

Projeto Escola

Documentação

Clóvis Lemes Ferreira Junior

<clovis.ferreira@serpro.gov.br>

Sobre o Projeto Escola	v
1. Instalação	1
2. Funcionalidades Utilizadas	3
2.1. Funcionalidade de Exceções	3
2.2. Funcionalidade de Mensagens	3
2.3. Funcionalidade de Configurações	3
2.4. Funcionalidade de Segurança	3
2.5. Funcionalidade de Relatório	4
2.6. Funcionalidade de Transação	4
2.7. Funcionalidade de Inicialização	4
2.8. Funcionalidade de E-mail	5
2.9. Funcionalidade de Paginação	5
3. Caso de Uso	7
3.1. Caso de Uso	7
4. Camadas	9
4.1. Camada de Persistência	9
4.2. Camada de Negócio	9
4.3. Camada de Visão	10

Sobre o Projeto Escola

Escola é um projeto de software construído sob o Framework Demoiselle, seu objetivo é prover uma aplicação de exemplo para estudo do framework e seus componentes. Suas funcionalidades foram projetadas de maneira que os principais recursos do framework fossem utilizados, portanto não há pretensão de criar uma ferramenta que atenda requisitos reais de uma aplicação comercial, mas um projeto que possua problemas capazes de serem resolvidos com uso do Framework Demoiselle e apresente estas soluções didaticamente.

Instalação

O código fonte pode ser encontrado no repositório do GitHub no endereço <https://github.com/demoiselle/laboratory/escola-2.3.1>

Ferramentas utilizadas:

- demoiselle-eclipse-3.7-SR2
- demoiselle-workspace
- demoiselle-maven2
- demoiselle-maven2-local-repo
- demoiselle-jboss-7.1

As ferramentas listadas acima fazem parte do Projeto Demoiselle Infra que pode ser encontrado no endereço <http://demoiselle.sourceforge.net/infra/1.2.4>

Funcionalidades Utilizadas

O Projeto Escola utiliza algumas funcionalidades do Framework Demoiselle 2.2.2 que ajudam na criação e facilitam o seu desenvolvimento. Abaixo temos uma breve descrição de como cada funcionalidade foi utilizada.

2.1. Funcionalidade de Exceções

Foi utilizado uma Exceção de Aplicação "@ApplicationException" chamada "EscolaException" para representar todos os erros acontecidos na aplicação. Uma exceção lançada na camada de Negócio ou Persistência pode ser transmitidas entre as camadas até a tela de apresentação "View" sem precisar programar essa transferência de mensagens entre as camadas, o Framework Demoiselle 2 traz essa facilidade para o desenvolvedor.



Nota

Referência:

http://demoiselle.sourceforge.net/docs/framework/reference/2.3.1/html_single/#excecao

2.2. Funcionalidade de Mensagens

Com a utilização do mecanismo de Mensagens do Framework Demoiselle 2 foi possível realizar a transmissão de mensagens entre as camadas.



Nota

Referência:

http://demoiselle.sourceforge.net/docs/framework/reference/2.3.1/html_single/#mensagem

2.3. Funcionalidade de Configurações

Utilizado para externalizar as configurações do JDBC (.properties) e dos Papeis dos Usuários (.xml)



Nota

Referência:

http://demoiselle.sourceforge.net/docs/framework/reference/2.3.1/html_single/#configuracao

2.4. Funcionalidade de Segurança

No Projeto Escola foi adicionado o Mecanismo de Segurança por método. Através do usuário logado na aplicação é possível saber quais são seus Papeis e através dele saber se o usuário logado tem permissão de execução do método. Todo o tratamento de Autenticação e Autorização do usuário é realizado pelo Framework Demoiselle.



Nota

Referência:

http://demoiselle.sourceforge.net/docs/framework/reference/2.3.1/html_single/#security

2.5. Funcionalidade de Relatório

Com a utilização do componente de relatórios do demoiselle (demoiselle-report) foi possível realizar a geração de relatórios em PDF e ODT de forma simples. Através de anotações e das classes do componente o trabalho de configuração, preparação, comunicação com os arquivos .jrxml ou .jasper foram abstraídos.



Nota

Referência:

<https://github.com/demoiselle/report/tree/2.1.0>

2.6. Funcionalidade de Transação

Com a utilização de uma única anotação "@Transactional" foi possível delegar todo o trabalho de inicializar a transação, verificação da transação e encerramento da transação para que o Framework Demoiselle, diminuindo consideravelmente o tempo de desenvolvimento.



Nota

Referência:

<http://demoiselle.sourceforge.net/docs/what-is-new/2.0.1/html/arquitetura.html#d0e214>

2.7. Funcionalidade de Inicialização

Onde antes se implementava "ServletContextListener" para executar algum trecho de código após a aplicação ser inicializada, com o Mecanismo de Inicialização é possível utilizar qualquer método de qualquer classe do seu projeto apenas utilizando a anotação "@Startup". No Projeto Escola essa anotação foi utilizada na classe DataSampleLoader para iniciar uma carga de registros na aplicação.




Nota

Referência:

http://demoiselle.sourceforge.net/docs/framework/reference/2.3.1/html_single/#inicializacao

2.8. Funcionalidade de E-mail

Outra facilidade na forma de componente foi a utilização do componente de e-mail do Demoiselle (demoiselle-mail). No Projeto Escola foi utilizado no envio de e-mail logo após o Professor cadastrar uma Nota no sistema, com poucas configurações já foi possível realizar o disparo dos e-mails.



