da ferramenta
- * Apresentação da ferramenta com todas as funcionalidades
- * Demonstração de exemplos
- * Realização de atividades
- * Avaliação das atividades desenvolvidas
- * Conclusões

Below the script editor is a "Tarefas" section with a form for adding tasks and a table of existing tasks.

Tarefas

Título:

Descrição:

Data Entrega:

Responsável:

Título	Data Entrega	Responsável
Criar apresentação sobre as funcionalidades da f...	10/11/2009	fabi
Criar apresentação exemplo	15/11/2009	fabi
Criar logo do curso	20/11/2009	joao

At the bottom of the window are two buttons: and .

Figura 28 – Tela de Cadastro de Roteiro

Nesta tela (Figura 28) o coordenador irá cadastrar as tarefas necessárias para lecionar o curso *PowerPoint* considerando os objetivos da unidade de aprendizagem. Essas tarefas ficarão sob a responsabilidade dos programadores e *designers*. A tela está ilustrada com três tarefas, duas sob a responsabilidade da *designer* fabi e uma sob a responsabilidade do programador joao.

Após as tarefas terem sido cadastradas, o usuário responsável pode visualizá-la na tela Assumir Tarefa (Figura 29). Clicando no botão “Assumir Tarefa” o sistema colocará a tarefa com o *status* de “Em andamento”. Abaixo da grade das tarefas existe uma grade com as respectivas ocorrências que seguem como padrão na maioria das telas. Estas ocorrências são gravadas toda vez que a tarefa for utilizada dentro do sistema.

Nome Projeto	Nome Unidade	Título	Descrição	Status	Data Entrega	Responsável
Curso Microsoft Office	Microsof PowerPoint	Criar apresentação sobre os princípios d...	Necessário a criação de a...	Aberta	10/11/2009	fabi
Curso Microsoft Office	Microsof PowerPoint	Criar apresentação das funcionalidades	Necessário a criação de a...	Aberta	10/11/2009	fabi

Data Ocorrência	Usuário	Descrição	Observação
08/11/0200	fabi	Tarefa Aberta	

Figura 29 – Tela Assumir Tarefas

Na tela Disponibilizar Material (Figura 30), pode-se observar uma grade com as tarefas assumidas e um botão chamado “Disponibilizar Material”. Quando selecionada a tarefa da grade e clicado neste botão, será aberta uma nova tela para disponibilizar o material dessa tarefa conforme ilustrado na Figura 31. Nesta tela o usuário seleciona o arquivo solicitado referente à tarefa e inclui uma observação. O sistema irá mandar este arquivo para o servidor (*upload*), mudará o *status* da tarefa para “Pronta para Revisão” e gravará uma ocorrência para esta tarefa.



Figura 30 – Tela Disponibilizar Material

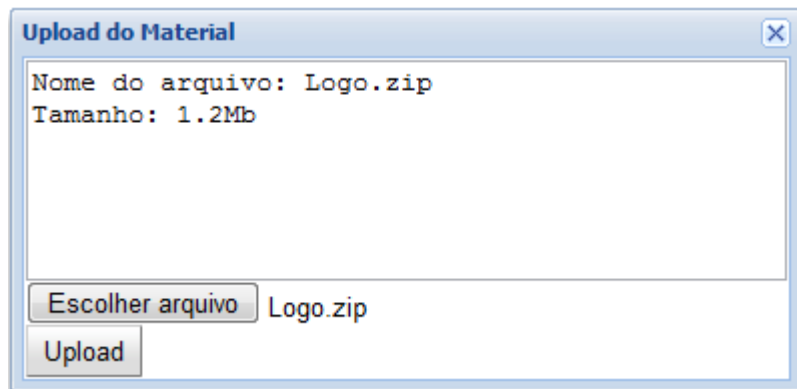


Figura 31 – Tela para disponibilizar o material da tarefa

Na tela Revisar Tarefas (Figura 32), o coordenador pode informar se a tarefa cujo material foi disponibilizado atende o solicitado ou não. Pode-se observar uma grade com as tarefas disponibilizadas e dois botões cujas funcionalidades são descritas a seguir:

