

UNIVERSIDADE FEDERAL DE SANTA CATARINA

**Uma Interface Gráfica Para o Ambiente Telis
Focada em Acessibilidade para Cegos**

Felipe Alexandre Ferreira

**Florianópolis
2011 / 1**

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

Uma Interface Gráfica Para o Ambiente Telis
Focada em Acessibilidade para Cegos

Felipe Alexandre Ferreira

Relatório de Trabalho de Conclusão de
Curso, apresentado como requisito para
aprovação na disciplina INE5434 -
Trabalho de Conclusão de Curso II
Orientador: Prof. Luiz Fernando Bier
Melgarejo

Florianópolis - SC

2011 / 1

Dedico este trabalho a todos os que constroem a ciência como uma ferramenta para atender as reais demandas do povo.

Agradecimentos

Agradeço a minha família, em especial minha mãe, por todo o apoio que me deu durante toda a vida, sem o qual eu não poderia chegar até aqui.

Agradeço ao meu orientador, professor Melgarejo, e aos membros da banca pelo apoio e compreensão, mesmo com as dificuldades e atrasos.

Agradeço a Walkiria, por todo apoio, amor e carinho que recebi.

Agradeço aos amigos, por todos os momentos divertidos que passamos juntos.

"O software privado é dependência e isso leva à colonização eletrônica. As empresas do software privado querem colonizar todos os países: eles tomaram os Estados Unidos, Europa e outros lugares do mundo"

Richard Stallman

Abstract

This work deals with the development of a Web graphical user interface for the Telis environment, respecting the general principles of accessibility and WCAG 2.0 standards, taking also into account usability principles. The objective of this work is to enable a blind person can use the Telis environment and contribute to enhancing the constructionist approach in the case of a blind person. To validate the developed tool was used validation tools and Total Validator Access Monitor for assessing compliance with the WCAG 2.0 guidelines and monitoring was also done by use of a blind user. At the end of development, the use by a blind user of developed tool has been proven feasible in field work. It was opted for not use any technology that relies on proprietary software to use, either server or client. It will also be addressed specific considerations on the possibilities and constraints of the didactic use of the tool for the blind Telis today and its potential use in future as this paper notes.

Keywords: Web accessibility, educational programming language, Constructivism, Constructionism, Web Application, Education for blind.

Resumo

Este trabalho trata do desenvolvimento de uma interface gráfica Web para o ambiente Telis, respeitando princípios gerais de acessibilidade e normas WCAG 2.0, levando também em conta princípios de usabilidade. O objetivo deste trabalho é viabilizar que uma pessoa cega possa utilizar o ambiente Telis e contribuir para potencializar a abordagem construcionista no caso de uma pessoa cega. Para validação da ferramenta desenvolvida foi utilizado ferramentas de validação Total Validator e Access Monitor para avaliar conformidade com as diretrizes WCAG 2.0 e também foi feito acompanhamento de uso por um usuário cego. Ao final do desenvolvimento, o uso por um cego da ferramenta desenvolvida foi comprovado viável em pesquisa de campo. Optou-se por não utilizar nenhuma tecnologia que dependa de software proprietário para a utilização, seja no servidor ou no cliente. Também é tratado considerações específicas sobre as possibilidades e impedimentos do uso didático da ferramenta Telis para cegos hoje e seu uso potencial no futuro, dado apontamentos deste trabalho.

Palavras-chave: Acessibilidade. Linguagem de programação educacional. Construtivismo. Construcionismo. Aplicação web. Educação de cegos

Lista de ilustrações

Ilustração 1 - Exemplo de caso de tarefa.....	44
Ilustração 2 - Editor de Apliques.....	48
Ilustração 3 - Editor de Modelos.....	49
Ilustração 4 - Editor de Sons.....	50
Diagrama 1 - Diagrama do modelo relacional utilizado.....	52
Diagrama 2 - Carregamento de modelos e apliques.....	54

Lista de tabelas

Tabela 1: Exemplo de medidas de usabilidade global.....	24
---	----

Lista de abreviaturas

AJAX – Asynchronous Javascript and XML

API – Application Programming Interface

ARIA – Accessible Rich Internet Application

ATAG – Authoring Tool Accessibility Guidelines

CSS – Cascading Style Sheets

DOM – Document Object Model

GNU GPL – GNU General Public License

HTML – Hypertext Markup Language

ISO – International Organization for Standardization

JAAPI – Java Accessibility API

JPA – Java Persistence API

RIA – Rich Internet Application

W3C – World Wide Web Consortium

WAI – Web Accessibility Initiative

WCAG – Web Content Accessibility Guidelines

XML – Extensible Markup Language

Sumário

1 Introdução.....	12
1.1 Organização do trabalho.....	12
1.2 Descrição do problema.....	12
1.3 Justificativa.....	13
1.4 Objetivos.....	15
1.4.1 Objetivos Gerais.....	15
1.4.2 Objetivos Específicos.....	15
2 Revisão Bibliográfica.....	17
2.1 Construtivismo.....	17
2.1.1 Construtivismo Cognitivo.....	17
2.1.2 Construcionismo.....	18
2.1.3 Construtivismo Social.....	18
2.1.4 Aprendizagem situada/ Cognição distribuída.....	19
2.2 Formação de conceitos em crianças cegas.....	19
2.2.1 Aquisição de conceitos por pessoas cegas.....	20
2.3 Representação espacial em crianças cegas.....	20
2.4 A Etnomatemática.....	21
2.5 Usabilidade.....	22
2.5.1 Avaliação informal de usabilidade e heurísticas.....	24
2.5.1.1 As heurísticas de Nielsen.....	25
2.6 Acessibilidade.....	26
2.6.1 Padrões de acessibilidade na web.....	28
2.6.1.1 WCAG Samurai.....	29
2.6.1.2 Web Content Accessibility Guidelines 2.0.....	32
2.6.1.3 Comparação entre WCAG 2.0 e WCAG+Samurai.....	33
2.6.1.4 Authoring Tool Accessibility Guidelines (ATAG).....	34
2.6.1.5 WAI-ARIA.....	35
2.6.2 Softwares leitores de tela.....	36
2.6.2.1 Orca.....	37
2.6.2.2 NVDA.....	37
2.6.2.3 Jaws.....	37
2.7 A linguagem e o ambiente Telis.....	37
2.7.1 Considerações sobre Telis e acessibilidade.....	38

2.8 Dix.....	39
2.8.1 Dix e acessibilidade.....	39
2.8.1.1 Análise da acessibilidade de páginas Dix.....	40
2.9 Ferramentas similares.....	41
3 Tecnicas de projeto de Interface.....	43
3.1 A abordagem The Usage-Centered Design.....	43
3.1.1 Modelo de papéis de usuários.....	43
3.1.2 Modelo de tarefas.....	44
3.1.2.1 Mapa de casos de tarefas.....	45
3.1.3 Modelo de conteúdo.....	45
3.2 Considerações sobre tecnicas de projeto de Interface.....	45
4 Desenvolvimento.....	46
4.1 Persistência e comunicação com o servidor.....	50
4.2 Problemas encontrados.....	54
5 Comentários sobre usabilidade baseado nas heurísticas de Nielsen.....	56
5.1 Visibilidade do estado do sistema.....	56
5.2 Relação entre sistema e mundo real.....	56
5.3 Controle e liberdade do usuário.....	56
5.4 Consistência e padrão.....	56
5.5 Prevenção de erros.....	57
5.6 Reconhecimento ao invés de memória.....	57
5.7 Flexibilidade e eficiência no uso.....	57
5.8 Design e estética minimalista.....	57
5.9 Ajuda usuários a reconhecer, diagnosticar e recuperar erros.....	57
5.10 Ajuda e documentação.....	58
6 Trabalhos Futuros e Considerações Finais.....	59
7 Referências bibliográficas.....	62
APÊNDICE A – Artigo sobre o trabalho desenvolvido.....	68
APÊNDICE B – Código-fonte da Applet Java (Somente partes novas).....	78
APÊNDICE C – Código-fonte das classes utilizadas por ambos, Applet e Servidor.....	146
APÊNDICE D – Código-fonte da aplicação Web.....	169

1 Introdução

1.1 Organização do trabalho

Este relatório está organizado da seguinte forma: Na seção 1, estarei fornecendo uma apresentação geral do trabalho. Na seção 2, farei a revisão bibliográfica, onde apresentarei conceitos que estão ligados e trazem implicações para esta monografia, independente de estarem ou não no domínio de conhecimento da ciência da computação. Incluindo aprendizagem em crianças cegas, Orientação espacial em cegos, , acessibilidade de software e web – princípios e normas, e uma apresentação do software Telis. No capítulo 3 irei discutir técnicas de projeto de interface. No capítulo 4 irei discutir sobre especificidades técnicas da aplicação desenvolvida e no capítulo 5 irei avaliar a aplicação de acordo com a usabilidade. No capítulo 6 irei avaliar a acessibilidade do programa obtido.

1.2 Descrição do problema

O computador tem sido uma ferramenta usada na educação em diferentes formas. Uma teoria educacional que tem grande foco no uso do computador é o Construcionismo, um importante ramo do construtivismo criado por Seymour Papert. Os construcionistas alegam que as construções que ocorrem no mundo tem um papel importante para as construções mentais (Papert, 1993, p. 142), e o computador é uma ferramenta útil devido a sua capacidade de representar idéias abstratas em formas concretas. Para implementar este conceito, Seymour Papert criou a linguagem Logo, uma linguagem de programação de propósitos educacionais, em que é possível, através de comandos da linguagem, controlar um cursor, a tartaruga gráfica, que é capaz de desenhar na tela (Papert, 1980, p. 55), uma abordagem claramente visual.

O Laboratório de Software Educacional (EDUGRAF) da Universidade Federal de Santa Catarina (UFSC) desenvolveu a linguagem e ambiente de desenvolvimento Telis, com principal objetivo suportar aprendizes que nunca tiveram contato com uma linguagem de programação. Telis tem objetivo de possibilitar uma abordagem construtivista, e fornece, além de recurso análogo a tartaruga gráfica da

linguagem logo, abstrações que permitem apresentar conceitos avançados da ciência da computação, como orientação a objetos, programação concorrente, programação distribuída, eventos, entre outros.

Entretanto, tendo em vista que o ambiente Telis não foi adaptado para garantir acessibilidade, pode-se perguntar: Quais as dificuldades para que um cego utilize esta ferramenta educacional? Estudando acessibilidade em software e questões específicas do aprendizado em cegos, foi analisada esta questão. Foi observado que o ambiente Telis encontra-se atualmente inacessível, e neste trabalho, foi desenvolvido uma nova interface gráfica com o usuário, a qual, diferente da versão anterior, além de acessível é também uma aplicação Web, respeitando as normas da Web Content Accessibility Guidelines. Este passo é fundamental para que um cego ou mesmo um usuário com certas outras deficiências pudessem usar a ferramenta. Este trabalho viabiliza o uso do ambiente Telis por um cego e de todo o subconjunto da linguagem que não é inviabilizado por sua inabilidade, por exemplo, para criar programas que utilizem teclado e sons. Entretanto, este trabalho não resolve o problema da acessibilidade em todos os diversos usos da ferramenta Telis. Questões como se há uma alternativa ao ator gráfico para representação espacial no caso de cegos, como produzir aplicativos acessíveis com Telis e que novas primitivas poderia-se adicionar a linguagem para melhor atender as necessidades deste público, foram brevemente tratadas como apontamentos para trabalhos futuros.

1.3 Justificativa

Uma recorrente preocupação do ensino regular refere-se modo de aprendizagem dos cegos e os recursos necessários para esta aprendizagem (LAPLANE e BATISTA, 2003, APUD BATISTA, 2005). Quanto ao uso de computadores por cegos, embora o uso de computadores por cegos e demais portadores de necessidades especiais, já é facilitado por um amplo leque de ferramentas de acessibilidade, aplicações ainda podem ser completa ou parcialmente inacessíveis.

No campo de ferramentas educacionais de enfoque construtivista, como o Telis, há um grande número de ferramentas inspiradas na linguagem Logo,

ferramenta criada por Seymour Papert para viabilizar seu foco particular do construtivismo, a que chamou construcionismo. Apesar de importante contribuição para a educação, a abordagem proposta por seymour papert parece fortemente visual, com grande ênfase na tartaruga gráfica. Um importante passo a ser dado portanto parece ser investigar as particularidades desta abordagem para cegos, ou seja, de que forma, na ausência de visão, estes usuários podem ter experiência semelhante na construção de conhecimento através do computador.

A importância de dar semelhante oportunidade a pessoas com diferentes dificuldades ou necessidades pessoais é baseada no princípio da isonomia, o que é ainda mais relevante pelo fato de a educação ser um direito básico reconhecido como direito fundamental na maioria dos países e como direito humano na Declaração Universal dos Direitos Humanos. Sendo que também é papel da universidade pública produzir conhecimento para atender as necessidades populares, fica evidente a relevância deste tema.

Telis tem particularidades que justificam seu foco nesta ferramenta para este trabalho. Embora há diversas alternativas de linguagem e ambiente de programação de propósitos educacionais, Telis é uma ferramenta avançada, e que apresenta conceitos modernos de ciência da computação como concorrência, computação distribuída, orientação a objetos, entre outros, e foi utilizado em aulas de fases iniciais de ciência da computação na Universidade Federal de Santa Catarina, e se mostrou avançado o suficiente para que alunos sem experiência anterior em programação fossem capazes de produzir programas em que tais conceitos estão presentes.

Outro forte motivo para o enfoque no ambiente Telis, é ser software livre e feito para produzir software livre, encorajando a livre produção e compartilhamento de conhecimento. Todas as aplicações produzidas em Telis que são publicadas apresentam um aviso de que o conteúdo é software livre sobre a licença GPL. O código Telis pode ser visto na mesma página gerada pela publicação e o aplicativo gerado pode ser facilmente importado em um ambiente Telis por qualquer pessoa e livremente editado. O software livre atende a interesses humanitários e coletivos por ser um modelo de produção baseado na coletivização do conhecimento, opondo-se a um modelo baseado no monopólio do conhecimento. Esta justificativa é ainda mais

coerente se novamente lembrarmos que a universidade pública deve produzir conhecimento que atenda a interesses populares.

Foi verificado que o ambiente Telis anterior a este trabalho é inacessível. Sendo assim, qualquer outra atividade envolvendo Telis para cegos ficaria limitada a este problema, e portanto a solução deste problema se mostrou o primeiro passo essencial para se atingir os objetivos deste trabalho. Optou-se pelo desenvolvimento de uma nova interface gráfica Web para o ambiente Telis. Tal abordagem tem a vantagem de poder utilizar de normas de acessibilidade da W3C, organização que normatiza padrões Web internacionais. Apresenta também a vantagem adicional de trazer benefícios também mesmo aos usuários de Telis que não tem necessidades especiais, mas que irão se beneficiar de ter o ambiente Telis com seus dados acessíveis de qualquer computador com Internet, e grande parte dos usuários apresenta experiência anterior com a Web.

Criar uma interface Web funcional, totalmente acessível e com boa usabilidade se mostrou tarefa suficiente para um trabalho de conclusão de curso, portanto demais apontamentos no sentido de melhor atingir aos objetivos gerais deste trabalho serão sugeridos como trabalhos futuros.

1.4 Objetivos

1.4.1 Objetivos Gerais

- Potencializar a abordagem construcionista na educação de pessoas com deficiência sem depender de software proprietário

1.4.2 Objetivos Específicos

- Desenvolver uma interface Gráfica para o Ambiente Telis acessível para cegos utilizando um software leitor de telas, sem depender de software proprietário
- Fornecer apontamentos para trabalhos futuros indicando quais futuras implementações podem contribuir para melhor atender as necessidades

específicas de usuários cegos ou com outras deficiência em relação a uma linguagem de programação de propósitos educacionais.

2 Revisão Bibliográfica

2.1 Construtivismo

Construtivismo é o nome dado a teorias de aprendizagem baseada em uma alternativa epistemológica das teorias objetivista do conhecimento (SWAN, 2005). Ainda segundo Karen Swan, tanto objetivistas quanto construtivistas acreditam em um mundo concreto, mas a diferença entre ambos se deve ao fato de que objetivistas acreditam que o significado é inerente ao mundo, enquanto os construtivistas acreditam que o sentido é construído em nossas mentes enquanto interagimos com o mundo físico, social e mental. Swan define as seguintes teorias construcionistas: Construcionismo Cognitivo, Construcionismo, Construcionismo Social (ou Socioconstrutivismo), e Aprendizado Situado/Cognição Distribuída. Irei conceituar cada uma a seguir com base em Swan.

2.1.1 Construtivismo Cognitivo

Esta teoria deriva da Epistemologia Genética de Jean Piaget (Swan, 2005). De acordo com Macedo (1994, p. 14) a Epistemologia Genética, hoje clássica e aceita generalizada

De acordo com Swan (2005), numa época dominada por teorias behavioristas, piaget estudou o desenvolvimento de estruturas mentais internas . Piaget acreditava que as crianças aprendem através de interações com o ambiente, pelo processamento cognitivo destas interações e o desenvolvimento de estruturas mentais, a que chamou de esquemas, para dar sentido a elas.

Piaget acreditava que o indivíduo tem 4 estágios de desenvolvimento (Swan, 2005, Piaget, 1971). O estágio sensório-motor é caracterizado pela compreensão cinestésica. No estágio pré-operatório, a criança adquire habilidade verbal, e organiza conhecimento em relação a si mesma. No estágio operacional concreto o conhecimento é organizado em categorias lógicas, mas continua ligado a experiência concreta. No estágio operacional formal, o conhecimento é abstraído da experiência e pode ocorrer raciocínio formal.

2.1.2 Construcionismo

O termo construcionismo foi cunhado por Seymour Papert, para se referir ao seu foco particular do construtivismo que, segundo o mesmo, atribui papel importante as construções no mundo como apoio para as construções mentais. A respeito disto, Papert escreveu:

Entendemos "construcionismo", como, incluindo, mas indo além, o que Piaget chamaria de "construtivismo". A palavra com o v expressa a teoria de que o conhecimento é construído pelo aprendiz, não fornecido pelo professor. A palavra com o n expressa a idéia, ainda, que isso acontece especialmente felizmente quando o aprendiz está engajado na construção de algo externo ou pelo menos compartilháveis... um castelo de areia, uma máquina, um programa de computador, um livro. Isso nos leva a um modelo usando um ciclo de internalização do que está fora, então a externalização do que está dentro e assim por diante. (Papert, 1990, tradução livre do autor)

Construcionistas atribuem importante papel as tecnologias de computação na educação, pois os computadores teriam a capacidade única de representar idéias abstratas em formas concretas e maleáveis. Papert desenvolveu a linguagem LOGO para testar essas idéias.

2.1.3 Construtivismo Social

De acordo com SWAN (2005), teorias de construtivismo social, são aquelas em que sua principal preocupação é a construção do conhecimento por meio das interações sociais. Estas teorias resultam principalmente da obra de Lev Vygotsky. Ainda de acordo com Swan, para Vygotsky, todo o aprendizado resulta das interações sociais nas instituições. O significado é construído socialmente e internalizado individualmente. Piaget focou na segunda parte do processo, enquanto Vygotsky focou na primeira. Para Vygotsky, habilidades cognitivas e padrões de

pensamentos não são determinados principalmente por fatores inatos, mas por atividades praticadas nas instituições sociais que o indivíduo vive.

Outro conceito importante de Vygotsky é a zona de desenvolvimento proximal (SWAN, 2005). Tal conceito se opõe as principais posições teóricas na época acerca da influência da aprendizagem no desenvolvimento (ROSA, 2011).

Para Vygotsky, todo desenvolvimento ocorre nesta zona, entre o que o indivíduo pode fazer por si mesmo (desenvolvimento real), e o que ele pode fazer sob a supervisão de um adulto (desenvolvimento potencial).

2.1.4 Aprendizagem situada/ Cognição distribuída

Segundo Swan (2005), este ramo tem origem no autor Alexei Leont'ev, seguidor de Vygotsky. Leont'ev acreditava que as construções de conhecimento, internas e externas, não podem ser entendidas separadas das ferramentas e culturas que a moldaram.

A aprendizagem situada é a corrente teórica que defende que o aprendizado ocorre é uma função da atividade, contexto e cultura em que ela ocorre, e portanto, inseparável das comunidades que a sustentam. Um conceito importante desta corrente teórica, segundo Swan (2005), é a participação periférica legítima, que trata de inserir os novatos gradualmente em comunidades de prática, rumo a um engajamento completo, onde construirão conhecimento de forma colaborativa com ajuda de experientes.

As teorias da cognição distribuída defendem que o pensamento, e portanto a aprendizagem, não ocorrem somente na mente do indivíduo, mas é distribuída socialmente entre os indivíduos, ferramentas e artefatos de uma cultura.

2.2 Formação de conceitos em crianças cegas

Batista (2005) levanta a questão de como a criança cega irá entender as noções apresentadas em sala de aula, e, apesar de recursos como o braile, como ela irá aprender noções como acidentes geográficos, o funcionamento do corpo humano, ou fazer distinções entre animais? Segundo Batista (2005), muitos educadores buscam formas alternativas para apresentar objetos que se assumem

serem conhecidos normalmente através da visão. Batista afirma que uma das respostas correntes na psicologia para conceituar conhecer está relacionada a aquisição de conceitos.

2.2.1 Aquisição de conceitos por pessoas cegas

Conforme Batista (2005), todas as implicações sobre aquisição de conceitos válidas para qualquer pessoa continuam válidas para cegos, embora devam ser acrescentadas questões específicas para este tema. Conforme Lewis (2003 apud Batista) e Warren (1994, apud Batista), as revisões mais recentes sobre o desenvolvimento de pessoas cegas não trazem mais dúvidas sobre a presença de capacidades conceituais, ao invés disso questionam em que aspectos diferem, e implicações destas diferenças. Em relação às capacidades dos cegos, Warren (1994 apud Batista), criticou a abordagem no estudo de cegos em que se compara capacidades de crianças cegas em relação às capacidades correspondentes de crianças videntes de mesma idade cronológica, propondo como alternativa o que chamou de abordagem diferencial, em que busca a explicação de diferenças com maior foco nos casos de alta aquisição, pois, para Warren, um caso de alta aquisição nos diz mais do que resultados médios, pois provam que possíveis dificuldades não são inerentes a cegueira.

2.3 Representação espacial em crianças cegas

Ochaíta, Huertas e Espinosa (1991) estudaram a capacidade de representação espacial de crianças cegas e suas relações com desenvolvimento e aprendizagem, com objetivo de encontrar resultados conclusivos, que segundo os autores não foi alcançado em trabalhos anteriores, as variáveis que determinam ou modulam a capacidade de representação espacial em pessoas cegas. Ochaíta, Huertas e Espinosa (1991) apontam que apesar de existir um bom número de investigações sobre o desenvolvimento da representação espacial que seguem o modelo construtivista, não haviam estudos que tomassem esta referência sobre a representação espacial por pessoas cegas.

Para esta investigação, Ochaíta, Huertas e Espinosa realizaram um experimento com dois grupos de cegos, os quais testaram a habilidade de aprenderem um trajeto em um máximo de 4 sessões e representá-los posteriormente. Um dos grupos tentava aprender um trajeto pequeno, composto por 7 pontos de referência, o outro grupo deveria aprender um trajeto similar porém maior. Para testar a representação espacial do trajeto, foram usadas 3 técnicas: maquetes; estimativa de distância; e a habilidade de passar por um novo trajeto composto por 3 dos 7 pontos de referência que estavam no trajeto anterior.

Desta experiência, para espaços pequenos o resultado encontrado forneceu:

- Diferença significativa entre idades
- Não há diferença significativa entre cegos precoces e tardios
- Houve grande diferenças de êxitos na primeira e última sessão
- Duração foi pouco sensível a aprendizagem

Quanto ao resultado em espaços grandes:

- Efeito da aprendizagem significativo nas maquetes e distâncias (exceto crianças pequenas)
- Idade foi relevante em todas as medidas utilizadas (principalmente no teste de modelar maquetes, e menos relevante em distâncias)
- Tendência de melhor representação em cegos tardias, principalmente em grupos de menor idade

Cruzando os dados de ambos os grupos, decorre que o tamanho do espaço não determina a capacidade de representação (tal associação encontrada na literatura anterior se deve a maior complexidade de espaços maiores, não ao tamanho propriamente dito), e que não houve diferença significativa em mobilidade (êxito no trajeto) e conduta espacial (avaliada pela quantidade de vezes que o sujeito pede ajuda).

2.4 A Etnomatemática

Um conceito que pode trazer contribuição para este trabalho, e que apresentarei a seguir, é a etnomatemática, especialmente na sua concepção mais ampla, não limitado a matemática. Primeiramente, é necessário observar que o termo etnomatemática foi e é usado com diferentes conceituações (FERREIRA,

2011, p. 5). De acordo com Ferreira (2011. p. 3), sua origem remonta a correntes educacionais que surgiram na década de 70 com o fracasso da Matemática Moderna, e que tinham em comum “a forte reação contra a existência de um currículo matemático de um só visão, como um conhecimento universal e caracterizado por divulgar verdades absolutas”. Mais precisamente, o Movimento de Etnomatemática surgiu no Brasil em 1975 a partir dos trabalhos de Ubiratan D'Ambrosio (D'AMBROSIO, 2011).

D'Ambrosio cunhou o termo a “etnomatemática” como junção das raízes “etno” (realidade natural e sociocultural), “matema” (conhecer, explicar, aprender), e “tica” (técnica), (D'AMBROSIO, 2002, 2011). Assim, D'Ambrosio deu origem a sua conceituação de etnomatemática, relacionado as diferentes formas e técnicas de entender, explicar, em diferentes contextos naturais e socioculturais da realidade. Para D'Ambrosio (2011), “se trata de um programa interdisciplinar, abarcando o que constitui o domínio das chamadas ciências da cognição, da epistemologia, da história, da sociologia e da difusão”. Ainda segundo D'Ambrosio (2011), “o Programa Etnomatemática tem como objetivo entender o ciclo do conhecimento em distintos ambientes”. Portanto, embora D'Ambrosio confirme que “O Programa Etnomatemática teve sua origem na busca de entender o fazer e o saber matemático de culturas marginalizadas.” (2011), não está limitada a disciplina reconhecida academicamente como Matemática, mas, pelo contrário, D'Ambrosio afirma que a disciplina Matemática é que é uma etnomatemática particular, que se desenvolveu na Europa com contribuições importantes de civilizações da África e do Oriente.

2.5 Usabilidade

De acordo com a norma ABNT ISO 9241, usabilidade é a “ Medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”. Outra norma que define usabilidade é a NBR ISO/IEC 9126 (2003, p. 9) que define usabilidade como: “Capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas.” De acordo com Vilella (p. 47, 2011):

“De uma maneira geral, os estudos de usabilidade em Ciência da Informação se propõem a ser um instrumento para viabilizar o desenvolvimento e/ou adequação de sistemas de informação a partir de uma perspectiva centrada no usuário, em suas necessidades, tarefas a serem executadas, condições de interação com o sistema, entre outros fatores. ”

A norma define ISO 9126 (p. 7), define também que a usabilidade é um dos atributos da qualidade de software, e define as seguintes métricas:

- Inteligibilidade: “Capacidade do produto de software de possibilitar ao usuário compreender se o software é apropriado e como ele pode ser usado para tarefas e condições de uso específicas. ”
- Apreensibilidade: “Capacidade do produto de software de possibilitar ao usuário aprender sua aplicação. ”
- Operacionalidade: “Capacidade do produto de software de possibilitar ao usuário operá-lo e controlá-lo. ”
- Atratividade: “Capacidade do produto de software de ser atraente ao usuário.”

A norma NBR ISO 9241-11 estabelece que normalmente é necessário ao menos uma medida para eficácia, eficiência e satisfação. Os quais são definidos como:

- “eficácia: Acurácia e completude com as quais usuários alcançam objetivos específicos. ”
- “eficiência: Recursos gastos em relação à acurácia e abrangência com as quais usuários atingem objetivos. ”
- “satisfação: Ausência do desconforto e presença de atitudes positivas para com o uso de um produto. ” (ISO 2002)

Convém que a própria norma avisa que:

“a importância relativa dos componentes de usabilidade depende do contexto de uso e das propostas para as quais a usabilidade está

sendo descrita, convém que não haja regra geral de como as medidas sejam escolhidas ou combinadas. ”

Além disto, o anexo B, informativo, da norma, sugere como exemplo as seguintes medidas de Usabilidade Global (para objetivos gerais do produto, não específicos):

Tabela 1: Exemplo de medidas de usabilidade global

Objetivos de usabilidade	Medidas de eficácia	Medidas de eficiência	Medidas de satisfação
Usabilidade global	Porcentagem de objetivos alcançados;	Tempo para completar uma tarefa;	Escala de satisfação;
	Porcentagem de usuários completando a tarefa com sucesso;	Tarefas completadas por unidade de tempo;	Frequência de uso;
	Média da acurácia de tarefas completadas	Custo monetário de realização da tarefa	Frequência de reclamações

Neste mesmo anexo, há exemplos de medidas para propriedades desejáveis, como facilidade de aprender, adequação a necessidades para usar facilmente, etc.

Técnicas de design de interface, sobre as quais são discorridas no capítulo 3, fornecem um meio formal de alcançar usabilidade.

2.5.1 Avaliação informal de usabilidade e heurísticas

De acordo com Nielsen (1990), avaliação heurística é uma forma de avaliação informal de usabilidade, na qual especialistas julgam uma interface de acordo com princípios pré-estabelecidos de usabilidade. Esta técnica pode ser usada em qualquer estágio do ciclo de desenvolvimento.

Outro método comum de teste de usabilidade, o Teste do Corredor ou teste da Intercepção de salão, é feito convidando aleatoriamente 5 pessoas para testar o software. De acordo com Nielsen (1990), a maioria dos problemas de usabilidade são encontrados em testes pequenos com até 5 pessoas.

2.5.1.1 As heurísticas de Nielsen

Esta é provavelmente a heurística mais utilizada no mundo. Se baseia em 10 itens de avaliação. Foi publicada em 1990 e atualizada em 1994. Seu autor, Jakob Nielsen, foi um importante pesquisador da área de usabilidade, ex-funcionário da Sun Microsystems, onde trabalhou grande parte do tempo no campo de usabilidade na Web, inclusive no website e intranet da Sun, foi autor de diversos livros sobre usabilidade e possui Ph.d em interação humano-computador da Universidade técnica da Dinamarca (NIELSEN, 2011). As heurísticas se baseiam em 294 erros mais comuns que o autor encontrava:

1. Visibilidade de estado do sistema: O usuário deve ser informado do que está acontecendo em tempo razoável
2. Relação entre o sistema e o mundo real: O sistema deve ter linguagem familiar com o usuário para evitar mal uso. Deve evitar termos técnicos e usar termos considerando-se que o usuário pode não saber nada do assunto
3. Controle do usuário e liberdade: Usuários frequentemente escolhem funções do sistema por engano. Forneça uma forma rápida e claramente marcada de sair do estado indesejado. Suporte botões avançar/voltar
4. Consistência e padrões: Siga as convenções da plataforma, evite usar diferentes termos, imagens, etc, para a mesma coisa
5. Prevenção de erros: Tente eliminar as condições que permitem que erro ocorra ou peça confirmação do usuário antes de executar sua ação.
6. Reconhecimento ao invés de lembrança: Minimize a carga de memória do usuário deixando objetos, ações e opções sempre visíveis, ao invés do usuário ter que lembrar aonde elas estão
7. Flexibilidade e eficiência de uso: Aceleradores podem acelerar a experiência para usuários experientes e novatos. Permita usuários personalizar ações frequentes
8. Estética e design minimalista: Diálogos não devem conter informação irrelevante. Cada informação irrelevante diminui a visibilidade da informação relevante.

9. Ajude o usuário a reconhecer, diagnosticar e recuperar-se de erros: Mensagens de erro devem ser expressadas em linguagem simples, ao invés de códigos, indicar precisamente o problema e sugerir uma solução
10. Ajuda e documentação: Embora seja melhor se o sistema puder ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Qualquer informação deve ser fácil de encontrar, focada nas tarefas do usuário, listar passos exatos e não ser muito grande.

2.6 Acessibilidade

De acordo com o portal Brasil.gov.br, portal oficial do Estado brasileiro, acessibilidade significa “a possibilidade de qualquer pessoa usufruir de todos os benefícios da vida em sociedade, entre eles o uso da internet. É o acesso a produtos, serviços e informações de forma irrestrita.” (2011). O decreto 5.296, publicado em 2 de dezembro de 2004, Art 8º, define acessibilidade como “condição para utilização, com segurança e autonomia, total ou assistida, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos serviços de transporte e dos dispositivos, sistemas e meios de comunicação e informação, por pessoa portadora de deficiência ou com mobilidade reduzida” (BRASIL, 2011). A Convenção das Nações Unidas sobre os Direitos de Pessoas com Deficiência, do qual o Brasil é signatário, reconhece como direito humano básico o acesso a tecnologias de informação e comunicação (NAÇÕES UNIDAS, 2011).

No caso de software, o conceito refere-se a não haver impedimentos para o uso de determinadas ferramentas no caso de usuários com deficiência. É notável que, neste contexto, a acessibilidade se relaciona com usabilidade. De acordo com Winckler e Pimenta (2002, p. 7), a acessibilidade é um foco da usabilidade para pessoas com deficiência. Podemos avaliar a acessibilidade com base nos princípios de usabilidade definido na seção anterior portanto, observando o caso específico de pessoas com deficiência. Por exemplo, um software incompatível com um leitor de telas é o caso extremo de falta de operabilidade para um usuário cego.

Pessoas com deficiência, em muitos casos, dependem de tecnologias assistivas para a utilização de computadores. De acordo com Kamaludin, Yatim e Nordin (2010), algumas tecnologias assistivas utilizadas por deficientes visuais são:

software de reconhecimento de voz; leitor de tela e texto; display Braille; teclado Braille, entre outros. Como neste trabalho o tema é restrito ao caso de uso de um usuário cego, será dada maior ênfase às ferramentas leitoras de tela, uma vez que é esta a tecnologia assistiva frequentemente utilizada por estes usuários (DELPIZZO, 2005). Vale notar, no entanto, que o desenvolvimento da nova interface gráfica seguiu por diretrizes W3C de acessibilidade que levam em conta diversas deficiências como cegueira e baixa visão, surdez e baixa audição, dificuldades de aprendizagem, limitações cognitivas, limitações de movimentos, incapacidade de fala, fotossensibilidade e suas combinações (W3C, 2008), embora tenha sido testado somente o caso de um usuário cego.

Pessoas com deficiência também frequentemente possuem dificuldades no uso de certos dispositivos de entrada. No caso de pessoas cegas por exemplo, assim como outras deficiências, não é possível utilizar o mouse para realizar ações que pessoas sem deficiência frequentemente realizam com o mouse, e em geral utiliza-se teclado (WEBAIM, 2011). Na verdade a independência de dispositivo de entrada é também um importante princípio para que seja possível acessar um mesmo conteúdo a partir de diferentes dispositivos como, smartphones, tablets, etc.

Thatcher (1997) ressalta a importância para a construção de uma aplicação acessível utilizar certos padrões do sistema operacional para que possa ser utilizada por diversos leitores de tela (e outras tecnologias assistivas). Com base nos mesmos argumentos, podemos inferir a importância de seguir padrões da plataforma Web para garantir acessibilidade¹. Para Thatcher, um software que não siga tais padrões não é acessível mesmo que funcione bem com um leitor de telas específico. Thatcher enunciou 3 princípios para um software ser acessível:

1 É notável o fato de que este artigo de Thatcher é de 1997 e fala sobre a construção de aplicações desktop acessíveis, sendo mais recente o fenômeno que tornou a Web uma plataforma de aplicativos ao invés de somente hipertextos. Embora a Web seja uma plataforma independente de sistema operacional, e portanto, a rigor, a afirmação de Thatcher não seria aplicável, fica evidente que tal importância é análoga a importância de se utilizar os padrões da plataforma Web (por exemplo, HTML válido, WCAG e WAI-ARIA), que serão discutidos posteriormente, para que tecnologias assistivas possam oferecer uma alternativa de uso. Esta nota seria desnecessária se o termo utilizado no artigo fosse plataforma ao invés de sistema operacional.

1. Toda ação com o mouse deve ter uma equivalente via teclado. Você deve ser capaz de navegar rapidamente, competitivamente, via teclado.
2. Ao navegar via teclado, deve-se garantir que o leitor de tela possa conhecer a posição e conteúdo do que está em foco, para permitir que o usuário possa saber aonde está.
3. O leitor de telas deve ser capaz de identificar o objeto que está em foco. Isto é importante para que um usuário cego possa saber como utilizar cada componente da GUI. Usuários que podem ver, identificam ao ver uma área de texto que podem escrever naquela área, ou identificam que uma caixa de seleção pode ou não ser checada. Cegos dependem do leitor de telas informar qual tipo de componente está em foco. Conforme Thatcher nos lembra, sendo usados padrões da plataforma, é responsabilidade da tecnologia assistiva informar isto.

Cabe informar que Thatcher se refere com maior ênfase aos leitores de tela pois para Thatcher, se um software poder ser usado com leitores de tela por seguir padrões da plataforma, certamente poderá ser utilizado por outras tecnologias assistivas. É importante no entanto observar que o artigo em que Thatcher enunciou estes princípios é de 1997, e portanto não considera a evolução no campo de acessibilidade desde então. Também não está claro a validade do texto para diferentes tipos de deficiência como epilepsia e deficiências cognitivas. Embora insuficientes, no entanto, estes princípios se mostram ainda válidos e ajudam a entender a necessidade do uso correto da tecnologia para fornecer semântica adequada a tecnologias assistivas, e inclusive a necessidade que levou ao desenvolvimento do WAI-ARIA, tratado na seção 2.6.1.5.

2.6.1 Padrões de acessibilidade na web

O World Wide Web Consortium (W3C), consórcio internacional que visa desenvolver padrões web (W3C, 2011), criou em 1997 a Web Accessibility Initiative (WAI), que desenvolve orientações amplamente aceitas como padrão internacional,

materiais de suporte e recursos (W3C, 2011). A WAI produziu em 1999 a recomendação WCAG (Web Content Accessibility Guidelines) 1.0, e em 2008, foi lançada a nova versão da recomendação, WCAG 2.0 (W3C, 2011) que é a principal recomendação vigente para acessibilidade na Web. Em 2008 também, um grupo de desenvolvedores descontentes com o processo de desenvolvimento da WCAG 2.0 e liderados por Joe Clark, um ex-colaborador do grupo de trabalho WCAG, lançaram uma errata para a WCAG 1.0, chamada de WCAG Samurai, com o intuito de ser uma alternativa à WCAG 2.0.

Além da recomendação WCAG, a WAI está produzindo uma especificação, para aplicações de Internet ricas, isto é, aplicações web com conteúdo dinâmico e interface avançadas, desenvolvida com HTML, Ajax, Javascript e tecnologias relacionadas, chamado Accessible Rich Internet Applications (WAI-ARIA - WAI-ARIA Overview, W3C, 2011). O WAI-ARIA está em fase de Candidato a Recomendação, conforme a própria recomendação publicada (W3C, 2011) e o documento WAI-ARIA FAQ (W3C, 2011), o que, de acordo com o mesmo, significa que já há um amplo consenso sobre o conteúdo técnico. Conforme este mesmo documento, WAI-ARIA já é suportado por diversos navegadores Web e tecnologias assistivas.

2.6.1.1 WCAG Samurai

De acordo com o sítio oficial, “O WCAG Samurai é um grupo de desenvolvedores liderados por Joe Clark que propõe uma lista de correções e extensões para as Recomendações para Acessibilidade do Conteúdo da Web 1.0” (WCAG SAMURAI, 2008). O documento portanto é uma errata para a WCAG 1.0, e não uma norma completa. De acordo com o documento, a errata WCAG Samurai é uma alternativa a WCAG 2.0 e não é possível estar de acordo com ambos. No sítio do grupo utiliza-se o termo WCAG+Samurai para se referir as normas WCAG 1.0 Corrigidas e este termo será utilizado doravante. No sítio deste grupo, a única informação sobre a motivação para se criar uma alternativa é um link para um artigo de Joe Clark publicado na revista A List Apart, no qual faz diversas críticas ao processo de desenvolvimento do WCAG 2.0 – que de acordo com Clark não é acessível e houve pouco tempo para partes interessadas comentar sobre WCAG 2.0 – e a própria norma que foi considerado extenso, confuso, e demasiadamente vago

e permissivo, incluindo alguns exemplos objetivos nos quais Clark identificou retrocessos. É importante observar, no entanto, que o artigo é de 2006, quando a WCAG 2.0 ainda se encontrava em desenvolvimento. A versão final da WCAG 2.0 só foi lançada em 2008.

WCAG Samurai foi desenvolvido em um processo fechado, com participação somente de convidados que não tem seu nome divulgado (exceto pelo Joe Clark), e submetido a dois revisores (com nomes e formação devidamente divulgados). Como justificativa, o documento de introdução apresenta apenas o argumento de que o modo de desenvolvimento da WCAG que reivindica-se aberto não funciona e encontra-se fechado. De acordo com o documento¹:

Porque o processo ostensivamente aberto do W3C atualmente encontra-se fechado: Ele é dominado por multinacionais; a opinião de qualquer um que não seja convidado pode ser e é ignorada; o status de expert-convidado tem sido recusado ou revogado; o Grupo de Trabalho pode reivindicar que houve "consenso" mesmo nas questões não resolvidas por discordâncias internas; o processo de trabalho em si é inacessível a pessoas com necessidades especiais, como surdos, por exemplo; os membros do Grupo de Trabalho para as WCAG têm atuado como tiranos.

O processo "aberto" do W3C simplesmente não funciona. Nós tentamos fazer algo mais.

Como ponto positivo desta errata em relação ao WCAG 2.0, nota-se sua objetividade. WCAG Samurai não usa termos imprecisos ou que possam dar margem a interpretação errôneas, tal como “evite”, ou “até que os agentes de usuário suportem”. Outra simplificação, WCAG Samurai abole a possibilidade de estar em conformidade em diferentes níveis de prioridade, aceitando as prioridades 1 e 2 da WCAG 1.0 corrigidas, e ignorando a prioridade 3. Em contraste com a WCAG 2.0, a errata WCAG Samurai é pouca extensa.

