INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA CAMPUS CAJAZEIRAS
UNIDADE ACADÊMICA DA ÁREA DE INFORMÁTICA
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

USO DOS PADRÕES DE PROJETO NO SISTEMA
(ApResSa - Aplicação para Reserva de Salas)

Aluísio José Pereira[1]
José Ferreira Vieira[2]
Zilderlan Leite da Silva[3]

CAJAZEIRAS-PB
2015.2

[1] aluisio1102@gmail.com
[2] joseferreiravieira123@gmail.com
[3] zilderlan123@gmail.com

SUMÁRIO

1 INTRODUÇÃO

Na forma de Documentação de software do Sistema ApResSa (Aplicação para Reserva de Salas), em vista a obter maior operacionalização e eficiência das atividades voltada para tal serviço temos a implantação deste sistema.

Como o meio acadêmico envolve uma série de atividades relacionadas ao domínio de ensino e Pesquisa. Com base nisso foi solicitado à criação de uma ferramenta capaz de tornar mais dinâmico a alocação de sala e materiais, para aplicação de aulas, entre outros.

Dentre suas diversas funcionalidades o sistema de início trará uma maior consolidação das informações referente a manter usuários, permitindo cadastrar, atualizar, remover usuários assim como datas. Disponibilizado em ambiente *WEB*, o sistema (ApResSa) promoverá maior facilidade em disponibilizar para os usuários cadastrado e com suas devidas permissões cada módulo respectivo a seu perfil.

O ApResSa foi idealizado para plataforma WEB, sendo toda suas operações executadas por meio de uma página acessada, com um endereço pré-definido (URL) por dado navegador (browser) instalado na máquina do usuário.

2 PASSOS INICIAIS

Para melhor execução deste sistema se faz necessário construir um banco de dados em um servidor PostgreSQL, e executar o arquivo **bancoPadroes.sql** que se encontra dentro de **resources** e subpasta **recursos**. Feito isso se faz necessário modificar as configurações do banco no já especificado caminho (no mesmo pacote - recursos) no arquivo **connection.properties**, descrevendo a **url** (caminho do banco seguido do nome no qual você usou ao criar o banco de dados) **user** (nome do usuário do servidor do banco) e **password** (senha do seu servidor do banco PostgresSQL).

Se faz também necessário um servidor de aplicação **Apache TOMCAT** se preferir, assim como **JDK 8.0** previamente instalada e configurada ambos compatíveis. E realizar todos os downloads para execução do projeto **maven.**

1. Criar um banco de dado (ex.: new tatabase... nome “sistemaApressa”), no seu ambiente PostgreSQL;
2. Execute o script (**bancoPadroes.sql**) de criação das tabelas no banco de dados criado;
3. Altear (URL, USER, PASSWORD) do arquivo **connection.properties** de acordo com o servidor posgreSQL utilizado que você usa.
4. Copie a pasta do **SistemaApressa**, para a pasta webapp do apache TOMCAT;
5. Execute o Start service do seu TOMCAT;
6. Abre o navegador (ex.: Google Chrome), acessa com a URL de caminho do seu Apache (usualmente para o TOMCAT é: http://localhost:8080/sistemaApressa/);
7. Realizar login com usuário e senha padrão: (usuário: admin@gmail.com, senha: admin153Chg%).

3 PADRÕES USADOS

No projeto do sistema ApResSa foi usado os padrões: criação – Abstract Factory, Factory Method; estruturação – DAO, MVC; comportamentais – Command, Front Controller, Observer, State., seguindo cada particularidade dos mesmo para melhor aplicação destes no projeto.

Abstract Factory - O padrão foi usado aqui para a criação de uma família de objetos que interagem por meio da interface não se fazendo necessário, deste modo, a criação de classe concreta devidamente especificada. O padrão faz a interação entre a interface que cria as relações entre as demais interfaces se comunicado entre cada uma.

Factory Method – O padrão aqui mencionado foi usado como forma de criar uma interface de objetos mas deixando as subclasses decidirem qual objeto criar. Tendo como motivação a estruturação de um frameworks, não sendo possível determinar qual elemento deve ser criado, sendo assim fazendo com que o instanciado faça a criação de uma instância específica, assim usados na criação das classe factory, factoryBD, criando deste modo os objetos DAO.

DAO – Data Access Object: usado para a criação de classes de dados capazes de estabelecer a relação necessária para os dados do banco usado. Este padrão atribui ao projeto a possibilidade de encapsular os mecanismos de acessos aos dados criados pelas interfaces, atribuindo também uma característica que permite que os mecanismos de acesso aos dados sejam alterados independentemente do código que utiliza os dados, deste modo temos: o acesso aos dados do banco através das classes DAO se responsabilizando assim por um objeto de domínio específico para cada operação finais na realização desejadas da aplicação.

MVC – Modelo Visão Controlador: promove uma situação capaz de separar a representação da informação da interação do usuário com ele próprio foi possível estabelecer em (modelo) a comunicação entre regra de negócio e aplicação das mesmas e gerar a mediação entre os modelos e a visão, e assim centrar a reusabilidade do código e separa o conceito. No projeto tem os controles recebendo entradas de requisição vinda dos métodos get/post da aplicação e desenvolvendo um estímulo do utilizador e decidindo como processá-la, invocando dado objeto para cada domínio de acordo com a lógica de negócio.

Command – Na aplicação este padrão foi usado de modo a estabelecer o encapsulamento de dadas solicitações para os objetos, e permitir: parametrizada diferentes as solicitações, enfileirando cada operação para ser executada e para poder desfazer uma operação, sendo este presente em todas as classes do pacote command desta aplicação.

Front Controller – Usado como forma de controlar as chamadas vindas dos recursos web no pacote controle classe AppControle e fazendo assim a manipulação web e a hierarquização dos comandos, estes recebem as solicitações HTTP (GET/POST), extraindo as informações necessárias e decidindo qual ação tomar e deste modo delegar a dado objeto a excursão da ação.

Observer – O padrão foi responsável neste projeto pelos mecanismos de notificação dos eventos. No pacote entidades com proposito de comunicar/notificar aos usuários possíveis mudanças especificas nos estados dos eventos a serem alocados e as disponibilidades dos recursos (sala de aula, materiais, etc.) quanto a possíveis exclusões ou mudanças.

State – Mediante a troca de estado de um objeto temos o uso deste padrão, nesta aplicação, de forma a realizar as necessárias trocas de estado das salas de aulas dos materiais, eventos, quanto a alocado/disponível/etc.

4 CONSIDERAÇÕES

O projeto ApResSa permitirá melhora as relações de serviço e administração dos mesmo para fins de alocação de sala de aula e uso dos principais recursos que nas mesmas se fizerem necessários para seus usuários. Proporcionar um serviço seguindo tais especificidades requer um nível de detalhamento e contato característico para cada usuários, deste modo, o sistema ApResSa irá proporcionar esta investida, por meio da elaboração deste projeto foi possível traçar uma abordagem capaz de gerir os recursos de forma a valorizar e melhorar os serviços prestados.