- a) **Baixar Arquivo Tarefa:** quando selecionada a tarefa da grade e clicado no botão, será efetuado o *download* do arquivo da tarefa que a representa;
- b) **Trâmite Tarefa:** quando selecionada a tarefa da grade e clicado no botão, será aberta uma nova tela (Figura 33). Nessa tela o usuário tem a opção de colocar a tarefa como “Concluída” ou “Com Problemas” e inserir uma observação. O sistema irá mudar o *status* da tarefa e incluirá uma ocorrência. Se a tarefa for colocada como “Com problemas”, o usuário terá que assumi-la novamente pela tela Assumir Tarefa (Figura 29).

Revisar Tarefas

Tarefas

Nome Projeto	Nome Unidade	Título	Status	Data Entrega	Responsável
Curso Microsoft Office	Microsof PowerPoint	Criar apresentação sobre os princípios da Mi...	Pronta para Revisão	10/11/2009	fabi

Ocorrências Tarefa

Data Ocorrência	Usuário	Descrição	Observação
10/11/2009	fabi	Tarefa Aberta	
10/11/2009	fabi	Status alterado para Em Andamento	
10/11/2009	fabi	Status alterado para Pronta para Revisão	

Figura 32 – Tela Revisar Tarefas

Tarefa com Problemas

Status:
 Com Problemas

Observação:
 Apresentação não está compatível com o que foi solicitado.

Figura 33 – Tela trâmite tarefa

A tela Disponibilizar Unidades (Figura 34) é usada pelo coordenador para disponibilizar o material referente a unidade de aprendizagem que está sob sua responsabilidade. A unidade de aprendizagem somente ficará visível após todas as tarefas referentes a esta unidade estiverem concluídas. Nesta tela pode-se observar uma grade com as unidades “Concluídas” e um botão chamado “Disponibilizar Material” que quando selecionada a unidade da grade e clicado no botão, será aberta uma nova tela para o coordenador poder fazer o *upload* do material dessa unidade, conforme ilustrado na Figura 35.