Como ponto negativo, ao contrário da WCAG 2.0, as recomendações da WCAG Samurai não são independentes da tecnologia usada. De acordo com o

1 Tradução de Maurício Samy Silva (acesso em 2011).

próprio documento, supõe-se que o desenvolvedor usa HTML com CSS “a maior parte do tempo” (WCAG SAMURAI, 2008). Grande parte das recomendações citam diretamente tags HTML que podem ou não ser usadas, embora a norma estabelece que “Flash, JavaScript e outras tecnologias interativas são permitidas contudo em todos os casos devem seguir as recomendações de acessibilidade correntes” (WCAG SAMURAI, 2008). Não há, entretanto, uma explicação de quais são as recomendações correntes de cada tecnologia, se referem-se a própria WCAG+Samurai, ou a normas específicas dos fabricantes das tecnologias. Uma vez que esta norma muitas vezes cita diretamente tags HTML que podem ou não ser usadas, ao invés de descrever o comportamento esperado ou proibido de forma independente de forma independente da tecnologia usada e de como foi implementado, o desenvolvedor pode carecer de normas aplicáveis para a tecnologia em uso.

Embora o fato de a norma citar diretamente a forma de implementação em muitos casos torna este documento mais objetivo que o WCAG 2.0, ele pode não se adequar bem em casos que se faz maior uso de outras tecnologias. Tal fato também tem o revés de poder ficar mais rapidamente defasado em relação a evolução da tecnologia, como novas versões do HTML e XHTML e a especificação WAI-ARIA. Outro ponto negativo é que a própria hipótese que o desenvolvedor está usando “HTML e CSS a maior parte do tempo” é vaga e traz imprecisão. Afinal, se estou usando mais outras tecnologias que HTML e CSS, as normas corrigidas pela WCAG Samurai se aplicam ou não? Se usar HTML e CSS “a maior parte do tempo” é um requisito para a WCAG corrigida pela WCAG Samurai ser aplicável, como saber se está sendo cumprido ou há uso demasiado de outras tecnologias?

Uma nota no WCAG Samurai também traz outro ponto contra, que é não contemplar deficiências cognitivas. De acordo com o documento:

Esta errata não faz uma correção substancial das diretrizes do WCAG 1 no que diz respeito às deficiências cognitivas. Conformidade com a WCAG+Samurai não pode gerar reivindicação de plena acessibilidade para pessoas com deficiências cognitivas.

Um último ponto negativo deste documento que gostaria de citar, em muitos casos, na norma WCAG+Samurai não fornece uma explicação sobre porque algo é

proibido ou obrigado. Por exemplo, a WCAG Samurai acrescenta a Diretriz 9 - “Projetar páginas considerando a independência de dispositivos”¹, a proibição aos atributos `tabindex` e `accesskey`, sem qualquer explicação do motivo de tais atributos estarem sendo proibidos por esta norma. Em especial, se o uso do atributo `accesskey`, que permite a criação de teclas de atalho para acessar componentes, não afeta as demais formas de se acessar o mesmo componente, porque criar uma forma de acesso adicional violaria a diretriz de independência de dispositivos?

2.6.1.2 Web Content Accessibility Guidelines 2.0

A Web Content Accessibility Guidelines (WCAG²) 2.0 é a norma em vigor para criação de conteúdo Web acessível criada pela W3C. Embora seja possível estar em conformidade com a WCAG 1.0 e 2.0, a recomendação para novos conteúdos é que sigam as diretrizes da WCAG 2.0. O documento fornece 12 recomendações, divididos em 4 princípios: Perceptível, Operável, Compreensível e Robusto. Cada recomendação por sua vez é fornecido com um conjunto de critérios de sucesso, que são declarações testáveis e não são de tecnologia específica. Os critérios de sucesso por sua vez, são classificados nos níveis A, AA e AAA. Uma página pode estar em conformidade com o WCAG 2.0 também em níveis A, AA e AAA. Uma página está em conformidade com a WCAG 2.0 em nível A se cumprir todos os critérios de sucesso de nível A. Uma página estará em conformidade com o nível AA se cumprir todos os critérios de sucesso de níveis A e AA. Finalmente, uma página estará em conformidade com o nível AAA se cumprir todos os critérios de sucesso de níveis A, AA e AAA. Além da seção com as recomendações propriamente, também são normativas as seções Conformidade – que indica como pode ser alcançada conformidade e como apresentar uma reivindicação de conformidade – e Glossário (que define os termos técnicos utilizados).

O documento WCAG 2.0 possui 4 documentos de apoio não-normativos. São eles³:

1 Termo utilizado na tradução da WCAG Samurai de Maurício Samy Silva (2011). No original: “Design for device-independence”.

2 Em português: “Recomendações de Acessibilidade para Conteúdo Web (WCAG) 2.0” (BECHARA, 2009)

3 Nomes em português dos documentos de acordo com tradução de Bechara (2009)

1. **How to Meet WCAG 2.0 (Como cumprir as WCAG 2.0):** Referência rápida. Fornece para cada critério de sucesso uma lista de links para técnicas e falhas relacionadas do documento Techniques for WCAG 2.0, além do próprio texto dos critérios do critério de sucesso ser repetido aqui. É possível filtrar as técnicas por tecnologias e níveis.
2. **Understanding WCAG 2.0 (Entendendo as WCAG 2.0):** Fornece uma pequena explicação para cada critério de sucesso ou recomendação, também contém redundantemente o texto das recomendações e critérios de sucesso, além dos links de técnicas e falhas que constam no documento How to Meet WCAG 2.0.
3. **Techniques for WCAG 2.0 (Técnicas para as WCAG 2.0):** Fornece uma lista de técnicas suficientes ou aconselhadas, em tecnologia específica, para se alcançar um ou mais critério de sucesso, e também casos de falhas comuns, em tecnologia específica. Cada técnica é apresentada em página específica com descrição, exemplos, código-fonte e teste. Uma falha ou técnica pode se relacionar a mais de um critério de sucesso e vice-versa. Este documento não está organizado por critérios de sucesso. Para se encontrar técnicas para atingir um determinado critério de sucesso, pode-se utilizar os links de um dos documentos anteriores
4. **The WCAG 2.0 Documents (Os documentos das WCAG 2.0):** Um diagrama e uma descrição sobre a forma como os documentos técnicos estão relacionados e ligados.

2.6.1.3 Comparação entre WCAG 2.0 e WCAG+Samurai

A norma WCAG 2.0 é desenvolvida pelo consórcio internacional responsável por criar os padrões Web, a W3C, em um processo que reivindica permitir a colaboração pública no processo. Por este fato é amplamente suportada e aceita como padrão. A errata WCAG Samurai porém, foi criada por um grupo de desenvolvedores anônimos liderados por um ex-colaborador do grupo de trabalho WCAG, Joe Clark. A participação no grupo é exclusiva por convite. Antes da publicação, o documento foi enviado para dois revisores independentes com formação na área e participação anterior no grupo de trabalho WCAG. Apesar de

Joe Clark argumentar que o processo aberto da W3C na verdade encontra-se fechado, não explica como um processo totalmente fechado pode ser melhor do que com alguma abertura, ainda que dominado por multinacionais e porque não tentou uma forma melhor de colaboração ao invés de descartar a colaboração.

O maior problema do WCAG+Samurai porém, é que, tal como as WCAG 1.0, continua fornecendo recomendações normativas de tecnologia específica e que portanto tendem a ficar defasado assim que houver uma nova versão da tecnologia referenciada. Recomendações de tecnologia específica também não evitam problemas semelhantes que podem ocorrer em outras tecnologias. As WCAG 2.0 porém não se referem a tecnologias específicas em suas diretrizes e critérios de sucesso normativos, normatizando comportamento ao invés de implementações e deixando as técnicas para tecnologia específica para um documento separado e não normativo.

Por as WCAG 2.0 serem a recomendação oficial da W3C, amplamente reconhecida e utilizada, suportada por ferramentas de validação, sem depender de tecnologia específica, e pelo WCAG Samurai não apresentar argumentos para optar-se por sua alternativa as WCAG 2.0, exceto por um defasado documento de 2006, optou-se neste trabalho pelo uso das normas WCAG 2.0.

2.6.1.4 Authoring Tool Accessibility Guidelines (ATAG)

A ATAG, ou diretrizes de acessibilidade para ferramentas de autoria, são as diretrizes de acessibilidade da W3C para ferramentas que produzem conteúdo Web. A versão 1.0 é a diretriz vigente, publicada em 2000, embora a versão 2.0 esteja em desenvolvimento, publicada como rascunho (Draft). O objetivo, definido no documento é que a ferramenta possa ser usada por pessoa com deficiência e que seja projetada para permitir, apoiar e promover a produção de conteúdo Web acessível.

Assim como a WCAG 2.0, as diretrizes da ATAG 2.0 são organizadas em princípios e compostos por critérios de sucesso checáveis. Como aplica-se Telis a conteúdo Web, Telis é uma ferramenta de autoria de conteúdo Web e portanto a norma ATAG é aplicável. Porém, o escopo deste trabalho foi delimitado para fornecer acessibilidade ao ambiente Telis, garantidas pela norma WCAG 2.0. Obter

conformidade com as ATAG porém, implica em também garantir acessibilidade do conteúdo produzido, o que implica em mudanças na linguagem também, sendo fora do escopo deste trabalho.

2.6.1.5 WAI-ARIA

A World Wide Web – originalmente apenas um sistema de hipertextos para a internet – está cada vez mais comportando aplicações ao invés de simples documentos. Tal fato foi possibilitado pelo surgimento de tecnologias que permitem gerar conteúdo dinâmico tanto do lado do cliente – como Javascript lançado em 1995 – como do lado do servidor com os scripts CGI. No entanto, foi em 2004 que foi cunhado o termo Web 2.0, para se referir ao fenômeno em que a internet se transformava em plataforma de aplicações. Hoje, ainda é notável a tendência com a recente onda de computação em nuvem.

Diversas aplicações de internet modernas utilizam diversas técnicas para modelar widgets análogos aos encontrados em aplicações desktop, além de atualizar partes da página e o estado destes componentes dinamicamente. Doravante será utilizado o termo aplicação de internet rica ou a sigla em inglês RIA (Rich Internet Application) para se referir a páginas que utilizam estas técnicas. O uso destas técnicas cria um problema para que tecnologias assistivas possam reconhecer estes componentes e fornecer suporte adequado. Frameworks de acessibilidade de interfaces gráfica com o usuário (GUI) de aplicações desktops legados fornecem uma interface de programação de aplicativos (doravante será utilizado a sigla em inglês API, de Application Programming Interface) e infraestrutura (WAI-ARIA Primer 1.0, W3C, 2010). Estas APIs constituem um contrato entre aplicações e tecnologias assistivas, para que tecnologias assistivas possam acessar conteúdo rico e dinâmico com a semântica apropriada para produzir uma alternativa utilizável (WAI-ARIA Primer 1.0, W3C, 2010). Não havia até então uma API equivalente na Web, e a especificação WAI-ARIA vem justamente atender esta necessidade. WAI-ARIA define uma API que permite através de atributos fornecer semântica adequada para que tecnologias assistivas possam fornecer forma alternativa de uso.

Para entender porque isto é necessário, imaginemos um usuário que necessita de um leitor de telas para navegar por uma página com componentes como botões pressionáveis, caixa de diálogo, barra de progresso, etc. Não existe um componente nativo no HTML equivalente a um botão com estados pressionado ou não pressionado, como os botões que podem ser encontrados em editores de texto (negrito, itálico, sublinhado, entre outros). Um desenvolvedor pode simular este componente de diversas formas utilizando componentes nativos do HTML. Uma forma trivial de fazer isto poderia ser ter duas imagens, uma para cada estado (e opcionalmente mais uma para o estado com foco), e utilizar Javascript para exibir a imagem correta. Mesmo que seja possível alcançá-lo e ativá-lo via teclado, sem WAI-ARIA um usuário cego usando um leitor de telas não será apropriadamente informado do estado do botão. Outro exemplo, uma barra de progresso em HTML 4 que indica o carregamento do site, e que portanto altera seu valor dinamicamente, dificilmente seria lida de forma adequada por uma tecnologia assistiva. Cabe salientar que HTML 5 já tem uma alternativa nativa para barras de progresso (a tag `progress`) que dispensa o uso de WAI-ARIA para este componente específico, por ser responsabilidade dos agentes de usuário a leitura correta dos parâmetros de elementos nativos.

Atributos ARIA podem ser atribuídos diretamente na marcação da página (HTML ou XHTML) ou através de linguagem de script. Atributos ARIA podem ser papéis (referente ao que é o objeto), propriedades e estados.

2.6.2 Softwares leitores de tela

Apesar de se buscar alcançar o suporte a uma grande diversidade de tecnologias assistivas através da conformidade com as WCAG 2.0, a interface Web desenvolvida foi testada com 3 leitores de tela: Orca, por ser o principal leitor de telas para GNU/Linux, parte do ambiente Gnome e disponível na maioria das distribuições GNU/Linux. NVDA, por ser o leitor de telas de código aberto com maior uso (WebAIM, 2009), embora seja disponível para plataforma Windows, e JAWS, que embora seja um software proprietário, é o leitor de telas utilizado pela ampla maioria dos usuários, 74% (WebAIM, 2009).

2.6.2.1 Orca

De acordo com seu site oficial (GNOME, 2010, tradução livre), “Orca é um leitor de telas livre, de código aberto, que fornece acesso a ambientes gráficos via combinações customizáveis pelo usuário de voz, braille e ampliação de tela”. Por ser parte do ambiente Gnome, está presente em um grande número de distribuições Linux. Neste trabalho usaremos o orca como leitor de telas padrão para testar a acessibilidade em softwares. Orca precisa de um sintetizador de voz, por software ou hardware, para ler a tela com voz. No Ubuntu 11.10, o sintetizador de voz instalado por padrão foi o espeak, que inclui voz em português.

2.6.2.2 NVDA

NVDA é um leitor de tela livre, fornecido em licença GPL versão 2, para plataforma Windows. É fornecido com sintetizador de voz livre espeak (também utilizado pelo Orca no Linux), mas pode ser acoplado a um sintetizador externo.

2.6.2.3 Jaws

É um leitor de telas líder de mercado, fornecido apenas para plataforma Windows, é software proprietário e pago

2.7 A linguagem e o ambiente Telis

Telis é uma linguagem de programação e um ambiente de desenvolvimento, desenvolvido pelo Laboratório de Software Educacional (EDUGRAF) da Universidade Federal de Santa Catarina (UFSC) (PIERI, et. al). Seu principal objetivo é suportar aprendizes, e foi modelado especialmente para quem nunca teve contato com linguagens de programação (PIERI, et. al). Segundo Pieri et. al [200x], Telis adota uma abordagem construtivista, em que o aprendiz é estimulado a construir seu próprio conhecimento através da prática (aprender fazendo). Ainda de acordo com Pieri et. al, devido a isso, é importante que o aprendiz se torne familiar com a linguagem rapidamente - e segundo Kelleher e Pausch (2005, APUD PIERI

et. al) a maioria das linguagens de propósitos gerais incluem um grande número de elementos sintáticos que tornam difícil o aprendizado por um iniciante, principalmente porque estes elementos não apresentam um significado óbvio. Devido ao fato de tais linguagens não se destinarem a aprendizes e sim a programadores profissionais, e portanto priorizem flexibilidade ao invés da simplicidade, mesmo tarefas simples podem exigir alto nível de conhecimento anterior.

Telis possui uma sintaxe simplificada e apresenta conceitos avançados da ciência da computação - tais como concorrência, distribuição, programação orientada a serviço, refatoração, etc.

2.7.1 Considerações sobre Telis e acessibilidade

Para se analisar a acessibilidade do Telis é preciso analisar tanto a interface com o usuário do ambiente, quanto a interface com o usuário que se pode construir em um aplicativo Telis. Com relação a interface, além das limitações que ocorrem nos aplicativos publicados também estarem presentes no aplicativo interativo do ambiente, foi identificado que o ambiente é quase completamente inacessível. Em teste pelo autor, o leitor de telas Orca não foi capaz de ler nenhum conteúdo da interface. Ainda que um leitor de telas pudesse ler, o software permanece inacessível por conta de ser quase completamente não-navegável via teclado, violando o princípio de acessibilidade discutido na seção 2.6. Ao abrir o Telis e tentar navegar utilizando o teclado verifica-se que se pode se atingir o foco da entrada de comandos e do botão executar. Mais nenhuma funcionalidade do software pode ser atingida, incluindo janela da pilha, editores, etc. Mesmo abrindo utilizando o mouse, verifiquei que quase todos os diálogos do ambiente permanecem não navegáveis via teclado e portanto inacessíveis. Uma nova interface com o usuário para o ambiente Telis será criada neste trabalho.

Com relação aos aplicativos (programas em Telis publicados, na forma de Applet Java), um programa, ou aplicativo, feito totalmente em Telis, sem componentes externos, pode ter como entrada da interface com o usuário estímulos emitidos pelo mouse ou teclado. Como saída da interface com o usuário, pode-se emitir sons e desenhar ou imprimir imagem ou texto na área visível do aplicativo. Como pode se

observar pela lista de primitivas Telis, não é possível inserir num aplique Telis componentes gráficos usuais como botões, caixa de texto, a não ser utilizando um componente externo (Ver item seguinte, Dix). Mesmo o texto impresso com a primitiva mostrar se mostrou inacessível para leitores de tela (não pode ser selecionável via teclado e o Orca não conseguiu ler no teste feito)

Embora esta abordagem pareça coerente com os objetivos do ambiente, por não haver sequer o conceito de componentes gráficos tais como widgets em Telis, as tecnologias assistivas pouco podem fazer para auxiliar um usuário a utilizar um aplique Telis, pois tecnologias assistivas precisam reconhecer componentes para fornecer uma alternativa de uso. Com relação a essas limitações, no entanto, é possível, num momento inicial do aprendizado, um usuário cego utilizar Telis se limitando ao uso de sons como saída, embora seja importante também o estudo de formas de produzir conteúdo acessível em Telis, a acessibilidade do ambiente constitui um avanço.

2.8 Dix

Dix é um framework para criar páginas Web, feito pelo EDUGRAF, com componentes que podem facilmente se comunicar com aplicações Telis ou Java. Dix consiste de um template XSL que permite escrever facilmente páginas em XML com sintaxe bastante simplificada comparando com HTML ou XML e em português; e uma API para comunicação com Telis ou Java. O desenvolvimento de aplicações Dix com Telis é simples. Bastando criar um ator cujo modelo utilize o molde ComunicávelComDix, que herdará desse molde primitivas para comunicação com componentes Dix da página.

2.8.1 Dix e acessibilidade

Dix é importante para nós, porque permite que as aplicações em Telis utilizem componentes conhecidos – no caso componentes fornecidos pelas tags HTML, para fornecer a interface com o usuário – e portanto estes componentes são reconhecidos pelas tecnologias assistivas. É preciso um olhar mais detalhado, no entanto, para avaliar e corrigir as limitações do Dix para fornecer interfaces com o o

usuário acessíveis. Para tal foi analisado, de acordo com as recomendações WCAG 2.0 (W3C, 2008), o código de páginas Dix transformadas.

2.8.1.1 Análise da acessibilidade de páginas Dix

Para uma análise objetiva, foi analisado a possibilidade e facilidade de produzir páginas Dix acessíveis de acordo com os critérios de sucesso, normatizados pela norma WCAG documento “Web Content Accessibility Guidelines 2.0” (W3C, 2008), no entanto foi notavelmente observado que Dix é inacessível. Aqui é relatado alguns dos problemas de acessibilidade do Dix para mostrar que Dix hoje ainda não pode ser usado para produzir uma interface gráfica acessível para um aplique Telis.

Dix viola princípios básicos da acessibilidade na Web e não fornece semântica adequada para a estrutura do documento. Um exemplo é que não é possível fornecer uma alternativa em texto para imagens quando não há legendas (por exemplo, utilizando-se o atributo alt, de acordo com a técnica H37 do documento Techniques for WCAG 2.0, W3C, 2010), Também não é possível fornecer semântica adequada para imagens que são meramente decorativas e não fazem parte do conteúdo. Estes problemas claramente violam o critério de sucesso 1.1.1 das WCAG 2.0. Esta não conformidade também leva a não conformidade com os documentos ATAG 1.0 (Checkpoint 1.1, W3C, 2000) e ATAG 2.0 (Guideline B.2.1, W3C, 2011). Outro problema de acessibilidade é ignorar a semântica das tags HTML. Html fornece meios de marcar parágrafos, títulos e subtítulos, entre outros, Porém há uma única tag para escrever texto em Dix, não sendo possível distinguir elementos da estrutura. Na verdade, em teste com leitor de telas, todas as etiquetas Dix forma confundidas com hiperlinks. Este problema viola a diretriz 1.3.1 das WCAG 2.0.

Mas outro problema refere-se ao fato de ser utilizado em Dix sempre posições absolutas por coordenadas geográficas obrigatoriamente para todos os componentes. Isto, além da questão de acessibilidade deste conteúdo, pois a ordem de apresentação pode ser diferente da ordem de leitura, violando a recomendação 1.3.2 das WCAG 2.0, causa o problema adicional de ser muito difícil para um cego produzir uma página em Dix com apresentação visual coerente.

Dix é portanto inacessível, e as modificações para garantir acessibilidade seriam extensas e fora do escopo deste trabalho. Porém, como é importante o uso de widgets que podem ser reconhecidos por tecnologias assistivas para produzir conteúdo acessível, uma possibilidade para resolver esta questão em Telis em trabalhos futuros é adaptar ou reescrever o Dix para garantir acessibilidade, de acordo com as ATAG.

2.9 Ferramentas similares

Neste trabalho, foi considerado como requisitos para a linguagem de programação utilizada e que determinaram a escolha do Telis: Fornecer todas os comandos da linguagem, ambiente e documentação em português; Ser software livre e não depender de software proprietário (inclusive sistema operacional); apresentar conceitos avançados como programação distribuída, concorrente, orientada a objetos, entre outros, de maneira simples, abstraindo aspectos técnicos.

Há diversos softwares proprietários para este propósito como Micromundos, Superlogo e Megalog, estão disponíveis em português e preenchem parcialmente alguns requisitos, mas são proprietários, fornecidos unicamente para sistema operacional Windows e foram descartados por este motivo.

As ferramentas Greenfoot, baseada em Java, (Programming Education Tools Group, 2011) e Kojo, baseada em Scala, (Kogics) são software livre, orientadas a objetos, possuem o conceito de ator e portanto abstraem a concorrência tal como Telis. Um diferencial destas linguagens é que, embora forneça abstrações de aspectos técnicos para o aprendiz, são na verdade um ambiente e framework de linguagens profissionais e permitem ao aprendiz a medida do progresso utilizar recursos avançados da linguagem, ao invés do usuário ter que migrar de uma linguagem educacional para uma linguagem de propósitos gerais com sintaxe diferente. O grupo que desenvolve o Greenfoot também possui outro ambiente de programação educacional chamado BlueJ, sendo o Greenfoot para atender a quem não tem nenhuma experiência anterior e o BlueJ uma ferramenta para quem deseja aprofundar o aprendizado.

A linguagem Squeak (SQUEAK, 2011) por sua vez, baseada em Smalltalk, é software livre, orientada a objetos, possuem o conceito de ator e tem a vantagem de estar disponível em português.

Para este trabalho, foi escolhida a linguagem Telis, embora o Squeak também possua boas características para esta necessidade. Telis atende aos requisitos enunciados anteriormente, permitindo abstrair aspectos técnicos para diversos conceitos como computação concorrente, distribuída, interface gráfica, etc. Telis é uma ferramenta desenvolvida pelo laboratório Edugraf e que já vem sendo utilizado em oficinas de programação e mesmo em aulas de ciência da computação na Universidade Federal de Santa Catarina, esta nova implementação de interface gráfica pode substituir a versão anterior, que de outra forma poderia precisar de solução alternativa para o caso de um usuário cego.

3 Técnicas de projeto de Interface

Existem diversas técnicas para projeto de interface que fornecem um meio formal para se atingir boa usabilidade.

3.1 A abordagem *The Usage-Centered Design*

Usage-Centered Design é uma abordagem para projeto de interface que foca nas intenções do usuário e padrões de uso. Ela analisa os usuários em termos dos papéis que desempenham e utiliza-se de casos de usos abstratos para análise de tarefas. Esta abordagem foi proposta por Larry Constantine e Lucy Lockwood. (CONSTANTINE, LOCKWOOD, 1999). De acordo com Constantine (2002), Usage-Centered design é guiado por três modelos abstratos simples e relativamente fechados: Um modelo de papéis de usuário, um modelo de tarefas e um modelo de conteúdo.

3.1.1 Modelo de papéis de usuários

Segundo Constantine (2002, p. 4), um modelo de papéis captura a característica notável dos papéis que usuários desempenham em relação ao sistema. De acordo com Constantine (2006, p. 5), um usuário interagindo com um sistema é denominado ator, termo herdado da engenharia de softwares. Atores cumprem papéis, sendo que um papel de usuário é uma abstração para representar o relacionamento entre usuários e um sistema. Ainda de acordo com Constantine (2006), um papel de usuário pode ser descrito pelo contexto que é executado, a forma característica em que é realizado e pelo critério de projeto evidente para o desempenho do papel. O exemplo a seguir é de Constantine (2006):

R01 - Current-Sales-and-Ticketing Role

context (of role performance): isolated in booth, likely facing queue of customers; final step in performance promotion and sales characteristics (of role performance): relatively simple task performed repeatedly with some training, prior experience likely; performed under some time pressure, which increases as show time

approaches

criteria (for support of role performance): simplified, highly efficient interaction; foolproof means of identifying customer, guarantee that all the right tickets are dispensed and received by the customer.

Usage-centered design foca em papéis ao invés de usuários por que as características dos papéis e seu relacionamento com o sistema tem mais relevancia imediata para o design do que as características da pessoa desempenhando o papel, e porque este relacionamento é um subconjunto de todos os aspectos possíveis do usuário.

3.1.2 Modelo de tarefas

Segundo Constantine (2001), o modelo de tarefas representa a estrutura do trabalho que os usuários precisam realizar em relação com um sistema. Um modelo de tarefas consiste de um conjunto de casos de tarefas (também chamados de casos de uso essenciais), e um mapa de inter-relacionamento entre esses casos. De acordo com Constantine (2006) um caso de tarefa, ou caso de uso essencial, representa uma única intenção do usuário em interação com um sistema. São escritos como um diálogo, com intenções do usuário e responsabilidades do sistema, em linguagem abstrata e independente de tecnologia. Casos de tarefas são tipicamente pequenos e focados em objetivos altamente específicos. Cenários complexos podem ser expressos construindo casos de tarefas compostos por outros. O exemplo a seguir foi retirado do artigo Activity Modeling (CONSTANTINE, 2006, p. 6):

Ilustração 1 - Exemplo de caso de tarefa

T01 - <u>Issuing-Held-Ticket(s) for Performance(s)</u>	
USER INTENTIONS	SYSTEM RESPONSIBILITIES
2. provide customer identification 4. confirm by selection	1. request customer identification 3. provide confirming details 5. print tickets with in-process and completion notification

3.1.2.1 Mapa de casos de tarefas

Os relacionamentos entre os casos de tarefas são representados por um mapa de tarefas. Os relacionamentos possíveis podem ser especialização (caso específico de um caso de tarefa), extensão (caso particular que adiciona intenção alternativa a um caso de tarefa), composição (caso de tarefa é realizado por meio de outros que o compõe) e afinidade (similaridades não totalmente claras).

3.1.3 Modelo de conteúdo

De acordo com Constantine (2001, P. 4), o modelo de conteúdo, também chamado de protótipo abstrato, representa, de forma abstrata, o conteúdo e organização da interface de usuário necessária para apoiar as tarefas identificadas. Protótipos abstratos podem ter muitas formas, e de acordo com Constantine, servem de ponte entre o modelo de tarefas e protótipos realísticos.

3.2 Considerações sobre técnicas de projeto de Interface

Técnicas formais para design de interface em geral possuem diversas etapas em que são produzidos diversos documentos. Infelizmente, não foi possível seguir uma técnica rígida para este intuito devido as condições disponíveis para este trabalho, principalmente pela questão de tempo e falta de um número maior de voluntários para testar usabilidade e acessibilidade.

4 Desenvolvimento

Como mostramos, a interface atual do ambiente é completamente inacessível. Optei neste trabalho por construir uma interface totalmente nova e WEB. Tal escolha se deu pelos seguintes motivos: A acessibilidade Web é amplamente discutida, com normas (WCAG) amplamente, além de que uma interface WEB torna possível a utilização do software “na nuvem”, via Internet, dispensando a necessidade de instalação, o que também abre caminho para o acesso através de outros dispositivos equipados com diferentes sistemas operacionais (por exemplo tablets).

Para desenvolver esta solução foi utilizado Java Enterprise Edition (Java EE), Java Persistence API (JPA) e MySQL no servidor e no lado do cliente foi utilizado Javascript, Applet Java, HTML e WAI-ARIA. Através de componentes nativos do HTML e Javascript foi possível oferecer um ambiente análogo a versão Desktop da aplicação, permitindo a navegação total sem que em qualquer momento seja necessário recarregar a página completamente. Isto foi possível utilizando técnicas de Javascript conhecidas como AJAX. Há diferentes responsabilidades para as tecnologias utilizadas: No Applet é executado o modelo da aplicação, Javascript manipula elementos da Interface (que por sua vez são componentes HTML nativos) e faz comunicação com o servidor para carregar e persistir dados do usuário. No container Java EE, por sua vez, se localizam Servlets e páginas JSP, chamadas por requisições AJAX, com intuito de persistir ou fornecer os dados do usuário. Aplices, modelos e agendas são persistidas em MySQL, enquanto Imagens e Sons são persistidos em sistema de arquivos, assim como usuários da aplicação e respectivas senhas. Também é responsabilidade do container Java EE o gerenciamento de usuários e fornecer segurança.

No Applet Java, é executado o modelo da aplicação, que se trata de código legado da última versão Telis como aplicação Desktop, responsável pela execução das primitivas da linguagem Telis, que podem ser digitados diretamente no interpretador do ambiente, ou pela execução de modelos e aplices. A área gráfica da Applet corresponde a área gráfica de um aplice em execução ou a área do aplice da aplicação Desktop, ou seja, é nesta área que é exibido a saída gráfica das primitivas executadas (por exemplo, comRastros, Visível, pintar, etc...).

Não foi portanto utilizado a área do applet para se incluir elementos do ambiente, como pilha, entrada de comandos, caixas de diálogo, e demais elementos dos editores. A razão disto, é que, embora uma aplicação Java pode ser acessível utilizando a API Java para acessibilidade, tal tecnologia tem como requisito suporte especial da tecnologia assistiva ao Java Access Bridge (ORACLE, 2011). Por outro lado, é uma responsabilidade da tecnologia assistiva lidar com Componentes HTML nativos, pois são padrão da plataforma Web. Portanto, para todos os demais componentes gráficos foi utilizados componentes nativos do HTML, utilizando-se Javascript quando a aplicação necessita manipula-los e tratar eventos. Também é uma responsabilidade do Javascript a comunicação com o servidor, através de requisições AJAX, relacionadas a persistência.

Notar que Telis não possui o conceito de componentes gráficos (conhecidos como Widgets), e portanto não há na área do applet elementos que possam ser legíveis por tecnologias assistivas, o que dispensou o uso da Java Accessibility API. A questão de acessibilidade nas aplicações desenvolvidas em Telis será tratada em trabalhos futuros.

Visualmente o resultado é uma aplicação utilizável em uma única página após o login, embora constituído de diversas telas, referente aos diferentes editores e ao interpretador. Ao se navegar por estas partes do software, é utilizado a propriedade hidden do HTML 5 para esconder o conteúdo não correspondente a parte que se deseja visualizar. Usando este atributo, além de evitar a visualização, se oferece semântica adequada para que tecnologias assistivas ignorem estes componentes não visíveis. Algumas das telas do programa estão disponíveis nas ilustrações 2, 3, e 4. Através de WAI-ARIA é fornecido semântica adequada para que tecnologias assistivas possam interpretar todo o ambiente como uma aplicação (atributo role configurado para application), com regiões atualizáveis (ARIA LIVE-REGION), e caixas de diálogo (role dialog).

Ilustração 2 - Editor de Apliques

Editor de Apliques

Apliques:

nome: novoAplique

Largura: 500 Altura: 500

Modelos disponíveis: **Modelos que serão instanciados:**

umModelo

Ilustração 4 - Editor de Sons



4.1 Persistência e comunicação com o servidor

Para persistência de imagens e sons, foi utilizado sistema de arquivos. A razão é a simplicidade de código e menor número de operações, uma vez estes dados já são arquivos em sua origem, fornecidos como tal pelo usuário. Armazenar estes dados no banco de dados constituiria apenas um passo adicional desnecessário nos momentos de persistir e recuperar tais dados. Para modelos, agendas e aplicativos porém, foi adotada outra solução, persistência em banco de dados através da Java Persistence API (JPA), também por motivo de simplicidade

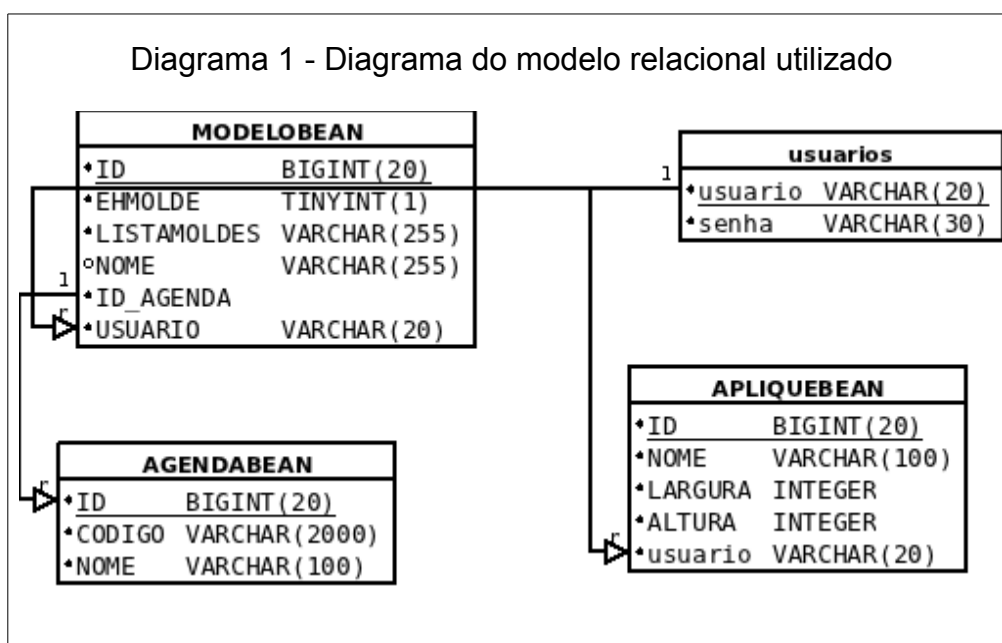
de código. A razão disto é que a origem destes dados não é arquivo, o que traz implicações adicionais devido a necessidade de conversão para um formato de arquivo e portanto, não tem a mesma trivialidade quanto a persistência de imagens e sons. A solução adotada porém, se mostrou trivial e elegante para a persistência e comunicação com o servidor.

Foi criado classes do tipo JavaBean para encapsular informações relevantes do modelo e possibilitar a comunicação com o servidor – AgendaBean para encapsular informações da agenda, ModeloBean para Modelos e Moldes, ApliqueBean para apliques e um Bean que encapsula todos os outros, AmbienteBean, utilizado na recuperação no início da sessão. Todas, exceto AmbienteBean, com as devidas anotações para persistência através da JPA. A conversão de objetos do modelo para JavaBeans e vice-versa é trivial, assim como a troca destes objetos entre o servidor e a applet também o é porque a mesma classe está disponível na aplicação servidor e na Applet, e a serialização de objetos é trivial em Java utilizando a classe ObjectOutputStream. Estes objetos são serializados para um array de bytes, e, como não é possível enviar dados binários diretamente em requisições e respostas HTTP, o dado binário é codificado como texto em base 64. Uma classe responsável por abstrair a serialização e deserialização dos objetos é incluída no servidor e na applet e utiliza uma biblioteca externa para a conversão array de bytes para base 64 e vice-versa. A serialização de objetos java simplifica a troca de objetos compostos de diferentes atributos, por ser trivial enviá-los numa única resposta ou requisição. Se um modelo do Telis, composto por um conjunto de agendas, tivesse que ser fornecido como resposta a uma requisição http através de outro método, teria que se criar uma forma de se separar as diversas informações, como código de cada agenda, título de cada agenda , título do modelo e moldes utilizados, possivelmente utilizando uma sintaxe XML própria.

O uso de JavaBeans, com anotações JPA, também tornou trivial a persistência e recuperação destes objetos, mesmo quando composto por atributos menores e relacionamentos, bastando a inserção de poucas anotações na classe, e o objeto pode ser persistido ou recuperado como um todo, ao invés de tratar separadamente a persistência e recuperação de cada atributo e relacionamento, como poderia ser necessário se não fosse disponível esta abstração.

Uma desvantagem em utilizar a serialização de objetos Java como formato de comunicação cliente-servidor para estes dados é a dependência da tecnologia Java em ambos os lados. Um formato texto ou XML poderia tornar transparente a migração de tecnologia em qualquer um dos lados. Mas o benefício da trivialidade é grande o suficiente para que esta questão seja tratada apenas quando houver a real necessidade, uma vez que não foi despendido muito tempo para o funcionamento atual. Os dados persistidos porém, não serão afetados por mudanças de tecnologia no servidor ou cliente, pois estão em um banco de dados MySQL em modelo relacional de alto nível, cujo manipulação é razoavelmente trivial através da maioria das tecnologias

No diagrama a seguir é apresentado o esquema de tabelas armazenado no banco de dados utilizado nesta aplicação.



O diagrama 2 é um diagrama de sequência que demonstra o procedimento de recuperação completa das agendas, moldes, modelos e apliques persistidos, como ocorre no momento em que a applet carrega.

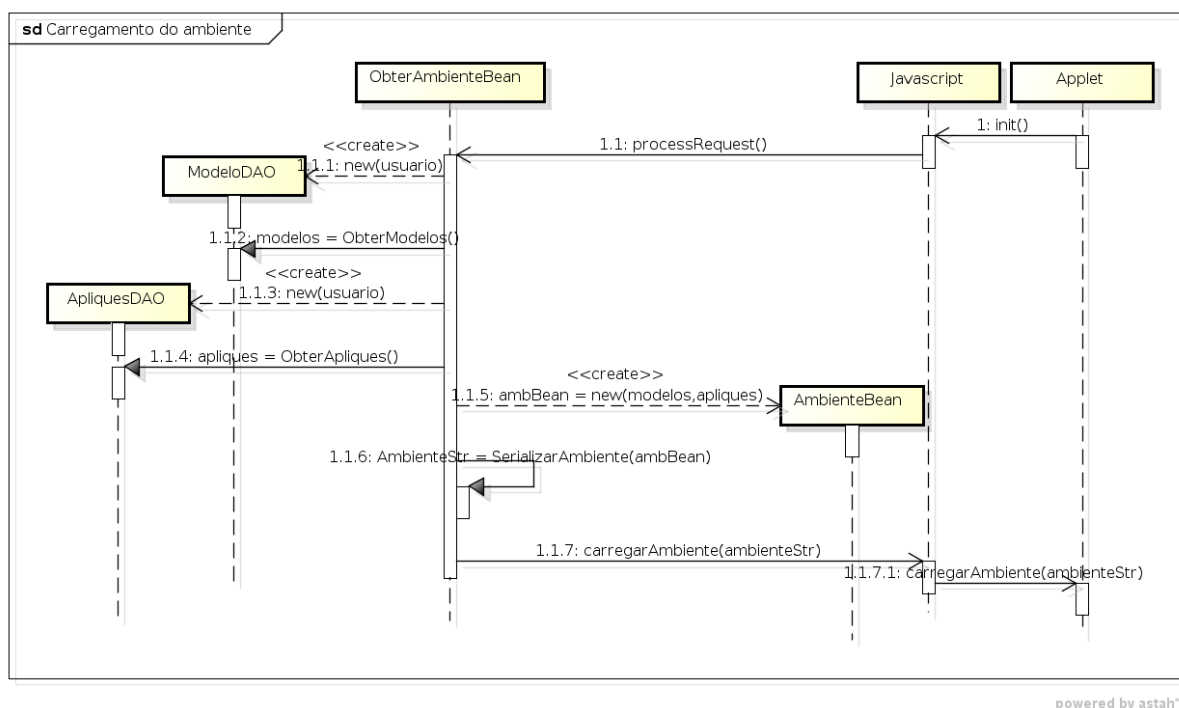
Notar que em passo simples, com uma única requisição foi possível recuperar todos os modelos, moldes, agendas e apliques persistidos. A sequência após ser enviado o retorno a applet, omitida do diagrama, é apenas deserializar o

texto recebido para obter a cópia do AmbienteBean, que fornece uma estrutura de dados do tipo Set contendo todos os modelos salvos, e outro Set para todos os apliques salvos, e a partir dos objetos JavaBeans, criar objetos do modelo da aplicação utilizando a API legada.

Já para dados de imagem e som, foi utilizada a biblioteca de upload de arquivos para recuperar os arquivos enviados pelo usuário, e então são mantidos em diretório específico do usuário para imagem ou som. O próprio servlet redireciona no servidor ao servlet que fornece a lista atual de URLs para as imagens e sons, para que o javascript possa atualizar o componente.

A persistência ocorre no momento em que modelos ou apliques são salvos ou são enviados imagens ou sons ao servidor. A recuperação de modelos e apliques porém ocorre apenas no momento que a applet inicia a execução e todos esses elementos são carregados em um único passo (exceto para imagens e sons que ocorre no momento do upload). Com exceção desta diferença, a recuperação é análoga a persistencia de um bean porém é feito o caminho inverso. Ao iniciar, a Applet Java invoca os métodos relacionados a recuperação dos modelos, moldes, agendas, apliques, imagens e sons persistidos. O Javascript então, faz uma requisição Ajax para o endereço de um Servlet responsável por fornecer no conteúdo da resposta um objeto serializado em base64 que encapsula todos os objetos JavaBeans do modelo. Esse texto é novamente convertido em JavaBean, e a partir destes é criado os objetos do modelo com a API legada. A recuperação de imagens ou sons porém, ocorre também sempre que é criado ou removido uma imagem ou som. Ela é fornecida por um servlet que fornece uma lista de urls de todas as imagens ou sons do usuário.