Nota

Referência:

<http://demoiselle.sourceforge.net/docs/components/2.2.0/html/mail-master.html>

2.9. Funcionalidade de Paginação

É fornecido em tempo de execução um contexto de paginação, o qual tem escopo de sessão e armazena a informação de paginação de cada entidade (i.e., bean) que necessite de tal mecanismo. Esse contexto é compartilhado entre as diversas camadas da aplicação, especificamente entre as camadas de visão e persistência. Dessa maneira, a paginação dos dados é transparente para a camada intermediária (i.e., negócio) e não interfere na modelagem das classes de um projeto.



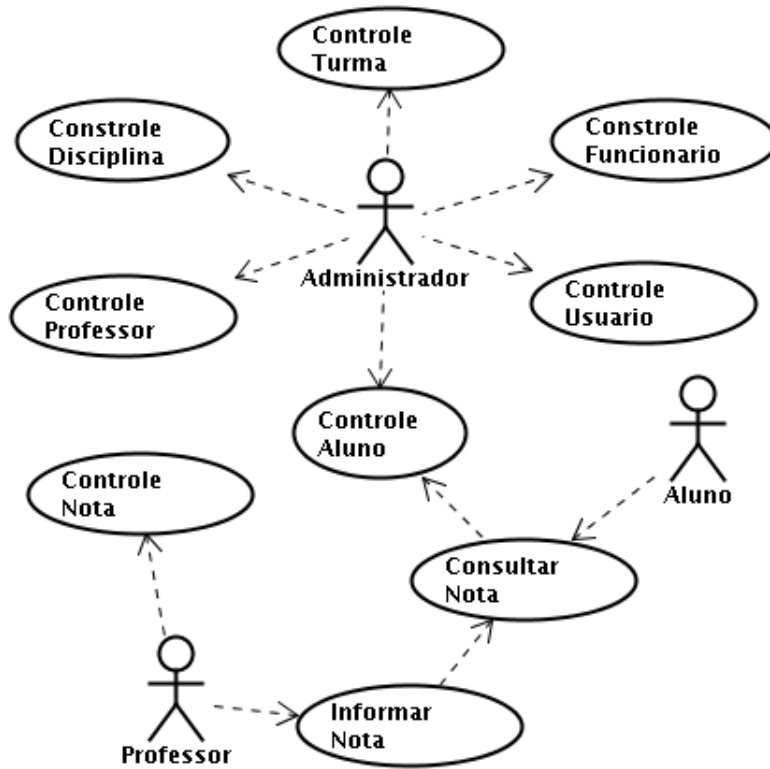
Nota

Referência:

http://demoiselle.sourceforge.net/docs/framework/reference/2.3.1/html_single/#paginacao

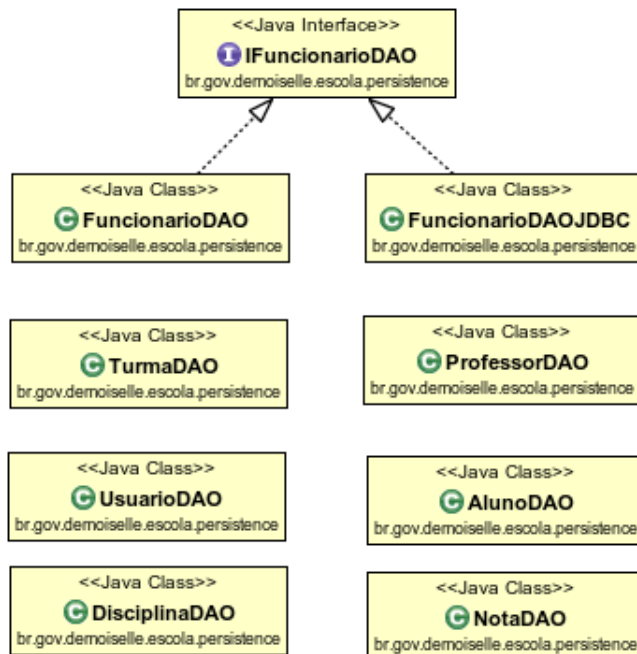
Caso de Uso

3.1. Caso de Uso

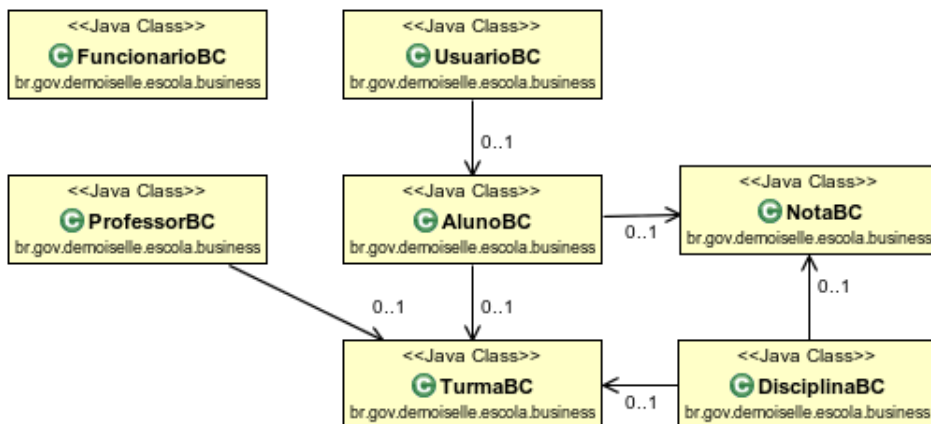


Camadas

4.1. Camada de Persistência



4.2. Camada de Negócio



4.3. Camada de Visão