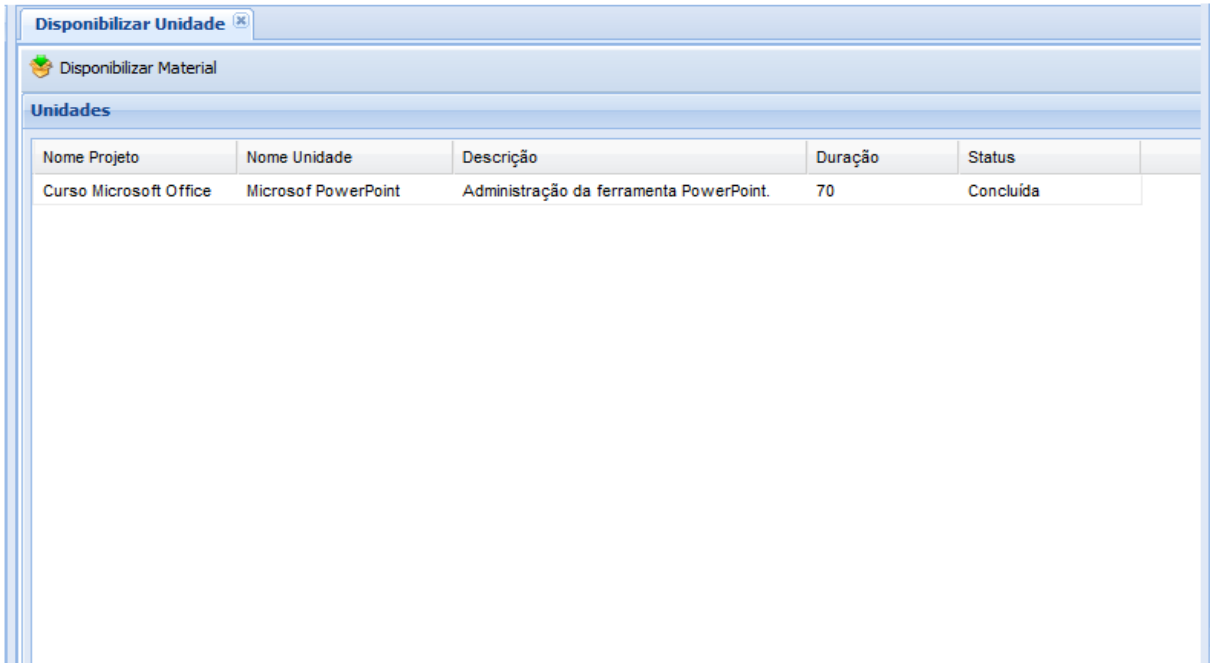


Figura 34 – Tela disponibilizar unidades revisão

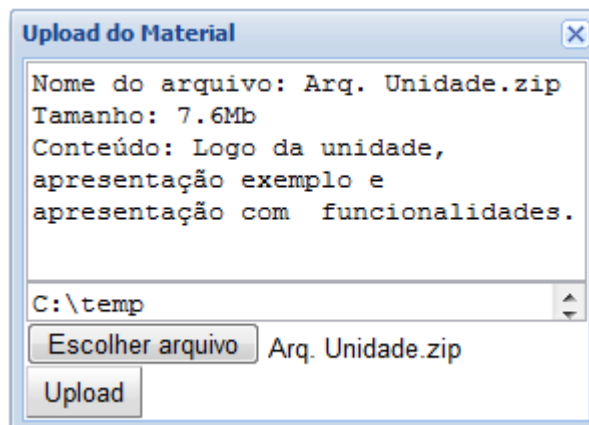


Figura 35 – Tela para disponibilizar o material da unidade

Nessa tela (Figura 35), o coordenador escolherá o arquivo, colocará uma observação e terá que informar o diretório raiz do arquivo que está sendo disponibilizado para ser utilizado futuramente na geração do arquivo no formato SCORM. O sistema irá mandar este arquivo para o servidor, mudará o *status* da unidade para “Pronta para Revisão” e gravará uma ocorrência para esta unidade.

A tela Validar Unidade de Aprendizagem (Figura 36), visualizada pelo professor conteudista, é usado para fazer o *download* do arquivo disponibilizado referente à unidade de aprendizagem e informar se os objetivos da unidade de aprendizagem foram concluídos ou não. Nesta tela pode-se observar uma grade com as unidades com *status* de “Pronta para Revisão” e dois botões cujas funcionalidades são descritas a seguir:

- a) Baixar arquivo unidade: quando selecionado a unidade da grade e clicado no botão, será efetuado o *download* do arquivo que representa a unidade.

- b) Trâmite unidade: quando selecionada a unidade da grade e clicado no botão, será aberta uma nova tela (Figura 37). Nessa tela são informados quais objetivos das unidades foram cumpridos. Se algum dos objetivos não for cumprido, o sistema colocará a unidade como “Com Problemas” e incluirá uma ocorrência. Se todos os objetivos foram cumpridos, o sistema colocará a unidade como “Concluída” e incluirá uma ocorrência.

The screenshot shows a window titled "Validar Unidades" with two main sections: "Unidades" and "Ocorrências Unidade".

Unidades

Nome Projeto	Nome Unidade	Status	Descrição	Duração
Curso Microsoft Office	Microsof PowerPoint	Pronta para Revisão	Leccionação da ferramenta PowerPoint.	70

Ocorrências Unidade

Data Ocorrência	Usuário	Descrição	Observação
09/11/2009	jckr	Status alterado para Pronta para Revisão	Tamanho total do arquivo igual a 6.7Mb.
09/11/2009	maria	Status alterado para Com Problemas	Faltou criar arquivo de exemplo.
11/11/2009	jckr	Status alterado para Pronta para Revisão	Nome do arquivo: Arq. Unidade.zip Tamanho: 7....