Diagrama 2 - Carregamento de modelos e aplicaes



4.2 Problemas encontrados

Foi encontrada problemas de diversas naturezas durante o processo. Nem todas está no escopo do trabalho resolver, como no caso de problemas com agentes de usuário.

Durante as últimas sessões houveram diversos casos em que uma determinada primitiva não funcionava conforme o manual. Esta é uma violação da heurística de Nielsen de número 10, e é um contra-senso para uma linguagem feita para se aprender fazendo. No entanto, correções no manual fogem ao escopo deste trabalho, ao invés disto, é apontado como trabalho futuro a revisão do mesmo.

Foi também observado um bug na leitura de páginas pelo agente de usuário Firefox utilizando o leitor de telas Orca, devidamente reportado no launchpad do ubuntu, sendo possível acompanhar seu estado em <https://bugs.launchpad.net/ubuntu/+source/gnome-orca/+bug/880117>. Sempre que um elemento de tag select com valor do atributo size maior que 1 fica vazio, a acessibilidade de todo o formulário fica comprometida, em alguns casos não lendo

componentes ou relacionando indevidamente o label do componente vazio indevidamente com outro componente. Como é um objetivo ser acessível utilizando Orca e Firefox, o problema teve que ser contornado inserindo um elemento option de valor nulo sempre que uma tag select for ficar vazia.

Foi identificado um grande problema com relação a captura de eventos do teclado. Sempre que se navega pela página web e o applet perde o foco, é impossível voltar ao applet navegando via teclado. O foco somente pode voltar com clique do mouse (inacessível) ou com a execução do método `requestFocus()` no applet ou em componentes swing. A solução foi garantir que o applet não seja alcançável via teclado usando a propriedade `tabindex="-1"` e utilizar um botão, alcançável que chama o método `requestFocus()` no applet. Vale lembrar que outra solução que pudesse alcançar o applet pelas teclas tab não seria acessível se o usuário não pudesse saber que está entrando na área da applet. Assim, foi testada a solução de um label chamar via javascript o método `requestFocus` no applet quando ocorresse o evento `onfocus` no label, porém, se mostrou inacessível visto que o applet ganha imediatamente o foco antes que o orca termine de ler o label, e a leitura é interrompida. Alguns agentes de usuário porém, acontece também o problema de não ser possível retornar o foco a página quando o foco entra na applet. Para contornar isto, foi tratada a tecla TAB para sempre chamar na página o método `focus` de um elemento apropriado.

Outra falha é que caixas de diálogo fornecidas pelos métodos javascript `alert`, `confirm` e `prompt`, apesar de serem fornecidos em caixa de diálogo nativa do navegador Web não eram lidas corretamente em alguns agentes de usuário. Este problema foi contornado substituindo seu uso por caixas de diálogo fornecidas pela biblioteca JQuery UI. Estas caixas de diálogo já tem nativamente configurado os atributos WAI-ARIA `aria-labelledby` para o título e `aria-describedby` para o conteúdo da caixa de diálogo. No entanto, foi verificado durante as sessões de validação com o usuário, que é esperado a leitura automática do conteúdo, e com esta configuração padrão isto não ocorreu nos agentes de usuário testados. Para que a leitura automática do conteúdo da caixa de diálogo, e não somente do título, ocorresse, foi necessário alterar o atributo `aria-labelledby` para incluir o conteúdo do diálogo.

5 Comentários sobre usabilidade baseado nas heurísticas de Nielsen

5.1 Visibilidade do estado do sistema

Não foi verificado no ambiente Telis alguma operação que possa manter o usuário não informado sobre o que está acontecendo por um período longo. No caso específico de conexões lentas o envio de imagens e sons para a versão Web desenvolvidas neste trabalho pode demorar, mas quando isto acontece o usuário é devidamente informado que a lista de imagens ou som está carregando. Não foi observado outra operação no ambiente que possa ter um longo tempo de espera, sendo necessário feedback.

5.2 Relação entre sistema e mundo real

De maneira geral, foi utilizado nesta nova versão os mesmos termos encontrados na versão legada sempre que possível. Vale ressaltar que Telis possui termos próprios em português de uso comum como modelos e agenda, para designar elementos da linguagem.

5.3 Controle e liberdade do usuário

Embora não exista opção desfazer nativa, em todos os navegadores Web modernos será possível utilizar as operações de desfazer e refazer para tratar de alterações acidentais indevidas no código da agendas. Cabe salientar também que sempre que se muda a agenda atual no editor e há alterações não salvas é perguntado se deseja-se salvar as mudanças. No momento não é possível porém desfazer operações de apagar elementos como moldes, agendas, imagens, sons. Estes erros são evitados com caixa de diálogo de confirmação.

5.4 Consistência e padrão

Não foi observado na ferramenta final problemas relacionados a diferentes palavras para se referir a um mesmo conceito ou termos diferentes do que o usuário

da plataforma está acostumado. No geral são usados termos comuns como “editores”, “salvar”, “novo”.

5.5 Prevenção de erros

Em todas as alterações importantes (apagar, renomear, criar novo, etc) é exibida uma caixa de diálogo de confirmação, evitando alterações acidentais.

5.6 Reconhecimento ao invés de memória

Todos os itens do ambiente Telis Web são visíveis na tela, e portanto não se verifica falha ao atingir esta heurística.

5.7 Flexibilidade e eficiência no uso

De maneira geral, o ambiente Telis não desperdiça eficiência, por exemplo, selecionando o texto ao se executar um comando no diálogo principal, para permitir a digitação de outro comando, ou selecionando nova agenda ou modelo após a mesma ser criada. A versão Web mantém as mesmas características.

5.8 Design e estética minimalista

A disposição dos componentes seja baseada na versão legada do Telis. Mensagens de erro se tornaram mais minimalistas não exibindo informações irrelevantes.

5.9 Ajuda usuários a reconhecer, diagnosticar e recuperar erros

As mensagens de erro em geral são fornecidas pelo modelo e foram legadas da versão anterior, não sendo avaliado portanto possíveis melhorias. No entanto, foi eliminado informações irrelevantes de diálogos de erro como a informação do estado da pilha (que já é visível na tela) e a exceção que foi disparada.

5.10 Ajuda e documentação

Telis possui documentação. Revisar a mesma está fora do escopo deste trabalho, embora foi observado durante o desenvolvimento diversas falhas. Para cumprimento desta heurística foi inserido um link para o manual.

6 Trabalhos Futuros e Considerações Finais

Este trabalho, conforme explicado anteriormente, não resolve todas as barreiras de acessibilidade relacionadas ao Telis, embora tenha sido um grande avanço fornecer um ambiente acessível, onde é possível o uso de pelo menos parte da linguagem cujas saídas não são afetadas pela deficiência do usuário. No entanto, para que se possa reivindicar o Telis como uma ferramenta acessível ainda é necessário um estudo para adequar as primitivas Telis a necessidade de usuários com diferentes tipos de deficiência. Telis não possui um meio de por exemplo emitir um texto acessível para tecnologias assistivas, ficando o usuário cego limitado a primitivas relacionadas a emitir sons e tratamento de teclado. Embora a primitiva JS possa ser utilizada para, através de código Javascript, acessar elementos acessíveis da página, é importante que isto possa ser feito utilizando uma primitiva nativa da linguagem. Usuários com outras deficiências e combinações tem necessidades diferentes. Pessoas com surdocegueira por exemplo, necessitam de um display braile para leitura do conteúdo da tela. Por não emitir conteúdo acessível, Telis não pode produzir qualquer conteúdo perceptível para um usuário com esta deficiência. Mesmo sem depender do conceito de componentes gráficos (widgets) acessíveis, seria importante durante o aprendizado buscar meios de se produzir saídas, especialmente texto, perceptíveis a usuários com diferentes tipos de deficiência.

Um passo além de produzir conteúdo perceptível para o usuário desenvolvendo com a linguagem (conforme tratado no parágrafo anterior), é também necessário um estudo sobre as formas de se produzir conteúdo acessível utilizando Telis, considerando os casos do usuário desenvolvendo ter ou não uma deficiência. A acessibilidade de aplicativos desktop ou Web é fornecida através do uso de componentes gráficos que podem ser reconhecidos por tecnologias assistivas para que se possa fornecer uma alternativa de uso. Telis não possui nativamente o conceito de componentes gráficos (conhecidos como Widgets) que possuem uma forma de uso comum, e portanto não há componentes a que se pode atribuir semântica. Interfaces em Telis são construídas através de primitivas que permitem aos atores emitir saída gráfica (pintar, riscar, entre outros) em áreas específicas do aplicação (dependendo de coordenadas), emitir sons, e tratar eventos de teclado e mouse. Embora possa ser uma alternativa válida no processo de aprendizagem,

esta forma de produzir conteúdo depende de coordenadas para posicionar elementos na tela (cegos não poderão produzir elementos visuais) e não possui semântica. Cabe informar que o laboratório Edugraf também desenvolveu uma tecnologia chamada Dix que permite o desenvolvimento de interfaces gráficas Web utilizando componentes nativos da especificação XHTML e é compatível com Telis. Porém, foi verificado neste trabalho que as páginas Dix não são acessíveis, não sendo possível fornecer semântica correta a elementos da página ou alternativa textual para todos os elementos, além de depender de coordenadas para posicionar elementos e não fornecer navegação adequada via teclado. É uma indicação para trabalhos futuros criar meios para se criar conteúdo acessível com Telis e obter conformidade com as ATAG.

Durante o processo foram observadas diversas falhas no manual. Conforme dito anteriormente, é uma contradição com o objetivo de ser um software que permite aprender fazendo e também é uma falha de usabilidade, portanto, a revisão da documentação é um importante trabalho futuro.

Outra possibilidade de estudo é relacionada a formas de representação espacial para usuários cegos com Telis. Telis possui o conceito de atores, com representação visual e posição na tela, e manipulado através de comandos da linguagem. É o mesmo conceito da tartaruga gráfica da linguagem Logo, na qual Seymour Papert (1980, 1993) sugere usos diversos na educação, como apresentar conceitos de geometria e física. Assim, o estudo de formas alternativas do uso de uma linguagem de programação educacional para apresentar conceitos que dependem de representação espacial no caso de aprendizes cegos é mais uma indicação para trabalhos futuros.

Um indicativo coerente de trabalho futuro é oferecer contribuições a comunidade de software livre para melhorias em geral relacionadas a acessibilidade. Por exemplo, melhorias no leitor de telas Orca ou sintetizador de voz espeak, entre outros.

Por fim, apesar de ainda ser necessário outros estudos relacionados a acessibilidade em ambiente Telis, o objetivo de fornecer um ambiente Telis acessível para o caso de uso de um usuário cego foi atingido, sendo que o usuário foi capaz de navegar sozinho pela interface e acessar diferentes componentes da interface gráfica sem auxílio externo, e sem dúvida o Telis se mostrou como

ferramenta viável no momento inicial de aprendizagem de programação de um usuário cego, e este resultado abre caminho para tornar viável novas pesquisas como os trabalhos futuros indicados. No entanto, com o progresso do usuário, as limitações da linguagem devem tornar-se latentes.

Também é um importante resultado o fato de não ter sido utilizado software proprietário em nenhum momento no processo, e tanto o lado do cliente como servidor não dependem de software proprietário, sendo coerente com o fato de Telis ser software livre e feito para produzir software livre.

7 Referências bibliográficas

ABNT, ABNT ISO 9241-11:2002

ACESSO DIGITAL (Brasil). **Acessibilidade Web: Custo ou Benefício?**. Vídeo em formato digital. Disponível em: <<http://acessodigital.net/video.html>>. Acesso em: 27 ago. 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 9241: **Requisitos Ergonômicos para Trabalho de Escritórios com Computadores**: Parte 11 – Orientações sobre Usabilidade. Rio de Janeiro, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 9126: **Engenharia de software - Qualidade de produto** . Rio de Janeiro, 2003.

BATISTA, Cecília Guarnieri. **Formação de Conceitos em Crianças Cegas**: Questões Teóricas e Implicações Educacionais. Psicologia: Teoria e Pesquisa, Campinas, n. , p.7-15, abr. 2005. Disponível em: <<http://www.scielo.br/pdf/ptp/v21n1/a03v21n1.pdf>>. Acesso em: 22 jun. 2011.

KOGICS. **Kojo**. Disponível em: <<http://kogics.net/sf:kojo>>. Acesso em: 01 dez. 2011.

OCHAÍTA, Esperanza; HUERTAS, Juan Antonio; ESPINOSA, Angeles. **Representación espacial en los niños ciegos**: una investigación sobre las principales variables que la determinan y los procedimientos de objetivación más adecuados. Infancia Y Aprendizaje, [madrid], v. 79, n. 53, p.53-79, 1991. Disponível em: <http://sid.usal.es/docs/F8/ART5336/representacion_espacial_ni%C3%B1os_ciegos.pdf>. Acesso em: 21 jun. 2011.

BRASIL. Decreto nº 5.296, de 2 de dezembro de 2004. Regulamenta as Leis nos 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências. **Diário Oficial [da República Federativa do Brasil]**, Brasília, DF, 03 dez. 2004. p. 5. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2004/decreto/d5296.htm>. Acesso em: 24 ago. 2011.

BRASIL. Portal Oficial do Estado Brasileiro. Governo Federal. **O que é acessibilidade?** Disponível em: <<http://www.brasil.gov.br/menu-de-apoio/apoio/perguntas-frequentes/o-que-e-acessibilidade>>. Acesso em: 24 ago. 2011.

BECHARA, Everaldo. **Recomendações de Acessibilidade para Conteúdo Web (WCAG) 2.0**: Recomendação W3C de 11 Dezembro de 2008. Traduzido em: 2 mar. 2009. Disponível em: <<http://www.ilearn.com.br/TR/WCAG20/>>. Acesso em: 04 dez. 2011.

CONSTANTINE, Larry L.. **Activity Modeling**: Toward a Pragmatic Integration of

Activity Theory with Usage-Centered Design. 2006. Disponível em: <<http://www.foruse.com/articles/activitymodeling.pdf>>. Acesso em: 16 out. 2011.

CONSTANTINE, Larry L.; LOCKWOOD, Lucy A. D.. **Usage-Centered Engineering for Web Applications**. 2001. Disponível em: <<http://www.sis.pitt.edu/~gray/INFSCI2510/docs/UIModeling/UsageCenteredDesign/webapplications.pdf>>. Acesso em: 16 out. 2011.

Constantine, L. "**Usage-Centered Software Engineering**: novos modelos, métodos e métricas." Em Purvis, M. (ed.) Engenharia de Software: Educação e Prática. Los Alamitos, CA: IEEE Computer Society Press, 1996.

D'AMBRÓSIO, Ubiratan. **Etnomatemática**: Elo entre as tradições e a modernidade. 2ª Edição. Belo Horizonte: Autêntica, 2002. 110 p. (Coleção Tendências em Educação Matemática)

D'AMBROSIO, Ubiratan. **Site oficial de Ubiratan D'Ambrosio**. Disponível em: <<http://vello.sites.uol.com.br/ubi.htm>>. Acesso em: 11 ago. 2011.

D'AMBROSIO, Ubiratan. **O programa etnomatemática**: história, metodologia e pedagogia. Disponível em: <<http://vello.sites.uol.com.br/program.htm>>. Acesso em: 11 ago. 2011.

FERREIRA, Eduardo Sebastiani. **O que é Etnomatemática**. Disponível em: <<http://www.ufrj.br/leprtrans/arquivos/etno.pdf>>. Acesso em: 11 Ago. 2011.

KAMALUDIN, Khairuddin; YATIM, Noor Faezah Mohd; NORDIN, Md. Jan. Refining technical and learning accessibility elements on e-learning for user with visual impairment. In: INFORMATION TECHNOLOGY (ITSIM), 2010 INTERNATIONAL SYMPOSIUM IN, 2., 2010, Kuala Lumpur. **Information Technology (ITSim), 2010 International Symposium in**. Kuala Lumpur: Ieee, 2010. p. 1 - 4. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5561331>>. Acesso em: 24 ago. 2011.

MACEDO, Lino de. **Ensaio Construtivistas**. 6. ed. São Paulo: Casa do Psicólogo, 1994. 170 p.

NAÇÕES UNIDAS. **Convention on the Rights of Persons with Disabilities**. Disponível em: <<http://www.un.org/disabilities/default.asp?navid=12&pid=150>>. Acesso em: 11 dez. 2011.

NIELSEN, Jakob. **About Jakob Nielsen**. Disponível em: <<http://www.useit.com/jakob/>>. Acesso em: 11 dez. 2011.

NIELSEN, J., ; MOLICH, R. (1990). **Heuristic evaluation of user interfaces**, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-25

PAPERT, Seymour. **Constructionism Learning**. Ed. Idit Harel. Cambridge, MA: MIT Media Laboratory, 1990.

PAPERT, Seymour. **The Children's Machine**. New York: Basic Books, 1993.

PAPERT, Seymour. **Mindstorms**. New York: Basic Books, 1980. 230 p.

PIAGET, Jean. **O Nascimento da Inteligência na Criança**. 10. ed. Lisboa: Delachaux & Niestlé S.a., 1971. 389 p. (Plural).

SWAN, Karen. **A constructivist model for thinking about learning online**. In: JOHN R. BOURNE (Org.). *Elements of Quality Online Education: Engaging Communities*. Needham: The Sloan Consortium, 2005. p. 13-30. Disponível em: <<http://books.google.com/books?id=wOnGofjH8BkC&lpg=PP1&dq=Elements%20of%20quality%20online%20education%3A%20engaging%20communities&hl=pt-BR&pg=PA13#v=onepage&q&f=false>>. Acesso em: 20 jun. 2011.

PIERI, Giovanni et al. Telis: a Programming Tool Set for Beginners. In: INTERNATIONAL INFORMATION AND TELECOMMUNICATION TECHNOLOGIES SYMPOSIUM, 8., 2009, Florianópolis. **Proceedings of the 8th International Information and Telecommunication Technologies Symposium**. Florianópolis: IEEE, 2009. p. 183 – 186.

PROGRAMMING EDUCATION TOOLS GROUP – Group at the School of Computing - Computing Education Research – University Of Kent. **Greenfoot**. Disponível em: <<http://www.greenfoot.org/>>. Acesso em: 01 dez. 2011.

PROGRAMMING EDUCATION TOOLS GROUP – Group at the School of Computing - Computing Education Research – University Of Kent. **BlueJ**. Disponível em: <<http://bluej.org/>>. Acesso em: 01 dez. 2011.

ROSA, Paulo Ricardo da Silva. **A Teoria de Vygotsky**. Disponível em: <http://www.dfi.ufms.br/prrosa/Pedagogia/Capitulo_5.pdf>. Acesso em: 10 ago. 2011

SQUEAK Site oficial. Disponível em <<http://www.squeak.org>>. Acesso em 01 dez. 2011

THATCHER, James. **What is Accessible Software?** IBM. Jul. 1997. Disponível em: <http://www.empowermentzone.com/acc_soft.txt>. Acesso em: 24 ago. 2011.

VILELLA, Renata Moutinho. **Conteúdo, Usabilidade e Funcionalidade: três dimensões para a avaliação de portais estaduais de Governo Eletrônico na We**. Escola de Ciência da Informação da UFMG, Belo Horizonte, 2003

VYGOTSKY, L. S. **Pensamento e linguagem**. Martins Fontes: São Paulo, 2000.

WOOD, 1997 WOOD, Larry E. **User Interface Design: Bridging the Gap from User Requirements to Design**. Boca Raton, Florida, 1997

WCAG SAMURAI. **WCAG Samurai**. Disponível em: <<http://wcagsamurai.org/>>. Acesso em: 04 dez. 2011.

WCAG SAMURAI. **WCAG Samurai Errata for Web Content Accessibility**

Guidelines (WCAG) 1.0. Publicado em: 26 Fev. 2008. Disponível em: <<http://wcagsamurai.org/errata/errata.html>>. Acesso em: 04 dez. 2011.

WCAG SAMURAI. **WCAG Samurai Errata para as Recomendações para Acessibilidade ao conteúdo da Web (WCAG) 1.0.** Publicado em 26 Fev. 2008. Traduzido por: Maurício Samy Silva. Disponível em: <<http://www.maujor.com/wcagsamurai/errata-list-pt-br.html>>. Acesso em: 04 dez. 2011.

WEBAIM. **Keyboard Accessibility.** Disponível em: <<http://webaim.org/techniques/keyboard/>>. Acesso em: 11 dez. 2011.

WEBAIM. **Survey of Preferences of Screen Readers Users.** 2009. Disponível em: <<http://webaim.org/projects/screenreadersurvey/>>. Acesso em: 02 dez. 2011.

WORLD WIDE WEB CONSORTIUM (W3C). **About W3C.** Disponível em: <<http://www.w3.org/Consortium/>>. Acesso em: 17 out. 2011.

WORLD WIDE WEB CONSORTIUM (W3C). **Authoring Tool Accessibility Guidelines 1.0:** W3C Recommendation 3 February 2000. Disponível em: <<http://www.w3.org/TR/ATAG10/>>. Acesso em: 29 out. 2011.

WORLD WIDE WEB CONSORTIUM (W3C). **Authoring Tool Accessibility Guidelines (ATAG) 2.0:** W3C Working Draft 21 July 2011. Disponível em: <<http://www.w3.org/TR/ATAG20/>>. Acesso em: 29 nov. 2011.

WORLD WIDE WEB CONSORTIUM (W3C). **Accessible Rich Internet Applications (WAI-ARIA) 1.0:** W3C Candidate Recommendation 18 January 2011. Disponível em: <<http://www.w3.org/TR/wai-aria/>>. Acesso em: 17 out. 2011

WORLD WIDE WEB CONSORTIUM. **Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification:** W3C Recommendation 07 June 2011. Disponível em: <<http://www.w3.org/TR/CSS2/>>. Acesso em: 12 dez. 2011.

World Wide Web CONSORTIUM (W3C). **Document Object Model (DOM).** Disponível em: <<http://www.w3.org/DOM/>>. Acesso em: 12 dez. 2011.

WORLD WIDE WEB CONSORTIUM (W3C). **WAI-ARIA FAQ:** What is the current status of WAI-ARIA development?. Disponível em: <<http://www.w3.org/WAI/aria/faq#status>>. Acesso em: 17 out. 2011.

WORLD WIDE WEB CONSORTIUM (W3C). **WAI-ARIA Overview.** Disponível em: <<http://www.w3.org/WAI/intro/aria.php>>. Acesso em: 17 out. 2011.

WORLD WIDE WEB CONSORTIUM (W3C). **Web Content Accessibility Guidelines (WCAG) 2.0:** W3C Recommendation 11 December 2008. Disponível em: <<http://www.w3.org/WAI/WCAG20/>>. Acesso em: 16 out. 2011.

Glossário

Agente de usuário: Qualquer programa que obtenha e apresente conteúdo da Web ao usuário, incluindo navegadores Web, plug-ins e tecnologias assistivas (W3C, 2008).

AJAX: Técnicas de Javascript e XML para consultar o servidor e atualizar dinamicamente partes de uma página Web sem que seja necessário recarregá-la completamente

Aplicação Desktop (como usado neste trabalho): Aplicação não web, que executa no sistema operacional do usuário.

Cascading Style Sheets (CSS): Linguagem de estilo da W3C para definir a apresentação de documentos em uma linguagem de marcação (W3C, 2011).

Document Object Model: Interface independente de plataforma e linguagem, criada pela W3C que permite a programas de script acessar e atualizar o conteúdo, estrutura e estilo de documentos (W3C, 2011).

Extensible Markup Language (XML): Linguagem de marcação da W3C para necessidades especiais.

GNU General Public License: Licença de software livre, escrita pela fundação do para o software livre para o projeto GNU e amplamente utilizada em demais softwares livres.

JavaBean: São componentes reutilizáveis para Java, construídos através de uma classe Java com uma convenção particular: A classe deve ser serializável, com construtor nulo e permitir acessar propriedades por métodos getters e setters.

Leitor de Telas: Tecnologia assistiva que interpreta o que está sendo mostrado na tela e reapresenta em outra saída, como através de um sintetizador de voz, para um dispositivo braile, entre outros.

Navegador Web: Software que permite apresentar e interagir com conteúdo Web

Software livre: Software que permite seu uso para qualquer propósito, permite estudar como o programa funciona e modificá-lo, e permite redistribuir cópias (FSF, GNU, 2011)

Tecnologia assistiva: Hardware ou software que funcionam como agente de usuário, ou juntamente com um agente de usuário convencional, para fornecer funcionalidades para atender a requisitos de pessoas com deficiências.

Web: Utilizado neste trabalho como forma abreviada de World Wide Web.

APÊNDICE A – Artigo sobre o trabalho desenvolvido

Uma Interface Gráfica Para o Ambiente Telis Focada em Acessibilidade para Cegos

Felipe A. Ferreira¹

¹Departamento de informática e estatística (INE) – Centro Tecnológico –
Universidade Federal de Santa Catarina (UFSC)

felipealexandref@gmail.com

Abstract. *This work deals with the development of a Web graphical user interface for the Telis environment, respecting the general principles of accessibility and WCAG 2.0 standards, taking also into account usability principles. The use by a blind user of developed tool has been proven feasible in field work. It was opted for not use any technology that relies on proprietary software to use, either server or client. It will also be addressed specific considerations on the possibilities and constraints of the didactic use of the tool for the blind Telis today and its potential use in future as this paper notes.*

Resumo. *Este trabalho trata do desenvolvimento de uma interface gráfica Web para o ambiente Telis, respeitando princípios gerais de acessibilidade e normas WCAG 2.0, levando também em conta princípios de usabilidade. O uso por um cego da ferramenta desenvolvida foi comprovado viável em pesquisa de campo. Optou-se por não utilizar nenhuma tecnologia que dependa de software proprietário para a utilização, seja no servidor ou no cliente. Também será tratado considerações específicas sobre as possibilidades e impedimentos do uso didático da ferramenta Telis para cegos hoje e seu uso potencial no futuro, dado apontamentos deste trabalho.*

1. Introdução

O Laboratório de Software Educacional (EDUGRAF) da Universidade Federal de Santa Catarina (UFSC) desenvolveu a linguagem e ambiente de desenvolvimento Telis, com principal objetivo suportar aprendizes que nunca tiveram contato com uma linguagem de

programação. Telis fornece abstrações que permitem apresentar de maneira simples conceitos avançados da ciência da computação, como orientação a objetos, computação distribuída, programação concorrente, orientação a eventos, interface gráfica, entre outros. O objetivo é favorecer uma abordagem construtivista, em que se aprende fazendo ao invés de um processo em que o aprendiz apenas recebe instruções do professor e as armazena (PIERI, et. al). Entretanto, até o desenvolvimento deste trabalho, o ambiente Telis legado encontra-se inacessível para cegos, e possivelmente usuários com outras deficiências, pois não foi desenvolvido para ser utilizado com tecnologias assistivas.

Neste trabalho, foi desenvolvido uma nova interface gráfica com o usuário (doravante GUI – do inglês graphical user interface) para o ambiente Telis, acessível e Web, de acordo com as recomendações vigentes da W3C para acessibilidade, publicadas no documento Web Content Accessibility Guidelines 2.0, com acessibilidade testada por um usuário cego aprendiz e com resultado positivo. Entretanto, apesar de o ambiente com a nova GUI ser acessível, nem todas as questões de acessibilidade relacionadas a Telis são concluídas nesse trabalho. Este trabalho se limita a acessibilidade do ambiente, e não do conteúdo gerado. Questões como se há uma alternativa ao ator gráfico para representação espacial no caso de cegos, como produzir apliques acessíveis com Telis e que novas primitivas poderia-se adicionar a linguagem para melhor atender as necessidades deste público, foram brevemente tratadas como apontamentos para trabalhos futuros.

2. Revisão Bibliográfica

Para implementar esta ferramenta foi feito pesquisas na literatura com relação a usabilidade e acessibilidade. Também foi pesquisado questões relacionadas ao aprendizado de pessoas cegas, que não chegaram a ter implicações diretas na implementação desta ferramenta específica, mas foi mantido como registro importante para o desenvolvimento dos trabalhos futuros.

2.1. Usabilidade

Usabilidade é, de acordo com ABNT ISO 9241 (2002) “Medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso”, ou de acordo com NBR ISO/IEC 9126 (2003,

p. 9) “Capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas.”.

Embora não seja um requisito deste trabalho melhorias de usabilidade em relação a versão legada do Telis, houve um esforço para evitar problemas de usabilidade. Assim, foi estudado técnicas para design de interface, porém, como não houve condições de tempo para aplicar uma abordagem formal de desenho de interface, foi desenvolvida uma interface análoga a versão legada do Ambiente Telis e feita avaliação heurística da mesma.

A avaliação heurística é um método informal de avaliação da usabilidade, na qual é avaliado uma lista de princípios pré-estabelecidos de usabilidade. Foi utilizado para este intuito as heurísticas de Nielsen. Seu autor, Jakob Nielsen, foi um importante pesquisador da área de usabilidade, ex-funcionário da Sun Microsystem, onde trabalhou grande parte do tempo no campo de usabilidade na Web, inclusive no website e intranet da Sun, foi autor de diversos livros sobre usabilidade e possui Ph.d em interação humano-computador da Universidade técnica da Dinamarca.

2.2. Acessibilidade

No contexto de software, acessibilidade poder ser entendido como um foco especial da usabilidade para o grupo específico de usuários com deficiência. Assim, acessibilidade trata de evitar barreiras que impedem ou prejudiquem o uso de um software ou conteúdo por uma pessoa com deficiência.

Pessoas com deficiência podem depender de tecnologias assistivas para usar computador, tais como reconhecimento de voz; leitor de tela e texto; display Braille; teclado Braille, entre outros [KAMALUDIN, YATIN e NORDIN, 2010] e frequentemente possuem limitação de uso com certos dispositivos como mouse. No caso específico de um usuário cego, frequentemente é utilizado um software leitor de telas [DELPIZZO, 2005] e utilizado teclado para ações que podem ser feitas com o mouse [WEBAIM, 2011].

2.2.1. Diretrizes de acessibilidade para conteúdo Web

Em 1997 a W3C criou a iniciativa para acessibilidade na Web (do inglês Web Accessibility Initiative – WAI), que consiste em diversos grupos de trabalho e grupos de interesses

especiais. Em 1999, o Grupo de Trabalho para as Diretrizes de Acessibilidade ao Conteúdo Web (do inglês Web Content Accessibility Guidelines Working Group – WCAG WG) publicou a primeira norma da W3C para acessibilidade na web, as Diretrizes de Acessibilidade ao Conteúdo Web (do inglês Web Content Accessibility Guidelines – WCAG). Uma nova versão do documento, WCAG 2.0, foi lançada em 2008, e é a recomendação vigente da W3C para novos conteúdos.

2.2.2. WAI-ARIA

Com o desenvolvimento da tecnologia, a Web, antes apenas uma rede de documentos de hipermídia, se tornou uma plataforma de aplicações, fenômeno ao qual foi cunhado o termo “Web 2.0” em 2004. Este fenômeno trouxe novas implicações para a acessibilidade na Web, devido ao comportamento dinâmico destas aplicações e do fato de frequentemente ser utilizado diversas técnicas para criar componentes de interface que não são nativamente presente na tecnologia utilizada, ou com comportamento diferente do esperado, como caixas de diálogo, regiões atualizadas dinamicamente, etc. Até então, não era possível fornecer semântica adequada para que a tecnologia assistiva possa fornecer uma alternativa de uso adequada a aplicações Web desenvolvida com tecnologia W3C (como HTML e XHTML e Javascript)¹. O suporte a tecnologias assistivas de aplicações desktop legadas é fornecido por uma infra-estrutura e API que permite fornecer a tecnologias assistivas semântica apropriada para o uso de aplicações dinâmicas e ricas, e o WAI-ARIA tem o intuito de fornecer esta API para aplicações Web [WAI-ARIA Primer 1.0, W3C, 2010].

O WAI-ARIA permite fornecer a semântica para componentes de interface não nativos ou com comportamento dinâmico através de atributos. O atributo role permite definir que uma região se comporta como aplicação ou como determinado elemento de interface. Demais atributos permitem definir estados (por exemplo, se determinado elemento está visível, pressionado, ativo, etc) e outros valores (por exemplo, estado atual de uma barra de progresso, outro componente que tem a descrição ou título do componente, etc).

3. Desenvolvimento

¹ Cabe salientar que conteúdos desenvolvidos com tecnologias como Java e Flash já possuem API para acessibilidade própria, mas que depende de suporte específico da tecnologia assistiva.

Este trabalho foi desenvolvido com utilizando tecnologias Javascript, AJAX, Java EE (páginas com tecnologias JSP e Servlets), Java Applet, Java Persistence API (JPA), MySQL, HTML e WAI-ARIA. Para componentes de interface foi utilizado componentes nativos da especificação HTML, exceto por caixas de diálogo em que foi utilizada a biblioteca JQuery UI, que por sua vez utiliza Javascript, elementos HTML e atributos WAI-ARIA.

O uso de Java Applet foi necessário para reuso de código-fonte do interpretador Telis e do modelo da aplicação legados. A área visual da applet é utilizada somente pelas primitivas (comandos da linguagem) que permitem manipular a área visível, a mesma área de um aplique Telis publicado, não sendo usado a applet para fornecer componentes de interface que são parte do ambiente Telis. Como Telis não tem o conceito de componentes de interface reutilizáveis (widgets), não há componentes a se atribuir qualquer semântica adequada e portanto a API de acessibilidade do Java (JAAPI) não foi utilizada. Isto no entanto traz implicações quanto a acessibilidade do conteúdo produzido, que será tratada em trabalhos futuros. No momento, a intenção é permitir que um usuário cego possa ter acesso completo ao ambiente e iniciar um aprendizado em programação através de primitivas da linguagem que permitem tratar eventos de teclado e emitir sons.

O Javascript é responsável para quando a aplicação necessita manipular componentes da página (atualizar a pilha e lista de atores por exemplo), ou tratar eventos (executar comando ou salvar algo, que pode gerar requisições AJAX no servidor e chamadas a métodos específicos na applet).

O servidor é responsável pela persistência. Para persistência, cria-se na applet objetos JavaBean equivalentes a elementos do modelo, que são serializados e transmitidos numa requisição AJAX para o servidor, onde são recuperados e persistidos em banco de dados através da JPA. Para recuperar estes dados, é feito o caminho inverso: Após o login de um usuário, é feito uma requisição ao servidor, que recupera os objetos JavaBeans persistidos através da JPA, encapsula-os em um único objeto, serializa-os e emite como página de resposta. Estes objetos JavaBeans são então recuperados na applet e convertidos em objetos do modelo. Nota-se que apesar de a comunicação cliente-servidor ocorrer com objetos java, devido a JPA, estes dados são persistidos em banco de dados relacional com modelagem

adequada e independente da tecnologia utilizada na aplicação. Esta conversão de objeto java em modelo relacional é abstraída pela JPA.

Para validação do resultado foi utilizado as ferramentas de validação automática Access Monitor e Total Validator com intuito de garantir o cumprimento das WCAG 2.0 em nível de conformidade AA, além da ferramenta de validação da W3C para validar a marcação HTML. Também foi feito verificado a leitura dos componentes de interface com 3 leitores de telas específicos e foi efetuado 3 sessões com um usuário cego sem experiência anterior de programação para validação de um caso real. Durante estas sessões foram encontrados pequenas falhas que foram corrigidas.

3.1. Falhas encontradas no processo de desenvolvimento

Foi encontrada problemas de naturezas diversas durante o processo. Por exemplo, durante as últimas sessões houveram diversos casos em que uma determinada primitiva não funcionava conforme o manual. Esta é uma violação da heurística de Nielsen de número 10, e é um contra-senso para uma linguagem feita para se aprender fazendo. No entanto, correções no manual fogem ao escopo deste trabalho, ao invés disto, é apontado como trabalho futuro a revisão do mesmo.

Outra falha é que caixas de diálogo fornecidas pelos métodos javascript alert, confirm e prompt, apesar de serem fornecidos em caixa de diálogo nativa do navegador não eram lidas corretamente em alguns agentes de usuário. Este problema foi contornado substituindo seu uso por caixas de diálogo fornecidas pela biblioteca JQuery UI. Estas caixas de diálogo já tem nativamente configurado os atributos WAI-ARIA aria-labelledby para o título e aria-describedby para o conteúdo da caixa de diálogo. No entanto, foi verificado durante as sessões de validação com o usuário, que é esperado a leitura automática do conteúdo, e com esta configuração padrão isto não ocorreu nos agentes de usuário testados. Para que a leitura automática do conteúdo da caixa de diálogo, e não somente do título, ocorresse, foi necessário alterar o atributo aria-labelledby para incluir o conteúdo do diálogo.

4. Trabalhos Futuros

Este trabalho, conforme explicado anteriormente, não resolve todas as barreiras de acessibilidade relacionadas ao Telis, embora tenha sido um grande avanço fornecer um ambiente acessível, onde é possível o uso de pelo menos parte da linguagem cujas saídas não são afetadas pela deficiência do usuário. No entanto, para que se possa reivindicar o Telis como uma ferramenta acessível ainda é necessário um estudo para adequar as primitivas Telis a necessidade de usuários com diferentes tipos de deficiência. Telis não possui um meio de por exemplo emitir um texto acessível para tecnologias assistivas, ficando o usuário cego limitado a primitivas relacionadas a emitir sons e tratamento de teclado. Embora a primitiva JS possa ser utilizada para, através de código Javascript, acessar elementos acessíveis da página, é importante que isto possa ser feito utilizando uma primitiva nativa da linguagem. Usuários com outras deficiências e combinações tem necessidades diferentes. Pessoas com surdocegueira por exemplo, necessitam de um display braile para leitura do conteúdo da tela. Por não emitir conteúdo acessível, Telis não pode produzir qualquer conteúdo perceptível para um usuário com esta deficiência. Mesmo sem depender do conceito de componentes gráficos (widgets) acessíveis, seria importante durante o aprendizado buscar meios de se produzir saídas, especialmente texto, perceptíveis a usuários com diferentes tipos de deficiência.

Um passo além de produzir conteúdo perceptível para o usuário desenvolvendo com a linguagem (conforme tratado no parágrafo anterior), é também necessário um estudo sobre as formas de se produzir conteúdo acessível utilizando Telis, considerando os casos do usuário desenvolvendo ter ou não uma deficiência. A acessibilidade de aplicativos desktop ou Web é fornecida através do uso de componentes gráficos que podem ser reconhecidos por tecnologias assistivas para que se possa fornecer uma alternativa de uso. Telis não possui nativamente o conceito de componentes gráficos (conhecidos como Widgets) que possuem uma forma de uso comum, e portanto não há componentes a que se pode atribuir semântica. Interfaces em Telis são construídas através de primitivas que permitem aos atores emitir saída gráfica (pintar, riscar, entre outros) em áreas específicas do aplique (dependendo de coordenadas), emitir sons, e tratar eventos de teclado e mouse. Embora possa ser uma alternativa válida no processo de aprendizagem, esta forma de produzir conteúdo depende de coordenadas para posicionar elementos na tela (cegos não poderão produzir elementos visuais) e não possui semântica. Cabe informar que o laboratório Edugraf também

desenvolveu uma tecnologia chamada Dix que permite o desenvolvimento de interfaces gráficas Web utilizando componentes nativos da especificação XHTML e é compatível com Telis. Porém, foi verificado neste trabalho que as páginas Dix não são acessíveis, não sendo possível fornecer semântica correta a elementos da página ou alternativa textual para todos os elementos, além de depender de coordenadas para posicionar elementos e não fornecer navegação adequada via teclado.

Durante o processo foram observadas diversas falhas no manual. Conforme dito anteriormente, é uma contradição com o objetivo de ser um software que permite aprender fazendo e também é uma falha de usabilidade, portanto, a revisão da documentação é um importante trabalho futuro.

Outra possibilidade de estudo é relacionada a formas de representação espacial para usuários cegos com Telis. Telis possui o conceito de atores, com representação visual e posição na tela, e manipulado através de comandos da linguagem. É o mesmo conceito da tartaruga gráfica da linguagem Logo, na qual Seymour Papert sugere usos diversos na educação, como apresentar conceitos de geometria e física. Assim, o estudo de formas alternativas do uso de uma linguagem de programação educacional para apresentar conceitos que dependem de representação espacial no caso de aprendizes cegos é mais uma indicação para trabalhos futuros.

5. Conclusão

Por fim, apesar de ainda ser necessário outros estudos relacionados a acessibilidade em ambiente Telis, o objetivo de fornecer um ambiente Telis acessível para o caso de uso de um usuário cego foi atingido, sendo que o usuário foi capaz de navegar sozinho pela interface e acessar diferentes componentes da interface gráfica sem auxílio externo, e sem dúvida o Telis se mostrou como ferramenta viável no momento inicial de aprendizagem de programação de um usuário cego, e este resultado abre caminho para tornar viável novas pesquisas como os trabalhos futuros indicados. No entanto, com o progresso do usuário, as limitações da linguagem devem tornar-se latentes.

Também é um importante resultado o fato de não ter sido utilizado software proprietário em nenhum momento no processo, e tanto o lado do cliente como servidor não

dependem de software proprietário, sendo coerente com o fato de Telis ser software livre e feito para produzir software livre.

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 9241: Requisitos Ergonômicos para Trabalho de Escritórios com Computadores: Parte 11 – Orientações sobre Usabilidade. Rio de Janeiro, 2002.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 9126: Engenharia de software - Qualidade de produto . Rio de Janeiro, 2003.

DELPIZZO, Graziela Naspolini; GHISI, Marcilene Aparecida Alberton; SILVA, Solange Cristina da. A tecnologia promovendo a inclusão de pessoas cegas no ensino superior a distância. In: CONGRESSO INTERNACIONAL DE EDUCAÇÃO A DISTÂNCIA, 12., 2005, Florianópolis. Disponível em: <<http://www.abed.org.br/congresso2005/por/pdf/038tcc3.pdf>>. Acesso em: 11 dez. 2011.

