

IMPLEMENTAÇÃO, APLICAÇÃO E AFERIÇÃO DE UM SISTEMA DE CONSULTAS ON-LINE COMO FERRAMENTA DIDÁTICA PARA EDUCADORES

Orientador: Profa. Dra. Ana Lúcia Nogueira de Camargo Harris

luharris@fec.unicamp.br

Aluno: Filipe Antônio Marques Falcetta

filipetonho@gmail.com

Financiadora: Pibic / SAE

Introdução

Este projeto de pesquisa dá continuidade a outro, intitulado “Desenvolvimento de um sistema de consulta *on-line* para dar suporte a estudos sobre *brises* para projetos arquitetônicos” (FALCETTA, 2006).

O objetivo principal desta pesquisa é o de desenvolver um sistema de consultas *on-line* que permita a inserção de diversos tipos de conteúdo – na forma de mídias: textos, imagens estáticas e animações, projetos em CAD, simulações etc. – por educadores das mais diversas áreas do conhecimento.

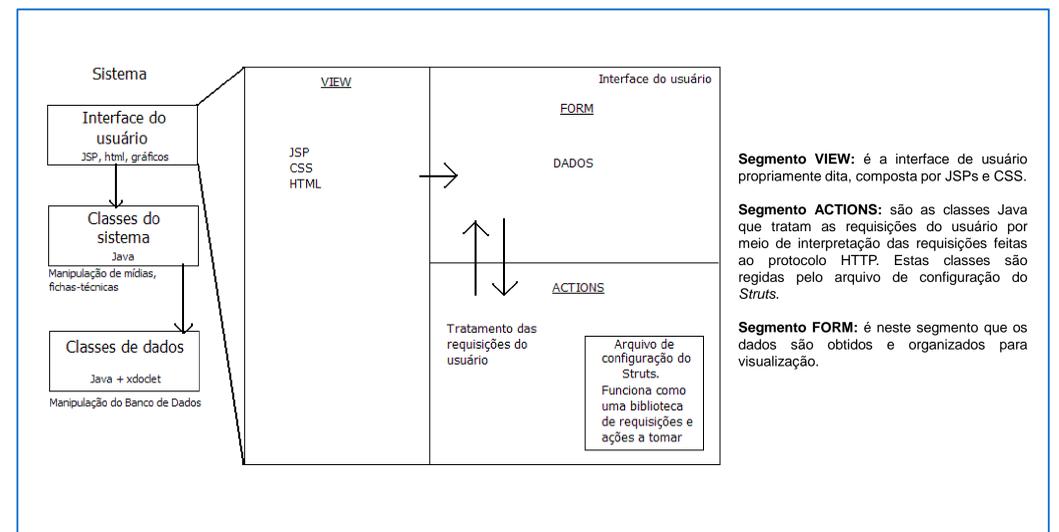
Como efeito de testes, o sistema foi implementado com conteúdo referente aos *brises-soleil*. Para este conteúdo, visualizou-se dois tipos de usuários: profissionais e estudantes da área de construção civil e arquitetura.

Metodologia

De modo análogo ao projeto anterior, os alicerces da pesquisa continuam sendo a orientação a objeto, linguagens dinâmicas Web (o sistema foi programado em Java), sistemas de bancos de dados relacionais (MySQL, por exemplo).

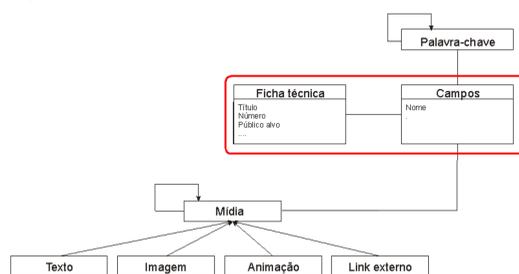
A fim de facilitar o processo de implementação do sistema, foram estudados diversos *frameworks* (kits de ferramentas que fornecem/geram automaticamente os códigos-fonte do sistema): *Struts*, *Xdoclet* e *Hibernate*.

- **Struts:** Largamente utilizado na confecção do sistema, permite obter uma ligação simplificada entre a base de dados (seja ela fixa ou não) e o usuário do sistema. (Um detalhe da estrutura do sistema pode ser vista nos esquemas ao lado)
- **Xdoclet:** gera automaticamente os códigos qualificadores das estruturas internas do sistema.
- **Hibernate:** escrito na forma de comentários e *tags* (etiquetas), gera automaticamente a estrutura do banco de dados relacional.



Resultados e análises

Em um sistema baseado na teoria de Orientação a Objeto (FALCETTA, 2006), deve-se escolher objetos-chave, que irão gerir e relacionar todo o conteúdo; desta forma, o projeto desenvolveu-se em torno uma ficha-técnica – macro-estrutura de ligação entre as mídias e o utilizador do sistema. Ela pode conter diversos tipos de conteúdo, controlados por microestruturas, os campos – permitem a ligação mídia – palavras-chave.