Figura 36 – Tela Validar Unidade de Aprendizagem

The screenshot shows a dialog box titled "Objetivos da Unidade de Aprendizagem" with a list of objectives and checkboxes. At the bottom, there are "Salvar" and "Cancelar" buttons.

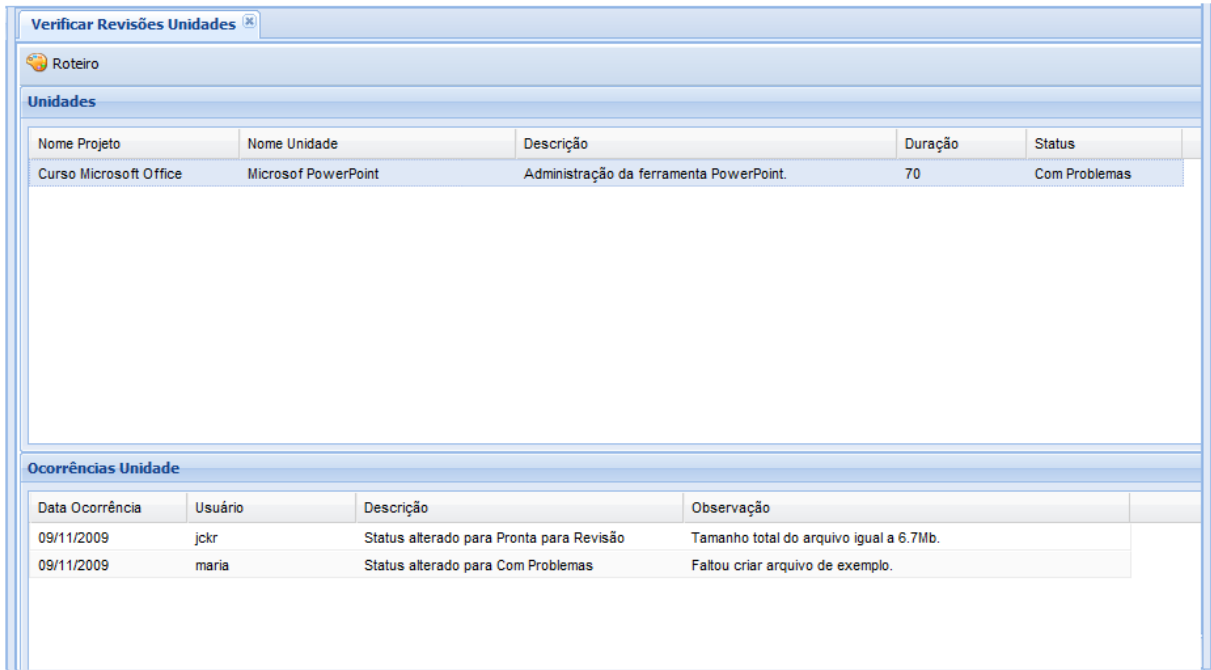
Objetivos da Unidade de Aprendizagem

- Objetivos
- Avaliar a atividade desenvolvida.
- Apresentar a ferramenta aos alunos.
- Demonstrar exemplos.
- Realizar atividades.

Buttons:

Figura 37 – Tela para informar quais objetivos foram cumpridos

A tela Verificar Revisões Unidades (Figura 38) serve para o coordenador incluir novas tarefas caso algum dos objetivos da unidade de aprendizagem não tiverem sido cumpridos. Nesta tela pode-se observar uma grade com as unidades com *status* de “Com Problemas” e um botão chamado “Roteiro”. Quando selecionada a unidade da grade e clicado no botão, será aberta uma nova tela do cadastro de roteiro (Figura 28) para poder incluir novas tarefas a serem realizadas.



Nome Projeto	Nome Unidade	Descrição	Duração	Status
Curso Microsoft Office	Microsof PowerPoint	Administração da ferramenta PowerPoint.	70	Com Problemas

Data Ocorrência	Usuário	Descrição	Observação
09/11/2009	jckr	Status alterado para Pronta para Revisão	Tamanho total do arquivo igual a 6.7Mb.
09/11/2009	maria	Status alterado para Com Problemas	Faltou criar arquivo de exemplo.