NIELSEN, J. (2011) “About Jakob Nielsen”, <http://www.useit.com/jakob/>

KAMALUDIN, Khairuddin; YATIM, Noor Faezah Mohd; NORDIN, Md. Jan. Refining technical and learning accessibility elements on e-learning for user with visual impairment. In: INFORMATION TECHNOLOGY (ITSIM), 2010 INTERNATIONAL SYMPOSIUM IN, 2., 2010, Kuala Lumpur. Information Technology (ITSim), 2010 International Symposium in. Kuala Lumpur: Ieee, 2010. p. 1 - 4. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5561331>>. Acesso em: 24 ago. 2011.

PAPERT, Seymour. The Children’s Machine. New York: Basic Books, 1993.

WEBAIM. Keyboard Accessibility. Disponível em: <<http://webaim.org/techniques/keyboard/>>. Acesso em: 11 dez. 2011.

APÊNDICE B – Código-fonte da Applet Java (Somente partes novas)

```
package web;

import java.applet.Applet;
import java.awt.BorderLayout;
import java.awt.Image;
import java.awt.Label;
import java.awt.MediaTracker;
import java.awt.Toolkit;
import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;

import javax.swing.JApplet;
import portablebeans.AmbienteBean;
import portablebeans.ApliqueBean;
import portablebeans.ModeloBean;
import portablebeans.Serializador;

import web.persistencia.ConversorParaBeans;
import web.persistencia.GerenteDePersistencia;
import web.publicacao.AssistentePublicacao;
import web.tratadores.TratadorAtores;
import web.tratadores.TratadorDicionarioDeModelos;
import web.utils.JsUtils;
import web.utils.Utils;

import netscape.javascript.JSObject;

import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.Aplique;
import br.ufsc.edugraf.telis.ambiente.ContextoDeEdicaoAMB;
import br.ufsc.edugraf.telis.ambiente.impl.edicao.EditorDeModelos;
import br.ufsc.edugraf.telis.ambiente.impl.edicao.ExcecaoDeMoldeInexistente;
import br.ufsc.edugraf.telis.maquina.ContextoDeEdicao;
```

```

import br.ufsc.edugraf.telis.maquina.ContextoDeExecucao;
import br.ufsc.edugraf.telis.maquina.ExcecaoEmTelis;
import br.ufsc.edugraf.telis.maquina.Modelo;
import br.ufsc.edugraf.telis.maquina.bidimensional.ContextoDeExecucao2d;
import br.ufsc.edugraf.telis.maquina.bidimensional.ImagemAWT;
import br.ufsc.edugraf.telis.maquina.linguagem.ContextoDeExecucaoTelis;
import br.ufsc.edugraf.telis.maquina.palavras.Som;

```

```

public class Principal extends JApplet {

```

```

    /**

```

```

     *

```

```

    */

```

```

    private static final long serialVersionUID = 2020000567372335796L;

```

```

    private Ambiente ambiente = null;

```

```

    private boolean estaParado = false;

```

```

    ContextoDeEdicao contextoDeEdicao;

```

```

    ContextoDeExecucao contextoDeExecucao;

```

```

    TratadorAtores tratadorAtores = null;

```

```

    private static Applet applet = null;

```

```

    public static Applet obterApplet() {

```

```

        return applet;

```

```

    }

```

```

    public Principal() {

```

```

        super();

```

```

    }

```

```

    public void start() {

```

```

        iniciar();

```

```

        JSObject win = JsUtils.getWindow(this);

```

```

        JsUtils.call(win, "appletStart");

```

```

        if(JsUtils.obterAppUrl()=="")

```



```

((AmbienteApplet)ambiente).fixarAppUrl("http://localhost:8084/TelisWeb");//
para debug
    applet = this;
}

public void init() {
    setLayout(new BorderLayout());
    ambiente = new AmbienteApplet(this);
    contextoDeEdicao = ambiente.obterContextoDeEdicao();
    contextoDeExecucao = ambiente.obterContextoDeExecucao();
    tratadorAtores = new TratadorAtores(this, contextoDeExecucao);
    TratadorDicionarioDeModelos tratadorModelos = new
TratadorDicionarioDeModelos(this, contextoDeEdicao);
    ambiente.obterEditorDeModelos().addObserver(tratadorModelos);
    JsUtils.fixarWin(this);
    applet = this;
    JsUtils.call("appletIniciou", contextoDeEdicao,
contextoDeExecucao, ambiente, ambiente.obterEditorDeModelos(),
ambiente.obterEditorDeApliques());
}

public void iniciar() {
    try {
        boolean reiniciar = true;
        try {
            if (estaParado) {
                ((ContextoDeExecucaoTelis)
contextoDeExecucao).continuarGeral();
                reiniciar = false;
            }
        } catch (Throwable e) {
            e.printStackTrace();
        }
        if (reiniciar) {
            ((ContextoDeExecucao2d)
contextoDeExecucao).iniciarTarefa();

```

```

        ((ContextoDeExecucao2d)
contextoDeExecucao).reiniciar();
    }
    } catch (Exception e) {
        e.printStackTrace();
    }
    validate();
    ((ContextoDeExecucao2d)
contextoDeExecucao).obterMundo().validate();
    estaParado = false;
}

public void stop() {
    try {
        ((ContextoDeExecucaoTelis)
contextoDeExecucao).pararGeral();
        estaParado = true;
    } catch (ExcecaoEmTelis ex) {
        System.out.println(ex);
    }
}

//Aplique:
public void selecionarAtor(Integer i) {
    tratadorAtores.selecionarAtor(i);
}

public void carregarSom(String filename, String palavra) {
    try {
        Som som = new Som(palavra, new URL(filename));
        contextoDeEdicao.incluirSom(som);
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (ExcecaoEmTelis e) {
        e.printStackTrace();
    }
}

```

```

    }
}

public void carregarImagem(String filename, String palavra) {
    try{
        Image image = null;
        try {
            image = Toolkit.getDefaultToolkit().createImage(new
URL(filename));
        } catch (MalformedURLException e1) {
            e1.printStackTrace();
        }

        MediaTracker mt = new MediaTracker(new Label());
        mt.addImage(image, 0);
        try {
            mt.waitForAll();
        } catch (InterruptedException ee) {
            ee.printStackTrace();
        }

        byte[] data = Utils.imageToByte(image);

        //getExtensao
        int posicaoPonto = filename.lastIndexOf('.');
        String extensao = (posicaoPonto < 0) ? "" :
filename.substring(posicaoPonto);

        ImagemAWT imagem = new ImagemAWT(palavra, filename, image,
data, extensao);
        try {
            contextoDeEdicao.incluirImagem(imagem);
        } catch (ExcecaoEmTelis e) {
            e.printStackTrace();
        }
    }catch(Exception e) {e.printStackTrace();}
}

```

```

    public void removerImagem(String nome) {
        try {
            contextoDeEdicao.removerImagem(nome);
        } catch (ExcecaoEmTelis e) {
            e.printStackTrace();
        }
    }

    public void removerSom(String nome) {
        try {
            contextoDeEdicao.removerSom(nome);
        } catch (ExcecaoEmTelis e) {
            e.printStackTrace();
        }
    }

    public AssistentePublicacao getPublicador() {
        return new AssistentePublicacao(ambiente,
JsUtils.getWindow(this));
    }

    public void salvarAgenda(final String agendaCod, final String moldes,
final String agendaLocal) {
        new Thread() {
            public void run() {
                EditorDeModelos editorDeModelos =
ambiente.obterEditorDeModelos();
                try {
                    editorDeModelos.salvarAgenda(agendaCod);
                } catch (ExcecaoEmTelis e) {
                    e.printStackTrace();
                }
            }
        } try {
            editorDeModelos.fixarListaDeMoldes(moldes);
            JsUtils.call("dialogoDeAlert", "Agenda salva
com sucesso", "Agenda salva");
        } catch (ExcecaoDeMoldeInexistente e) {
            e.printStackTrace();
        }
    }

```

```

    ambiente.obterGerenteDeInterface().mostrarMensagemDeErro(null, e);
    } catch (ExcecaoEmTelis e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    ambiente.obterGerenteDeInterface().mostrarMensagemDeErro(null, e);
    }

    editorDeModelos.fixarNomeDaAgendaLocalAtual(agendaLocal);
    JsUtils.call("persisteModelo");

    }
    }.start();
}

public void removerAplique() {
    new Thread() {
        public void run() {
            try {
                ambiente.obterEditorDeApliques().remover();
            } catch (ExcecaoEmTelis e) {

                ambiente.obterGerenteDeInterface().mostrarMensagemDeErro(null, e);
                e.printStackTrace();
            }
        }
    }.start();
}

public void importarAplique() {
    new Thread() {
        public void run() {
            try {
                ambiente.importarApliqueWeb();
            } catch (ExcecaoEmTelis e) {

                ambiente.obterGerenteDeInterface().mostrarMensagemDeErro(null, e);
            }
        }
    }.start();
}

```

```

        e.printStackTrace();
    }
}

}.start();
}

public void carregarAmbienteSeguro(String ambienteStr) {
    try {
        carregarAmbiente(ambienteStr);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public void carregarAmbiente(String ambienteStr) throws IOException,
ClassNotFoundException {
    if(ambienteStr==null)
        return;
    if(ambienteStr.equals(""))
        return;
    AmbienteBean ambienteBean = (AmbienteBean)
    Serializador.deserializar(ambienteStr);
    GerenteDePersistencia persistencia = new
    GerenteDePersistencia(ambiente);
    try {
        persistencia.carregarAmbiente(ambienteBean);
    } catch (ExcecaoEmTelis e) {
        e.printStackTrace();
    }
}

public String serializaApliqueAtual() {
    return
    serializaApliqueDeNome(ambiente.obterEditorDeApliques().obterNomeDoApliqueA
    tual());
}

```

```

    public String serializaApliqueDeNome(String nomeAplique) {
        ContextoDeEdicaoAMB contexto =
(ContextoDeEdicaoAMB)contextoDeEdicao;
        Aplique aplique = contexto.obterAplique(nomeAplique);
        ApliqueBean apliqueBean =
ConversorParaBeans.aplique2bean(aplique);
        try {
            String serial = Serializador.serializar(apliqueBean);
            return URLEncoder.encode("dado", "UTF-8") + "=" +
URLEncoder.encode(serial, "UTF-8");
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }
    }

    public String serializaModeloAtual() {
        try {
            System.out.println("Serializando...");
            String nomeModelo =
ambiente.obterEditorDeModelos().obterNomeDoModeloAtual();
            System.out.println("Serializando modelo "+ nomeModelo);
            return serializaModeloDeNome(nomeModelo);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    public String serializaModeloDeNome(String nome) {
        Modelo modelo =
ambiente.obterContextoDeEdicao().obterModelo(nome);
        System.out.println("Modelo encontrado");
        return serializaModelo(modelo);
    }

    public String serializaModelo(Modelo modeloAtual) {

```

```
        ModeloBean modeloAtualBean =
ConversorParaBeans.modelo2Bean(modeloAtual);
        System.out.println("Modelo " + modeloAtualBean.getNome() + "
convertido para bean");
        try {
            String serial = Serializador.serializar(modeloAtualBean);
            return URLEncoder.encode("dato", "UTF-8") + "=" +
URLEncoder.encode(serial, "UTF-8");
        } catch (IOException e) {
            e.printStackTrace();
            return null;
        }
    }
}
```



```
package web;

import java.applet.Applet;
import java.io.FileNotFoundException;
import java.io.UnsupportedEncodingException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.URLEncoder;

import web.utils.JsUtils;

import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.Configuracoes;
import br.ufsc.edugraf.telis.ambiente.ConfiguracoesDoAmbiente;
import br.ufsc.edugraf.telis.ambiente.ContextoDeEdicaoAMB;
import br.ufsc.edugraf.telis.ambiente.ContextoDeExecucaoAMB;
import br.ufsc.edugraf.telis.ambiente.GerenteDeInterface;
import br.ufsc.edugraf.telis.ambiente.IEditorDeApliques;
import br.ufsc.edugraf.telis.ambiente.IEditorDeImagens;
import br.ufsc.edugraf.telis.ambiente.IEditorDeSons;
import br.ufsc.edugraf.telis.ambiente.IVisualizador;
import br.ufsc.edugraf.telis.ambiente.impl.ConfiguracoesImpl;
import br.ufsc.edugraf.telis.ambiente.impl.Importador;
import br.ufsc.edugraf.telis.ambiente.impl.edicao.EditorDeApliques;
import br.ufsc.edugraf.telis.ambiente.impl.edicao.EditorDeModelos;
import br.ufsc.edugraf.telis.maquina.ExcecaoEmTelis;
import br.ufsc.edugraf.telis.maquina.palavras.Giria;

public class AmbienteApplet extends Ambiente {

    Configuracoes configuracoes = new ConfiguracoesImpl();
    ContextoDeEdicaoAMB contextoDeEdicao = null;
    ContextoDeExecucaoAMB contextoDeExecucao = null;
    EditorDeModelos editorDeModelos = null;
    GerenteDeInterface gInterface = null;
    IEditorDeApliques editorDeApliques = null;

    Giria ultimaGiria = null;
```

```

    private String appUrl = "";

    public AmbienteApplet(Applet applet) {
        super();
        contextoDeEdicao = new ContextoDeEdicaoAMB(this);
        contextoDeExecucao = new
ContextoDeExecucaoWeb(applet.getWidth(), applet.getHeight(), this, contextoDeE
dicao);

        applet.add("Center", contextoDeExecucao.obterMundo());
        editorDeModelos = new EditorDeModelos(this);
        editorDeApliques = new EditorDeApliques(this);
        gInterface = criarGerenteDeInterface(applet);
        ((EditorDeApliques) editorDeApliques).fixarMediadorUI(new
MediadorApliquesApplet(this, JsUtils.getWindow(applet)));
        configuracoes.fixerPropriedade("publicar.diretorio", "/");
    }

    private GerenteDeInterface criarGerenteDeInterface(Applet
applet) {

        return new GerenteDeInterfaceApplet(this,
JsUtils.getWindow(applet));
    }

    @Override
    public void fixarGerenteDeInterface(GerenteDeInterface ger) {
        gInterface = ger;
    }

    @Override
    public GerenteDeInterface obterGerenteDeInterface() {
        return gInterface;
    }

    @Override
    public void continuar() throws ExcecaoEmTelis {

    }

```

```

@Override
    public void executarNoAtorDidatico(String texto) {
        try {
            if ((ultimaGiria == null) || (!
texto.equals(ultimaGiria.obterTexto())) {
                Giria pultimaGiria = new Giria(texto, "");
                contextoDeEdicao.compilarGiria(pultimaGiria);
                ultimaGiria = pultimaGiria;
            }
            contextoDeExecucao.executarGiria(ultimaGiria);
        }
        catch (ExcecaoEmTelis ex) {
            String errorMsg = "Erro ao executar texto: " +
texto;// + "\n" + ex.getMessage();
            gInterface.mostrarMensagem(null, errorMsg,
errorMsg);
        }
    }

@Override
    public void exportar() throws ExcecaoEmTelis {
        // TODO Auto-generated method stub
    }

@Override
    public void fechar() throws ExcecaoEmTelis {
    }

@Override
    public void trocar() throws ExcecaoEmTelis {
    }

@Override
    public void importar() throws ExcecaoEmTelis {
    }

```

```
}

@Override
public void importarAplique() throws ExcecaoEmTelis {

}

@Override
public void novo() throws ExcecaoEmTelis {

}

@Override
public Configuracoes obterConfiguracoes() {
    return configuracoes;
}

@Override
public ConfiguracoesDoAmbiente obterConfiguracoesDoAmbiente() {
    return null;
}

@Override
public ContextoDeEdicaoAMB obterContextoDeEdicao() {
    return contextoDeEdicao;
}

@Override
public ContextoDeExecucaoAMB obterContextoDeExecucao() {
    return contextoDeExecucao;
}

@Override
public IEditorDeApliques obterEditorDeApliques() {
    return editorDeApliques;
}

@Override
```

```
public EditorDeModelos obterEditorDeModelos() {  
    return editorDeModelos;  
}  
  
@Override  
public IEditorDeImagens obterEditorDeImagens() {  
    return null;  
}  
  
@Override  
public IEditorDeSons obterEditorDeSons() {  
    return null;  
}  
  
@Override  
public IVisualizador obterVizualizadorDePrimitivas() {  
    return null;  
}  
  
@Override  
public void parar() throws ExcecaoEmTelis {  
  
}  
  
@Override  
public void reiniciar() throws ExcecaoEmTelis {  
  
}  
  
@Override  
public void salvar() throws ExcecaoEmTelis {  
  
}  
  
@Override  
public void senha() throws ExcecaoEmTelis {  
  
}
```

```

@Override
    public void terminar() throws ExcecaoEmTelis {

    }

    public void fixarAppUrl(String url) {
        appUrl = url;
        JsUtils.fixarAppUrl(url);
    }

    @Override
    public void importarApliqueWeb() throws ExcecaoEmTelis {
        final String uri =
gInterface.mostrarDialogoDePergunta(null, "URL", "Digite a url do
aplique:", "http://telis.edugraf.ufsc.br/apliques/");
        String localUri = uri;
        try {
            localUri = appUrl + "/ExternoServlet?url=" +
URLLEncoder.encode(uri, "UTF-8");
        } catch (UnsupportedEncodingException e1) {
            e1.printStackTrace();
        }
        final Importador importador = new Importador(this, null);
        try {
            importador.importarAplique(new URI(localUri));
        } catch (URISyntaxException e) {
            gInterface.mostrarMensagemDeErro(null, e);
            e.printStackTrace();
        } catch (FileNotFoundException e) {
            gInterface.mostrarMensagemDeErro(null, e);
            e.printStackTrace();
        }
    }
}

```

```
package web;

import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.ContextoDeExecucaoAMB;
import br.ufsc.edugraf.telis.maquina.ContextoDeEdicao;
import br.ufsc.edugraf.telis.maquina.ambiente.script.AdaptadorJavaScript;

public class ContextoDeExecucaoWeb extends ContextoDeExecucaoAMB {

    public ContextoDeExecucaoWeb(int largMundo, int altMundo,
        Ambiente ambiente, ContextoDeEdicao ce) {
        super(largMundo, altMundo, ambiente, ce);
    }

    public String executarJS(String comando) {
        AdaptadorJavaScript adaptador =
        AdaptadorJavaScript.obterInstancia(Principal.obterApplet());
        String result = adaptador.executar(comando);
        if(result != null)
            result = result.replace("\n", "\\n");
        return result;
    }
}
```

```

package web;

import java.awt.Frame;
import java.io.IOException;
import java.util.Properties;

import javax.swing.JOptionPane;

import web.utils JsUtils;

import netscape.javascript.JSObject;

import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.Configuracoes;
import br.ufsc.edugraf.telis.ambiente.ContextoDeExecucaoAplique;
import br.ufsc.edugraf.telis.ambiente.GerenteDeInterface;
import br.ufsc.edugraf.telis.ambiente.igrafica.JanelaDeAplique;
import br.ufsc.edugraf.telis.maquina.Escopo;
import br.ufsc.edugraf.telis.maquina.palavras.Lista;

//TODO: TIRAR Impl
public class GerenteDeInterfaceApplet extends GerenteDeInterface {

    JSObject win;

    public GerenteDeInterfaceApplet(Ambiente _ambiente, JSObject win) {
        super(_ambiente);
        this.win = win;
    }

    @Override
    /**
     * Feito para ser usado TEMPORARIAMENTE pelo editor de modelos no
ambiente web.
     * Não pode ser usado em outro lugar
     */
    public int mostrarMensagem(Frame janela, String e, String m, String
b1, String b2) {
        return mostrarMensagem(janela, e,m,b1,b2,"");
    }

```



```

    }

    @Override
    public int mostrarMensagem(Frame janela, String e, String m, String
b1, String b2, String b3) {
        JsUtils.call("dialogoDeMensagem", m,e,b1,b2,b3);
        Object resposta = aguardarResposta();
        if(resposta instanceof Integer)
            return (Integer) resposta;
        return (int)(double)(Double) resposta; //TODO: Estava vindo
double em alguns casos, ver certinho como é esta conversão do javascript e
porque não vem sempre integer

    }

    @Override
    public void mostrarMensagem(Frame janela, String e, String m) {
        mostrarMensagemAlert(m, "aviso");
    }

    @Override
    public void mostrarMensagemDeErro(Frame janela, Throwable ex) {
        ex.printStackTrace();
        mostrarMensagemAlert(ex.toString() + '\n' + ex.getMessage(),
"aviso");
    }

    public void mostrarNovaJanelaDeAplique(
        ContextoDeExecucaoAplique novoContexto) {
        win.eval("alert('O aplique iniciará em nova janela');");
        JanelaDeAplique ja = new JanelaDeAplique(obterAmbiente(),
novoContexto);
        ja.show();
    }

    @Override
    public Properties mostrarDialogoDePublicar(String nomeDoAplique,

```

```

        String projeto, String caminho, String usuário, char[]
senha)

        throws IOException {
        final Configuracoes prop =
obterAmbiente().obterConfiguracoes();

        String raiz = prop.obterPropriedadeSD("publicar.raiz",
"/foruns/Telis/Apliques");
        String diretorio =
prop.obterPropriedadeSD("publicar.diretorio", raiz);
        String modo = prop.obterPropriedadeSD("publicar.tecnica",
"FTP");
        String maquina =
br.ufsc.edugraf.telis.maquina.Versao.obterVersao();

        Object[] args = {nomeDoAplique, projeto, caminho, usuário,
senha};
        JSObject retorno = (JSObject) win.call("factoryDialogPublicar",
args);

        nomeDoAplique = (String) retorno.getMember("aplique");
        projeto = (String) retorno.getMember("projeto");
        diretorio = (String) retorno.getMember("caminho");
        usuário = (String) retorno.getMember("usuario");
        String strSenha = (String) retorno.getMember("senha");
        senha = strSenha.toCharArray();

        Properties p = new Properties();
        p.put("usuário", usuário);
        p.put("senha", senha);
        p.put("diretorio", diretorio);
        p.put("arquivo", projeto);
        p.put("descricao", nomeDoAplique);
        p.put("maquina", maquina);
        p.put("modo", modo);

        return p;
}

```

```

@Override
    public String mostrarDialogoDePergunta(Frame janela, String título,
String mensagem, String valorPadrão) {
        JsUtils.call("dialogoDePergunta", título, mensagem,
valorPadrão);
        return (String) aguardarResposta();
    }

@Override
    public Lista mostrarDialogoDeErroDeExecucao(Exception erro, Escopo
escopo, Lista lista, int ip) {
        erro.printStackTrace();
        mostrarMensagemAlert("Erro de execução: " + erro.getMessage() +
"\ntentando executar:" + lista.toString(), "Erro de execução");
        return null;
    }

    private void mostrarMensagemAlert(String msg, String titulo) {

        if(win!=null) {
            JsUtils.call("dialogoDeAlert",msg,titulo);
            aguardarResposta();
        } else {
            JOptionPane.showMessageDialog(null, msg);
        }
    }

    private Object aguardarResposta() {
        boolean respondeu = false;
        while(!respondeu) {
            respondeu = (Boolean) JsUtils.call("jaRespondeu");
        }
        return JsUtils.call("obterResposta");
    }
}

```

```

package web;

import java.util.Observable;
import java.util.Observer;
import java.util.Vector;

import web.utils.JsUtils;

import netscape.javascript.JSObject;
import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.Aplique;
import br.ufsc.edugraf.telis.ambiente.IEditorDeApliquesUI;
import
br.ufsc.edugraf.telis.ambiente.igrafica.editores.IMediadorApliquesUI;
import br.ufsc.edugraf.telis.maquina.Modelo;
import br.ufsc.edugraf.telis.util.estruturasDeDados.Dicionario;

public class MediadorApliquesApplet implements IMediadorApliquesUI {

    private JSObject win;
    private final String jsNome =
"document.getElementById('textNomeAplique')";
    private final String jsLargura =
"document.getElementById('textLargura')";
    private final String jsAltura =
"document.getElementById('textAltura')";
    private final String jsApliques =
"document.getElementById('sApliques')";
    private final String jsAtores =
"document.getElementById('sModelosInstanciados')";//lista de modelos
instanciados (a da direita)
    private final String jsModDisp =
"document.getElementById('sModelosDisponiveis')";
    private Object[] vazio = new Object[0];//para chamar funções sem
argumento, sem precisar dar new toda hora
    public Dicionario atores = null;//substitui
editorUi.obterAtoresDoAplique//TODO PRIVATE

```

```

    private Dicionario apliques; //substitui
editorUI.obterDispDicionario()

    private Ambiente ambiente;

    private int atorSelecionado = -1;
    private int apliqueSelecionado = -1;

    public MediatorApliquesApplet(Ambiente ambiente, JSObject win) {
        this.win = win;
        this.ambiente = ambiente;
        apliques = ambiente.obterContextoDeEdicao().obterApliques();
        apliques.addObserver(new ObserverAplique());

ambiente.obterContextoDeEdicao().obterDicionarioDeModelos().addObserver(new
ObserverModelosDisponiveis());
    }

    private class ObserverAplique implements Observer{
        @Override
        public void update(Observable arg0, Object arg1) {
            atualizarListaDeApliques();
            if(apliqueSelecionado == -1 && apliques.tamanho() > 0)
                selecionarAplique(0);
        }
    }

    private class ObserverModelosDisponiveis implements Observer {
        @Override
        public void update(Observable o, Object arg) {
            atualizaModelosDisponiveis();
        }
    }

    private void atualizaModelosDisponiveis() {
        JsUtils.call("limpaSModelosDisponiveis");
        Dicionario modelos =
ambiente.obterContextoDeEdicao().obterDicionarioDeModelos();

```

```

        for(int i = 0; i < modelos.tamanho(); i++) {
            Modelo modelo = (Modelo) modelos.obterElemento(i);
            if(!modelo.éMolde())
                if(win!=null)
                    win.eval(jsModDisp + ".add(new Option('" +
modelo.obterNome() + "', '" + String.valueOf(i) + "'))");
        }
        JsUtils.call(win, "invalidateSModelosDisponiveis");
    }

    @Override
    /**
     * Não faz nada em web, se comunica direto com página, sem editorUI
     */
    public void fixarUI(IEditorDeApliquesUI editorUI) {
    }

    /**
     * Seta a propriedade valor de um objeto javascript como uma string
passada em valor
     * @param elementoJs
     * @param valor
     */
    private void setarValor(String elementoJs, String valor) {
        if(win!=null)
            win.eval(elementoJs + ".value = '" + valor + "'");
    }

    private String obterValor(String elementoJs) {
        if(win!=null)
            return (String) win.eval(elementoJs + ".value");
        return "";
    }

    @Override
    public void fixarNome(String nome) {
        setarValor(jsNome, nome);
    }

```

```

@Override
public String obterNome() {
    return obterValor(jsNome);
}

@Override
public void fixarLargura(String largura) {
    setarValor(jsLargura, largura);
}

@Override
public String obterLargura() {
    return obterValor(jsLargura);
}

@Override
public void fixarAltura(String altura) {
    setarValor(jsAltura, altura);
}

@Override
public String obterAltura() {
    return obterValor(jsAltura);
}

@Override
public void fixarDicionarioDeAtores(Dicionario dicionario) {
    if(atores!=null)
        atores.deleteObserver(observerAtores);
    atores = dicionario;
    atores.addObserver(observerAtores);
}

private ObserverAtores observerAtores = new ObserverAtores();
private class ObserverAtores implements Observer {

    @Override

```

```

        public void update(Observable arg0, Object arg1) {
            atualizarAtoresDoAplique();
        }

    }

    @Override
    /**
     * Força view a atualizar lista
     */
    public void atualizarAtoresDoAplique() {
        //TESTE
        Vector vetor = atores.obterObjetos();
        atualizarListaNaInterface(jsAtores, vetor);
        win.call("invalidateSModelosInstanciados",vazio);
    }

    public void atualizarListaNaInterface(String jsElement, Vector vetor)
    {

        if(win == null)
            return;

        win.eval(jsElement + ".length = 0;");
        for(int i = 0; i < vetor.size(); i++) {
            win.eval(jsElement + ".add(new Option('" +
vetor.get(i).toString() + "',' + String.valueOf(i) + '));");
        }
    }

    @Override
    public int obterTamanhoDoDicionario() {
        return apliques.tamanho();
    }

    @Override
    public String obterNomeDoApliqueSelecionado() {
        if(apliques.obterObjetos().size() == 0)
            return null;
    }

```



```

        if(apliqueSelecionado < 0 || aplicaselecionado >=
apliques.tamanho())
        {
            return null;
        }
        Aplique aplicaselecionado =
(Aplique)apliques.obterElemento(aplicaselecionado);
        return aplicaselecionado.obterNome();
    }

    @Override
    public int obterIndiceDoApliqueSelecionado() {
        return aplicaselecionado;
    }

    @Override
    public void selecionarAplique(String n) {
        @SuppressWarnings("unchecked")
        Vector<Object> vetor = aplicaselecionado.obterObjetos();
        for (int i = 0; i < vetor.size(); i++) {
            if (vetor.get(i).toString().equals(n)) {
                selecionarAplique(i);
                return;
            }
        }
    }

    private void selecionarEmSelect(String jsSelect, int i) {
        if(win!=null)
            win.eval(jsSelect + ".value=" + String.valueOf(i) +
        "");
    }

    private void onchange(String jsSelect) {
        if(win!=null)
            win.eval(jsSelect + ".onchange();");
    }

    @Override

```

```

    public void selecionarAplique(int i) {
        apliqueSelecionado = i;
        selecionarEmSelect(jsApliques,i);
        onchange(jsApliques);
    }

    public void fixarSelecionado(int i) {
        apliqueSelecionado = i;
    }

    @Override
    public boolean mostrarDialogoDePublicaçãoLocal(String
diretorioDePublicacao) {
        return
ambiente.obterGerenteDeInterface().mostrarDialogoDePublicaçãoLocal(diretori
oDePublicacao, null);
    }

    @Override
    public int obterIndiceDoAtorSelecionado() {
        return atorSelecionado;
    }

    @Override
    public int obterTamanhoDoDicionarioDeAtores() {
        return atores.tamanho();
    }

    @Override
    public void selecionarAtor(int i) {
        atorSelecionado = i;
    }

    @Override
    public void atualizarListaDeApliques() {
        Vector vetor = apliques.obterObjetos();
        atualizarListaNaInterface(jsApliques, vetor);
        JsUtils.call(win, "invalidateApliques");
    }

```

```
}

@Override
public void ApliqueSetValueIsAdjusting(boolean b) {
}

@Override
public void fixarAlterado(boolean alt) {
}

}
```

```

package web.persistencia;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import portablebeans.AgendaBean;
import portablebeans.ApliqueBean;
import portablebeans.ModeloBean;
import br.ufsc.edugraf.telis.ambiente.Aplique;
import br.ufsc.edugraf.telis.maquina.Agenda;
import br.ufsc.edugraf.telis.maquina.Modelo;

public class ConversorParaBeans {

    public static AgendaBean agenda2Bean(Agenda agenda) {
        AgendaBean agendaBean = new AgendaBean();
        agendaBean.setCodigo(agenda.obterTexto());
        agendaBean.setNome(agenda.obterNome());
        return agendaBean;
    }

    public static ModeloBean modelo2Bean(Modelo modelo) {
        ModeloBean modeloBean = new ModeloBean();
        modeloBean.setNome(modelo.obterNome());

        modeloBean.setMoldado(modelo.obterListaOrdenadaDeNomesDosMoldes());
        modeloBean.setEhMolde(modelo.éMolde());

        //get agendas bean
        Iterator<Agenda> it = modelo.obterIteradorDeAgendasLocais();
        while(it.hasNext())
            modeloBean.addAgenda(agenda2Bean(it.next()));
        return modeloBean;
    }

    public static ApliqueBean aplique2bean(Aplique aplique) {
        ApliqueBean bean = new ApliqueBean();
        bean.setNome(aplique.obterNome());
    }

```

```
bean.setAltura(Integer.valueOf(aplique.obterAltura()));
bean.setLargura(Integer.valueOf(aplique.obterLargura()));

List<String> modelos = new ArrayList<String>();
for(int i = 0; i < aplicue.atores.tamanho(); i++)
    modelos.add(aplique.atores.obterDescritor(i));

bean.setModelosStr(modelos);

return bean;
}
}
```

```
package web.persistencia;

import java.io.IOException;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

import portablebeans.AgendaBean;
import portablebeans.AmbienteBean;
import portablebeans.ApliqueBean;
import portablebeans.ModeloBean;
import portablebeans.Serializador;

import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.Aplique;
import br.ufsc.edugraf.telis.maquina.ContextoDeEdicao;
import br.ufsc.edugraf.telis.maquina.ContextoDeExecucao;
import br.ufsc.edugraf.telis.maquina.ExcecaoEmTelis;
import br.ufsc.edugraf.telis.ambiente.ContextoDeEdicaoAMB;

import web.utils.JsUtils;

public class GerenteDePersistencia {

    Ambiente ambiente;
    ContextoDeEdicao contextoDeEdicao;
    ContextoDeExecucao contextoDeExecucao;
    public GerenteDePersistencia(Ambiente ambiente) {
        this.ambiente = ambiente;
        this.contextoDeEdicao = ambiente.obterContextoDeEdicao();
        this.contextoDeExecucao = ambiente.obterContextoDeExecucao();
    }
}
```

```

    public void carregarAmbiente(AmbienteBean ambiente) throws
ExcecaoEmTelis {
        if(ambiente == null)
            return;
        System.out.println("\nCarregando modelos...\n");

        carregarModelos(Collections.synchronizedSet(ambiente.getModelos()));
        System.out.println("\nCarregando apliques...\n");
        carregarApliques(ambiente.getApliquesComoLista());

        JsUtils.eval("document.getElementById('sApliques').onChange()");
    }

    private void carregarApliques(List<ApliqueBean> listaDeApliques) throws
ExcecaoEmTelis {
        for(ApliqueBean apliqueBean: listaDeApliques) {
            Aplique aplique = new Aplique(apliqueBean.getNome(),
String.valueOf(apliqueBean.getLargura()),
String.valueOf(apliqueBean.getAltura()), contextoDeEdicao);

            ((ContextoDeEdicaoAMB)contextoDeEdicao).adicionarAplique(aplique);

            ((ContextoDeEdicaoAMB)contextoDeEdicao).obterApliques().notifyObservers();
            this.ambiente.obterEditorDeApliques().atualizar();
            aplique.adicionarAtores(apliqueBean.getModelosStr().toArray());
        }
    }

    private void carregarModelos(Set<ModeloBean> modelos) throws
ExcecaoEmTelis {
        if(modelos==null)
            return;
        Set<ModeloBean> importados = new HashSet<ModeloBean>();
        Set<ModeloBean> importadosNovos = new HashSet<ModeloBean>();

```

```

        int i = 0;
        while(importados.size() < modelos.size()) {
            System.out.println("Iteração: " + String.valueOf(++i));
            for(ModeloBean modeloBean: modelos) {
                System.out.println("    Importando: " +
                modeloBean.getNome());
                System.out.println("    com moldes: " +
                modeloBean.getMoldado());
                for(String moldeNome: modeloBean.getMoldado())
                    if(!setContemModeloDeNome(modelos,
                moldeNome)){
                        if(!
                setContemModeloDeNome(importadosNovos, moldeNome)) {
                            ModeloBean novo =
                criaModeloPadrao(moldeNome);

                ambiente.executarNoAtorDidatico(novo.toString());
                            importadosNovos.add(novo);
                        }
                    }
                if(moldesForamImportados(modeloBean, importados,
                importadosNovos) && naoFoiImportado(importados, importadosNovos,
                modeloBean)) {

                ambiente.executarNoAtorDidatico(modeloBean.toString());
                            importados.add(modeloBean);
                        }
                    }
            }
        }

        private boolean naoFoiImportado(Set<ModeloBean> importados,
        Set<ModeloBean> importadosNovos,
        ModeloBean modeloBean) {

```



```

        return !(importados.contains(modeloBean) ||
importadosNovos.contains(modeloBean));
    }

```

```

    private ModeloBean criaModeloPadrao(String moldeNome) {
        ModeloBean novo = new ModeloBean();
        AgendaBean agenda = new AgendaBean();
        agenda.setNome("iniciar");
        agenda.setCodigo("");
        novo.addAgenda(agenda);
        novo.setNome(moldeNome);
        novo.setListaMoldes(new ArrayList<ModeloBean>(0));
        novo.setEhMolde(true);
        novo.setMoldado(new ArrayList<String>(0));
        return novo;
    }

```

```

    private boolean setContemModeloDeNome(Set<ModeloBean> modelos,
        String moldeNome) {
        for(ModeloBean modelo: modelos)
            if(modelo.getNome().equals(moldeNome))
                return true;
        return false;
    }

```

```

    private boolean moldesForamImportados(ModeloBean modeloBean,
        Set<ModeloBean> importados, Set<ModeloBean>
importadosNovos) {
        if(modeloBean.getMoldado() == null)
            return true;
        if(modeloBean.getMoldado().size() == 0)
            return true;
    }

```

```

        Set<String> nomesDeModelosImportados = new
HashSet<String>(importados.size());
        for(ModeloBean modeloImportado: importados)
            nomesDeModelosImportados.add(modeloImportado.getNome());
        for(ModeloBean modeloImportado: importadosNovos)
            nomesDeModelosImportados.add(modeloImportado.getNome());

        String[] nomeMoldes = new
String[modeloBean.getMoldado().size()];
        modeloBean.getMoldado().toArray(nomeMoldes);
        int numDependenciaResolvidas = 0;
        for(String nomeModelo: nomeMoldes)
            if(nomesDeModelosImportados.contains(nomeModelo))
                numDependenciaResolvidas++;
        return numDependenciaResolvidas == nomeMoldes.length;
    }

    public void persistirAplique(Aplique aplique) throws IOException {
        ApliqueBean bean = ConversorParaBeans.aplique2bean(aplique);
        String apliqueSerializado = Serializador.serializar(bean);
        String parametro = URLEncoder.encode("aplique", "UTF-8") + "="
+ URLEncoder.encode(apliqueSerializado, "UTF-8");
        JsUtils.call("persistirApliqueSerializado", parametro);
    }
}

```

```

package web.publicacao;

import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;

import javax.xml.transform.Source;
import javax.xml.transform.stream.StreamSource;

import web.utils.JsUtils;
import netscape.javascript.JSObject;
import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.Aplique;
import br.ufsc.edugraf.telis.ambiente.impl.PublicadorDeApliques;

public class AssistentePublicacao {

    private FileManager fm;
    private Ambiente ambiente;

    public AssistentePublicacao(Ambiente ambiente, JSObject win) {
        fm = new FileManager();
        this.ambiente = ambiente;
    }

    public void publicar(Aplique aplique, String nomeDiretorio) {
        try {
            ContainerVirtualWeb diretorio = new
ContainerVirtualWeb(nomeDiretorio, fm);
            PublicadorDeApliques publicadorApliques = new
PublicadorDeApliquesWeb(ambiente,aplique,diretorio,null);
            publicadorApliques.publicarHTML();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```
protected Source obterStreamSource() {
    Source transformação = null;
    try {
        String CAMINHO_DA_TRANSFORMAÇÃO = JsUtils.obterAppUrl() +
        "/configuracoes/xmi.transformacao.xml";
        transformação = new StreamSource(new
        URL(CAMINHO_DA_TRANSFORMAÇÃO).openStream());
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    return transformação;
}
}
```

```

package web.publicacao;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.PrintWriter;

import br.ufsc.edugraf.telis.hiperForum.servidor.ContainerVirtual;
import
br.ufsc.edugraf.telis.hiperForum.servidor.FiltroDeListagemDeArquivos;
import br.ufsc.edugraf.telis.hiperForum.servidor.HFException;
import br.ufsc.edugraf.telis.hiperForum.servidor.Menu;
import br.ufsc.edugraf.telis.hiperForum.servidor.ObjetoVirtual;

/**
 * Esta classe representa um diretório no servidor do Telis Acessível e foi
criada para manter a compatibilidade
 * com o PublicadorDeAplicques existente
 * @author Felipe A. Ferreira
 *
 */
public class ContainerVirtualWeb implements ContainerVirtual, ObjetoVirtual
{

    String diretorio = "";
    ContainerVirtualWeb pai = null;
    FileManager filemanager = null;

    public FileManager getFileManager() {
        return filemanager;
    }

    @Override
    public String obterCaminho() {
        if(pai == null)
            return "/" + diretorio;
        return pai.obterCaminho() + "/" + diretorio;
    }

```

```

    }

    /**
     * este construtor é usado para o diretório raiz
     */
    public ContainerVirtualWeb(FileManager filemanager) {
        this.filemanager = filemanager;
    }

    public ContainerVirtualWeb(String diretorio, FileManager fm) {
        filemanager = fm;
        this.diretorio = diretorio;
        filemanager.criarDiretorio(diretorio);
    }

    public ContainerVirtualWeb(String diretorio, FileManager fm,
ContainerVirtualWeb pai) {
        this.diretorio = diretorio;
        this.pai = pai;
        filemanager = fm;
        filemanager.criarDiretorio(this.obterCaminho());
    }

    @Override
    public ObjetoVirtual criarSubdiretorio(String nome) throws
IOException {
        return new ContainerVirtualWeb(nome, filemanager, this);
    }

    @Override
    public ObjetoVirtual criarArquivo(String nome) throws IOException {
        return new ObjetoVirtualArquivo(nome, this, filemanager);
    }

    ///////////////////////////////////
    // Os métodos abaixo não são usados, apenas precisam ser criados por
causa da interface

```

```

@Override
public void descrever(String _descricao) throws IOException {
    // TODO Auto-generated method stub

}

@Override
public OutputStream escrever(long offset) throws IOException {
    // TODO Auto-generated method stub
    return null;
}

@Override
public void escrever(InputStream in, int mode, long offset)
    throws IOException {
    // TODO Auto-generated method stub

}

@Override
public void escrever(InputStream in, long offset) throws IOException
{
    // TODO Auto-generated method stub

}

@Override
public void fixarAtributo(String chave, String valor) throws
HFException {
    // TODO Auto-generated method stub

}

@Override
public void fixarInscritos(String d) throws HFException, IOException
{
    // TODO Auto-generated method stub