A interação sistema-BD se realiza com as classes DAO, que possibilita a manipulação dos dados dentro do sistema de banco de dados: inclusão, exclusão, atualização, busca etc. Estas classes, embora programadas em Java, independem da linguagem do sistema, pois seguem um padrão de formação. Cabe ao *Hibernate* o mapeamento e direcionamento dos dados enviados pelo usuário às tabelas do BD.

Conclusões

Por fim, pôde ser concebida a interface do usuário na linguagem JSP (páginas que mesclam html estático e campos dinâmicos Java – em *tags* específicas).

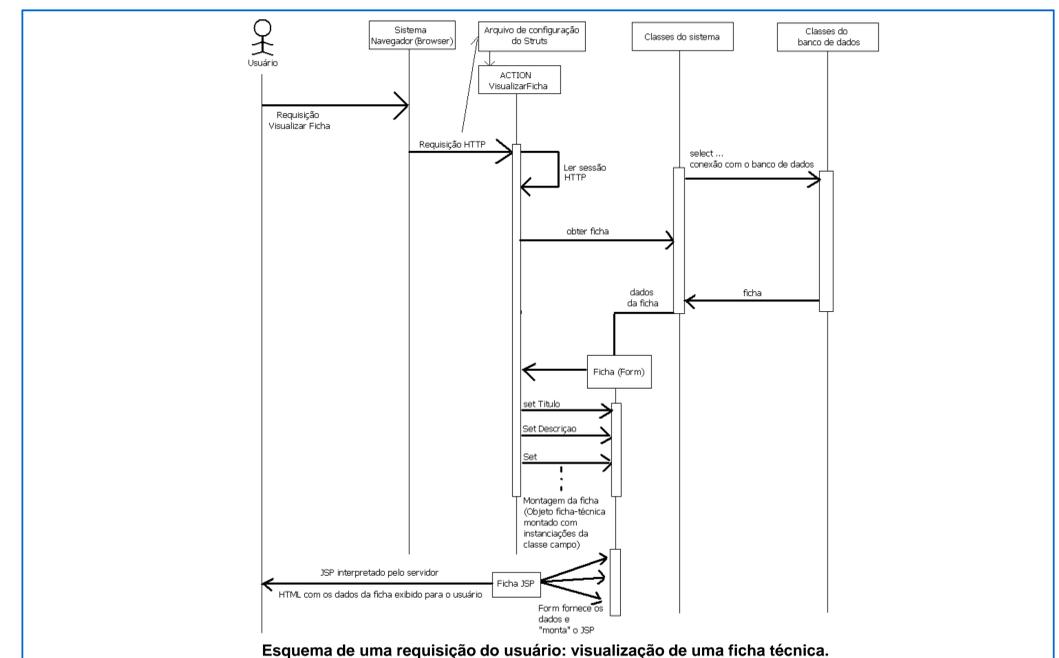
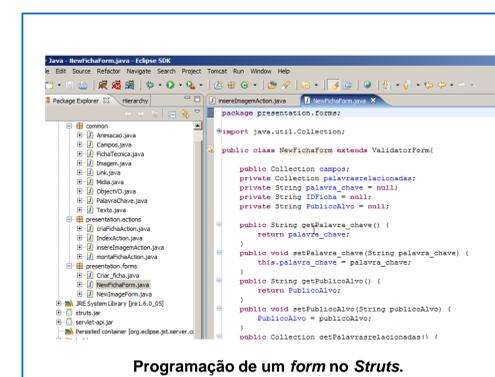
Tais arquivos incluem *taglibs* do *Struts*, que vinculam o arquivo visualizado no navegador Web à estrutura Java montada anteriormente, endereçando **forms** com **actions** e direcionando os dados inseridos pelo usuário às classes do sistema.

O sistema deverá ser hospedado em um servidor Web para eventuais testes e verificação da estabilidade e funcionalidade do mesmo.

Esta sendo desenvolvida uma documentação para facilitar posterior utilização da estrutura-base do sistema por educadores de diversas áreas do conhecimento.

Agradecimentos

À Profa. Dra. Ana Lúcia N. de Camargo Harris, pelo seu empenho e dedicação ao projeto. Ao André Constantino, do NIED (Unicamp), pelo apoio no processo de programação e desenvolvimento do sistema.

```

package presentation.form;

import java.util.Collection;

public class NewFichaForm extends ValidatorForm {
    public Collection campos;
    private Collection palavrasRelacionadas;
    private String palavra_chave = null;
    private String IDTitulo = null;
    private String PublicaAlvo = null;

    public String getPalavra_chave() {
        return palavra_chave;
    }

    public void setPalavra_chave(String palavra_chave) {
        this.palavra_chave = palavra_chave;
    }

    public String getPublicaAlvo() {
        return PublicaAlvo;
    }

    public void setPublicaAlvo(String publicaAlvo) {
        PublicaAlvo = publicaAlvo;
    }

    public Collection getPalavrasRelacionadas() {

```