Figura 38 – Caso de uso verificar revisões unidades

A tela Gerar Arquivo SCORM (Figura 39) é utilizada pelo coordenador para fazer a geração do arquivo referente à unidade de aprendizagem no formato SCORM. Nessa tela pode-se observar uma grade com as unidades de aprendizagem com *status* “Concluída” e um botão chamado “Gerar Arquivo SCORM”. Quando selecionada a unidade da grade e clicado no botão, será gerado um arquivo no formato SCORM no diretório “C:\Uploads\SCORM”. O arquivo gerado pode ser visualizado na Figura 40.

Gerar Arquivo SCORM				
Unidades				
Nome Projeto	Nome Unidade	Status	Descrição	Duração
Curso Microsoft Office	Unidade 02	Concluída	Descricao da unidade 02	322

Ocorrências Unidade			
Data Ocorrência	Usuário	Descrição	Observação
15/12/2009	jckr	Unidade Aberta	
16/12/2009	jckr	Status alterado para Pronta para Revisão	teste unidade
16/12/2009	maria	Status alterado para Concluída	

Figura 39 – Tela Gerar Arquivo SCORM

```

- <manifest xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.xsd
  http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p3.xsd http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd http://www.imsglobal.org/xsd/imsss
  imsss_v1p0.xsd" xmlns="http://www.imsglobal.org/xsd/imscp_v1p1" xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
  xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3" xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3"
  xmlns:imsss="http://www.imsglobal.org/xsd/imsss" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <organizations default="Curso Microsoft Office">
- <organization identifier="Curso Microsoft Office">
  <title>Curso Microsoft Office</title>
- <item identifier="Unidade 02" identifierref="RES-Unidade 02" isVisible="true">
  <title>Unidade 02</title>
  + <imsss:sequencing>
  </item>
- <imsss:sequencing>
  <imsss:controlMode choice="true" choiceExit="true" flow="true" forwardOnly="false" useCurrentAttemptObjectiveInfo="true"
    useCurrentAttemptProgressInfo="true" />
  + <imsss:objectives>
  <randomizationControls randomizationTiming="never" selectCount="0" reorderChildren="false" selectionTiming="never" />
  <deliveryControls tracked="true" completionSetByContent="false" objectiveSetByContent="false" />
  <constrainedChoiceConsiderations preventActivation="false" constrainChoice="false" />
  <rollupConsiderations requiredForSatisfied="always" requiredForNotSatisfied="always" requiredForCompleted="always" requiredForIncomplete="always"
    measureSatisfactionIfActive="true" />
  </imsss:sequencing>
  </organization>
</organizations>
- <resources>
  <resource identifier="RES-Unidade 02" type="webcontent" href="Unidade 02/c:UploadSCORM" xml:base="" adlcp:scormType="sco" />
</resources>
</manifest>

```

Figura 40 – Arquivo gerado SCORM

3.4 RESULTADOS E DISCUSSÃO

Os resultados encontrados com o término da ferramenta são muito satisfatórios, pois com a utilização do Processo Sophia em conjunto com a matriz de *design* instrucional nas fases de planejamento e produção e utilizando SCORM para a fase de distribuição, atendem os objetivos iniciais.

Quanto à utilização do Processo Sophia para normalização do processo de produção, pode-se observar que a ferramenta proposta não contempla totalmente a etapa de distribuição, pois ela prevê a avaliação do objeto pela WebTutoria e os alunos, visando obter *feedback* para melhoria contínua.

A coluna período da matriz de *design* instrucional não foi utilizada na ferramenta pois diz respeito à execução dos OAs e não ao processo de produção. Já a coluna ferramentas não foi utilizada pois diz respeito aos bastidores da produção dos OAs.

No Quadro 21 é apresentado um comparativo das características da ferramenta desenvolvida com os trabalhos correlatos. Estas características foram encontradas em suas documentações. Pode-se observar que a ferramenta desenvolvida não possui a geração de relatórios com históricos da produção. Este seria um próximo estágio para garantir conformidades entre os dados. Outro fato interessante no comparativo das características é que a ferramenta proposta é a única que utiliza o formato SCORM para distribuição dos OAs.