```

```

    }

    @Override
    public InputStream ler(long offset) throws IOException {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void ler(OutputStream out, int mode, long offset) throws
IOException {
        // TODO Auto-generated method stub

    }

    @Override
    public void ler(OutputStream out, long offset) throws IOException {
        // TODO Auto-generated method stub

    }

    @Override
    public String obterAtributo(String chave) throws HException {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public String obterDescricao() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public String obterInscritos() throws HException, IOException {
        // TODO Auto-generated method stub
        return null;
    }

```



```
}

@Override
public String obterMimeType() {
    // TODO Auto-generated method stub
    return null;
}

@Override
public String obterNome() {
    // TODO Auto-generated method stub
    return null;
}

@Override
public ContainerVirtual obterSuperior() throws IOException {
    // TODO Auto-generated method stub
    return null;
}

@Override
public boolean possoAdministrar() throws IOException {
    // TODO Auto-generated method stub
    return false;
}

@Override
public boolean possoEditar() throws IOException {
    // TODO Auto-generated method stub
    return false;
}

@Override
public boolean possoEscrever() throws IOException {
    // TODO Auto-generated method stub
    return false;
}
```

```
@Override
public boolean possivel() throws IOException {
    // TODO Auto-generated method stub
    return false;
}

@Override
public void remover() throws IOException {
    // TODO Auto-generated method stub

}

@Override
public void renomearPara(String dest) throws IOException {
    // TODO Auto-generated method stub

}

@Override
public long tamanho() throws IOException {
    // TODO Auto-generated method stub
    return 0;
}

@Override
public long ultimaAlteracao() throws IOException {
    // TODO Auto-generated method stub
    return 0;
}

@Override
public ObjetoVirtual criarObjeto(String nome, String tipo)
    throws IOException {
    // TODO Auto-generated method stub
    return null;
}

@Override
```

```

    public void listar(PrintWriter out, String patt, boolean verbose)
        throws IOException {
        // TODO Auto-generated method stub

    }

    @Override
    public ObjetoVirtual[] listar(FiltroDeListagemDeArquivos patt)
        throws IOException {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void mudarOrdem(String[] itens) throws HFException,
    IOException {
        // TODO Auto-generated method stub

    }

    @Override
    public Menu obterMenu() throws HFException, IOException {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public boolean possoListar() throws IOException {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public void acessar(String caminho) throws IOException {
        // TODO Auto-generated method stub

    }

```

```
@Override
    public void criarEstruturaDeDiretorios(String caminho) throws
IOException {
    // TODO Auto-generated method stub

}

@Override
    public String[] listarArquivosEPastas(String caminho) throws
IOException {
    // TODO Auto-generated method stub
    return null;
}

}
```

```

package web.publicacao;

import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;

import javax.xml.transform.stream.StreamSource;

import web.utils.JsUtils;
import br.ufsc.edugraf.telis.ambiente.impl.ConversorDeCodigo;

public class ConversorDeCodigoWeb extends ConversorDeCodigo {

    private static String CAMINHO_DA_TRANSFORMAÇÃO = "";

    public ConversorDeCodigoWeb(String códigoXml) {
        super(códigoXml);
        CAMINHO_DA_TRANSFORMAÇÃO = JsUtils.obterAppUrl() +
        "/configuracoes/xmlParaCodigoTelis.xsl";
    }

    @Override
    protected StreamSource getStreamSource() {
        StreamSource arquivoXslt = null;
        try {
            arquivoXslt = new StreamSource(new
URL(CAMINHO_DA_TRANSFORMAÇÃO).openStream());
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return arquivoXslt;
    }
}

```

```

package web.publicacao;

import java.io.IOException;
import java.net.MalformedURLException;
import java.net.URL;

import javax.xml.transform.Source;
import javax.xml.transform.stream.StreamSource;

import web.utils.JsUtils;
import br.ufsc.edugraf.telis.ambiente.impl.ConversorDeTelisParaXMI;

public class ConversorDeTelisParaXMIWeb extends ConversorDeTelisParaXMI {

    private static String CAMINHO_DA_TRANSFORMAÇÃO = "";

    public ConversorDeTelisParaXMIWeb(String _conteúdoDoAplique) {
        super(_conteúdoDoAplique);
        CAMINHO_DA_TRANSFORMAÇÃO = JsUtils.obterAppUrl() +
"/configuracoes/xmi.transformacao.xml";
    }

    @Override
    protected Source obterStreamSource() {
        Source transformação = null;
        try {
            transformação = new StreamSource(new
URL(CAMINHO_DA_TRANSFORMAÇÃO).openStream());
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return transformação;
    }
}

```

```

package web.publicacao;

import web.utils.JsUtils;

public class FileManager {

    public FileManager() {
    }

    public void criarDiretorio(String nome) {
        JsUtils.call("criarDiretorio", nome);
    }

    public void criarArquivo(String caminhoCompleto) {
        JsUtils.call("criarArquivo", caminhoCompleto);
    }

    /**
     * Este método escreve texto em um arquivo. Se arquivo já existir é
sobreescrito. Se não existir, cria o arquivo, podendo criar todos os
diretórios
     * necessários para isso caso o caminho esteja numa pasta que não
exista
     * @param caminho
     * @param conteúdo
     */
    public void escreverArquivo(String caminho, String conteudo) {
        JsUtils.call("escreverArquivo", caminho, conteudo);
    }

    public void copiarArquivo(String origem, String destino) {
        JsUtils.call("copiarArquivo", origem, destino);
    }
}

```

```
package web.publicacao;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import web.utils.Utills;

import br.ufsc.edugraf.telis.hiperForum.servidor.ContainerVirtual;
import br.ufsc.edugraf.telis.hiperForum.servidor.HFException;
import br.ufsc.edugraf.telis.hiperForum.servidor.ObjetoVirtual;

public class ObjetoVirtualArquivo implements ObjetoVirtual {

    private String nome;
    private ContainerVirtualWeb pai;
    private FileManager filemanager;

    public ObjetoVirtualArquivo(String nome, ContainerVirtualWeb pai,
FileManager filemanager) {
        this.nome = nome;
        this.pai = pai;
        this.filemanager = filemanager;
        filemanager.criarArquivo(this.obterCaminho());
    }

    @Override
    public String obterCaminho() {
        return pai.obterCaminho() + '/' + nome;
    }

    @Override
    public void escrever(InputStream in, int mode, long offset)
        throws IOException {
        escrever(in,offset);
    }
}
```



```

@Override
public void escrever(InputStream in, long offset) throws IOException
{
    String conteúdo = Utils.convertStreamToString(in);
    filemanager.escreverArquivo(obterCaminho(), conteúdo);
}

////////////////////////////////////
// Daqui para baixo não é usado, apenas está na interface

@Override
public void descrever(String _descricao) throws IOException {
    // TODO Auto-generated method stub
}

@Override
public void fixarAtributo(String chave, String valor) throws
HFException {
    // TODO Auto-generated method stub
}

@Override
public void fixarInscritos(String d) throws HFException, IOException
{
    // TODO Auto-generated method stub
}

@Override
public InputStream ler(long offset) throws IOException {
    // TODO Auto-generated method stub
    return null;
}

@Override

```

```

        public void ler(OutputStream out, int mode, long offset) throws
IOException {
            // TODO Auto-generated method stub

        }

        @Override
        public void ler(OutputStream out, long offset) throws IOException {
            // TODO Auto-generated method stub

        }

        @Override
        public String obterAtributo(String chave) throws HException {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public String obterDescricao() {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public String obterInscritos() throws HException, IOException {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public String obterMimeType() {
            // TODO Auto-generated method stub
            return null;
        }

        @Override
        public String obterNome() {

```

```
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public ContainerVirtual obterSuperior() throws IOException {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public boolean possoAdministrar() throws IOException {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean possoEditar() throws IOException {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean possoEscrever() throws IOException {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean possoLer() throws IOException {
        // TODO Auto-generated method stub
        return false;
    }

    @Override
    public boolean possoListar() throws IOException {
        // TODO Auto-generated method stub
        return false;
    }
}
```

```
}

@Override
public void remover() throws IOException {
    // TODO Auto-generated method stub

}

@Override
public void renomearPara(String dest) throws IOException {
    // TODO Auto-generated method stub

}

@Override
public long tamanho() throws IOException {
    // TODO Auto-generated method stub
    return 0;
}

@Override
public long ultimaAlteracao() throws IOException {
    // TODO Auto-generated method stub
    return 0;
}

@Override
public OutputStream escrever(long offset) throws IOException {
    // TODO Auto-generated method stub
    return null;
}

}
```

```

package web.publicacao;

import java.io.IOException;
import br.ufsc.edugraf.telis.ambiente.Ambiente;
import br.ufsc.edugraf.telis.ambiente.Aplique;
import br.ufsc.edugraf.telis.ambiente.Progresso;
import br.ufsc.edugraf.telis.ambiente.impl.ConversorDeCodigo;
import br.ufsc.edugraf.telis.ambiente.impl.ConversorDeTelisParaXMI;
import br.ufsc.edugraf.telis.ambiente.impl.PublicadorDeApliques;
import br.ufsc.edugraf.telis.hiperForum.servidor.ContainerVirtual;
import br.ufsc.edugraf.telis.hiperForum.servidor.ObjetoVirtual;
import br.ufsc.edugraf.telis.maquina.bidimensional.ImagemAWT;
import br.ufsc.edugraf.telis.maquina.palavras.Som;

public class PublicadorDeApliquesWeb extends PublicadorDeApliques {

    public PublicadorDeApliquesWeb(Ambiente ambiente, Aplique aplicacao,
        ContainerVirtual pasta, Progresso progressoUI)
        throws IOException {
        super(ambiente, aplicacao, pasta, false, progressoUI);
        // TODO Auto-generated constructor stub
    }

    @Override
    protected ConversorDeCodigo conversorFactory(String
conteúdoDoAplique) {
        ConversorDeCodigo conversor = new ConversorDeCodigoWeb
(conteúdoDoAplique);
        return conversor;
    }

    @Override
    protected ConversorDeTelisParaXMI obterConversorXMI(String
conteúdoDoAplique) {
        ConversorDeTelisParaXMI conversor = new
ConversorDeTelisParaXMIWeb(conteúdoDoAplique);
        return conversor;
    }
}

```

```

@Override
protected void copiarArquivoDeRecurso(String nomeDoArquivo) {

((ContainerVirtualWeb)pastaDeRecursos).getFileManager().copiarArquivo("conf
iguracoes/" + nomeDoArquivo, pastaDeRecursos.obterCaminho() + '/' +
nomeDoArquivo );
}

@Override
protected String obterVersaoDaMaquina() {
    return "telisacessivel.jar";
}

@Override
protected String obterDiretorioMaquinas() {
    return "/TelisWeb/maquinas/";
}

//Figuras e sons do ambiente já estão no servidor, por isso os
métodos escreverImagem e escreverSom foram sobrescritos, para que a
publicação não precise
//de novo escrever imagem e som, o que significaria um novo upload do
cliente para o servidor de cada imagem e som utilizado
//overhead desnecessário

@Override
protected void escreverImagem(ImagemAWT im, ObjetoVirtual oimagem)
    throws IOException {
    String origem = "icones/" + im.nome + im.obterExtensão();
    String destino = oimagem.obterCaminho();

((ContainerVirtualWeb)pastaDeRecursos).getFileManager().copiarArquivo(origem, destino);
}

@Override
protected void escreverSom(Som somAtual, ObjetoVirtual arquivoDeSom)

```

```
        throws IOException {  
            String origem = "sons/" + somAtual.obterNome() +  
somAtual.obterExtensão();  
            String destino = arquivoDeSom.obterCaminho();  
  
            ((ContainerVirtualWeb)pastaDeRecursos).getFileManager().copiarArquivo(origem,  
destino);  
        }  
    }  
}
```

```

package web.tratadores;

import java.applet.Applet;
import java.util.Observable;
import java.util.Observer;
import java.util.Vector;

import netscape.javascript.JSException;
import netscape.javascript.JSObject;
import br.ufsc.edugraf.telis.maquina.ContextoDeExecucao;
import br.ufsc.edugraf.telis.maquina.ExcecaoEmTelis;
import br.ufsc.edugraf.telis.maquina.Pilha;
import br.ufsc.edugraf.telis.maquina.bidimensional.Cursor2d;
import br.ufsc.edugraf.telis.maquina.palavras.Palavra;
import br.ufsc.edugraf.telis.util.estruturasDeDados.Dicionario;

public class TratadorAtores implements Observer{

    private Dicionario dicAtores = null;
    private Integer atorSelecionado = -1;
    private Applet applet;
    Pilha pilha;

    public TratadorAtores(Applet applet, ContextoDeExecucao
contextoDeExecucao) {
        this.applet = applet;
        dicAtores = contextoDeExecucao.obterAtores();
        dicAtores.addObserver(this);
    }

    @Override
    public void update(Observable o, Object arg) {
        Dicionario dicionario = (Dicionario) o;
        @SuppressWarnings("rawtypes")
        Vector objs = dicionario.obterObjetos();
        if(dicionario.tamanho() == 0)
            selecionarAtor(null);
        else if(atorSelecionado == null || pilha == null) {

```



```

        selecionarAtor(0);
    }
    try {
        JSObject win = JSObject.getWindow(applet);
        if(win!=null)

win.eval("document.getElementById('sAtores').options.length = 0");
        for(int i = 0; i < objs.size(); i++) {

win.eval("document.getElementById('sAtores').options.add( new
Option('\" + objs.get(i) + "\",'\" + i + \"')");
        }
    } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } finally {
        atualizarPilhaNaInterface();
    }
}

public void selecionarAtor(Integer i) {
    int numAtores = dicAtores.tamanho();
    if(i == null) {
        atorSelecioneado = null;
    }
    else if (i >=0 && i<numAtores) {
        atorSelecioneado = i;
        Cursor2d ator = (Cursor2d) dicAtores.obterElemento(i);
        if(pilha!= null)
            pilha.deleteObserver(tratadorPilha);
        pilha = ator.obterExecutor().obterEscopo().obterPilha();
        pilha.addObserver(tratadorPilha);
        atualizarPilhaNaInterface();
    }
    else if(numAtores > 0)
        selecionarAtor(0);
    else
        selecionarAtor(null);
}

```

```

}

TratadorPilha tratadorPilha = new TratadorPilha();
private class TratadorPilha implements Observer {

    @Override
    public void update(Observable o, Object arg) {
        atualizarPilhaNaInterface();
    }

}

private void atualizarPilhaNaInterface() {
    if(pilha == null || atorSelecioneado == null) {
        return;
    }
    try
    {
        JSObject win = JSObject.getWindow(applet);
        if(win!=null)

            win.eval("document.getElementById('sPilha').options.length = 0");

        for (int ind = 0; (ind < pilha.obterTamanho()) && (ind < 50);
ind++)
        {
            try {
                Palavra p = (Palavra) pilha.peekPalavra(ind);

                win.eval("document.getElementById('sPilha').options.add( new
Option('\" + p.toString() + '\", '\" + p.toString() + '\'))");
            } catch (ExcecaoEmTelis e) {
                e.printStackTrace();
            }
        }
        Object[] arg = new Object[0];
        win.call("invalidatePainelInterp", arg);
    }
}

```

```
    } catch (Exception e) {e.printStackTrace();} //try-catch para  
possibilitar debug fora do navegador, jsoject dispara exceção  
    }  
}
```

```

package web.tratadores;

import java.applet.Applet;
import java.util.List;
import java.util.Observable;
import java.util.Observer;

import br.ufsc.edugraf.telis.maquina.ContextoDeEdicao;

import netscape.javascript.JSObject;

public class TratadorDicionarioDeModelos implements Observer {

    Applet applet;
    private ContextoDeEdicao contextoDeEdicao;

    public TratadorDicionarioDeModelos(Applet applet, ContextoDeEdicao
contextoDeEdicao) {
        this.applet = applet;
        this.contextoDeEdicao = contextoDeEdicao;
        contextoDeEdicao.adicionarObservadorNosModelos(this);
    }

    @Override
    public void update(Observable arg0, Object arg1) {
        Object[] arg = new Object[1];
        List<String> modelos = contextoDeEdicao.obterNomesDeModelos();
        arg[0] = modelos;
        try {
            JSObject win = JSObject.getWindow(applet);
            win.call("atualizaModelosDeLista", arg);
        }
        catch (Exception e) {
            System.out.println("TratadorDicionarioDeModelos: erro
atualizando. Lista: ");
            System.out.println(modelos);
            e.printStackTrace();
        }
    }
}

```

}
}

```

package web.utils;

import java.applet.Applet;

import netscape.javascript.JSObject;

public class JsUtils {

    private static JSObject win = null;
    private static String appUrl = "";

    public static String obterAppUrl() {
        return appUrl;
    }

    public static void fixarAppUrl(String url) {
        appUrl = url;
    }

    public static void fixarWin(Applet applet) {
        win = getWindow(applet);
    }

    @SuppressWarnings("finally")
    public static JSObject getWindow(Applet applet) {
        JSObject win = null;
        try {
            win = JSObject.getWindow(applet);
        } catch (netscape.javascript.JSException e) {
            System.out.println("JSObject não conseguiu pegar
janela");
        } finally {
            return win;
        }
    }

    public static Object call(JSObject win, String method, Object ...
args) {

```

```

        if(win!=null)
            return win.call(method, args);
        return null;
    }

    /**
     * deprecated, é pra usar static mesmo
     * @param win
     */
    @Deprecated
    public JsUtils(JSObject win) {
        JsUtils.win = win;
    }

    public static Object call(String method, Object ... args) {
        return call(win, method, args);
    }

    public static void eval(String str) {
        if(win!=null)
            win.eval(str);
    }
}

```

```
package web.utils;

import java.awt.Graphics;
import java.awt.Image;
import java.awt.image.BufferedImage;
import java.io.BufferedReader;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.StringWriter;
import java.io.Writer;

import javax.imageio.ImageIO;

//Esta classe teve como base o exemplo:
//http://www.exampledepot.com/egs/java.io/file2bytearray.html

public class Utils {

    public static byte[] getBytesFromFile(File file) throws IOException {
        InputStream is = new FileInputStream(file);

        // Get the size of the file
        long length = file.length();

        if (length > Integer.MAX_VALUE) {
            // File is too large
        }

        // Create the byte array to hold the data
        byte[] bytes = new byte[(int)length];

        // Read in the bytes
        int offset = 0;
```



```

        int numRead = 0;
        while (offset < bytes.length
                && (numRead=is.read(bytes, offset, bytes.length-
offset)) >= 0) {
            offset += numRead;
        }

        // Ensure all the bytes have been read in
        if (offset < bytes.length) {
            throw new IOException("Could not completely read file
"+file.getName());
        }

        // Close the input stream and return bytes
        is.close();
        return bytes;
    }

    public static byte[] getBytesFromFilename(String filename) throws
IOException {
        return getBytesFromFile(new File(filename));
    }

    public static byte[] imageToByte(Image image) {

        BufferedImage bi = new
BufferedImage(image.getWidth(null),image.getHeight(null),BufferedImage.TYPE
_INT_RGB);

        Graphics bg = bi.getGraphics();
        bg.drawImage(image, 0, 0, null);
        bg.dispose();

        ByteArrayOutputStream buff = new ByteArrayOutputStream();
        try {
            ImageIO.write(bi, "JPG", buff);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

```

```

        return buff.toByteArray();
    }

    public static String convertStreamToString(InputStream is)
        throws IOException {
        /*
         * To convert the InputStream to String we use the
         * Reader.read(char[] buffer) method. We iterate until the
         * Reader return -1 which means there's no more data to
         * read. We use the StringWriter class to produce the string.
         */
        if (is != null) {
            Writer writer = new StringWriter();

            char[] buffer = new char[1024];
            try {
                Reader reader = new BufferedReader(
                    new InputStreamReader(is, "UTF-8"));
                int n;
                while ((n = reader.read(buffer)) != -1) {
                    writer.write(buffer, 0, n);
                }
            } finally {
                is.close();
            }
            return writer.toString();
        } else {
            return "";
        }
    }
}

```

APÊNDICE C – Código-fonte das classes utilizadas por ambos, Applet e Servidor

```
package utils;

public class Utils {

    /**
     * Compara se dois objetos são iguais tratando o caso de o primeiro
ser nulo
     *
     * @param obj1
     * @param obj2
     * @return verdadeiro se ambos são nulos ou se obj1.equals(obj2)
     */
    public static boolean equals(Object obj1, Object obj2) {
        return obj1 == null? obj2 == null: obj1.equals(obj2);
    }
}
```

```

package portablebeans;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

import biz.source_code.base64Coder.Base64Coder;

public class Serializador implements Serializable{

    /**
     *
     */
    private static final long serialVersionUID = 4145163217455748406L;

    public static String serializar(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        //oos.flush();
        oos.close();

        String result = new
String(Base64Coder.encode(baos.toByteArray()));
        baos.close();
        return result;
    }

    public static Object deserializar(String str) throws IOException,
ClassNotFoundException {
        byte[] data = Base64Coder.decode(str);
        ObjectInputStream ois = new ObjectInputStream(new
ByteArrayInputStream(data));

        Object result = ois.readObject();
    }

```

```
        ois.close();  
        return result;  
    }  
}
```

```

package portablebeans;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToMany;
import javax.persistence.OneToOne;
import javax.persistence.CascadeType;
import javax.persistence.Transient;

import utils.Utills;
/**
 *
 * @author felipe
 */
@Entity
public class ModeloBean implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -6503922070374483075L;

    private String nome = null;

    @OneToOne(cascade=CascadeType.ALL, fetch=FetchType.EAGER,
mappedBy="modelo")
    private Map<String, AgendaBean> agendas = null;
    boolean ehMolde = false;

```

```

@Transient
private List<String> listaMoldesStr = null;
private String usuario = null;
@ManyToMany
private List<ModeloBean> listaMoldes = null;

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id = null;

    public Long getId() {
        return id;
    }

public void setId(Long id) {
    this.id = id;
}

@Override
public int hashCode() {

    if(id != null)
        return id.hashCode();

    int hash = 0;

    if(ehMolde)
        hash += 1;

    if(agendas != null)
        hash += agendas.hashCode();

    if(listaMoldesStr != null)
        hash += listaMoldesStr.hashCode();

    return hash;
}

```

```

@Override
public boolean equals(Object object) {
    if(object == null)
        return false;

    if (!(object instanceof ModeloBean))
        return false;

    ModeloBean other = (ModeloBean) object;

    if(this.id != null)
        return this.id.equals(other.id);
    //meu id é nulo

    return this.ehMolde == other.ehMolde &&
    Utils.equals(this.listaMoldesStr, other.listaMoldesStr) &&
    Utils.equals(this.id, other.id) && Utils.equals(this.agendas,
    other.agendas) && Utils.equals(this.nome, other.nome); // tirei revisao
    para contornar recursao infinita (revisao chamaria equals do modelo de
    volta)

}

public ModeloBean() {
    super();
    agendas = new HashMap<String, AgendaBean>();
}

/**
 * @return the nome
 */
public String getNome() {
    return nome;
}

/**
 * @param nome the nome to set

```



```

    */
    public void setNome(String nome) {
        this.nome = nome;
    }

    /**
     * @return the agendas
     */
    public Map<String, AgendaBean> getAgendas() {
        return agendas;
    }

    /**
     * @param agendas the agendas to set
     */
    public void setAgendas(Map<String, AgendaBean> agendas) {
        Iterator<Entry<String, AgendaBean>> it =
agendas.entrySet().iterator();
        while(it.hasNext())
            it.next().getValue().setModelo(this);
        this.agendas = agendas;
    }

    @Transient
    public void addAgenda(AgendaBean agenda) {
        agenda.setModelo(this);
        agendas.put(agenda.getNome(), agenda);
    }

    @Transient
    public AgendaBean obterAgenda(String nome) {
        return agendas.get(nome);
    }

    @Transient
    public Map<String, String> getAgendasCod() {
        Map<String, String> result = new HashMap<String, String>();
        for(Entry<String, AgendaBean> entrada: agendas.entrySet())

```

```

        result.put(entrada.getKey(), entrada.getValue().getCodigo());
    }
    return result;
}

@Transient
private String getCodAgendas() {
    String cod = "";
    for(Entry<String, AgendaBean> entrada: agendas.entrySet())
        cod += entrada.getValue();
    return cod;
}

@Transient
private String getMoldadoComoString() {
    String result = "";
    for(String nome: listaMoldesStr)
        result += nome + " ";
    if(result.length() > 1)
        return result.substring(0, result.length()-1);
    return "";
}

@Override
public String toString() {
    String moldado = getMoldadoComoString();
    return ehMolde ? "[" + moldado + "] \\" + nome + "\" [" +
getCodAgendas() + "] incluirMolde" : "[" + moldado + "] \\" + nome + "\" [" +
+ getCodAgendas() + "] incluirModelo";
}

/**
 * Compara com modelo de revisão anterior e seta na revisão atual
 * \(this\) ids de
 * agendas que não mudaram. Atualiza próprio id se continua tudo igual
 * @param modelo
 * @return
 * @return true se modelo foi modificado e precisa ser atualizado (id
continua null,

```

```

    * false se não precisa ser atualizado (id foi setado com id do
    modeloAntigo)
    */
    @Transient
    public boolean atualizaIds(ModeloBean modeloAntigo) {
        if(modeloAntigo==null) {//não havia modelo antigo
            this.setId(null);
            return true;
        }
        //Se número de agendas for iguais, parte-se da hipótese que as
        agendas são iguais, e então ao encontrar
        //uma agenda que mudou, atualiza este boolean
        boolean todasAsAgendasSaoIguais = this.agendas.size() ==
        modeloAntigo.agendas.size();
        for(Entry<String,AgendaBean> entrada: agendas.entrySet()) {
            AgendaBean agenda = entrada.getValue();
            AgendaBean agendaAntiga =
            modeloAntigo.obterAgendaDeNome(agenda.getNome());
            if(agendaAntiga == null) {
                todasAsAgendasSaoIguais = false;
                agenda.setId(null);
            } else if(agenda.ehSimilar(agendaAntiga))
                agenda.setId(agendaAntiga.getId());
            else {
                todasAsAgendasSaoIguais=false;
                agenda.setId(null);
            }
        }

        boolean revisaoMudou = !todasAsAgendasSaoIguais;
        if(this.getMoldado() != modeloAntigo.getMoldado())
            revisaoMudou = true;

        if(!revisaoMudou)
            this.setId(modeloAntigo.getId());
        else
            this.setId(null);
        return revisaoMudou;
    }

```

```
}

@Transient
public AgendaBean obterAgendaDeNome(String nome) {
    return agendas.get(nome);
}

public boolean ehMolde() {
    return ehMolde;
}

public void setEhMolde(boolean ehMolde) {
    this.ehMolde = ehMolde;
}

@Transient
public List<String> getMoldado() {
    return listaMoldesStr;
}

@Transient
public void setMoldado(List<String> moldado) {
    listaMoldesStr = moldado;
}

public String getUsuario() {
    return usuario;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}

public List<ModeloBean> getListaMoldes() {
    return listaMoldes;
}

public void setListaMoldes(List<ModeloBean> listaMoldes) {
```

```
        this.listaMoldes = listaMoldes;
        listaMoldesStr = new ArrayList<String>();
        for(ModeloBean molde: listaMoldes)
            listaMoldesStr.add(molde.getNome());
    }

    @Transient
    public void addNomeMolde(String nomeMolde) {
        if(listaMoldesStr == null)
            listaMoldesStr = new ArrayList<String>();
        listaMoldesStr.add(nomeMolde);
    }

    @Transient
    public void criaListaMoldesStr() {
        listaMoldesStr = new ArrayList<String>();
        for(ModeloBean molde: listaMoldes)
            listaMoldesStr.add(molde.getNome());
    }

    @Transient
    public void addMolde(ModeloBean molde) {
        if(listaMoldes == null)
            listaMoldes = new ArrayList<ModeloBean>();
        listaMoldes.add(molde);
    }
}
```

```

package portablebeans;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
import javax.persistence.*;

import utils.Utills;

@Entity
public class ApliqueBean implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 3200253330689194890L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id = null;
    private String usuario = null;
    private String nome = null;
    private int altura = 0;
    private int largura = 0;
    @ManyToMany(cascade=CascadeType.ALL, fetch=FetchType.EAGER)// Ver se
isso vale quando persiste aplique só ou quando persiste modelo tb
    //Esta lista é transient por que na serialização o que vai é a lista
    de strings, menor overhead. mas na hora de persistir servidor tem que
    atualizar a
    //lista de modelosbean pela lista de string para fazer os
    relacionamentos corretos.
    private List<ModeloBean> modelos = null;
    //Do applet para o servidor, é enviado a lista de beans nula, para
    evitar overhead
    //e passa modelos pela lista de strings
    //no servidor, é recuperado os beans para jpa fazer relacionamento
    correto
    @Transient
    private List<String> modelosStr = null;

```

```

@Override
public boolean equals(Object object) {
    if(object == null)
        return false;

    if (!(object instanceof ApliqueBean))
        return false;

    ApliqueBean other = (ApliqueBean) object;

    if(this.id != null)
        return this.id.equals(other.id);
    //meu id é nulo

    return Utils.equals(this.usuario, other.usuario) &&
        Utils.equals(this.nome, other.nome) && Utils.equals(this.modelos,
        other.modelos) && largura == other.largura && altura == other.altura;
}

@Transient
public void addModeloStr(String nomeModelo) {
    if(modelosStr == null)
        modelosStr = new ArrayList<String>();
    modelosStr.add(nomeModelo);
}

@Transient
public void addModelo(ModeloBean modelo) {
    if(modelos == null)
        modelos = new ArrayList<ModeloBean>();
    modelos.add(modelo);
}

@Transient
public void criaModelosStr() {
    if(modelos == null) {

```

```
        modelosStr = new ArrayList<String>(0);
        return;
    }

    modelosStr = new ArrayList<String>(modelos.size());
    for(ModeloBean modelo: modelos)
        modelosStr.add(getNome());
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

public String getUsuario() {
    return usuario;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public int getAltura() {
    return altura;
}

public void setAltura(int altura) {
```



```

        this.altura = altura;
    }

    public int getLargura() {
        return largura;
    }

    public void setLargura(int largura) {
        this.largura = largura;
    }

    public List<ModeloBean> getModelos() {
        return modelos;
    }

    public void setModelos(List<ModeloBean> modelos) {
        this.modelos = modelos;

        if(modelos == null)
            modelosStr = null;
        else {
            modelosStr = new ArrayList<String>(modelos.size());
            for(ModeloBean modelo: modelos)
                modelosStr.add(modelo.getNome());
        }
    }

    @Transient
    public List<String> getModelosStr() {
        if(modelosStr == null && modelos != null) {
            List<String> result = new ArrayList<String>();
            for(ModeloBean modelo: modelos)
                result.add(modelo.getNome());
            return result;
        }
        return modelosStr;
    }

```

```

@Transient
public void setModelosStr(List<String> modelosStr) {
    this.modelosStr = modelosStr;
}

public ModeloBean obterModeloDeNome(String nome) {
    for(ModeloBean modelo: modelos)
        if(modelo.getNome().equals(nome))
            return modelo;
    return null;
}

public boolean atualizaIds(ApliqueBean aplicacaoAntiga) {
    if(aplicacaoAntiga == null) {
        this.id = null;
        return true;
    }
    if(this.modelos.size() != aplicacaoAntiga.modelos.size()) {
        this.id = null;
        return true;
    }
    boolean aplicacaoMudou= false;
    for(ModeloBean modelo: modelos) {
        ModeloBean modeloAntigo=
aplicacaoAntiga.obterModeloDeNome(modelo.getNome());
        if(!Utils.equals(modelo, modeloAntigo)) {
            this.id = null;
            return true;
        }
    }
    this.id = aplicacaoAntiga.id;
    return false;
}
}

```

```

package portablebeans;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Map;
import java.util.Set;

import utils.Utills;

public class AmbienteBean implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 7595554164810809642L;

    private Set<ModeloBean> modelos;

    /**
     * Guarda a quantidade de vezes que um ator repete.
     */
    private Map<ApliqueBean, Integer> apliques;

    public AmbienteBean() {
        modelos = new HashSet<ModeloBean>();
        apliques = new HashMap<ApliqueBean, Integer>();
    }

    public AmbienteBean(Set<ModeloBean> modelos, Map<ApliqueBean,
Integer> apliques) {
        this.modelos = modelos;
        this.apliques = apliques;
    }

    public AmbienteBean(Set<ModeloBean> modelos) {

```

```

        this.modelos = modelos;
        apliques = new HashMap<ApliqueBean, Integer>();
    }

    public AmbienteBean(Map<ApliqueBean, Integer> apliques) {
        this.modelos = new HashSet<ModeloBean>();
        this.apliques = apliques;
    }

    public AmbienteBean(List<ModeloBean> modelos, List<ApliqueBean>
listaApliques) {
        this.modelos = new HashSet<ModeloBean>(modelos);
        criaMapaDeApliquesAPartirDeLista(listaApliques);
    }

    private void criaMapaDeApliquesAPartirDeLista(
        List<ApliqueBean> listaApliques) {
        this.apliques = new HashMap<ApliqueBean,
Integer>(listaApliques.size());
        for(ApliqueBean aplique: listaApliques) {
            Integer contAtual = apliques.get(aplique);
            apliques.put(aplique, contAtual == null? 1 : contAtual+
+);
        }
    }

    public AmbienteBean(Set<ModeloBean> modelos, List<ApliqueBean>
listaApliques) {
        this.modelos = modelos;
        criaMapaDeApliquesAPartirDeLista(listaApliques);
    }

    public Set<ModeloBean> getModelos() {
        return modelos;
    }

    public List<ApliqueBean> getApliquesComoLista() {

```

```

        List<ApliqueBean> result = new
ArrayList<ApliqueBean>(apliques.size());
        for(ApliqueBean aplique: apliques.keySet())
            for(int i = 0; i < apliques.get(aplique); i++)
                result.add(aplique);
        return result;
    }

    public Map<ApliqueBean, Integer> getApliques() {
        return apliques;
    }

    @Override
    public boolean equals(Object obj) {
        if(obj == null)
            return false;
        if(!(obj instanceof AmbienteBean))
            return false;
        AmbienteBean outro = (AmbienteBean)obj;
        return Utils.equals(this.modelos, outro.modelos) &&
Utils.equals(this.apliques, outro.apliques);
    }

    @Override
    public int hashCode() {
        if(modelos==null)
            return (apliques==null)? 0 : apliques.hashCode();
        if(apliques==null)
            return modelos.hashCode();
        return modelos.hashCode() + apliques.hashCode();
    }
}

```

```

package portablebeans;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.ManyToOne;
import javax.persistence.Transient;

import utils.Utils;
/**
 *
 * @author felipe
 */
@Entity
public class AgendaBean implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 7942447944263473403L;
    private String nome = null;
    private String codigo = null;

    @ManyToOne
    private ModeloBean modelo = null;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id = null;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {

```

```

        this.id = id;
    }

    @Override
    public int hashCode() {

        if(this.id != null)
            return id.hashCode();

        int hash = 0;

        if(nome != null)
            hash += nome.hashCode();

        if(codigo != null)
            hash += codigo.hashCode();

        //não tem modelo porque daria stackoverflow, e equals também não pode
        ter modelo

        return hash;
    }

    @Override
    public boolean equals(Object object) {

        if(object == null)
            return false;

        if (!(object instanceof AgendaBean)) {
            return false;
        }

        AgendaBean other = (AgendaBean) object;

        if(this.id != null)
            return this.id.equals(other.id);
    }

```

```

        return this.ehSimilarInsegura(other);
    }

    /**
     * Usada pelo modelo para saber se uma agenda deve ser atualizada em
     nova revisão
     * @param agenda
     * @return agenda tem mesmo código e nome
     */
    @Transient
    public boolean ehSimilar(AgendaBean other) {
        return other == null ? false : ehSimilarInsegura(other);
    }

    @Transient
    private boolean ehSimilarInsegura(AgendaBean other) {
        return Utils.equals(this.nome, other.nome) &&
        Utils.equals(this.codigo, other.codigo); //tirei modelo para evitar stack
        overflow
    }

    /**
     * @return the nome
     */
    public String getNome() {
        return nome;
    }

    /**
     * @param nome the nome to set
     */
    public void setNome(String nome) {
        this.nome = nome;
    }

    /**
     * @return the codigo

```



```
    */  
    public String getCodigo() {  
        return codigo;  
    }  
  
    /**  
     * @param codigo the codigo to set  
     */  
    public void setCodigo(String codigo) {  
        this.codigo = codigo;  
    }  
  
    @Override  
    public String toString() {  
        return "'" + nome + "\" [" + codigo + "] incluirAgenda ";  
    }  
  
    /**  
     * @return the modelo  
     */  
    public ModeloBean getModelo() {  
        return modelo;  
    }  
  
    /**  
     * @param modelo the modelo to set  
     */  
    public void setModelo(ModeloBean modelo) {  
        this.modelo = modelo;  
    }  
}
```

APÊNDICE D – Código-fonte da aplicação Web

TelisWeb/web/app.jsp

```

1 <%@page import="sessao.BdToHomeFile"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <%@page contentType="text/html" pageEncoding="UTF-8" import="java.util.Map"%>
4 <%
5     BdToHomeFile bd = (BdToHomeFile) session.getAttribute("bd2home");
6     session.setAttribute("Imagens", bd.getImagens());
7     //session.setAttribute("Modelos", bd.getModelos());
8 %>
9 <!DOCTYPE html>
10 <html lang="pt-br">
11     <head>
12         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
13         <title>Interpretador do Telis Acessível</title>
14         <script>
15
16             appUrl = "${appUrl}";//editorDeImagens e editorDeSons usa.Isso pode ser
desnecessário se os .js fossem também jsp
17             srcHome = "${strSrcHome}";
18
19             //Função chamada pelo applet no init, para passar parametros iniciais do applet para
javascript
20             function appletIniciou(_contextoDeEdicao, _contextoDeExecucao, _ambiente,
_editorDeModelos, _editorDeApliques) {
21                 carregouModelos = false;
22                 contextoDeEdicao = _contextoDeEdicao;
23                 contextoDeExecucao = _contextoDeExecucao;
24                 ambiente = _ambiente;
25                 ambiente.fixarAppUrl(appUrl);
26                 editorDeModelos = _editorDeModelos;
27                 editorDeApliques = _editorDeApliques;
28                 carregarImagens();
29                 carregarSons();
30
31                 invalidateSModelos();
32

```

```

33     iniciado = true;
34 }
35
36 //chamado no start()
37 function appletStart() { //Isso é necessário porque se modelos forem carregados antes
do start as vezes dá bug!
38     if(!carregouModelos) { //não vou precisar mais quando eu não estiver mais usando a
gíria incluirModelo para incluir um modelo!
39         carregarModelos();
40         carregarAplices();
41     }
42     document.getElementById('contentLink').focus();
43     ambiente.executarNoAtorDidatico(['TECLADO "teclaPressionada" 10 número]
["entradaFocus()" js] quandoDito');
44     ambiente.executarNoAtorDidatico(["aviso" [textoMensagem]S associar
"dialogoDeAlert(\\\" @textoMensagem]S concatenar \"\\\",\\\"mensagem do aplique\\\")\" concatenar js]
incluirAgenda');
45     ambiente.executarNoAtorDidatico(["aviso" Texto] [ultimo aviso] quandoDito');
46 }
47
48 function entradaFocus() {
49     document.getElementById('entrada').focus();
50     document.getElementById('entrada').select();
51 }
52
53 //scripts para o interpretador
54
55 var activeElem, interpDiv, imagensDiv, modelosDiv;
56
57
58 window.onload=function() {
59     interpDiv = document.getElementById('interpretadorDiv');
60     imagensDiv = document.getElementById('editorImagemDiv');
61     modelosDiv = document.getElementById('editorMoldesDiv');
62     sonsDiv = document.getElementById('editorSomDiv');
63     apliquesDiv = document.getElementById('editorAplicesDiv');
64     onloadInterp();
65     onloadImagens();

```

```

66     onloadSons();
67     onloadModelos();
68     onloadAppliques();
69
70     onloadDialogs();
71
72 }
73
74 //TODO tirar um arquivo JS dessas funções, tipo gerenciadordeabas.js
75 function swapElem(elem) {
76     hideElem(activeElem);
77     showElem(elem);
78     // activeElem.style.display = 'none';
79     // elem.style.display = '';
80     // activeElem.hidden = true;
81     // elem.hidden = false;
82     activeElem = elem;
83 }
84
85 function hideElem(elem) {
86     interpDiv = document.getElementById('interpretadorDiv');
87     if(elem == interpDiv) {
88         interpApplet = document.getElementById('interpretadorDivApplet');
89         interpFormEsq = document.getElementById('interpretadorDivFormEsq');
90         interpFormBaixo = document.getElementById('interpretadorDivFormBaixo');
91         var interpretadorDivAjuda = document.getElementById('interpretadorDivAjuda');
92
93         interpFormEsq.hidden = true; //html5 acessability
94         //http://dev.w3.org/html5/spec/editing.html#the-hidden-attribute
95         interpFormEsq.style.display = 'none';
96         interpFormBaixo.hidden = true;
97         interpFormBaixo.style.display = 'none';
98         interpretadorDivAjuda.hidden = true;
99         interpretadorDivAjuda.style.display = 'none';
100
101         interpApplet.style.visibility = "hidden";
102         interpApplet.width = 0;
103         interpApplet.height = 0;

```

```

103
104     aplique.style.visibility = "hidden";
105     //aplique.width = 0;
106     //aplique.height = 0;
107
108     interpDiv.style.visibility = "hidden";
109 } else {
110     elem.style.display = 'none';
111     elem.hidden = true; //html5 acessibility
112     http://dev.w3.org/html5/spec/editing.html#the-hidden-attribute
113 }
114
115 function showElem(elem) {
116     interpDiv = document.getElementById('interpretadorDiv');
117     if(elem == interpDiv) {
118         interpApplet = document.getElementById('interpretadorDivApplet');
119         interpFormEsq = document.getElementById('interpretadorDivFormEsq');
120         interpFormBaixo = document.getElementById('interpretadorDivFormBaixo');
121         var interpretadorDivAjuda = document.getElementById('interpretadorDivAjuda');
122
123         interpFormEsq.hidden = false;
124         interpFormEsq.style.display = "";
125
126         interpFormBaixo.hidden = false;
127         interpFormBaixo.style.display = "";
128
129         interpretadorDivAjuda.hidden = false;
130         interpretadorDivAjuda.style.display = "";
131
132         interpApplet.style.visibility = "visible";
133         interpApplet.width = 500;
134         interpApplet.height = 500;
135
136         aplique.style.visibility = "visible";
137         //aplique.width = 500;
138         //aplique.height = 500;
139

```

```

140         interpDiv.style.visibility = "visible";
141     } else {
142         elem.style.display = "";
143         elem.hidden = false;
144     }
145 }
146
147     function preventSubmit(){return false;}
148
149     </script>
150     <link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css"
151     rel="stylesheet" type="text/css"/>
152     <script type="text/javascript" src="jquery-1.6.2.min.js"></script>
153     <script type="text/javascript" src="jquery-ui-1.8.16.custom.min.js"></script>
154     <script type="text/javascript" src="dialogs.js"></script>
155     <script type="text/javascript" src="editordeapliques.js?<
156     %out.print(java.util.Calendar.getInstance().getTimeInMillis());//chuncho pro browser não fazer cache
157     do arquivo js%>"></script>
158     <script type="text/javascript" src="editordemodelos.js?<
159     %out.print(java.util.Calendar.getInstance().getTimeInMillis());//chuncho pro browser não fazer cache
160     do arquivo js%>"></script>
161     <script type="text/javascript" src="editordesons.js?<
162     %out.print(java.util.Calendar.getInstance().getTimeInMillis());%>"></script>
163     <script type="text/javascript" src="editordeimagens.js?<
164     %out.print(java.util.Calendar.getInstance().getTimeInMillis());//chuncho pro browser não fazer cache
165     do arquivo js%>"></script>
166     <script type="text/javascript" src="utils.js?<
167     %out.print(java.util.Calendar.getInstance().getTimeInMillis());%>"></script>
168     <script type="text/javascript" src="interpretador.js?<
169     %out.print(java.util.Calendar.getInstance().getTimeInMillis());%>"></script>
170     <script type="text/javascript" src="FileUpload.js?<
171     %out.print(java.util.Calendar.getInstance().getTimeInMillis());%>"></script>
172     <style type="text/css">
173         #interpretadorDiv {
174             position:relative; left:0px; top:0px;
175         }
176     /*         #entradaInterp {

```

```
167         position: absolute;
168         left: 0px;
169         top: 530px;
170     }*/
171 #entrada {
172     width: 500px;
173 }
174 #interpretadorDivApplet {
175     position: absolute;
176     left: 200px;
177     top: 200px;
178 }
179
180 /* #thumb {
181     position: absolute;
182     left: 150px;
183     top: 20%;
184 }*/
185
186 /* input#playpause {
187     top: 0px;
188     left: 0px;
189     width: 34px;
190     height: 30px;
191     background-color: transparent;
192     background-image: url('player_play.png');
193     color: transparent;
194     border-right-color: #848484; */
195
196 .painel {
197     float: left;
198     margin: 2px;
199 }
200
201 #interpretadorDivFormBaixo {
202     clear: both;
203 }
204
```

```

205     .divClear{
206         clear:both;
207     }
208
209     #codAgenda {
210         width: 600px;
211         height: 400px;
212     }
213     .selectEdModelos {
214         width: 150px;
215         height: 400px;
216     }
217     .listaInterp {
218         width:130px;
219         height: 250px;
220     }
221     #painelModelosAplique {
222         display: table;
223     }
224     .painelCelula {
225         float: left;
226         margin: 2px;
227         display: table-cell;
228     }
229     #botoesCentraisAplique {
230         vertical-align: middle;
231     }
232 </style>
233 </head>
234 <body role="application">
235     <div id="geral"><!-- Serve para esconder tudo quando for mostrar um erro !-->
236         <header id="cabecalho">
237             <a href="logout">Finalizar Sessão</a> | <a href="GerenciarUsuario.jsp">Gerenciar
238             Usuário</a><br/>
239             <nav>
240                 <ul>
241                     <li><a href="#content" id="contentLink">Ir para aba atual</a></li>

```



```

241         <li><a href="#InterpretadorTitulo"
onclick="javascript:swapElem(interpDiv)">Interpretador</a></li>
242         <li><a href="#ImagensTitulo"
onclick="javascript:swapElem(imagensDiv)">Imagens</a></li>
243         <li><a href="#ModelosTitulo"
onclick="javascript:swapElem(modelosDiv)">Modelos</a></li>
244         <li><a href="#SonsTitulo"
onclick="javascript:swapElem(sonsDiv)">Sons</a></li>
245         <li><a href="#ApliquesTitulo"
onclick="javascript:swapElem(apliquesDiv)">Apliques</a></li>
246     </ul>
247 </nav>
248 </header>
249 <div id="content">
250     <div id="interpretadorDiv" class="divClear">
251         <h1 id="InterpretadorTitulo">Interpretador</h1>
252         <div id="interpretadorDivAjuda">
253             <p>
254                 Use as seguintes teclas de acesso:
255             </p>
256             <ul>
257                 <li>P: vai para pilha</li>
258                 <li>A: vai para atores</li>
259                 <li>E: vai para entrada de comandos</li>
260                 <li>X: Executa linha de comando</li>
261             </ul>
262         </div>
263         <div id="interpretadorDivFormEsq" class="painelCelula">
264             <form name="interpretador" onsubmit="preventSubmit">
265                 <label id="labelPilha" for="sPilha" accesskey="p">Pilha: </label><br/>
266                 <select id="sPilha" size="4" class="listaInterp" aria-
live="polite"></select><br/>
267                 <label id="labelSAtores" for="sAtores" accesskey="a">Atores:
</label><br/>
268                 <select id="sAtores" size="4" class="listaInterp" aria-
live="polite"></select>
269                 <input type="submit" hidden="hidden"/><!-- Validador wcag 2.0 exige
botão submit !-->

```

```

270         </form>
271     </div>
272     <div id="interpretadorDivFormBaixo" class="painelCelula">
273         <form name="entradaInterpretador" id="entradaInterpretador">
274             <label for="entrada" accesskey="e">Entrada de comandos</label>
275             <input id="entrada" type="text" title="Entrada de comandos" size="100"/>
276             <input type="submit" value="Executar" accesskey="x"/>
277             <input type="button" id="btFocarApplet" value="Ir para aplique"
onclick="contextoDeExecucao.obterMundo().requestFocus()"/>
278         </form>
279     </div>
280     <div id="interpretadorDivApplet" class="painelCelula">
281         <object id="umModelo" type="application/x-java-applet" width="500"
height="500" tabindex="-1">
282             <param name="archive" value="<
%out.print(getServletContext().getInitParameter("telisVmJar"));%><
%out.print("?"+java.util.Calendar.getInstance().getTimeInMillis());%>"/>
283             <param name="code" value="<
%out.print(getServletContext().getInitParameter("appletMainClass"));%>"/>
284             <param name="cache_option" value="no"/>
285             <param name="mayscript" value="mayscript"/>
286             Área do Aplique Telis
287         </object><!-- MUDAR PARA RELATIVO!! !-->
288
289     </div>
290 </div>
291
292     <div id="editorImagemDiv" class="divClear">
293         <h1 id="ImagensTitulo">Carregador de Imagens</h1>
294         <form name="CarregalImagens" id="CarregalImagens" action="upload"
method="post" enctype="multipart/form-data">
295             <div id="painelEdImagem" class="painel">
296                 <label for="listaImagens">Imagens</label><br/>
297                 <select id="listaImagens" size="30" class="selectEdModelos" aria-
live="polite">
298                     <c:forEach var="imagemItem" items="{Imagens}">
299                         <option value="{imagemItem.value}" title="{imagemItem.value}">${
{imagemItem.key}}</option>

```

```

300         </c:forEach>
301     </select>
302 </div>
303 
304 <br/>
305 <label for="file">Enviar arquivo:</label>
306 <input name="file" id="file" type="file" onchange="uploadAjax()"/>
307 <input type="hidden" name="filename" id="filename"/>
308 <input type="hidden" name="tipo" id="tipo" value="imagem"/>
309 <input type="button" id="btRemoverImg" value="Remover Arquivo"
onclick="deletaImagem()"/>
310 <input type="submit" hidden="hidden"/><!-- Validador wcag 2.0 exige botão
submit !-->
311 </form>
312
313 </div>
314
315 <div id="editorSomDiv" class="divClear">
316 <h1 id="SonsTitulo">Carregador de Sons</h1>
317 <p>Use a tecla de acesso t para tocar ítem selecionado</p>
318 <form name="formSons" id="formSons" action="upload" method="post"
enctype="multipart/form-data">
319 <label for="sSons">Sons</label>
320 <select id="sSons" size="20" class="selectEdModelos" aria-live="polite">
321 <%@include file="geradoresComponentes/optionsSelectSom.jsp" %>
322 </select>
323 <input type="button" id="playpause" value="Tocar" Title="Tocar"
onclick="playAudio()" accesskey="t"/>
324 <label for="fileSom">Enviar arquivo:</label>
325 <input name="file" id="fileSom" type="file" onchange="uploadSndAjax()"/>
326 <input type="hidden" name="filename" id="filenameSom"/>
327 <input type="hidden" name="tipo" id="tipoSom" value="som"/>
328 <input type="button" id="btRemoverSom" value="Remover Som"
onclick="deletaSom()"/>
329
330 <audio id="amostraSom" src="Blank.wav" preload="auto"
tabindex="0"></audio>

```

```

331         <input type="submit" hidden="hidden"/><!-- Validador wcag 2.0 exige botão
submit !-->
332     </form>
333
334 </div>
335
336 <div id="editorMoldesDiv" class="divClear">
337     <h1 id="ModelosTitulo">Editor de Modelos, Moldes e Agendas</h1>
338     <p>Utilize as seguintes telcas de acesso:</p>
339     <ul>
340         <li>N: Novo modelo</li>
341         <li>m: Ativar Modelo</li>
342         <li>g: Nova agenda</li>
343         <li>S: Salvar agenda</li>
344     </ul>
345     <form id="editorMoldesForm">
346         <div id="esqMoldes">
347             <div id="painelModelos" class="painel">
348                 <label for="sModelos">Modelos e Moldes:</label><br>
349                 <select id="sModelos" class="selectEdModelos" size="4" aria-
live="polite">
350
351                 </select><br>
352                 <input type="button" id="btAddModelo" value="Novo Modelo"
onclick="addModelo()" accesskey="n"/><br>
353                 <input type="button" id="btAddMolde" value="Novo Molde"
onclick="addMolde()"/><br>
354                 <input type="button" id="btRemoverModelo" value="Remover
Molde/Modelo" onclick="excluirModelo()"/><br>
355                 <input type="button" id="btAtivarModelo" value="Ativar Modelo"
onclick="editorDeModelos.ativarModelo()" accesskey="m"/>
356             </div>
357             <div id="painelAgendas" class="painel">
358                 <label for="sAgendas">Agendas do Modelo:</label><br>
359                 <select id="sAgendas" class="selectEdModelos" size="4" aria-
live="polite">
360
361                 </select><br>
362                 <input type="button" id="btAddAgenda" value="Nova Agenda"
onclick="addAgenda()" accesskey="g"/><br>

```

```

361         <input type="button" id="btSalvarAgenda" value="Salvar"
onclick="salvarAgenda()" accesskey="s"/><br/>
362         <input type="button" id="btRemoverAgenda" value="Remover Agenda"
onclick="removerAgenda()"/>
363     </div>
364 </div>
365 <label for="inputMoldesUsados">Moldes usados pelo modelo:</label><br/>
366 <input id="inputMoldesUsados" type="text" onkeypress="return
noenter(event)" aria-live="polite"/><br/>
367 <label for="codAgenda">Código da agenda:</label><br/>
368 <textarea id="codAgenda" rows="200" cols="200" accesskey="c" aria-
live="polite"></textarea>
369 <input type="submit" hidden="hidden"/><!-- Validador wcag 2.0 exige botão
submit !-->
370 </form>
371 </div>
372 <div id="editorAppliquesDiv" class="divClear">
373 <h1 id="AppliquesTitulo">Editor de Appliques</h1>
374 <form>
375 <div class="painel">
376 <label for="sAppliques" id="labelSAppliques">Appliques: </label>
377 <br/>
378 <select id="sAppliques" size="4" class="selectEdModelos" aria-live="polite">
379
380 </select>
381 <br/>
382 <input type="button" id="btCriarApplique" value="Criar Applique"/>
383 <br/>
384 <input type="button" id="btRemoverApplique" value="Remover Applique"/>
385 <br/>
386 <input type="button" id="btImportarApplique" value="Importar Applique"/>
387 </div>
388 <div class="painel">
389 <div>
390 <label for="textNomeApplique">Nome: </label>
391 <input id="textNomeApplique" type="text" aria-live="polite"/>
392 <input id="btSalvarApplique" type="button" value="Salvar" title="Salvar
nome,largura e altura"/>

```

```

393         <input id="btAtivarAplique" type="button" value="Ativar"/>
394     <br/>
395     <input id="btPublicarLocal" type="button" value="Publicar localmente"/>
396     <input id="btPublicarWeb" type="button" value="Publicar na Web"/>
397 </div>
398 <div>
399     <label for="textLargura">Largura: </label>
400     <input id="textLargura" type="text"/>
401     <label for="textAltura">Altura: </label>
402     <input id="textAltura" type="text"/>
403 </div>
404 <div id="painelModelosAplique"><!--div com modelos disponíveis e modelos
instanciados-->
405     <div class="painelCelula"><!--div com modelos disponíveis-->
406         <label for="sModelosDisponiveis">Modelos disponíveis: </label>
407         <br/>
408         <select id="sModelosDisponiveis" size="4" class="selectEdModelos"
multiple="multiple" aria-live="polite"></select>
409     </div>
410     <div id="botoesCentraisAplique" class="painelCelula">
411         <input id="btAddModelosAplique" type="button" value="Adicionar a
modelos instanciados"/>
412         <br/>
413         <input id="btRemModelosAplique" type="button" value="Remover de
modelos instanciados"/>
414     </div>
415     <div class="painelCelula"><!--div com modelos que serão instanciados-->
416         <label for="sModelosInstanciados">Modelos que serão instanciados:
</label>
417         <br/>
418         <select id="sModelosInstanciados" size="4" class="selectEdModelos"
multiple="multiple" aria-live="polite"></select>
419     </div>
420 </div>
421 </div>
422 <input type="submit" hidden="hidden"/><!-- Validador wcag 2.0 exige botão
submit !-->
423 </form>

```

```
424         </div><!-- Fim editor de aplicaes !-->
425     </div><!-- fim div content !-->
426 </div><!-- fim div geral !-->
427
428 <div id="alertDiv" aria-describedby="msgAlert" aria-labelledby="msgAlert">
429     <p id="msgAlert">Mensagem do erro</p>
430 </div>
431
432 <div id="divPrompt" title="Aviso!">
433     <label for="textPrompt" id="labelPrompt"></label>
434     <input type="text" id="textPrompt"></input>
435 </div>
436 </body>
437 </html>
```

TelisWeb/web/login.jsp

```

1 <%@page contentType="text/html" pageEncoding="UTF-8"%><%
2 String user = request.getParameter("user");
3 boolean isError = request.getParameter("user") != null;%>
4 <!DOCTYPE html>
5 <%--
6 Document : login
7 Created on : 19/09/2011, 23:38:56
8 Author : felipe
9 --%>
10 <html lang="pt-br" xml:lang="pt-br">
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
13 <title><% if (isError) { %>
14 Erro - usuário ou senha inválidos<% } %>Página de Login</title>
15 <script>
16 function cadastrar() {
17 document.form1.action="/TelisWeb/cadastro"
18 document.form1.submit();
19 return true;
20 }
21 </script>
22 </head>
23 <body>
24 <h1>Telis - Página de login</h1>
25 <% if(isError) { %>
26 <strong>Erro - usuário ou senha inválidos</strong><br/>
27 <%}%>
28 <p>Olá! Bem vindo ao Ambiente Telis Web. Por favor faça o login abaixo. A
página a seguir pode reproduzir sons automaticamente. É necessário suporte a javascript e Java
Applet para prosseguir</p>
29 <form name="form1" id="form1" action="ValidaLogin" method="post">
30 <label for="user" accesskey="u">Usuario: </label>
31 <input id="user" name="j_username" type="text" title="Nome de usuário"
tabindex="1" <% if (isError) { %> value="<%=user%>"<%}%>/><br/>
32 <label for="pass" accesskey="p">Senha: </label>

```



```
33      <input id="pass" name="j_password" type="password" title="Senha"
tabindex="2"/><br/>
34      <input type="submit" value = "Entrar" tabindex="3"/>
35      <button type="button" id="btCadastro"
onclick="cadastrar()">Cadastrar</button>
36  </form>
37  </body>
38 </html>
```

TelisWeb/web/GerenciarUsuario.jsp

```
1 <%--
2   Document   : GerenciarUsuario
3   Created on : 01/12/2011, 12:44:09
4   Author    : felipe
5 --%>
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10  <head>
11    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12    <title>Gerenciador de conta de usuário</title>
13  </head>
14  <body>
15    <h1>Gerenciador de conta de usuário</h1>
16    <form action="MudarSenha" method="post">
17      <label for="novaSenha">Nova senha: </label>
18      <input type="password" id="novaSenha" name="novaSenha"></input>
19      <input type="submit"></input>
20    </form>
21    <br/>
22    <a href="RemoverConta">Remover esta conta</a>
23
24  </body>
25 </html>
```

TelisWeb/web/cadastro.html

```
1 <!--
2 To change this template, choose Tools | Templates
3 and open the template in the editor.
4 -->
5 <!DOCTYPE html>
6 <html lang="pt-br" xml:lang="pt-br">
7   <head>
8     <title>Página de Login</title>
9     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
10  </head>
11  <body>
12    <form action="/TelisWeb/cadastro" method="get">
13      <label for="user" accesskey="u">Usuario: </label>
14      <input id="user" name="user" type="text" title="Nome de usuário"
tabindex="1"/><br/>
15      <label for="pass" accesskey="p">Senha: </label>
16      <input id="pass" name="pass" type="password" title="Senha"
tabindex="2"/><br/>
17      <input type="submit" tabindex="3" name="bt1"/><input type="submit"
tabindex="3" name="bt2"/>
18    </form>
19  </body>
20 </html>
```

TelisWeb/web/dialog.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
5   <title>Página de Login</title>
6   <link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-
ui.css" rel="stylesheet" type="text/css"/>
7   <script type="text/javascript" src="jquery-1.6.2.min.js"></script>
8   <script type="text/javascript" src="jquery-ui-1.8.16.custom.min.js"></script>
9
10  <script>
11
12
13  $(document).ready(function () {
14    $("#divPrompt").dialog({
15
16                                autoOpen: false,
17                                buttons: { "Ok": function() { $(this).dialog("close");
18
19                                closeText: 'hide',
20                                modal:true,
21                                closeOnEscape: false,
22                                title: "título tardiu"
23                                });
24  });
25
26  function showDialog() {
27    //document.getElementById('msg').focus();
28    //document.getElementById('divPrompt').focus();
29    $('#divPrompt').dialog('open');
30    document.getElementById('textPrompt').style.visibility = "hidden";
31    //document.getElementById('divPrompt').focus();
32    //document.getElementById('msg').focus();
33  }
34  </script>
35 </head>

```

```
35 <body style="font-size:62.5%;">
36   <p>
37     blablabla
38     <button onclick="showDialog()">mostrar dialog</button>
39   </p>
40   <!--<div id="divPrompt" title="Aviso!">
41     I'm in a dialog
42
43     <input type="text"></input>
44   </div>-->
45
46 <p>teste</p>
47 <div id="divPrompt" title="Aviso do diálogo" aria-describedby="msg" tabindex="-1">
48   <p id="msg">testando este texto</p>
49   <!--      <label for="textPrompt" id="labelPrompt">label</label>
50     <input type="text" id="textPrompt">-->
51   </div>
52 </body>
53 </html>
```

TelisWeb/web/utils.js

```

1 /**
2  * Retorna um array com as opções selecionadas de um select. Útil para select com
  atributo multiple="multiple"
3  */
4  function pegaSelecionadosDeSelect(select){
5      var selecionados = new Array();
6      var count = 0;
7      for(var i = 0; i < select.length; i++) {
8          if(select[i].selected){
9              selecionados[count] = select[i].text;
10             count++;
11         }
12     }
13     return selecionados
14 }
15
16 //Isto só está sendo necessário pelo bug firefox+orca que para de ler quando o select
  fica vazio.
17 /**
18  * Método genérico para consistência dos select, único parametro obrigatório é uma
  lista (select)
19  * que será desabilitada se tiver zero parametros (e chunchada por causa do bug) ou
  habilitada caso contrário
20  *
21  * @param lista lista a ser invalidada, por default, se tamnho == 0, é tornada disable
  e adicionado um option nulo (por causa do bug), e caso contrário, tudo isso é desfeito
22  * @param labelLista referência para um label, que pode receber um novo valor
  quando select está ativo ou inativo (ver próximos 2 parametros), opcional.
23  * @param activeTxt O label passado pela referência no parâmetro anterior,
  labelLista, receberá esse valor caso seja considerado válido e esse valor tenha sido passado.
  opcional
24  * @param inactiveTxt
25  * @param regraInvalidar
26  */
27  function invalidateSelect(lista, labelLista,activeTxt, inactiveTxt, toSwitch,
  regraInvalidar, funcaoAoDesabilitar, funcaoAoHabilitar) {

```

```
28  var disable;
29  if(regraInvalidar == null) {
30      regraInvalidar = false;
31  }
32  if(lista.length == 0 || regraInvalidar) {
33      lista.add(new Option());
34      lista.disabled = true;
35      if(inactiveTxt!=null) {
36          labelLista.tabIndex = 0;
37          labelLista.textContent = inactiveTxt;
38      }
39      disable = true;
40  } else {
41      lista.disabled = false;
42      if(activeTxt != null) {
43          labelLista.tabIndex = -1;
44          labelLista.textContent = activeTxt;
45      }
46      for(i = 0; i < lista.length; i++)
47          if(lista[i].value == "") {
48              lista.remove(i);
49              break;
50          }
51      disable = false;
52  }
53  if(toSwitch != null) {
54      switchState(toSwitch,disable);
55  }
56 }
57
58
59 function switchState(objetos, state) {
60     for(var i = 0; i < objetos.length; i++) {
61         objetos[i].disabled = state;
62     }
63 }
64
65 function stringEstaEmLista(lista,nome) {
```

```
66  var i = 0;
67  //verifica se modelo está na lista
68  var size = lista.size();
69  for(i = 0; i < size; i++) {
70      if(lista.get(i) == nome)
71          break;
72  }
73  return i!= size;
74 }
75
76 function selecionaNomeEmSelect(nome, elem) {
77     elem.value=nome;
78 }
79
80
81 function getXmlHttp() {
82     var xmlhttp;
83     if (window.XMLHttpRequest)
84     { // code for IE7+, Firefox, Chrome, Opera, Safari
85         xmlhttp=new XMLHttpRequest();
86     }
87     else
88     { // code for IE6, IE5, não foi testado mas tá aí
89         xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
90     }
91     return xmlhttp;
92 }
93
94 //deprecated
95 function ajaxAtualizaComponente(componente, endereco)
96 {
97     var xmlhttp = getXmlHttp();
98     xmlhttp.onreadystatechange=function()
99     {
100         if (xmlhttp.readyState==4 && xmlhttp.status==200)
101         {
102             componente.innerHTML = xmlhttp.responseText;
103         }
```



```

104 }
105 xmlhttp.open("GET", endereco,true);
106 xmlhttp.send();
107 }
108
109 function ajaxAtualizaComponenteInvalida(componente, endereco, invalidate)
110 {
111     var xmlhttp = getXmlHttp();
112     xmlhttp.onreadystatechange=function()
113     {
114         if (xmlhttp.readyState==4 && xmlhttp.status==200)
115         {
116             componente.innerHTML = xmlhttp.responseText;
117             invalidate();
118         }
119     }
120     xmlhttp.open("GET", endereco,true);
121     xmlhttp.send();
122 }
123
124 function ajaxGetExecutaFuncaoComResposta(url,funcResposta)
125 {
126     var xmlhttp = getXmlHttp();
127     xmlhttp.onreadystatechange=function()
128     {
129         if (xmlhttp.readyState==4 && xmlhttp.status==200)
130         {
131             funcResposta(xmlhttp.responseText);
132         }
133     }
134     xmlhttp.open("GET", url ,true);
135     xmlhttp.send();
136 }
137
138 /**
139  *@param url : url da requisição, sem parametros
140  *@param params : parametros da requisição na forma
    "chave1=valor1&chave2=valor2..." sem o ? no início

```

141 ***@param** funcResposta : função que recebe a resposta (página) em caso de sucesso, pode ser nula

142 ***@param** funcErro: função que recebe CÓDIGO DE ERRO E RESPOSTA em caso de erro, precisa de dois parametros. função pode ser nula

```

143 */
144 function ajaxPostExecutaFuncaoComResposta(url,params,funcResposta, funcErro)
145 {
146     var xmlhttp = getXmlHttp();
147     xmlhttp.open("POST", url, true);
148
149     xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
150     xmlhttp.setRequestHeader("Content-length", params.length);
151     xmlhttp.setRequestHeader("Connection", "close");
152
153     xmlhttp.onreadystatechange = function() {
154         if(xmlhttp.readyState == 4) {
155             if(xmlhttp.status == 200 && funcResposta != null)
156                 funcResposta(xmlhttp.responseText);
157             else if(xmlhttp.status != 200 && funcErro != null)
158                 funcErro(xmlhttp.responseText, xmlhttp.status);
159         }
160     }
161     xmlhttp.send(params);
162 }
163
164 /**
165  * Retorna um array de componentes dado um array de IDs
166  */
167 function getByIdArray(arrayIds) {
168     var result = new Array();
169     for(var i = 0; i < arrayIds.length; i++) {
170         result[i] = document.getElementById(arrayIds[i]);
171     }
172     return result;
173 }
174
175 function selectTemValorValido(select) {
176     return !selectTemValorInvalido(select);

```

```
177 }
178
179 function selectTemValorInvalido(select) {
180     return select.selectedIndex == -1 ;
181 }
182
183 function updateFilename(file,filename)
184 {
185     var pathname = file.value;
186     var leafname= pathname.split("\\").pop().split("/").pop();
187     filename.value = leafname;
188 }
189
190 function enviarGet(url) {
191     var xreq = getXmlHttp()
192     xreq.open("GET", url,true);
193     xreq.send();
194 }
195
196 function noenter(e, func) {
197     var key;
198     if (window.event)
199         key = window.event.keyCode;
200     else
201         key = e.which;
202
203     if (key == 13) {
204         if(func!=null)
205             func();
206         return false;
207     }
208     else
209         return true;
210 }
211
212 //este método é usado quando preciso passar a referência da função abaixo como
parametro em outra
213 function executarNoAtorDidatico(giria) {
```

```
214 ambiente.executarNoAtorDidatico(giria);
215 }
216
217
218
219 //daqui para baixo avaliar criar um arquivo separado filemanager.js
220 function escreverArquivo(filename, conteudo) {
221     enviarGet("WriterServlet?content=" + encodeURIComponent(conteudo) + "&filename=" +
filename);
222 }
223
224 function criarDiretorio(name) {
225     enviarGet("ManageFileServlet?action=createDir&filename=" + name);
226 }
227
228 //Se arquivo não existir não faz nada. para forçar a escrita de um arquivo vazio
(mesmo sobrescrevendo outro arquivo), usar escreverArquivo com conteúdo vazio
229 function criarArquivo(name) {
230     enviarGet("ManageFileServlet?action=createFile&filename=" + name);
231 }
232
233 function copiarArquivo(origem,destino) {
234     enviarGet("CopiadorServlet?origem=" + origem + "&destino=" + destino);
235 }
```

TelisWeb/web/editordeapliques.js

```

1 function onloadAppliques() {
2     btCriarAplique = document.getElementById('btCriarAplique');
3     //editorDeAppliques = ambiente.obterEditorDeAppliques();
4     btCriarAplique.onclick = function() {
5
6         //TODO: Acho que deveria ter um prompt aqui, mas o comportamento do telis
hoje é adiciona diretamente como novoApliqueXX
7         //está hardcoded no adicionar, tem que mecher no código
8         //var nome = window.prompt("Forneça um nome para o novo aplique",
"novoAplique");
9         ambiente.obterEditorDeAppliques().adicionar();
10    }
11
12    btRemoverAplique = document.getElementById('btRemoverAplique');
13    btRemoverAplique.onclick = function() {
14        var sAppliques = document.getElementById('sAppliques');
15        enviarGet("RemoverAplique?aplique=" +
sAppliques[sAppliques.selectedIndex].text);
16        aplique.removerAplique();
17    }
18
19    sAppliques = document.getElementById('sAppliques');
20    sAppliques.onchange = function() {
21
22        ambiente.obterEditorDeAppliques().obterMediadorUI().fixarSelecionado(sAppliques.selectedIndex);
23        ambiente.obterEditorDeAppliques().mudancaDelItem();
24        ambiente.obterEditorDeAppliques().atualizar();
25        invalidateSModelosDisponiveis();
26        editorDeAppliques.obterMediadorUI().atualizarAtoresDoAplique();
27    }
28
29    sModelosDisponiveis = document.getElementById('sModelosDisponiveis');
30    sModelosDisponiveis.onchange = invalidateBtAddModelosAplique;
31
32    sModelosInstanciados = document.getElementById('sModelosInstanciados');

```

```

33 sModelosInstanciados.onChange = invalidateBtRemModelosAplique;
34
35 btAddModelosAplique = document.getElementById('btAddModelosAplique');
36 btAddModelosAplique.onclick = function() {
37
ambiente.obterEditorDeApliques().adicionarAtores(pegaselecionadosDeSelect(sModelosDisponiveis
));
38 }
39
40 btRemModelosAplique = document.getElementById('btRemModelosAplique');
41 btRemModelosAplique.onclick = function() {
42
ambiente.obterEditorDeApliques().removerAtores(pegaselecionadosDeSelect(sModelosInstanciados
));
43 }
44
45 btSalvarAplique = document.getElementById('btSalvarAplique');
46 btSalvarAplique.onclick = function() {
47     editorDeApliques.salvar();
48     persistirApliqueAtual();
49 }
50
51 btAtivarAplique = document.getElementById('btAtivarAplique');
52 btAtivarAplique.onclick = function() {
53     ambiente.obterEditorDeApliques().executar();
54 }
55
56 document.getElementById('btImportarAplique').onclick = function() {
57 //     try {
58 //         ambiente.importarApliqueWeb();
59 //     } catch(e) {
60 //         alert('Erro importando aplique');
61 //     }
62     aplique.importarAplique();
63 }
64
65 document.getElementById('btPublicarLocal').onclick = function() {

```

```

66     var apliquePublicar =
contextoDeEdicao.obterAplique(editorDeApliques.obterNomeDoApliqueAtual());
67     var assistente = new
aplique.Packages.web.publicacao.AssistentePublicacao(ambiente>window);
68     var dir = prompt("Por favor, informe um subdiretório para publicacao: ",
apliquePublicar.obterNome());
69     assistente.publicar(apliquePublicar, "apliques/" + dir);
70     alert("Aplique publicado em: " + appUrl + "/apliques/" + dir + "/" +
apliquePublicar.obterNome() + ".xml");
71 }
72
73 invalidateApliques();
74 }
75
76 function invalidateApliques() {
77     var arrayInvalidate =
getByIdArray(['textNomeAplique','btSalvarAplique','btAtivarAplique','btPublicarLocal','btPublicarWeb
','textLargura','textAltura','btRemoverAplique','sModelosDisponiveis']);
78     invalidateSelect(sApliques,document.getElementById('labelSApliques'), 'Apliques: ',
'Lista vazia de apliques', arrayInvalidate);
79     invalidateSModelosDisponiveis();
80     invalidateBtAddModelosAplique();
81     invalidateBtRemModelosAplique();
82     invalidateSModelosInstanciados();
83 }
84
85 //function persistirApliques() {
86 //     var apliques = contextoDeEdicao.obterApliques();
87 //     for(var i = 0; i < apliques.tamanho(); i++) {
88 //         persistirAplique(apliques.obterElemento(i));
89 //     }
90 //}
91
92 function persistirApliqueAtual() {
93     var url = appUrl + "/PersistirAplique";
94     var modeloSerial;
95
96     modeloSerial = aplique.serializaApliqueAtual();

```

```
97
ajaxPostExecutaFuncaoComResposta(url,modeloSerial,vazia,erroPersistindoModelo);
98 }
99
100 function carregarAppliques() {
101     document.getElementById('sAppliques').onchange();
102 }
103
104 function carregarAppliquesDeLista(listaAppliquesStr) {
105     var listaAppliques = listaAppliquesStr.split('\n');
106     for(var i = 0; i < listaAppliques.length; i++) {
107         var nome = listaAppliques[i];
108         if(nome != "") {
109             ajaxGetExecutaFuncaoComResposta(srcHome + "/appliques/" +
nome,carregarApplique);
110         }
111     }
112 }
113
114 function carregarApplique(apliqueProp) {
115     var props = appliqueProp.split('\n');
116
117     var i = 0;
118     line = props[i];
119     while(line!="modelos:") {
120         var prop = line.split('=');
121         switch(prop[0]) {
122             case "nome":
123                 var nome = prop[1];
124                 break;
125             case "largura":
126                 var largura = prop[1];
127                 break;
128             case "altura":
129                 var altura = prop[1];
130                 break;
131         }
132         i++;
```



```

133     line = props[i];
134 }
135 i++;
136 line = props[i];
137 var modelos = new Array();
138 while(line!="fim modelos") {
139     modelos.push(line);
140     i++;
141     line = props[i];
142 }
143
144 var novoAplique = new
aplique.Packages.br.ufsc.edugraf.telis.ambiente.Aplique(nome,largura,altura,contextoDeEdicao);
145
146 ambiente.obterContextoDeEdicao().adicionarAplique(novoAplique);
147 ambiente.obterContextoDeEdicao().obterApliques().notifyObservers();
148 ambiente.obterEditorDeApliques().atualizar();
149 novoAplique.adicionarAtores(modelos);
150 novoAplique.atores.notifyObservers();
151 }
152
153 function obterListaModelosInstanciados(atores) {
154     result = "";
155     for(var i = 0; i < atores.tamanho(); i++) {
156         result += atores.obterDescritor(i) + '\n'; //Pelo código hoje não terá diferença
entre isso e atores.obterElemento(i).toString()
157     }
158     return result;
159 }
160 //executado pelo invalidateSModelosDisponíveis ou pelo invalidateApliques
161 function invalidateBtAddModelosAplique() {
162     btAddModelosAplique=document.getElementById('btAddModelosAplique');
163     sModelosDisponiveis = document.getElementById('sModelosDisponiveis');
164     btAddModelosAplique.disabled = selectTemValorInvalido(sModelosDisponiveis) ||
selectTemValorInvalido(sApliques);//sem aplique ou sem modelo selecionado
165 }
166
167 //executado pelo *invalidateSModelosInstanciados* ou pelo invalidateApliques

```

```

168 function invalidateBtRemModelosAplique() {
169     btRemModelosAplique.disabled = selectTemValorInvalido(sApliques) ||
selectTemValorInvalido(document.getElementById('sModelosInstanciados'));
170 }
171
172 function invalidateSModelosDisponiveis() {
173     sApliques = document.getElementById('sApliques');
174     invalidateSelect(document.getElementById('sModelosDisponiveis'), null, null, null,
null, selectTemValorInvalido(sApliques));
175     invalidateBtAddModelosAplique();
176 }
177
178 function invalidateSModelosInstanciados() {
179     invalidateSelect(document.getElementById('sModelosInstanciados'));
180     invalidateBtRemModelosAplique();
181 }
182
183 //Avaliar se precisa, serve para o publicador do edugraf. não é usado para publicar no
mesmo servidor do telis acessível
184 function DialogPublicar(nomeDoAplique, nomeDoProjeto, caminho, usuario, senha) {
185     this.caminho = prompt("Caminho: ", caminho);
186     this.projeto = prompt("Projeto: ", nomeDoProjeto);
187     this.aplique = prompt("Aplique: ", nomeDoAplique);
188     this.usuario = prompt("Usuario: ", usuario);
189     this.senha = prompt("Senha: ", senha);
190 }
191
192 function factoryDialogPublicar(nomeDoAplique, nomeDoProjeto, caminho, usuario,
senha) {
193     return new DialogPublicar(nomeDoAplique, nomeDoProjeto, caminho, usuario,
senha);
194 }
195
196 function limpaSModelosDisponiveis() {
197     document.getElementById('sModelosDisponiveis').length = 0;
198 }

```

TelisWeb/web/editordeimagens.js

```
1 function carregarImagens()
2 {
3     combobox=document.getElementById('listaImagens');
4     for(i = 0; i<combobox.length; i = i + 1) {
5         imageToTelis(combobox[i].value, combobox[i].text);
6     }
7     return true;
8 }
9
10 function imageToTelis(relativeUrl, imageName)
11 {
12     imageUrl = appUrl + '/' + relativeUrl;
13     aplique.carregarImagem(imageUrl, imageName);
14 }
15
16 function onloadImagens()
17 {
18     thumb=document.getElementById('thumb');
19     combobox=document.getElementById('listaImagens');
20     file = document.getElementById('file');//[TODO] RETIRAR
21     filename = document.getElementById('filename');//[TODO] RETIRAR
22
23     combobox.onchange=function()
24     {
25         thumb.src= this.value;
26     }
27 }
28
29 function uploadAjax()
30 {
31     var file = document.getElementById('file');
32     var filename = document.getElementById('filename');
33     updateFilename(file,filename);
34     fileUpload(document.getElementById('CarregarImagens'),15,'<strong>Carregando
Imagem</strong>',uploadFinished);
35 }
```

```
36
37 function deletaImagem()
38 {
39     nomeImgDel = combobox[combobox.selectedIndex].text;
40     ajaxAtualizaComponente(document.getElementById('listaImagens'),
"ManageFileServlet?action=delete&filename=" + combobox.value +
"&response=UpdateImageServlet");
41     aplique.removerImagem(nomeImgDel);
42 }
43
44 function uploadFinished(data) {
45     document.getElementById('listaImagens').innerHTML = data;
46     carregarImagens();
47 }
```

TelisWeb/web/editordemodelos.js

```

1 /**
2  * Carrega modelos persistidos em arquivo e inclui no aplique com primitiva
incluirModelo ou incluirMolde
3  * a princípio é chamado só no início da aplicação, após o applet carregar
4  */
5  function carregarModelos() {
6      ajaxGetExecutaFuncaoComResposta(appUrl +
"/ObterAmbienteBean",setaAmbienteBeanNoApplet);
7
8      function setaAmbienteBeanNoApplet(ambienteBean) {
9          aplique.carregarAmbienteSeguro(ambienteBean);
10     }
11 }
12
13 function carregarModelosBD() {
14
15     ajaxGetExecutaFuncaoComResposta("geradoresComponentes/geradorDeCodigoDeModelosBD.jsp",
16     executarNoAtorDidatico);
17 }
18
19 function atualizaESelectionaAgendas() {os
20     atualizaAgendas();
21     if(editorDeModelos.obterAgendaLocalAtual()==null) {
22         sAgendas.selectedIndex = 0;
23         sAgendas.onChange();
24     }
25     else {
26         selecionaNomeEmSelect(editorDeModelos.obterNomeDaAgendaLocalAtual(),
sAgendas);
27         codAgenda.value = editorDeModelos.obterAgendaLocalAtual().obterTexto();
28     }
29 }
30
31 function invalidateBtAtivarModelos() {
32     if(document.getElementById('sModelos').options.selectedIndex == -1)
33         document.getElementById('btAtivarModelo').disabled = true;

```

32 `else if(editorDeModelos.modeloAtualÉMolde())` //a pergunta pode gerar excessão se
 não houver modelo atual

```

33        document.getElementById('btAtivarModelo').disabled = true;
34    else
35        document.getElementById('btAtivarModelo').disabled = false;
36 }
37
38 function onloadModelos() {
39    sModelos=document.getElementById('sModelos');
40    moldesText = document.getElementById('inputMoldesUsados');
41    sAgendas = document.getElementById('sAgendas');
42    codAgenda = document.getElementById('codAgenda');
43
44    sModelos.onChange=function()
45    {
46        function execDepois() {
47            editorDeModelos.fixarNomeDoModeloAtual(sModelos.value);
48
49            invalidateBtAtivarModelos();
50
51            //atualizaESelectionaAgendas();
52            atualizaAgendas();
53            //sAgendas.onChange();
54            sAgendas.selectedIndex = 0;
55            sAgendas.onChange();
56        }
57
58        perguntaSalvarMudancas(execDepois);
59    }
60
61    sAgendas.onChange=function()
62    {
63        function executarDepois() {
64            perguntaSalvaSeMudar();
65            editorDeModelos.fixarNomeDaAgendaLocalAtual(sAgendas.value);
66            codAgenda.value = editorDeModelos.obterAgendaLocalAtual().obterTexto();

```

```

67         var btRemoverAgenda =
document.getElementById('btRemoverAgenda');//isso é necessário porque a variável global pode
ainda não ter sido inicializada.
68         if(sAgendas.value == 'iniciar')
69             btRemoverAgenda.disabled = true;
70         else
71             btRemoverAgenda.disabled = false;
72     }
73
74     perguntaSalvarMudancas(executarDepois);
75
76 }
77 }//end onload
78
79 function excluirModelo() {
80     var mensagem = editorDeModelos.modeloAtualÉMolde() ? ('O molde ' +
editorDeModelos.obterNomeDoModeloAtual() + ' será removido. Ele será removido da lista de
molde de qualquer modelo que o utilizar. Deseja prosseguir?')
81     : ('Deseja remover o modelo selecionado? ' +
editorDeModelos.obterNomeDoModeloAtual());
82
83     function seDeletar() {
84         var modelo = sModelos[sModelos.selectedIndex].value;
85         editorDeModelos.removerModelo();
86         enviarGet("RemoverModelo?modelo=" + modelo);
87     }
88
89     perguntaSimples("Sim", "Não", seDeletar, null, "Confirmar exclusão", mensagem);
90 }
91
92 function perguntaSalvarMudancas(executarAposPergunta) {
93
94     try {
95         if(editorDeModelos.obterListaDeMoldesComoString() != moldesText.value ||
editorDeModelos.obterAgendaLocalAtual().obterTexto() != codAgenda.value ) {
96
97             function seSim() {
98                 editorDeModelos.salvarAgenda(codAgenda.value)

```