Características	Processo OA	GROA	Ferramenta desenvolvida
Interface web	X	X	X
Validação de permissões, acessos	X		X
Distribuição dos OAs	X	X	X
Acompanhamento da produção	X		X
Recuperação dos OAs		X	X
Validação dos OAs conforme os objetivos	X		X
Geração de relatórios com históricos		X	
Utilização de normalizações na produção	X		X
Geração no formato SCORM			X

Quadro 21 – Comparativo entre as ferramentas

4 CONCLUSÕES

Com o notável crescimento da educação à distância nos últimos anos, a ferramenta desenvolvida deverá ser mais uma opção para auxiliar a produção e distribuição dos OAs.

A inspiração da ferramenta veio de Filatro (2008), pois ela especificou uma matriz de *design* instrucional que pode ser usada na fase de planejamento dos OAs, e Pessoa e Benitti (2008), que propuseram uma normalização no processo de produção de OAs.

No início do desenvolvimento da ferramenta foram elencadas funcionalidades que deveria atender: elencar papéis aos usuários, preenchimento da matriz de *design* instrucional, criação de roteiros e tarefas relacionadas aos desenvolvedores, disponibilizar OAs para revisão, validação dos OAs e geração de arquivos no formato SCORM. Ao término, pode-se observar que as funcionalidades foram criadas, atingindo os objetivos.

As tecnologias utilizadas para construção se mostraram eficientes, em especial a utilização do *framework* GWT-Ext. Com ele pode-se criar interfaces “amigáveis”, obtendo um código legível e com grande variedade de componentes.

No processo de desenvolvimento foram utilizados os paradigmas de orientação a objetos, implementado na linguagem Java. O uso desta tecnologia gerou diversas dificuldades e desafios na implementação devido a pouca experiência no ambiente de desenvolvimento. Apesar desses contratemplos os objetivos iniciais propostos foram alcançados com êxito.

Esta ferramenta possui algumas limitações. Somente é possível acessar o sistema utilizando-se os navegadores *Google Chrome* e *Mozilla Firefox*. A utilização do navegador *Internet Explorer* distorce os componentes criados pelo GWT-Ext, ficando inviável sua utilização. Essa limitação é do próprio *framework*.

A pasta de armazenamento do material das tarefas e do arquivo no formato SCORM está fixa no servidor (“C:\Uploads\SCORM”), ou seja, não é possível usuários da ferramenta alterar o diretório de localização dos arquivos.

4.1 EXTENSÕES

Como sugestão para trabalhos futuros podem ser desenvolvidas ou aperfeiçoadas as seguintes funcionalidade:

- a) desenvolver a geração de relatórios históricos e estatísticos;
- b) implementar uma forma de envio de email ao responsável quando determinada tarefa ou unidade de aprendizagem não for atendida;
- c) permitir a configuração das pastas de armazenamento dos arquivos;
- d) compatibilizar a ferramenta com o *Internet Explorer*;
- e) alterar o esquema de guardar os arquivos de upload, para que o sistema trate o nome do arquivo;
- f) na geração do arquivo SCORM, poder seleccionar mais de uma unidade de aprendizagem para compor o curso no formato SCORM.

REFERÊNCIAS BIBLIOGRÁFICAS

ADVANCED DISTRIBUTED LEARNING. [S.1.], 2009. Disponível em: <<http://www.adlnet.org>>. Acesso em: 22 mar. 2009.

ARAGÃO, M. L. Ajax em Java com Google Web Toolkit. **Revista Guj**, São Paulo, v. 18, n. 22, p. 3-25, nov. 2008.

BETTIO, R. W.; MARTINS, A. **Objetos de aprendizado**: um novo modelo direcionado ao ensino a distância. [S.1.], 2004. Disponível em: <<http://www.universia.com.br/materia/materia.jsp?id=5938>>. Acesso em: 24 mar. 2009.