```

99         persisteModelo();
100        executarAposPergunta();
101    }
102    function seNao() {
103        executarAposPergunta();
104    }
105
106    perguntaSimples("Sim", "Não", seSim, seNao, "Salvar modificações?", 'Agenda
ou modelo modificado, deseja salvar antes de mudar?');
107
108    } else
109        executarAposPergunta();
110    } catch(e) {
111        executarAposPergunta();
112    }
113 }
114
115 function addModelo()
116 {
117
118     function executarDepoisDaPergunta() {
119         var sugestao = editorDeModelos.obterSugestãoDeNomeParaNovoModelo();
120         function aoResponder() {
121             resposta = obterResposta();
122             editorDeModelos.criarModelo(resposta);
123             //nesta altura o aplique, ouvindo os modelos, deve chamar de volta a página
para atualizar a lista de modelos e agendas!
124             editorDeModelos.fixarNomeDoModeloAtual(resposta);
125         }
126         dialogDePerguntaComCallback("Informar modelo", "Por favor, informe o nome
do novo modelo", sugestao, aoResponder);
127     }
128
129
130     perguntaSalvarMudancas(executarDepoisDaPergunta);
131 }
132
133 function addMolde() {

```



```

134 function executarDepoisDaPergunta() {
135     var sugestao = editorDeModelos.obterSugestãoDeNomeParaNovoMolde();
136     function aoResponder() {
137         model = obterResposta();
138         if(model != null) {
139             editorDeModelos.criarMolde(model);
140             //nesta altura o aplique, ouvindo os modelos, deve chamar de volta a
página para atualizar a lista de modelos e agendas!
141             editorDeModelos.fixarNomeDoModeloAtual(model);
142         }
143     }
144     dialogDePerguntaComCallback("Informar modelo", "Por favor, informe o nome
do novo molde", sugestao, aoResponder);
145 }
146
147 perguntaSalvarMudancas(executarDepoisDaPergunta);
148 }
149
150 function addAgenda()
151 {
152     function executarDepoisDaPergunta() {
153         var nAgenda = editorDeModelos.obterSugestãoDeNomeParaNovaAgenda();
154         function aoResponder() {
155             nAgenda = obterResposta();
156             if(nAgenda != null) {
157                 editorDeModelos.criarAgendaLocal(nAgenda);
158                 atualizaAgendas();
159                 sAgendas.value=nAgenda;
160                 editorDeModelos.fixarNomeDaAgendaLocalAtual(nAgenda);
161             }
162             btRemoverAgenda = document.getElementById('btRemoverAgenda');
163             if(sAgendas.value == 'iniciar')
164                 btRemoverAgenda.disabled = true;
165             else
166                 btRemoverAgenda.disabled = false;
167         }
168         dialogDePerguntaComCallback("Informe nome da agenda", "Por favor, informe o
nome da nova agenda", nAgenda, aoResponder);

```

```

169 }
170
171 perguntaSalvarMudancas(executarDepoisDaPergunta);
172 }
173
174 //Salvamento no Bd ocorre quando o applet invocar a função persisteModelos
175 function salvarAgenda()
176 {
177     aplique.salvarAgenda(codAgenda.value, moldesText.value, sAgendas.value);
178 }
179
180 /**
181  *esta função irá retornar uma linha a ser escrita no arquivo listaModelos, que
    armazena nomes dos modelos persistidos, se é modelo ou molde, e moldes moldados
182  */
183 function obtemLinhaProp(nome,modelo) {
184     var propModelos;
185     if(modelo.éMolde())
186         propModelos = "molde:";
187     else
188         propModelos = "modelo:";
189     propModelos += nome + ":" + modelo.obterMoldesComoString() + '\n';
190     return propModelos;
191 }
192
193 function atualizaModelosDeLista(list)
194 {
195     sModelos=document.getElementById('sModelos');
196     sModelos.length = 0;
197     if(list.size() > 0) {
198         for(i=0; i < list.size(); i++) {
199             var valor = list.get(i);
200             var texto;
201             if(contextoDeEdicao.obterModelo(valor).éMolde())
202                 texto = valor + " (MOLDE)";
203             else
204                 texto = valor + " (MODELO)";
205             sModelos.add(new Option(texto, valor));

```

```

206     }
207 }
208 invalidateSModelos();
209
210 selecionaNomeEmSelect(editorDeModelos.obterNomeDaAgendaLocalAtual(),
sAgendas)
211     atualizaAgendaAtual(editorDeModelos.obterNomeDaAgendaLocalAtual());
212
213     return true;
214 }
215
216 function atualizaModelos()
217 {
218     atualizaModelosDeLista(contextoDeEdicao.obterNomesDeModelos());
219 }
220
221 function atualizaAgendas() {
222     modeloAtual = sModelos.value;
223     if(modeloAtual != null) {
224         //editorDeModelos.fixarNomeDoModeloAtual(modeloAtual);//loop infinito
225
226         atualizaAgendasDeLista(editorDeModelos.obterModeloAtual().obterNomeDasAgendasLocais());
227     }
228     moldesText.value = editorDeModelos.obterListaDeMoldesComoString();
229 }
230 function atualizaAgendasDeLista(list)
231 {
232     sAgendas.length = 0;
233     for(i=0; i < list.size(); i++) {
234         sAgendas.add(new Option(list.get(i), list.get(i)));
235     }
236 }
237
238 function setDisabledModelosComponents(disabled) {
239     sModelos.disabled = disabled;
240     sAgendas.disabled = disabled;
241     moldesText = document.getElementById('inputMoldesUsados');

```

```

242   moldesText.disabled = disabled;
243   codAgenda = document.getElementById('codAgenda');
244   codAgenda.disabled = disabled;
245   var btRemoverModelo = document.getElementById('btRemoverModelo');
246   btRemoverModelo.disabled=disabled;
247   var btAddAgenda = document.getElementById('btAddAgenda');
248   btAddAgenda.disabled = disabled;
249   var btSalvarAgenda = document.getElementById('btSalvarAgenda');
250   btSalvarAgenda.disabled = disabled;
251   document.getElementById('btRemoverAgenda').disabled = disabled;
252   document.getElementById('btAtivarModelo').disabled = disabled;
253 }
254
255 /**
256  * Desabilita componentes quando lista fica vazia e habilita-os de volta quando não
257  */
258 function invalidateSModelos() {
259   sModelos = document.getElementById('sModelos');
260   sAgendas = document.getElementById('sAgendas');
261   if(sModelos.length == 0) {
262     sAgendas.length = 0;
263     sAgendas.add(new Option()); //Workaround para Bug Orca+firefox com lista vazia
264     setDisabledModelosComponents(true);
265   } else {
266     if(sModelos.disabled)
267       setDisabledModelosComponents(false);
268
269     var modeloAtual = editorDeModelos.obterNomeDoModeloAtual();
270     var modelos = contextoDeEdicao.obterNomesDeModelos();
271
272     if(stringEstaEmLista(modelos,modeloAtual)) {
273       if(modeloAtual!=null)
274         atualizaModeloAtual(modeloAtual);
275     } else {
276       sModelos.selectedIndex=0;
277       sModelos.onChange();
278     }
279   }

```

```

280 }
281
282 function atualizaModeloAtual(nome) {
283     selecionaNomeEmSelect(nome,sModelos);
284     atualizaESelecionaAgendas();
285     listaAtualizada=true
286 }
287
288 function atualizaAgendaAtual(nome) {
289     selecionaNomeEmSelect(nome,sAgendas);
290     codAgenda.value = editorDeModelos.obterAgendaLocalAtual().obterTexto();
291 }
292
293 function perguntaSalvaSeMudar() {
294     try {
295         if(editorDeModelos.obterListaDeMoldesComoString() != moldesText.value ||
editorDeModelos.obterAgendaLocalAtual().obterTexto() != codAgenda.value ) {
296             if(confirm('Agenda ou modelo modificado, deseja salvar antes de mudar?')) {
297                 editorDeModelos.salvarAgenda(codAgenda.value)
298                 persisteModelo();
299             }
300         }
301     } catch(e) {}
302 }
303
304 function removerAgenda() {
305
306
307     function seDeletar() {
308         var agenda = sAgendas.value;
309         editorDeModelos.removerAgendaLocal();
310         var modelo = sModelos[sModelos.selectedIndex].value;
311         atualizaAgendas();
312         sAgendas.selectedIndex = 0;
313         //sAgendas.onChange();
314         editorDeModelos.fixarNomeDaAgendaLocalAtual(sAgendas.value);
315         codAgenda.value = editorDeModelos.obterAgendaLocalAtual().obterTexto();
316         var btRemoverAgenda = document.getElementById('btRemoverAgenda');

```

```

317     if(sAgendas.value == 'iniciar')
318         btRemoverAgenda.disabled = true;
319     else
320         btRemoverAgenda.disabled = false;
321 }
322
323 var mensagem = "Remover a agenda local selecionada " +
editorDeModelos.obterNomeDaAgendaLocalAtual() + '?';
324
325 perguntaSimples("Sim", "Não", seDeletar, null, "Confirmar exclusão", mensagem);
326 }
327
328
329 /**
330  * Função a ser chamada pela applet depois de executado persistirRevisaoBd
331  * Se nome da função mudar, precisa mudar na applet!
332  */
333 function persistirRevisaoSerializada(param) {
334     var url = appUrl + "/persistirRevisao";
335
ajaxPostExecutaFuncaoComResposta(url,param,persistiuRevisao,erroPersistindoRevisao);
336 }
337
338 function erroPersistindoModelo(conteudoPaginaResposta, codigoErro) {
339     dialogoDeAlert("Servidor retornou erro de código " + codigoErro + " ao
salvar<br/>" + conteudoPaginaResposta , "erro");
340 }
341
342 function vazia() {
343
344 }
345 function persisteModelo() {
346     var url = appUrl + "/PersistirModelo";
347     var modeloSerial;
348     modeloSerial = aplique.serializaModeloAtual();
349
ajaxPostExecutaFuncaoComResposta(url,modeloSerial,vazia,erroPersistindoModelo);
350 }

```

TelisWeb/web/editordesons.js

```
1 function carregarSons()
2 {
3   sSons = document.getElementById('sSons');
4   for(i = 0; i<sSons.length; i = i + 1) {
5     soundToTelis(sSons[i].value, sSons[i].text);
6   }
7   invalidateSom();
8   return true;
9 }
10
11 function soundToTelis(relativeUrl, soundName)
12 {
13   soundUrl = appUrl + '/' + relativeUrl;
14   aplique.carregarSom(soundUrl, soundName);
15 }
16
17 function playAudio() {
18   playpause = document.getElementById('playpause');
19   audio.play();
20   playpause.setAttribute('title', 'Pause');
21 }
22
23 function onloadSons() {
24   sSons= document.getElementById('sSons');
25   audio = document.getElementById('amostraSom');
26   playpause = document.getElementById('playpause');
27
28   sSons.onchange = function() {
29     document.getElementById('amostraSom').src = this.value;
30   }
31
32   sSons.onkeypress = function(e) {
33     var key;
34     if (window.event)
35       key = window.event.keyCode;
36     else
```

```

37     key = e.which;
38
39     if (key == 13) {
40         playAudio();
41         return false;
42     }
43     else
44         return true;
45 }
46 }
47
48 function uploadSndAjax()
49 {
50     var file = document.getElementById('fileSom');
51     var filename = document.getElementById('filenameSom');
52     var uploadSndFinished = function(data) {
53         document.getElementById('sSons').innerHTML = data;
54         carregarSons();
55     }
56     updateFilename(file,filename);
57     fileUpload(document.getElementById('formSons'),15,'<strong>Carregando
Sons</strong>',uploadSndFinished);
58     if(sSons.disabled) {
59         sSons.remove(0);
60         sSons.disabled = false;
61         document.getElementById('btRemoverSom').disabled = false;
62     }
63 }
64 }
65
66 function deletaSom()
67 {
68     nomeSomDel = sSons[sSons.selectedIndex].text;
69     ajaxAtualizaComponenteInvalida(sSons, "ManageFileServlet?
action=delete&filename=" + sSons.value +
"&response=geradoresComponentes/optionsSelectSom.jsp", invalidateSom);
70     aplique.removerSom(nomeSomDel); //[TODO]
71     if(sSons.length == 0) {

```



```
72     sSons.add(new Option());
73     sSons.disabled = true;
74     document.getElementById('btRemoverSom').disabled = true;
75 }
76 }
77
78 function invalidateSom()
79 {
80     if(sSons.length == 0) {
81         sSons.add(new Option());
82         sSons.disabled = true;
83         document.getElementById('btRemoverSom').disabled = true;
84         playpause.disabled= true;
85     } else {
86         for(i = 0; i < sSons.length; i++) {
87             if(sSons[0].value == "") {
88                 sSons.remove(i);
89                 break;
90             }
91         }
92         sSons.enabled = false;
93         document.getElementById('btRemoverSom').disabled = false;
94         playpause.disabled=false;
95     }
96 }
```

TelisWeb/web/interpretador.js

```
1 function onloadInterp() {
2     activeElem = document.getElementById('interpretadorDiv');
3     hideElem(imagensDiv);
4     hideElem(modelosDiv);
5     hideElem(sonsDiv);
6     hideElem(apliquesDiv);
7     aplicue = document.getElementById('umModelo');
8
9     sPilha = document.getElementById('sPilha');
10    sAtores = document.getElementById('sAtores');
11    sAtores.onchange = function() {
12        aplicue.selecionarAtor(document.getElementById('sAtores').value);
13    }
14
15    document.getElementById('entradaInterpretador').onsubmit = function () {
16        executarTextoDoInput();
17        return false;
18    }
19
20    document.getElementsByName('interpretador')[0].onsubmit = function () {
21        return false;
22    }
23 }
24
25 function executarTextoDoInput() {
26     if(document.getElementById('entrada').value!="") // para bug do internet explorer 8
27         sendText(document.getElementById('entrada').value);
28 }
29
30 function sendText(texto) {
31     ambiente.executarNoAtorDidatico(texto);
32     document.getElementById('entrada').focus();
33     document.getElementById('entrada').select();
34 }
35
36 function tratarEnterInterpretador(e) {
```

```
37  var key;
38  if (window.event)
39      key = window.event.keyCode;
40  else
41      key = e.which;
42
43  if (key == 13) {
44      executarTextoDoInput();
45      return false;
46  }
47  else
48      return true;
49 }
50
51 function invalidatePainelInterp() {
52
invalidateSelect(document.getElementById('sPilha'),document.getElementById('labelPilha'), "Pilha:
", "Pilha Vazia");
53     invalidateSelect(document.getElementById('sAtores'),
document.getElementById('labelSAtores'), "Atores:", "Sem atores");
54 }
```

TelisWeb/src/java/login/ValidaLogin.java

```

1 package login;
2
3 import java.io.IOException;
4 import javax.servlet.RequestDispatcher;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import utils.UserDAO;
10
11 /**
12  *
13  * @author felipe
14  */
15 public class ValidaLogin extends HttpServlet {
16
17     /**
18      * Processes requests for both HTTP GET and
19      * POST methods.
20      * @param request servlet request
21      * @param response servlet response
22      * @throws ServletException if a servlet-specific error occurs
23      * @throws IOException if an I/O error occurs
24      */
25     protected void processRequest(HttpServletRequest request, HttpServletResponse
26     response)
27         throws ServletException, IOException {
28         response.setContentType("text/html;charset=UTF-8");
29
30         String user = request.getParameter("j_username");
31         RequestDispatcher rd;
32         if(valida(user, request.getParameter("j_password"))) {
33             request.getSession().setAttribute("user", user);
34             rd = request.getRequestDispatcher("/IniciaSessao");
35         }
36         else

```

```

35     rd = request.getRequestDispatcher("/login.jsp?user=" + user);
36     rd.forward(request,response);
37 }
38
39 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
40 /**
41  * Handles the HTTP <code>GET</code> method.
42  * @param request servlet request
43  * @param response servlet response
44  * @throws ServletException if a servlet-specific error occurs
45  * @throws IOException if an I/O error occurs
46  */
47 @Override
48 protected void doGet(HttpServletRequest request, HttpServletResponse response)
49     throws ServletException, IOException {
50     processRequest(request, response);
51 }
52
53 /**
54  * Handles the HTTP <code>POST</code> method.
55  * @param request servlet request
56  * @param response servlet response
57  * @throws ServletException if a servlet-specific error occurs
58  * @throws IOException if an I/O error occurs
59  */
60 @Override
61 protected void doPost(HttpServletRequest request, HttpServletResponse response)
62     throws ServletException, IOException {
63     processRequest(request, response);
64 }
65
66 /**
67  * Returns a short description of the servlet.
68  * @return a String containing servlet description
69  */
70 @Override
71 public String getServletInfo() {

```

```
72     return "Short description";
73 }// </editor-fold>
74
75 private boolean valida(String user, String pass) {
76     UserDao userDao = new UserDao(getServletContext());
77
78     boolean validou = userDao.validaLogin(user, pass);
79     return validou;
80 }
81 }
```

TelisWeb/src/java/persistencia/ApliqueDAO.java

```

1 package persistencia;
2
3 import java.io.Serializable;
4 import java.util.List;
5 import javax.persistence.*;
6 import portablebeans.ApliqueBean;
7 /**
8  *
9  * @author felipe
10 */
11 public class ApliqueDAO implements Serializable{
12     private static final String PERSISTENCE_UNIT_NAME = "TelisWebPU";
13     private EntityManagerFactory factory;
14     private String usuario;
15
16     private static final long serialVersionUID = 1L;
17
18     public ApliqueDAO(String usuario) {
19         this.usuario = usuario;
20         factory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
21     }
22
23     private ApliqueBean getApliqueDeNome(EntityManager em, String nome) {
24         Query q = em.createQuery("SELECT m FROM ApliqueBean m WHERE m.usuario =
25 "" + usuario + "" and m.nome="" + nome + """).setMaxResults(1);
26         try {
27             return (ApliqueBean) q.getSingleResult();
28         } catch (javax.persistence.NoResultException e) {
29             return null;
30         }
31
32     public void removerAplique(String nomeAplique) {
33         EntityManager em = factory.createEntityManager();
34         em.getTransaction().begin();
35         ApliqueBean aplique = getApliqueDeNome(em, nomeAplique);

```

```

36     if(aplique == null)
37         return;
38     em.remove(aplique);
39     em.getTransaction().commit();
40     em.close();
41 }
42
43 public void persistirAplique(ApliqueBean aplicue) {
44     aplicue.setUsuario(usuario); //Nunca grava binário de outro usuário
45     EntityManager em = factory.createEntityManager();
46     em.getTransaction().begin();
47     ApliqueBean aplicueAntigo = getApliqueDeNome(em, aplicue.getNome());
48     if(aplicueAntigo != null)
49         aplicue.setId(aplicueAntigo.getId());
50     ModeloDAO modeloDAO = new ModeloDAO(usuario);
51     for(String nomeDep: aplicue.getModelosStr())
52         aplicue.addModelo(modeloDAO.getModeloDeNome(em, nomeDep));
53     em.merge(aplicue);
54     em.getTransaction().commit();
55     em.close();
56 }
57
58 public List<ApliqueBean> obterListaDeApliques() {
59     EntityManager em = factory.createEntityManager();
60     em.getTransaction().begin();
61     Query q = em.createQuery("SELECT a FROM ApliqueBean a WHERE a.usuario = "
+ usuario + "'");
62     return q.getResultList();
63 }
64 }

```


TelisWeb/src/java/persistencia/ModeloDAO.java

```
1 package persistencia;
2
3 import java.io.Serializable;
4 import java.util.ArrayList;
5 import java.util.HashSet;
6 import java.util.List;
7 import java.util.Set;
8 import javax.persistence.Query;
9 import javax.persistence.EntityManager;
10 import javax.persistence.EntityManagerFactory;
11 import javax.persistence.Persistence;
12 import portablebeans.AgendaBean;
13 import portablebeans.ModeloBean;
14
15 /**
16  *
17  * @author felipe
18  */
19 public class ModeloDAO implements Serializable{
20     private static final String PERSISTENCE_UNIT_NAME = "TelisWebPU";
21     private EntityManagerFactory factory;
22     private String usuario;
23
24     private static final long serialVersionUID = 1L;
25
26     public ModeloDAO(String usuario) {
27         factory = Persistence.createEntityManagerFactory(PERSISTENCE_UNIT_NAME);
28         this.usuario = usuario;
29     }
30
31     public ModeloBean getModeloDeNome(String nome) {
32         EntityManager em = factory.createEntityManager();
33         em.getTransaction().begin();
34         ModeloBean result = getModeloDeNome(em, nome);
35         em.close();
36         return result;
```

```

37     }
38
39     private void converteDependenciasParaBeans(ModeloBean modeloNovo,
EntityManager em) {
40         //Converter lista de moldes de string para beans
41         for(String nomeMolde: modeloNovo.getMoldado()) {
42             ModeloBean molde = getModeloDeNome(em, nomeMolde);
43             if(molde == null) {
44                 molde = new ModeloBean();
45                 molde.setNome(nomeMolde);
46                 molde.setUsuario(usuario);
47                 molde.setEhMolde(true);
48                 AgendaBean agenda = new AgendaBean();
49                 agenda.setNome("iniciar");
50                 agenda.setCodigo("");
51             }
52             modeloNovo.addMolde(molde);
53         }
54     }
55
56     public ModeloBean getModeloDeNome(EntityManager em, String nome) {
57         Query q = em.createQuery("SELECT m FROM ModeloBean m WHERE m.usuario
= '' + usuario + '' and m.nome='' + nome + ''").setMaxResults(1);
58         try {
59             return (ModeloBean) q.getSingleResult();
60         } catch (javax.persistence.NoResultException e) {
61             return null;
62         }
63     }
64
65     public Set<ModeloBean> getTodosOsModelos() {
66         return new HashSet<ModeloBean>(getModelosComoLista());
67     }
68
69     public List<ModeloBean> getModelosComoLista() {
70         List<ModeloBean> result = getModelosComoListaBd();
71         for(ModeloBean modelo: result)
72             modelo.criaListaMoldesStr();

```

```

73     return result;
74 }
75
76 private List<ModeloBean> getModelosComoListaBd() {
77     EntityManager em = factory.createEntityManager();
78     em.getTransaction().begin();
79     Query q = em.createQuery("SELECT m FROM ModeloBean m WHERE m.usuario
= '' + usuario + ''");
80     try {
81         return q.getResultList();
82     } catch (javax.persistence.NoResultException e) {
83         return new ArrayList<ModeloBean>(0);
84     } finally {
85         em.close();
86     }
87 }
88
89 public void persistirModelo(ModeloBean modeloNovo) {
90     modeloNovo.setUsuario(usuario);
91     EntityManager em = factory.createEntityManager();
92     em.getTransaction().begin();
93     ModeloBean modeloAntigo = getModeloDeNome(em, modeloNovo.getNome());
94
95     boolean modeloMudou = modeloNovo.atualizaIds(modeloAntigo);
96     if(!modeloMudou)
97         return;
98
99     if(modeloAntigo!=null)
100         modeloNovo.setId(modeloAntigo.getId());
101     converteDependenciasParaBeans(modeloNovo, em);
102
103     em.merge(modeloNovo);
104     em.getTransaction().commit();
105     em.close();
106 }
107
108 public void removeModelo(String nome) {
109     EntityManager em = factory.createEntityManager();

```

```
110     em.getTransaction().begin();
111     ModeloBean modelo = getModeloDeNome(em, nome);
112     if(modelo == null)
113         return;
114     em.remove(modelo);
115     em.getTransaction().commit();
116     em.close();
117 }
118 }
```

TelisWeb/src/java/persistencia/ObterAmbienteBean.java

```

1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package persistencia;
6
7 import java.io.IOException;
8 import java.io.PrintWriter;
9 import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import portablebeans.AmbienteBean;
14 import portablebeans.Serializador;
15
16 /**
17  *
18  * @author felipe
19  */
20 @WebServlet(name = "ObterAmbienteBean", urlPatterns = {"/ObterAmbienteBean"})
21 public class ObterAmbienteBean extends HttpServlet {
22
23     /**
24      * Processes requests for both HTTP GET and
25      * POST methods.
26      *
27      * @param request servlet request
28      * @param response servlet response
29      * @throws ServletException if a servlet-specific error occurs
30      * @throws IOException if an I/O error occurs
31      */
32     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
33         throws ServletException, IOException {
34         response.setContentType("text/html;charset=UTF-8");
35         String user = (String) request.getSession().getAttribute("user");
36         ApliqueDAO aplicacaoDAO = new ApliqueDAO(user);

```

```

35     ModeloDAO modeloDAO = new ModeloDAO(user);
36     AmbienteBean ambiente = new
AmbienteBean(modeloDAO.getModelosComoLista(), apliqueDAO.obterListaDeApliques());
37     String result = Serializador.serializar(ambiente);
38     PrintWriter out = response.getWriter();
39     try {
40         out.print(result);
41         /* TODO output your page here
42         out.println("<html>");
43         out.println("<head>");
44         out.println("<title>Servlet ObterAmbienteBean</title>");
45         out.println("</head>");
46         out.println("<body>");
47         out.println("<h1>Servlet ObterAmbienteBean at " + request.getContextPath ()
+ "</h1>");
48         out.println("</body>");
49         out.println("</html>");
50         */
51     } finally {
52         out.close();
53     }
54 }
55
56 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
57 /**
58  * Handles the HTTP <code>GET</code> method.
59  * @param request servlet request
60  * @param response servlet response
61  * @throws ServletException if a servlet-specific error occurs
62  * @throws IOException if an I/O error occurs
63  */
64 @Override
65 protected void doGet(HttpServletRequest request, HttpServletResponse response)
66     throws ServletException, IOException {
67     processRequest(request, response);
68 }
69

```

```
70  /**
71   * Handles the HTTP <code>POST</code> method.
72   * @param request servlet request
73   * @param response servlet response
74   * @throws ServletException if a servlet-specific error occurs
75   * @throws IOException if an I/O error occurs
76   */
77  @Override
78  protected void doPost(HttpServletRequest request, HttpServletResponse response)
79      throws ServletException, IOException {
80      processRequest(request, response);
81  }
82
83  /**
84   * Returns a short description of the servlet.
85   * @return a String containing servlet description
86   */
87  @Override
88  public String getServletInfo() {
89      return "Short description";
90  } // </editor-fold>
91 }
```

TelisWeb/src/java/persistencia/PersistirAplique.java

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package persistencia;
6
7  import java.io.IOException;
8  import java.io.PrintWriter;
9  import java.util.logging.Level;
10 import java.util.logging.Logger;
11 import javax.servlet.ServletException;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15 import portablebeans.ApliqueBean;
16 import portablebeans.Serializador;
17
18 /**
19 *
20 * @author felipe
21 */
22 public class PersistirAplique extends HttpServlet {
23
24     /**
25      * Processes requests for both HTTP GET and
26      * POST methods.
27      *
28      * @param request servlet request
29      * @param response servlet response
30      * @throws ServletException if a servlet-specific error occurs
31      * @throws IOException if an I/O error occurs
32      */
33     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
34         throws ServletException, IOException {
35         response.setContentType("text/html;charset=UTF-8");
36         String dado = request.getParameter("dado");

```



```

35     String user = (String) request.getSession().getAttribute("user");
36     PrintWriter out = response.getWriter();
37     try {
38         ApliqueBean aplique = (ApliqueBean) Serializador.deserializar(dado);
39         ApliqueDAO apliqueDAO = new ApliqueDAO(user);
40         apliqueDAO.persistirAplique(aplique);
41     } catch (Exception e) {
42         log(e.getMessage());
43         Logger.getLogger(PersistirModelo.class.getName()).log(Level.SEVERE, null, e);
44     } finally {
45         out.close();
46     }
47 }
48
49 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
50 /**
51  * Handles the HTTP <code>GET</code> method.
52  * @param request servlet request
53  * @param response servlet response
54  * @throws ServletException if a servlet-specific error occurs
55  * @throws IOException if an I/O error occurs
56  */
57 @Override
58 protected void doGet(HttpServletRequest request, HttpServletResponse response)
59     throws ServletException, IOException {
60     processRequest(request, response);
61 }
62
63 /**
64  * Handles the HTTP <code>POST</code> method.
65  * @param request servlet request
66  * @param response servlet response
67  * @throws ServletException if a servlet-specific error occurs
68  * @throws IOException if an I/O error occurs
69  */
70 @Override
71 protected void doPost(HttpServletRequest request, HttpServletResponse response)

```

```
72     throws ServletException, IOException {
73     processRequest(request, response);
74 }
75
76 /**
77  * Returns a short description of the servlet.
78  * @return a String containing servlet description
79  */
80 @Override
81 public String getServletInfo() {
82     return "Short description";
83 } // </editor-fold>
84 }
```

TelisWeb/src/java/persistencia/PersistirModelo.java

```

1 package persistencia;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import portablebeans.ModeloBean;
10 import portablebeans.Serializador;
11 import java.util.logging.Level;
12 import java.util.logging.Logger;
13
14 /**
15  *
16  * @author felipe
17  */
18 public class PersistirModelo extends HttpServlet {
19
20     /**
21      * Processes requests for both HTTP GET and
22      * POST methods.
23      * @param request servlet request
24      * @param response servlet response
25      * @throws ServletException if a servlet-specific error occurs
26      * @throws IOException if an I/O error occurs
27      */
28     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
29         throws ServletException, IOException {
30         response.setContentType("text/html;charset=UTF-8");
31         String dado = request.getParameter("dado");
32
33         PrintWriter out = response.getWriter();
34         try {
35             ModeloBean modelo = (ModeloBean) Serializador.deserializar(dado);

```

```

35      ModeloDAO modeloDAO = new ModeloDAO((String)
request.getSession().getAttribute("user"));
36      modeloDAO.persistirModelo(modelo);
37      } catch(ClassNotFoundException e) {
38          log(e.getMessage());
39          Logger.getLogger(PersistirModelo.class.getName()).log(Level.SEVERE, null, e);
40          response.sendError(500);
41          e.printStackTrace(out);
42      } finally {
43          out.close();
44      }
45  }
46
47  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
48  /**
49   * Handles the HTTP <code>GET</code> method.
50   * @param request servlet request
51   * @param response servlet response
52   * @throws ServletException if a servlet-specific error occurs
53   * @throws IOException if an I/O error occurs
54   */
55  @Override
56  protected void doGet(HttpServletRequest request, HttpServletResponse response)
57      throws ServletException, IOException {
58      processRequest(request, response);
59  }
60
61  /**
62   * Handles the HTTP <code>POST</code> method.
63   * @param request servlet request
64   * @param response servlet response
65   * @throws ServletException if a servlet-specific error occurs
66   * @throws IOException if an I/O error occurs
67   */
68  @Override
69  protected void doPost(HttpServletRequest request, HttpServletResponse response)
70      throws ServletException, IOException {

```

```
71     processRequest(request, response);
72 }
73
74 /**
75  * Returns a short description of the servlet.
76  * @return a String containing servlet description
77  */
78 @Override
79 public String getServletInfo() {
80     return "Short description";
81 }// </editor-fold>
82 }
```

TelisWeb/src/java/persistencia/RemoverAplique.java

```

1 package persistencia;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import java.util.logging.Level;
6 import java.util.logging.Logger;
7 import javax.servlet.ServletException;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 /**
13  *
14  * @author felipe
15  */
16 public class RemoverAplique extends HttpServlet {
17
18     /**
19      * Processes requests for both HTTP GET and
20      * POST methods.
21      * @param request servlet request
22      * @param response servlet response
23      * @throws ServletException if a servlet-specific error occurs
24      * @throws IOException if an I/O error occurs
25      */
26     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
27         throws ServletException, IOException {
28         response.setContentType("text/html;charset=UTF-8");
29         String nomeAplique = request.getParameter("aplique");
30
31         PrintWriter out = response.getWriter();
32         try {
33             ApliqueDAO apliqueDAO = new ApliqueDAO((String)
request.getSession().getAttribute("user"));
34             apliqueDAO.removerAplique(nomeAplique);

```

```

34     } catch(Exception e) {
35         log(e.getMessage());
36         Logger.getLogger(PersistirModelo.class.getName()).log(Level.SEVERE, null, e);
37     } finally {
38         out.close();
39     }
40 }
41
42 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
43 /**
44  * Handles the HTTP <code>GET</code> method.
45  * @param request servlet request
46  * @param response servlet response
47  * @throws ServletException if a servlet-specific error occurs
48  * @throws IOException if an I/O error occurs
49  */
50 @Override
51 protected void doGet(HttpServletRequest request, HttpServletResponse response)
52     throws ServletException, IOException {
53     processRequest(request, response);
54 }
55
56 /**
57  * Handles the HTTP <code>POST</code> method.
58  * @param request servlet request
59  * @param response servlet response
60  * @throws ServletException if a servlet-specific error occurs
61  * @throws IOException if an I/O error occurs
62  */
63 @Override
64 protected void doPost(HttpServletRequest request, HttpServletResponse response)
65     throws ServletException, IOException {
66     processRequest(request, response);
67 }
68
69 /**
70  * Returns a short description of the servlet.

```

```
71  * @return a String containing servlet description
72  */
73  @Override
74  public String getServletInfo() {
75      return "Short description";
76  }// </editor-fold>
77 }
```


TelisWeb/src/java/persistencia/RemoverModelo.java

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package persistencia;
6
7  import java.io.IOException;
8  import javax.servlet.ServletException;
9  import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 /**
14  *
15  * @author felipe
16  */
17 public class RemoverModelo extends HttpServlet {
18
19     /**
20      * Processes requests for both HTTP GET and
21      * POST methods.
22      * @param request servlet request
23      * @param response servlet response
24      * @throws ServletException if a servlet-specific error occurs
25      * @throws IOException if an I/O error occurs
26      */
27     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
28         throws ServletException, IOException {
29         response.setContentType("text/html;charset=UTF-8");
30
31         String modelo = request.getParameter("modelo");
32         ModeloDAO modeloDAO = new ModeloDAO((String)
request.getSession().getAttribute("user"));
33         modeloDAO.removeModelo(modelo);
34     }

```

```

34
35 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
36 /**
37  * Handles the HTTP <code>GET</code> method.
38  * @param request servlet request
39  * @param response servlet response
40  * @throws ServletException if a servlet-specific error occurs
41  * @throws IOException if an I/O error occurs
42  */
43 @Override
44 protected void doGet(HttpServletRequest request, HttpServletResponse response)
45     throws ServletException, IOException {
46     processRequest(request, response);
47 }
48
49 /**
50  * Handles the HTTP <code>POST</code> method.
51  * @param request servlet request
52  * @param response servlet response
53  * @throws ServletException if a servlet-specific error occurs
54  * @throws IOException if an I/O error occurs
55  */
56 @Override
57 protected void doPost(HttpServletRequest request, HttpServletResponse response)
58     throws ServletException, IOException {
59     processRequest(request, response);
60 }
61
62 /**
63  * Returns a short description of the servlet.
64  * @return a String containing servlet description
65  */
66 @Override
67 public String getServletInfo() {
68     return "Short description";
69 } // </editor-fold>
70 }

```

TelisWeb/src/java/persistencia/UserDAO.java

```

1 package persistencia;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.logging.Level;
9 import java.util.logging.Logger;
10 import javax.servlet.ServletContext;
11
12 /**
13  *
14  * @author felipe
15  */
16 public class UserDAO {
17     private ServletContext config;
18     private Connection connection;
19     private Statement statement;
20
21     private static final int TAMANHO_MINIMO_USUARIO = 4;
22     private static final int TAMANHO_MINIMO_SENHA = 4;
23
24     public UserDAO(ServletContext config) {
25         this.config = config;
26     }
27
28     private void criaStatement() throws ClassNotFoundException, SQLException {
29         Class.forName( config.getInitParameter( "databaseDriver" ) );
30         connection = DriverManager.getConnection(
31             config.getInitParameter( "databaseName" ),
32             config.getInitParameter( "username" ),
33             config.getInitParameter( "password" ) );
34         statement = connection.createStatement();
35     }
36

```

```

37 private void fechaStatement() {
38     try {
39         if(connection != null)
40             connection.commit();
41     } catch (SQLException ex) {
42         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
43     } finally {
44         try {
45             if(statement != null)
46                 statement.close();
47         } catch (SQLException ex) {
48             Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
49         } finally {
50             try {
51                 if(connection != null)
52                     connection.close();
53             } catch (SQLException ex) {
54                 Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
55             }
56         }
57     }
58 }
59
60 public boolean validaLogin(String usuario, String senha) {
61     try {
62         criaStatement();
63         ResultSet resultado = statement.executeQuery("SELECT * FROM telis.usuarios
64 WHERE usuario=" + usuario + " AND senha = +" + senha + "");
65         if(resultado == null)
66             return false;
67         return resultado.next();//true se encontrou algo, falso se retornou vazio
68     } catch (ClassNotFoundException ex) {
69         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
70         return false;
71     } catch (SQLException ex) {
72         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
73         return false;
74     } finally {

```

```

74     fechaStatement();
75 }
76 }
77
78 //C
79 public void CadastrarUsuario(String usuario, String senha) throws
UsuarioExisteException, LoginPequenoException{
80     if(usuario.length() < TAMANHO_MINIMO_USUARIO || senha.length() <
TAMANHO_MINIMO_SENHA)
81         throw new LoginPequenoException(usuario, senha);
82     try {
83         criaStatement();
84         statement.executeUpdate("INSERT INTO usuarios (usuario,senha) VALUES (\\"
+ usuario + "\",\\" + senha + "\")");
85         statement.executeUpdate("INSERT INTO papeis (usuario,papel) VALUES (\\" +
usuario + "\",\\"usuario\")");
86     } catch (com.mysql.jdbc.exceptions.MySQLIntegrityConstraintViolationException
e) {
87         throw new UsuarioExisteException(usuario);
88     } catch (ClassNotFoundException ex) {
89         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
90     } catch (SQLException ex) {
91         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
92     } finally {
93         fechaStatement();
94     }
95 }
96
97
98 //U
99 public void mudarSenha (String usuario, String senhaNova) throws
LoginPequenoException {
100     if(senhaNova.length() < TAMANHO_MINIMO_SENHA)
101         throw new LoginPequenoException(usuario, senhaNova);
102     try {
103         criaStatement();
104         statement.executeUpdate("UPDATE usuarios set senha=\\" + senhaNova +
\\" WHERE usuario=\\" + usuario + "\");

```

```

105     } catch (ClassNotFoundException ex) {
106         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
107     } catch (SQLException ex) {
108         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
109     } finally {
110         fechaStatement();
111     }
112 }
113
114 //D
115 public void removerUsuario(String usuario) {
116     try {
117         criaStatement();
118         statement.executeUpdate("DELETE FROM telis.usuarios WHERE usuario=\"\" +
usuario + "\"");
119     } catch (ClassNotFoundException ex) {
120         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
121     } catch (SQLException ex) {
122         Logger.getLogger(UserDAO.class.getName()).log(Level.SEVERE, null, ex);
123     } finally {
124         fechaStatement();
125     }
126 }
127
128 public class UsuarioExisteException extends Exception {
129     String usuario;
130     public UsuarioExisteException(String usuario) {
131         super("Usuario " + usuario + "existe, impossível cadastrar novo.");
132         this.usuario = usuario;
133     }
134 }
135
136 public class LoginPequenoException extends Exception {
137
138     private boolean usuarioPequeno, senhaPequena;
139
140     private LoginPequenoException(String usuario, String senha) {
141         super("Usuario ou senha pequenos demais");

```

```
142     usuarioPequeno = usuario.length() < TAMANHO_MINIMO_USUARIO;  
143     senhaPequena = senha.length() < TAMANHO_MINIMO_SENHA;  
144     assert (usuarioPequeno||senhaPequena);  
145 }  
146  
147 }  
148 }
```

TelisWeb/src/java/utils/MudarSenha.java

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package utils;
6
7  import java.io.IOException;
8  import java.io.PrintWriter;
9  import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import persistencia.UserDAO;
14 import persistencia.UserDAO.LoginPequenoException;
15
16 /**
17  *
18  * @author felipe
19  */
20 public class MudarSenha extends HttpServlet {
21
22     private UserDAO userDAO;
23
24     /**
25      * Processes requests for both HTTP GET and
26      * POST methods.
27      *
28      * @param request servlet request
29      * @param response servlet response
30      * @throws ServletException if a servlet-specific error occurs
31      * @throws IOException if an I/O error occurs
32      */
33     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
34         throws ServletException, IOException {
35         response.setContentType("text/html;charset=UTF-8");
36         userDAO = new UserDAO(getServletContext());

```



```

35     String usuario = (String) request.getSession().getAttribute("user");
36     String novaSenha = request.getParameter("novaSenha");
37     PrintWriter out = response.getWriter();
38     try {
39         userDao.mudarSenha(usuario, novaSenha);
40         out.println("<html>");
41         out.println("<head>");
42         out.println("<title>Senha salva com sucesso</title>");
43         out.println("</head>");
44         out.println("<body>");
45         out.println("<h1>Senha salva com sucesso</h1>");
46         out.println("</body>");
47         out.println("</html>");
48     } catch (LoginPequenoException ex) {
49         out.println("<html>");
50         out.println("<head>");
51         out.println("<title>Erro salvando nova senha</title>");
52         out.println("</head>");
53         out.println("<body>");
54         out.println("<h1>Senha muito curta</h1>");
55         out.println("</body>");
56         out.println("</html>");
57     } finally {
58         out.close();
59     }
60 }
61
62 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
63 /**
64  * Handles the HTTP <code>GET</code> method.
65  * @param request servlet request
66  * @param response servlet response
67  * @throws ServletException if a servlet-specific error occurs
68  * @throws IOException if an I/O error occurs
69  */
70 @Override
71 protected void doGet(HttpServletRequest request, HttpServletResponse response)