FILATRO, A. **Design instrucional na prática**. 1. ed. São Paulo: Pearson Editora, 2008. 192 p.

GUILLERMO, O. E. P.; TAROUCO, L. M. R.; ENDRES, L. A. M. Desenvolvimento de objetos educacionais: experimento em hidráulica. **Revista Novas Tecnologias na Educação**, Porto Alegre, v. 3, n. 2, p. 1-12, nov. 2005.

KRATZ, R. A. et al. Fábrica de adequação de objetos de aprendizagem. **Revista Brasileira de Informática na Educação**, São Leopoldo, v. 15, n. 3, p. 25-38, dez. 2007.

LTSC IEEE. **Standard for information technology: education and training systems – learning objects and metadata**. [S.1.], 2007. Disponível em: <<http://ltsc.ieee.org/wg12/>>. Acesso em: 25 mar. 2009.

LUFER, S. **Essencial RPC**. [S.1.], 2008. Disponível em: <<https://woc.ipca.pt/est/class/getmaterial.do?idclass=445&idyear=12>>. Acesso em: 24 out. 2009.

MIRANDA, R. M. **GROA**: um gerenciador de repositórios de objetos de aprendizagem. 2004. Tese (Doutorado em Ciências da Computação) – Faculdade de Ciências da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.

PESSOA, M. C.; BENITTI, F. B. V. Proposta de um processo para produção de objetos de aprendizagem. **Revista Hífen**, Santa Catarina, v. 32, n. 62, p. 172-180, dez. 2008.

RIVED. **Rede interativa virtual de educação**. [S.1.], 2009. Disponível em: <<http://www.rived.mec.gov.br/>>. Acesso em: 24 mar. 2009.

SAMPAIO, M. C. **Material sobre UML**: disciplina de sistemas de informação II. [S.1.], 2002. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/estado/diag_estados.htm>. Acesso em: 30 out. 2009.

SANTOS, R. S. **Pela primeira vez mais da metade da população já teve acesso ao computador.** [S.l.], 2008. Disponível em:
<<http://www.cgi.br/publicacoes/artigos/artigo50.htm>>. Acesso em: 24 mar. 2009.

VAHLDICK, A. **CELINE:** um modelo para utilização e adaptação de conteúdo SCORM em ambientes inteligentes. 2008. 138 f. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Itajaí, Itajaí, 2008.

VAHLDICK, A.; RAABE, A.L.A. Adaptação de conteúdo SCORM em ambientes inteligentes de aprendizagem. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 19, 2008, Fortaleza. Anais... Fortaleza: SBC, 2008. CD-ROM.

APÊNDICE A – Dicionário de dados

O dicionário de dados tem por objetivo detalhar a estrutura do banco de dados da ferramenta. Os tipos de dados de cada campo foram padronizados na seguinte forma:

- integer: caracteriza um valor numérico do valor inteiro, onde seu tamanho é definido pelo valor entre parênteses;
- varchar: lista de caracteres, onde seu tamanho é definido pelo valor entre parênteses;
- date: caracteriza um campo do tipo data.

O dicionário de dados utiliza o seguinte padrão: nome da tabela, descrição do campo, nome do campo, tipo do campo, se é chave primaria, se é obrigatório e se é chave estrangeira. No quadro 22, pode ser visualizado dicionário de dados da ferramenta.

USUARIO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
NICK	Nick	Varchar(30)	Não	Sim	Não
SENHA	Senha	Varchar(25)	Não	Não	Não
NOME	Nome completo	Varchar(80)	Não	Não	Não
EMAIL	Email	Varchar(50)	Não	Não	Não
IDF_PAPÉL	Papel do Usuário	Integer(10)	Não	Sim	SIM
PAPEIS					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
PAPÉL	Papel	Varchar(30)	Não	Sim	Não
PROJETO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
NOME	Nome do Projeto	Varchar(80)	Não	Sim	Não
ASSUNTO	Assunto	Varchar(280)	Não	Não	Não
DATA_INICIO	Data Inicio Proj.	Date	Não	Sim	Não
DATA_ENTREGA	Data Entrega Proj.	Date	Não	Não	Não
OBSERVACOES	Observacoes	Varchar(250)	Não	Não	Não

IDF_USUARIO	Usuário do Projeto	Integer(10)	Não	Sim	Sim
IDF_CONTEUDISTA	Conteudista	Integer(10)	Não	Sim	Sim
UNIDADE					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
NOME	Nome do Projeto	Varchar(80)	Não	Sim	Não
DESCRICAO	Descrição	Varchar(380)	Não	Não	Não
DURACAO	Duração	Varchar(80)	Não	Não	Não
IDF_PROJETO	Projeto da Unidade	Integer(10)	Não	Sim	Sim
STATUS	Estado da Unidade	Varchar(40)	Não	Sim	Não
NOME_ARQUIVO	Nome do Arquivo	Varchar(280)	Não	Não	Não
ARQUIVO_RAIZ	Arquivo Raiz	Varchar(260)	Não	Não	Não
ATIVIDADE					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
DESCRICAO	Descrição	Varchar(80)	Não	Sim	Não
IDF_UNIDADE	Unid. da Avaliação	Integer(10)	Não	Sim	Sim
AVALIACAO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
IDF_UNIDADE	Unid. da Avaliação	Integer(10)	Não	Sim	Sim
IDF_TIPO_AVAL	Tipo de Avaliacao	Integer(10)	Não	Sim	Sim
DESCRICAO	Descrição	Varchar(280)	Não	Sim	Não
TIPO_AVALIACAO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
TIPO	Tipo Avaliação	Varchar(80)	Não	Sim	Não
CONTEUDO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
DESCRICAO	Descricao	Varchar(180)	Não	Sim	Não
IDF_UNIDADE	Unid. do conteúdo	Integer(10)	Não	Sim	Sim

OBJETIVO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
DESCRICA0	Descrição	Varchar(280)	Não	Sim	Não
IDF_UNIDADE	Unid. do Objetivo	Integer(10)	Não	Sim	Sim
ROTEIRO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
TEXTO	Script do Roteiro	Varchar(280)	Não	Sim	Não
IDF_UNIDADE	Unid. do Roteiro	Integer(10)	Não	Sim	Sim
OCORRENCIA_UNIDADE					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
DESCRICA0	Nome do Projeto	Varchar(320)	Não	Sim	Não
DATA	Data da Ocorrência	Date	Não	Sim	Não
OBSERVACA0	Observação	Varchar(360)	Não	Não	Não
IDF_USUARIO	Usuário da Ocorr.	Integer(10)	Não	Sim	Sim
IDF_UNIDADE	Ocorrência da Unid	Integer(10)	Não	Sim	Sim
OCORRENCIA_TAREFA					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
DESCRICA0	Nome do Projeto	Varchar(320)	Não	Sim	Não
DATA	Data da Ocorrência	Date	Não	Sim	Não
OBSERVACA0	Observação	Varchar(360)	Não	Não	Não
IDF_USUARIO	Usuário da Ocorr.	Integer(10)	Não	Sim	Sim
IDF_TAREFA	Ocorr. Da tarefa	Integer(10)	Não	Sim	Sim
ATIVIDADE_ROTEIRO					
Descrição do Campo	Campo	Tipo	PK	Obrigatório	FK
ID	Identificação	Integer(10)	Sim	Sim	Não
IDF_ROTEIRO	Roteiro	Integer(10)	Não	Sim	Sim
TITULO	Título da Tarefa	Varchar(120)	Não	Sim	Não
DESCRICA0	Descrição	Varchar(240)	Não	Não	Não

DATA_ENTREGA	Data de Entrega	Date	Não	Sim	Não
IDF_RESP	Responsável	Integer(10)	Não	Sim	Sim
STATUS	Estado da Tarefa	Varchar(40)	Não	Sim	Não
NOME_ARQUIVO	Nome do Arquivo	Varchar(240)	Não	Não	Não

Quadro 22 - Dicionário de Dados