```

```
72         throws ServletException, IOException {
73     processRequest(request, response);
74 }
75
76 /**
77  * Handles the HTTP <code>POST</code> method.
78  * @param request servlet request
79  * @param response servlet response
80  * @throws ServletException if a servlet-specific error occurs
81  * @throws IOException if an I/O error occurs
82  */
83 @Override
84 protected void doPost(HttpServletRequest request, HttpServletResponse response)
85     throws ServletException, IOException {
86     processRequest(request, response);
87 }
88
89 /**
90  * Returns a short description of the servlet.
91  * @return a String containing servlet description
92  */
93 @Override
94 public String getServletInfo() {
95     return "Short description";
96 } // </editor-fold>
97 }
```

TelisWeb/src/java/Utils/RemoverConta.java

```

1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package utils;
6
7  import persistencia.UserDAO;
8  import java.io.IOException;
9  import java.io.PrintWriter;
10 import javax.servlet.ServletException;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14
15 /**
16  *
17  * @author felipe
18  */
19 public class RemoverConta extends HttpServlet {
20
21     private UserDAO userDAO;
22     /**
23      * Processes requests for both HTTP GET and
24      * POST methods.
25      * @param request servlet request
26      * @param response servlet response
27      * @throws ServletException if a servlet-specific error occurs
28      * @throws IOException if an I/O error occurs
29      */
30     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
31         throws ServletException, IOException {
32         response.setContentType("text/html;charset=UTF-8");
33         userDAO = new UserDAO(this.getServletContext());
34         String usuario = (String) request.getSession().getAttribute("user");
35         userDAO.removerUsuario(usuario);

```

```

35     PrintWriter out = response.getWriter();
36     try {
37         out.println("<html>");
38         out.println("<head>");
39         out.println("<title>Usuario removido</title>");
40         out.println("</head>");
41         out.println("<body>");
42         out.println("<h1>Usuario removido</h1>");
43         out.println("</body>");
44         out.println("</html>");
45     } finally {
46         out.close();
47     }
48 }
49
50 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
51 /**
52  * Handles the HTTP <code>GET</code> method.
53  * @param request servlet request
54  * @param response servlet response
55  * @throws ServletException if a servlet-specific error occurs
56  * @throws IOException if an I/O error occurs
57  */
58 @Override
59 protected void doGet(HttpServletRequest request, HttpServletResponse response)
60     throws ServletException, IOException {
61     processRequest(request, response);
62 }
63
64 /**
65  * Handles the HTTP <code>POST</code> method.
66  * @param request servlet request
67  * @param response servlet response
68  * @throws ServletException if a servlet-specific error occurs
69  * @throws IOException if an I/O error occurs
70  */
71 @Override

```

```
72  protected void doPost(HttpServletRequest request, HttpServletResponse response)
73      throws ServletException, IOException {
74      processRequest(request, response);
75  }
76
77  /**
78   * Returns a short description of the servlet.
79   * @return a String containing servlet description
80   */
81  @Override
82  public String getServletInfo() {
83      return "Short description";
84  } // </editor-fold>
85 }
```

TelisWeb/src/java/Utils/Bytefile.java

```
1 package utils;
2
3 import java.io.File;
4 import java.io.FileInputStream;
5 import java.io.FileNotFoundException;
6 import java.io.FileOutputStream;
7 import java.io.IOException;
8 import java.io.InputStream;
9
10 /**
11  *
12  * @author felipe
13  */
14 public class Bytefile {
15     public static byte[] getBytesFromFile(File file) throws IOException {
16         InputStream is = new FileInputStream(file);
17
18         // Get the size of the file
19         long length = file.length();
20
21         if (length > Integer.MAX_VALUE) {
22             // File is too large
23         }
24
25         // Create the byte array to hold the data
26         byte[] bytes = new byte[(int)length];
27
28         // Read in the bytes
29         int offset = 0;
30         int numRead = 0;
31         while (offset < bytes.length
32             && (numRead=is.read(bytes, offset, bytes.length-offset)) >= 0) {
33             offset += numRead;
34         }
35
36         // Ensure all the bytes have been read in
```

```
37     if (offset < bytes.length) {
38         throw new IOException("Could not completely read file "+file.getName());
39     }
40
41     // Close the input stream and return bytes
42     is.close();
43     return bytes;
44 }
45
46 public static byte[] getBytesFromFilename(String filename) throws IOException {
47     return getBytesFromFile(new File(filename));
48 }
49
50 public static void writeBytesInFile(File file, byte[] data) throws
FileNotFoundException, IOException
51 {
52     FileOutputStream fos = new FileOutputStream(file);
53     fos.write(data);
54     fos.flush();
55     fos.close();
56 }
57
58 public static File getFileFromBytes(String filename, byte[] data) throws
FileNotFoundException, IOException
59 {
60     File file = new File(filename);
61     writeBytesInFile(file, data);
62     return file;
63 }
64 }
```

TelisWeb/src/java/Utils/CadastroUsuariosServlet.java

```

1 package utils;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import java.sql.SQLException;
6 import java.util.logging.Level;
7 import java.util.logging.Logger;
8 import javax.servlet.ServletException;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12 import persistencia.UserDAO;
13 import persistencia.UserDAO.LoginPequenoException;
14 import persistencia.UserDAO.UsuarioExisteException;
15
16 /**
17  *
18  * @author felipe
19  */
20 public class CadastroUsuariosServlet extends HttpServlet {
21
22     private static final long serialVersionUID = 1L;
23     private UserDAO userDAO;
24
25
26     /**
27      * Processes requests for both HTTP GET and
28      * POST methods.
29      * @param request servlet request
30      * @param response servlet response
31      * @throws ServletException if a servlet-specific error occurs
32      * @throws IOException if an I/O error occurs
33      */
34     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)

```



```

34         throws ServletException, IOException, ClassNotFoundException, SQLException
{
35     response.setContentType("text/html;charset=UTF-8");
36     userDAO = new UserDAO(this.getServletContext());
37     String user = request.getParameter("_username");
38     String pass = request.getParameter("_password");
39
40
41     PrintWriter out = response.getWriter();
42     try {
43         userDAO.CadastrarUsuario(user, pass);
44         imprimePaginaSimples(out, "Cadastro realizado com sucesso", "Cadastro
realizado com sucesso.");
45     } catch (UsuarioExisteException e) {
46         imprimePaginaSimples(out, "Falha ao cadastrar - Usuário Existe", "Erro,
usuário já existe");
47     } catch (LoginPequenoException e) {
48         imprimePaginaSimples(out, "Falha ao cadastrar - login ou senha curtos",
"Erro, utilize pelo menos 4 caracteres para usuario e senha");
49     } finally {
50         out.close();
51     }
52 }
53
54 private void imprimePaginaSimples(PrintWriter out, String title, String h1) {
55     out.println("<html>");
56     out.println("<head>");
57     out.println("<title>" + title + "</title>");
58     out.println("</head>");
59     out.println("<body>");
60     out.println("<h1>" + h1 + "</h1>");
61     out.println("<p><a href=\"IniciaSessao\">voltar</a></p>");
62     out.println("</body>");
63     out.println("</html>");
64 }
65
66 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">

```

```

67  /**
68   * Handles the HTTP <code>GET</code> method.
69   * @param request servlet request
70   * @param response servlet response
71   * @throws ServletException if a servlet-specific error occurs
72   * @throws IOException if an I/O error occurs
73   */
74  @Override
75  protected void doGet(HttpServletRequest request, HttpServletResponse response)
76      throws ServletException, IOException {
77      try {
78          try {
79              processRequest(request, response);
80          } catch (SQLException ex) {
81
82              Logger.getLogger(CadastroUsuariosServlet.class.getName()).log(Level.SEVERE, null, ex);
83          } catch (ClassNotFoundException ex) {
84
85              Logger.getLogger(CadastroUsuariosServlet.class.getName()).log(Level.SEVERE, null, ex);
86          }
87      }
88  /**
89   * Handles the HTTP <code>POST</code> method.
90   * @param request servlet request
91   * @param response servlet response
92   * @throws ServletException if a servlet-specific error occurs
93   * @throws IOException if an I/O error occurs
94   */
95  @Override
96  protected void doPost(HttpServletRequest request, HttpServletResponse response)
97      throws ServletException, IOException {
98      try {
99          try {
100              processRequest(request, response);
101          } catch (SQLException ex) {

```

```
102
Logger.getLogger(CadastroUsuariosServlet.class.getName()).log(Level.SEVERE, null, ex);
103     }
104     } catch (ClassNotFoundException ex) {
105
Logger.getLogger(CadastroUsuariosServlet.class.getName()).log(Level.SEVERE, null, ex);
106     }
107 }
108
109 /**
110  * Returns a short description of the servlet.
111  * @return a String containing servlet description
112  */
113 @Override
114 public String getServletInfo() {
115     return "Short description";
116 } // </editor-fold>
117 }
```

TelisWeb/src/java/utils/CopiadorServlet.java

```

1 package utils;
2
3 import java.io.File;
4 import java.io.FileInputStream;
5 import java.io.FileOutputStream;
6 import java.io.IOException;
7 import java.io.InputStream;
8 import java.io.OutputStream;
9 import java.io.PrintWriter;
10 import javax.servlet.ServletException;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14 import javax.servlet.http.HttpSession;
15 import sessao.BdToHomeFile;
16
17 /**
18  *
19  * @author felipe
20  */
21 public class CopiadorServlet extends HttpServlet {
22
23     /**
24      * Processes requests for both HTTP GET and
25      * POST methods.
26      * @param request servlet request
27      * @param response servlet response
28      * @throws ServletException if a servlet-specific error occurs
29      * @throws IOException if an I/O error occurs
30      */
31     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
32         throws ServletException, IOException {
33         response.setContentType("text/html;charset=UTF-8");
34         HttpSession session = request.getSession();

```

```

35
36 BdToHomeFile bd = (BdToHomeFile) session.getAttribute("bd2home");
37 String origemStr = request.getParameter("origem");
38 String destinoStr = request.getParameter("destino");
39
40 File origem = new File(bd.getContextDir(),origemStr);
41 File destino = new File(bd.getContextDir(),destinoStr);
42
43 //boolean entrouNolf = (destino.getParentFile()==null);
44
45 if(!destino.exists()) {
46     destino.getParentFile().mkdirs();
47     destino.createNewFile();
48 }
49
50 copiar(origem,destino);
51
52 PrintWriter out = response.getWriter();
53 try {
54 //     if(entrouNolf) {
55 //         out.println("parentFile retornou nulo");
56 //     }
57     /* TODO output your page here
58     out.println("<html>");
59     out.println("<head>");
60     out.println("<title>Servlet CopiadorServlet</title>");
61     out.println("</head>");
62     out.println("<body>");
63     out.println("<h1>Servlet CopiadorServlet at " + request.getContextPath () +
"</h1>");
64     out.println("</body>");
65     out.println("</html>");
66     */
67 } finally {
68     out.close();
69 }
70 }
71

```

```

72 void copiar(File fonte, File destino) throws IOException{
73     InputStream in = new FileInputStream(fonte);
74     OutputStream out = new FileOutputStream(destino);
75
76     byte[] buf = new byte[1024];
77     int len;
78     while((len = in.read(buf)) > 0){
79         out.write(buf, 0, len);
80     }
81     in.close();
82     out.close();
83 }
84
85 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
86 /**
87  * Handles the HTTP <code>GET</code> method.
88  * @param request servlet request
89  * @param response servlet response
90  * @throws ServletException if a servlet-specific error occurs
91  * @throws IOException if an I/O error occurs
92  */
93 @Override
94 protected void doGet(HttpServletRequest request, HttpServletResponse response)
95     throws ServletException, IOException {
96     processRequest(request, response);
97 }
98
99 /**
100  * Handles the HTTP <code>POST</code> method.
101  * @param request servlet request
102  * @param response servlet response
103  * @throws ServletException if a servlet-specific error occurs
104  * @throws IOException if an I/O error occurs
105  */
106 @Override
107 protected void doPost(HttpServletRequest request, HttpServletResponse response)
108     throws ServletException, IOException {

```

```
109     processRequest(request, response);
110 }
111
112 /**
113  * Returns a short description of the servlet.
114  * @return a String containing servlet description
115  */
116 @Override
117 public String getServletInfo() {
118     return "Short description";
119 }// </editor-fold>
120 }
```

TelisWeb/src/java/Utils/UploadServlet2.java

```

1 package utils;
2 //baseado no exemplo http://srikanthtechnologies.com/blog/java/uploadfile.aspx
3 import java.io.File;
4 import java.io.IOException;
5 import java.util.Iterator;
6 import java.util.List;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9 import javax.servlet.RequestDispatcher;
10 import javax.servlet.ServletException;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14
15
16 import org.apache.commons.fileupload.FileItem;
17 import org.apache.commons.fileupload.FileUploadException;
18 import org.apache.commons.fileupload.disk.DiskFileItemFactory;
19 import org.apache.commons.fileupload.servlet.ServletFileUpload;
20
21 //@WebServlet(name="UploadServlet", urlPatterns={"/upload"}) // specify
urlPattern for servlet
22 //@MultipartConfig // specifies servlet takes
multipart/form-data
23 public class UploadServlet2 extends HttpServlet {
24
25     @Override
26     protected void doPost(HttpServletRequest request, HttpServletResponse response)
27         throws ServletException, IOException {
28         DiskFileItemFactory factory = new DiskFileItemFactory();
29         ServletFileUpload upload = new ServletFileUpload(factory);
30         try {
31             List fileItems = upload.parseRequest(request);
32             Iterator i = fileItems.iterator();
33             File file;
34

```



```

35     FileItem fiFile = null;
36     String tipo = "";
37     String filename = "";
38
39     while (i.hasNext()) {
40         FileItem fi = (FileItem) i.next();
41         if (!fi.isFormField()) {
42             // Get the uploaded file parameters
43             String fieldName = fi.getFieldName();
44             String FullFileName = fi.getName();
45             String contentType = fi.getContentType();
46             boolean isInMemory = fi.isInMemory();
47             long sizeInBytes = fi.getSize();
48             // Write the file
49             if (FullFileName.lastIndexOf("\\") >= 0) {
50                 filename = FullFileName.substring(FullFileName.lastIndexOf("\\"));
51             } else {
52                 filename = FullFileName.substring(FullFileName.lastIndexOf("\\") + 1);
53             }
54             if (filename.lastIndexOf("/") >= 0) {
55                 filename = filename.substring(FullFileName.lastIndexOf("/"));
56             } else {
57                 filename = filename.substring(FullFileName.lastIndexOf("/") + 1);
58             }
59             fiFile = fi;
60
61         } else {
62             if (fi.getFieldName().equals("tipo")) {
63                 tipo = fi.getString("utf-8");
64             }
65         }
66     } // end while
67
68     //escrever arquivo
69     File home;
70     String url = "";
71     if (tipo.equals("imagem")) {
72         home = (File) request.getSession().getAttribute("imageHome");

```

```

73         url = "UpdateImageServlet";
74     } else if (tipo.equals("som")) {
75         home = (File) request.getSession().getAttribute("sndHome");
76         url = "geradoresComponentes/optionsSelectSom.jsp";
77     } else {
78         home = (File) request.getSession().getAttribute("contextDir");
79     }
80     file = new File(home, filename);
81     try {
82         fiFile.write(file);
83     } catch (Exception ex) {
84         Logger.getLogger(UploadServlet2.class.getName()).log(Level.SEVERE, null,
ex);
85         java.io.PrintWriter out = response.getWriter( );
86         out.println("Exceção escrevendo: ");
87         out.println(ex.getMessage());
88         out.println(ex);
89     }
90
91     RequestDispatcher rd = request.getRequestDispatcher(url);
92     rd.forward(request, response);
93 } catch (FileUploadException ex) {
94     Logger.getLogger(UploadServlet2.class.getName()).log(Level.SEVERE, null,
ex);
95     java.io.PrintWriter out = response.getWriter( );
96     out.println("Exceção escrevendo: ");
97     out.println(ex.getMessage());
98     out.println(ex);
99 }
100
101 } // end of doPost()
102 } // end of UploadServlet

```

TelisWeb/src/java/sessao/BdDataGetter.java

```
1 /**
2  * Esta classe pega os arquivos do banco de dados e devolve num mapa de <String,
Bytes[]>
3  */
4  package sessao;
5
6  import java.io.Serializable;
7  import java.util.Map;
8
9  /**
10   *
11   * @author felipe
12   */
13  public abstract class BdDataGetter implements Serializable {
14
15      private static final long serialVersionUID = 1L;
16
17      public abstract Map<String, byte[]> getImages();
18      public abstract Map<String, byte[]> getSounds();
19      public abstract Map<String, byte[]> getTextures();
20  }
```

TelisWeb/src/java/sessao/BdDataGetterFake.java

```
1 /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5 package sessao;
6
7 import java.io.File;
8 import java.io.IOException;
9 import java.io.Serializable;
10 import java.util.Collections;
11 import java.util.HashMap;
12 import java.util.Map;
13 import java.util.logging.Level;
14 import java.util.logging.Logger;
15 import utils.Bytefile;
16
17 /**
18  *
19  * @author felipe
20  */
21 public class BdDataGetterFake extends BdDataGetter implements Serializable {
22
23     File contextDir;
24
25     public BdDataGetterFake(File contextDir) {
26         this.contextDir = contextDir;
27     }
28
29     @Override
30     public Map<String, byte[]> getImages() {
31
32         HashMap<String, byte[]> mapa = new HashMap<String, byte[]>();
33         try {
34             File imageDir = new File(contextDir, "icones");
35             for(File imageFile: imageDir.listFiles())
36                 mapa.put(imageFile.getName(), Bytefile.getBytesFromFile(imageFile));
```

```

37     } catch (IOException ex) {
38         Logger.getLogger(BdDataGetterFake.class.getName()).log(Level.SEVERE, null,
ex);
39     }
40     return Collections.synchronizedMap(mapa);
41 }
42
43 @Override
44 public Map<String, byte[]> getSounds() {
45     HashMap<String, byte[]> mapa = new HashMap<String, byte[]>();
46     try {
47         File imageDir = new File(contextDir, "sons");
48         for(File imageFile: imageDir.listFiles())
49             mapa.put(imageFile.getName(), Bytefile.getBytesFromFile(imageFile));
50     } catch (IOException ex) {
51         Logger.getLogger(BdDataGetterFake.class.getName()).log(Level.SEVERE, null,
ex);
52     }
53     return Collections.synchronizedMap(mapa);
54 }
55
56 @Override
57 public Map<String, byte[]> getTextures() {
58     throw new UnsupportedOperationException("Not supported yet.");
59 }
60
61 }

```

TelisWeb/src/java/sessao/BdToHomeFile.java

```
1 package sessao;
2
3 import java.io.BufferedReader;
4 import java.io.File;
5 import java.io.FileNotFoundException;
6 import java.io.FileReader;
7 import java.io.IOException;
8 import java.io.Serializable;
9 import java.util.Collections;
10 import java.util.HashMap;
11 import java.util.Iterator;
12 import java.util.Map;
13 import java.util.Map.Entry;
14 import java.util.Set;
15 import java.util.logging.Level;
16 import java.util.logging.Logger;
17 import utils.Bytefile;
18
19 /**
20  *
21  * @author felipe
22  */
23 public class BdToHomeFile implements Serializable{
24     private static final long serialVersionUID = 1L;
25
26     String user;
27     BdDataGetter diskDataGetter;//para defaults only
28     File homeDir, imageHome, soundHome, srcHome, contextDir, usersDir;
29
30     private File createDir(File dir, String filename) throws Exception {
31         File f = new File(dir, filename);
32         if(!f.exists())
33             f.mkdirs();
34         return f;
35     }
36 }
```

```
37 public BdToHomeFile(String user, String webhome) {
38     contextDir = new File(webhome);
39     diskDataGetter = new BdDataGetterFake(contextDir);
40     this.user = user;
41
42 }
43
44 public void removeImageFile(String filename) {
45     File f = new File(imageHome, filename);
46     f.delete();
47 }
48
49 public void removeFile(String filename) {
50     File f = new File(contextDir, filename);
51     if(f.isDirectory())
52         removeDir(f);
53     else
54         f.delete();
55 }
56
57 public void removeDir(String dirname) {
58     File dir = new File(contextDir, dirname);
59     removeDir(dir);
60 }
61
62 public void removeDir(File dir) {
63     for (File file : dir.listFiles())
64         if (file.isDirectory())
65             removeDir(file);
66     else
67         file.delete();
68     dir.delete();
69 }
70
71 public void bd2home() throws FileNotFoundException, IOException, Exception {
72     usersDir = createDir(contextDir, "users");
73     homeDir = createDir(usersDir, user);
74     imageHome = createDir(homeDir, "Imagens");
```

```
75     soundHome = createDir(homeDir, "Sons");
76     srcHome = createDir(homeDir, "Fontes");
77     copyImagesToHome();//imagens default + imagens banco de dados
78     copySoundsToHome();//sons default + sons do banco de dados
79 }
80
81 public File getImageHome() {
82     return imageHome;
83 }
84
85 public File getSrcHome() {
86     return srcHome;
87 }
88
89 public File getHome() {
90     return homeDir;
91 }
92
93 public File getUsersDir() {
94     return usersDir;
95 }
96
97 public File createDir(String name) {
98     try {
99         return createDir(contextDir, name);
100     } catch (Exception ex) {
101         Logger.getLogger(BdToHomeFile.class.getName()).log(Level.SEVERE, null, ex);
102         return null;
103     }
104 }
105
106 public File createEmptyFile(String name) {
107     File f = new File(contextDir, name);
108     try {
109         if(!f.exists())
110             f.createNewFile();
111     } catch (IOException ex) {
112         Logger.getLogger(BdToHomeFile.class.getName()).log(Level.SEVERE, null, ex);
```



```

113     }
114     return f;
115 }
116
117 public String getNomeSemExtensao(File f) {
118     String name = f.getName();
119     int posPonto = name.lastIndexOf('.');
120     if(posPonto < 0)
121         posPonto = 0;
122     return name.substring(0, posPonto);
123 }
124
125 public Map<String, String> getDirMap(File dir) throws FileNotFoundException,
IOException {
126     Map<String, String> result = Collections.synchronizedMap(new
HashMap<String,String>(dir.list().length));
127     for(File imageFile: dir.listFiles()) {
128         result.put(getNomeSemExtensao(imageFile), usersDir.getName() + '/' + user
+ '/' + dir.getName() + '/' + imageFile.getName());
129     }
130     return result;
131 }
132
133 public Map<String, String> getImages() throws FileNotFoundException,
IOException {
134     return getDirMap(imageHome);
135 }
136
137 public Map<String, String> getSounds() throws FileNotFoundException,
IOException {
138     return getDirMap(soundHome);
139 }
140
141 /**
142  *
143  * @param binarios mapa contendo nome de conteúdo de arquivos de imagens ou
som
144  * @param tipoHome imageHome ou soundHome

```

```

145  * @throws FileNotFoundException
146  * @throws IOException
147  */
148  private void escreveBinarios(Map<String, byte[]> binarios, File pasta) throws
FileNotFoundException, IOException {
    149      Set<Entry<String, byte[]>> entrySet = binarios.entrySet();
    150      int i = 0;
    151      String filename;
    152      for(Iterator<Entry<String, byte[]>> iterator = entrySet.iterator();
iterator.hasNext(); i++)
    153      {
    154          Entry<String, byte[]> entry = iterator.next();
    155          filename = entry.getKey();
    156          byte[] value = entry.getValue();
    157          File image = new File(pasta,filename);
    158          Bytefile.writeBytesInFile(image, value);
    159      }
    160  }
    161
    162
    163
    164  private void copyImagesToHome() throws FileNotFoundException, IOException {
    165      escreveBinarios(diskDataGetter.getImages(),imageHome); //Defaults
    166  }
    167
    168  private void copySoundsToHome() throws FileNotFoundException, IOException {
    169      escreveBinarios(diskDataGetter.getSounds(),soundHome);
    170  }
    171
    172  private String leArquivo(File file) {
    173      try {
    174          String result = "";
    175          BufferedReader reader = new BufferedReader(new FileReader(file));
    176          String line;
    177          while((line= reader.readLine()) != null) {
    178              result += line;
    179              result += '\n';
    180          }

```

```
181         return result;
182     } catch (FileNotFoundException ex) {
183         Logger.getLogger(BdToHomeFile.class.getName()).log(Level.SEVERE, null, ex);
184         return "";
185     } catch (IOException ex) {
186         return "";
187     }
188
189 }
190
191 public File getSndHome() {
192     return soundHome;
193 }
194
195 public File getContextDir() {
196     return contextDir;
197 }
198
199 }
```

TelisWeb/src/java/sessao/ConfigUsuario.java

```

1 package sessao;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.http.HttpServlet;
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 /**
10  *
11  * @author felipe
12  */
13 public class ConfigUsuario extends HttpServlet {
14
15     /**
16      * Processes requests for both HTTP GET and
17      * POST methods.
18      * @param request servlet request
19      * @param response servlet response
20      * @throws ServletException if a servlet-specific error occurs
21      * @throws IOException if an I/O error occurs
22      */
23     protected void processRequest(HttpServletRequest request, HttpServletResponse
24     response)
25         throws ServletException, IOException {
26         response.setContentType("text/html;charset=UTF-8");
27     }
28     // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
29     sign on the left to edit the code.">
30     /**
31      * Handles the HTTP GET method.
32      * @param request servlet request
33      * @param response servlet response
34      * @throws ServletException if a servlet-specific error occurs

```

```
34  * @throws IOException if an I/O error occurs
35  */
36  @Override
37  protected void doGet(HttpServletRequest request, HttpServletResponse response)
38      throws ServletException, IOException {
39      processRequest(request, response);
40  }
41
42  /**
43   * Handles the HTTP <code>POST</code> method.
44   * @param request servlet request
45   * @param response servlet response
46   * @throws ServletException if a servlet-specific error occurs
47   * @throws IOException if an I/O error occurs
48   */
49  @Override
50  protected void doPost(HttpServletRequest request, HttpServletResponse response)
51      throws ServletException, IOException {
52      processRequest(request, response);
53  }
54
55  /**
56   * Returns a short description of the servlet.
57   * @return a String containing servlet description
58   */
59  @Override
60  public String getServletInfo() {
61      return "Short description";
62  } // </editor-fold>
63 }
```

TelisWeb/src/java/sessao/ConfigUsuarioPeloContainer.java

```

1 package sessao;
2
3 import java.io.IOException;
4 import javax.servlet.RequestDispatcher;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10 /**
11  *
12  * @author felipe
13  */
14 public class ConfigUsuarioPeloContainer extends HttpServlet {
15
16     /**
17      * Processes requests for both HTTP GET and
18      * POST methods.
19      * @param request servlet request
20      * @param response servlet response
21      * @throws ServletException if a servlet-specific error occurs
22      * @throws IOException if an I/O error occurs
23      */
24     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
25         throws ServletException, IOException {
26         response.setContentType("text/html;charset=UTF-8");
27         request.getSession().setAttribute("user", request.getRemoteUser());
28         RequestDispatcher rd = request.getRequestDispatcher("IniciaSessao");
29         rd.forward(request,response);
30     }
31     // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
32     /**
33      * Handles the HTTP GET method.

```

```

34  * @param request servlet request
35  * @param response servlet response
36  * @throws ServletException if a servlet-specific error occurs
37  * @throws IOException if an I/O error occurs
38  */
39  @Override
40  protected void doGet(HttpServletRequest request, HttpServletResponse response)
41      throws ServletException, IOException {
42      processRequest(request, response);
43  }
44
45  /**
46   * Handles the HTTP <code>POST</code> method.
47   * @param request servlet request
48   * @param response servlet response
49   * @throws ServletException if a servlet-specific error occurs
50   * @throws IOException if an I/O error occurs
51   */
52  @Override
53  protected void doPost(HttpServletRequest request, HttpServletResponse response)
54      throws ServletException, IOException {
55      processRequest(request, response);
56  }
57
58  /**
59   * Returns a short description of the servlet.
60   * @return a String containing servlet description
61   */
62  @Override
63  public String getServletInfo() {
64      return "Short description";
65  } // </editor-fold>
66 }

```

TelisWeb/src/java/sessao/IniciaSessao.java

```

1 /*
2  * Servlet para carregar no início da sessão arquivos pessoais (imagens, sons e
3  * código fonte para odiretório temp do usuário
4  */
5 package sessao;
6
7 import java.io.IOException;
8 import java.io.PrintWriter;
9 import java.util.Map;
10 import javax.servlet.ServletException;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14 import javax.servlet.RequestDispatcher;
15 import javax.servlet.http.HttpSession;
16
17
18 /**
19  *
20  * @author felipe
21  */
22 //@WebServlet(name = "IniciaSessao", urlPatterns = {"/IniciaSessao"})
23 public class IniciaSessao extends HttpServlet {
24
25     /**
26      * Processes requests for both HTTP <code>GET</code> and
27      * <code>POST</code> methods.
28      *
29      * @param request servlet request
30      * @param response servlet response
31      * @throws ServletException if a servlet-specific error occurs
32      * @throws IOException if an I/O error occurs
33      */
34     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
35         throws ServletException, IOException {
36         response.setContentType("text/html;charset=UTF-8");

```



```

35     PrintWriter out = response.getWriter();
36     out.println("Carregando página<br/>");
37     HttpSession session = request.getSession();
38     String user = (String) session.getAttribute("user");
39     BdToHomeFile bd = new BdToHomeFile(user,
this.getServletContext().getRealPath(""));
40     try {
41         bd.bd2home();//cria diretórios e copia defaults
42         Map<String, String> imageFiles = bd.getImages();
43         session.setAttribute("bd2home", bd);
44         session.setAttribute("imageHome", bd.getImageHome());
45         session.setAttribute("contextDir", bd.getContextDir());
46         session.setAttribute("srcHome", bd.getSrcHome());
47         session.setAttribute("sndHome", bd.getSndHome());
48         session.setAttribute("strSrcHome", bd.getUsersDir().getName() + '/' +
bd.getHome().getName() + '/' + bd.getSrcHome().getName());
49         session.setAttribute("strImgHome", bd.getUsersDir().getName() + '/' +
bd.getHome().getName() + '/' + bd.getImageHome().getName());
50         session.setAttribute("strSndHome", bd.getUsersDir().getName() + '/' +
bd.getHome().getName() + '/' + bd.getSndHome().getName());
51         session.setAttribute("Imagens", imageFiles);
52         session.setAttribute("appUrl", getAppUrl(request));
53         RequestDispatcher rd = request.getRequestDispatcher("app.jsp");
54         rd.forward(request, response);
55     } catch (Exception e) {
56         out.println("Erro");
57         out.println(e.getLocalizedMessage());
58     }
59     finally {
60         out.close();
61     }
62 }
63
64 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
65 /**
66  * Handles the HTTP <code>GET</code> method.
67  * @param request servlet request

```

```

68  * @param response servlet response
69  * @throws ServletException if a servlet-specific error occurs
70  * @throws IOException if an I/O error occurs
71  */
72  @Override
73  protected void doGet(HttpServletRequest request, HttpServletResponse response)
74      throws ServletException, IOException {
75      processRequest(request, response);
76  }
77
78  /**
79   * Handles the HTTP <code>POST</code> method.
80   * @param request servlet request
81   * @param response servlet response
82   * @throws ServletException if a servlet-specific error occurs
83   * @throws IOException if an I/O error occurs
84   */
85  @Override
86  protected void doPost(HttpServletRequest request, HttpServletResponse response)
87      throws ServletException, IOException {
88      processRequest(request, response);
89  }
90
91  /**
92   * Returns a short description of the servlet.
93   * @return a String containing servlet description
94   */
95  @Override
96  public String getServletInfo() {
97      return "Short description";
98  } // </editor-fold>
99
100 private String getAppUrl(HttpServletRequest req) {
101     String scheme = req.getScheme(); // http
102     String serverName = req.getServerName(); // hostname.com
103     int serverPort = req.getServerPort(); // 80
104     String contextPath = req.getContextPath(); // /mywebapp
105     // String servletPath = req.getServletPath(); // /servlet/MyServlet

```

```
106 //      String pathInfo = req.getPathInfo();      // /a/b;c=123
107 //      String queryString = req.getQueryString();      // d=789
108      String url = scheme+"://"+serverName+": "+serverPort+contextPath;
109      return url;
110  }
111 }
```

TelisWeb/src/java/sessao/LogouServlet.java

```

1 package sessao;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.http.HttpServlet;
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 /**
10  *
11  * @author felipe
12  */
13 public class LogouServlet extends HttpServlet {
14
15     /**
16      * Processes requests for both HTTP GET and
17      * POST methods.
18      * @param request servlet request
19      * @param response servlet response
20      * @throws ServletException if a servlet-specific error occurs
21      * @throws IOException if an I/O error occurs
22      */
23     protected void processRequest(HttpServletRequest request, HttpServletResponse
24     response)
25         throws ServletException, IOException {
26         response.setContentType("text/html;charset=UTF-8");
27         request.getSession(false).invalidate();
28         response.sendRedirect(request.getContextPath());
29     }
30
31     // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
32     sign on the left to edit the code.">
33     /**
34      * Handles the HTTP GET method.
35      * @param request servlet request
36      * @param response servlet response

```

```

34  * @throws ServletException if a servlet-specific error occurs
35  * @throws IOException if an I/O error occurs
36  */
37  @Override
38  protected void doGet(HttpServletRequest request, HttpServletResponse response)
39      throws ServletException, IOException {
40      processRequest(request, response);
41  }
42
43  /**
44   * Handles the HTTP <code>POST</code> method.
45   * @param request servlet request
46   * @param response servlet response
47   * @throws ServletException if a servlet-specific error occurs
48   * @throws IOException if an I/O error occurs
49   */
50  @Override
51  protected void doPost(HttpServletRequest request, HttpServletResponse response)
52      throws ServletException, IOException {
53      processRequest(request, response);
54  }
55
56  /**
57   * Returns a short description of the servlet.
58   * @return a String containing servlet description
59   */
60  @Override
61  public String getServletInfo() {
62      return "Short description";
63  } // </editor-fold>
64 }

```

TelisWeb/src/java/sessao/ManageFileServlet.java

```

1 package sessao;
2
3 import java.io.IOException;
4 import javax.servlet.RequestDispatcher;
5 import javax.servlet.ServletException;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import javax.servlet.http.HttpSession;
10
11 /**
12  *
13  * @author felipe
14  */
15 public class ManageFileServlet extends HttpServlet {
16
17     /**
18      * Processes requests for both HTTP GET and
19      * POST methods.
20      * @param request servlet request
21      * @param response servlet response
22      * @throws ServletException if a servlet-specific error occurs
23      * @throws IOException if an I/O error occurs
24      */
25     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
26         throws ServletException, IOException {
27         response.setContentType("text/html;charset=UTF-8");
28         String file = request.getParameter("filename");
29         HttpSession session = request.getSession();
30         BdToHomeFile bd = (BdToHomeFile) session.getAttribute("bd2home");
31         String action = request.getParameter("action");
32         if(action.equals("delete"))
33             bd.removeFile(file);
34         else if(action.equals("createFile"))
35             bd.createEmptyFile(file);

```

```

35     else if(action.equals("createDir"))
36         bd.createDir(file);
37     String url = request.getParameter("response");
38     if(url != null)
39         if(!url.equals("")) {
40             RequestDispatcher rd = request.getRequestDispatcher(url);
41             rd.forward(request, response);
42         }
43     }
44
45     // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
46     /**
47      * Handles the HTTP <code>GET</code> method.
48      * @param request servlet request
49      * @param response servlet response
50      * @throws ServletException if a servlet-specific error occurs
51      * @throws IOException if an I/O error occurs
52      */
53     @Override
54     protected void doGet(HttpServletRequest request, HttpServletResponse response)
55         throws ServletException, IOException {
56         processRequest(request, response);
57     }
58
59     /**
60      * Handles the HTTP <code>POST</code> method.
61      * @param request servlet request
62      * @param response servlet response
63      * @throws ServletException if a servlet-specific error occurs
64      * @throws IOException if an I/O error occurs
65      */
66     @Override
67     protected void doPost(HttpServletRequest request, HttpServletResponse response)
68         throws ServletException, IOException {
69         processRequest(request, response);
70     }
71

```

```
72  /**
73   * Returns a short description of the servlet.
74   * @return a String containing servlet description
75   */
76  @Override
77  public String getServletInfo() {
78      return "Short description";
79  }// </editor-fold>
80 }
```


TelisWeb/src/java/sessao/UpdateImageServlet.java

```

1 package sessao;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import java.util.Iterator;
6 import java.util.Map;
7 import java.util.Map.Entry;
8 import java.util.Set;
9 import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import javax.servlet.http.HttpSession;
14
15 /**
16  *
17  * @author felipe
18  */
19 public class UpdateImageServlet extends HttpServlet {
20
21     /**
22      * Processes requests for both HTTP GET and
23      * POST methods.
24      * @param request servlet request
25      * @param response servlet response
26      * @throws ServletException if a servlet-specific error occurs
27      * @throws IOException if an I/O error occurs
28      */
29     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
30         throws ServletException, IOException {
31         response.setContentType("text/html;charset=UTF-8");
32
33         HttpSession session = request.getSession();
34         BdToHomeFile bd = (BdToHomeFile) session.getAttribute("bd2home");
35         Map<String, String> images = bd.getImages();

```

```

35     Set<Entry<String, String>> entrySet = images.entrySet();
36
37     PrintWriter out = response.getWriter();
38     try {
39         Entry<String, String> next;
40         for(Iterator<Entry<String, String>> iterator = entrySet.iterator();
iterator.hasNext(); ) {
41             next = iterator.next();
42             out.println("<option value=\"" + next.getValue() + "\" title=\"" +
next.getValue() + "\">" + next.getKey() + "</option>");
43         }
44     }finally {
45         out.close();
46     }
47 }
48
49 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
50 /**
51  * Handles the HTTP <code>GET</code> method.
52  * @param request servlet request
53  * @param response servlet response
54  * @throws ServletException if a servlet-specific error occurs
55  * @throws IOException if an I/O error occurs
56  */
57 @Override
58 protected void doGet(HttpServletRequest request, HttpServletResponse response)
59     throws ServletException, IOException {
60     processRequest(request, response);
61 }
62
63 /**
64  * Handles the HTTP <code>POST</code> method.
65  * @param request servlet request
66  * @param response servlet response
67  * @throws ServletException if a servlet-specific error occurs
68  * @throws IOException if an I/O error occurs
69  */

```

```
70  @Override
71  protected void doPost(HttpServletRequest request, HttpServletResponse response)
72      throws ServletException, IOException {
73      processRequest(request, response);
74  }
75
76  /**
77   * Returns a short description of the servlet.
78   * @return a String containing servlet description
79   */
80  @Override
81  public String getServletInfo() {
82      return "Short description";
83  } // </editor-fold>
84 }
```

TelisWeb/src/java/sessao/WriterServlet.java

```

1 package sessao;
2
3 import java.io.File;
4 import java.io.FileOutputStream;
5 import java.io.IOException;
6 import java.io.OutputStreamWriter;
7 import java.io.PrintWriter;
8 import java.net.URLDecoder;
9 import javax.servlet.ServletException;
10 import javax.servlet.http.HttpServlet;
11 import javax.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import javax.servlet.http.HttpSession;
14
15 /**
16  *
17  * @author felipe
18  */
19 public class WriterServlet extends HttpServlet {
20
21     /**
22      * Processes requests for both HTTP GET and
23      * POST methods.
24      * @param request servlet request
25      * @param response servlet response
26      * @throws ServletException if a servlet-specific error occurs
27      * @throws IOException if an I/O error occurs
28      */
29     protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
30         throws ServletException, IOException {
31         HttpSession session = request.getSession();
32         BdToHomeFile bd = (BdToHomeFile) session.getAttribute("bd2home");
33         String filename = request.getParameter("filename");
34         String content = request.getParameter("content");
35         if(content==null)

```

```

35     content="";
36     content = URLDecoder.decode(content, "utf-8");
37     File f = new File(bd.getContextDir(),filename);
38     if(!f.exists()) {
39         f.getParentFile().mkdirs();
40         f.createNewFile();
41     }
42
43     FileOutputStream fos = new FileOutputStream(f);
44     OutputStreamWriter outf = new OutputStreamWriter(fos);
45
46     outf.write(content);
47     outf.flush();
48     outf.close();
49
50     response.setContentType("text/html;charset=UTF-8");
51     PrintWriter out = response.getWriter();
52     try {
53         out.println("<html>");
54         out.println("<head>");
55         out.println("<title>Servlet WriterServlet</title>");
56         out.println("</head>");
57         out.println("<body>");
58         out.println("<h1>Servlet WriterServlet at " + request.getContextPath () +
"</h1>");
59         out.println("<p>file: " + filename + "</p>");
60         out.println("<p>file: " + f.getAbsolutePath() + "</p>");
61         out.println("<p>conteúdo: " + content + "</p>");
62         out.println("</body>");
63         out.println("</html>");
64     } finally {
65         out.close();
66     }
67 }
68
69 // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the +
sign on the left to edit the code.">
70 /**

```

```

71  * Handles the HTTP <code>GET</code> method.
72  * @param request servlet request
73  * @param response servlet response
74  * @throws ServletException if a servlet-specific error occurs
75  * @throws IOException if an I/O error occurs
76  */
77  @Override
78  protected void doGet(HttpServletRequest request, HttpServletResponse response)
79      throws ServletException, IOException {
80      processRequest(request, response);
81  }
82
83  /**
84   * Handles the HTTP <code>POST</code> method.
85   * @param request servlet request
86   * @param response servlet response
87   * @throws ServletException if a servlet-specific error occurs
88   * @throws IOException if an I/O error occurs
89   */
90  @Override
91  protected void doPost(HttpServletRequest request, HttpServletResponse response)
92      throws ServletException, IOException {
93      processRequest(request, response);
94  }
95
96  /**
97   * Returns a short description of the servlet.
98   * @return a String containing servlet description
99   */
100  @Override
101  public String getServletInfo() {
102      return "Short description";
103  } // </editor-fold>
104 }

```