INTRODUÇÃO

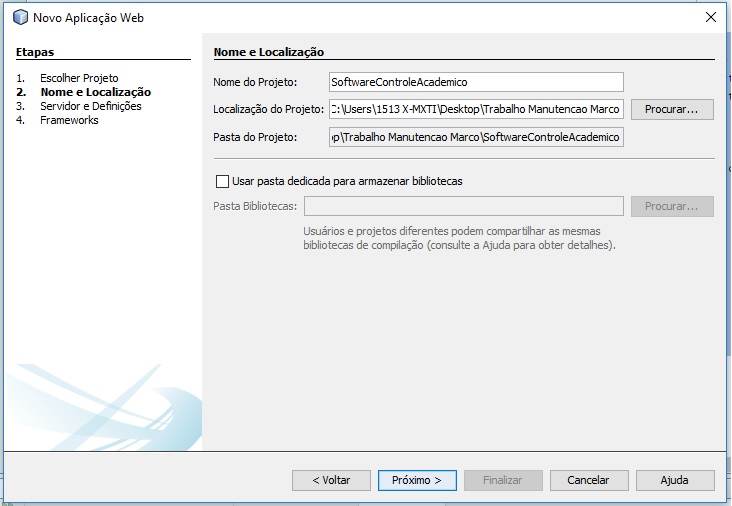
O seguinte documento irá apresentar os passos realizados durante a manutenção do software de controle acadêmico, também conhecido como SCA.

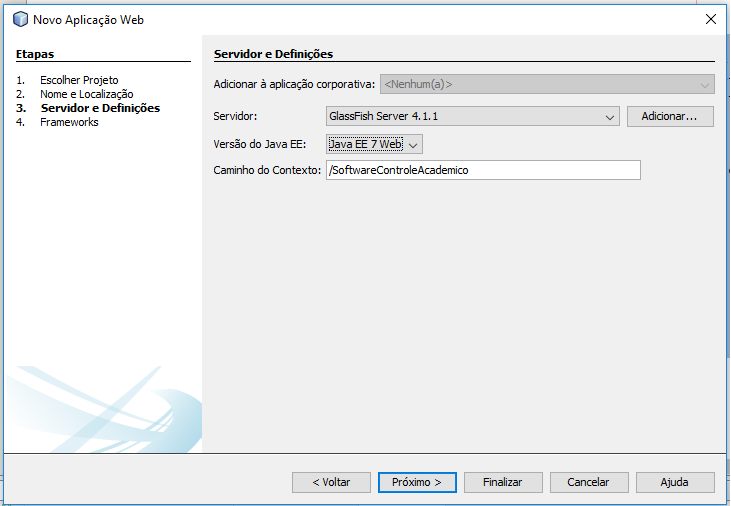
PASSO 01 – BANCO

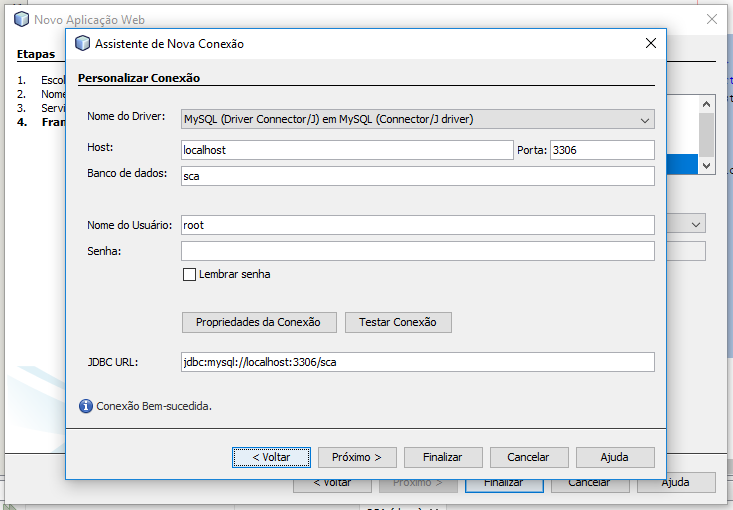
O seguinte projeto utiliza um banco de dados para persistir as informações adicionadas no programa. Antes de iniciar o projeto foi realizado a instalação do XAMP, iniciados os serviços do Apache e do MySql. Criado também um banco de dados vazio com o nome de `sca`.

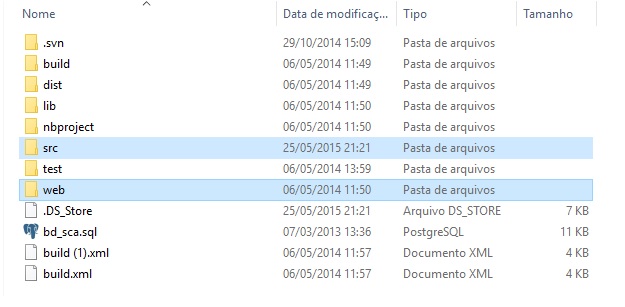
PASSO 02 – INICIALIZAÇÃO

A inicialização do projeto não foi realizada pela ferramenta de importação do NetBeans. Em outras tentativas de inicialização do projeto foi observado erro na execução do programa, portanto, a inicialização se deu através da criação de um novo projeto e posterior copia dos arquivos encontrados nos diretórios SRC e WEB do projeto original.



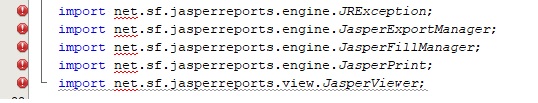






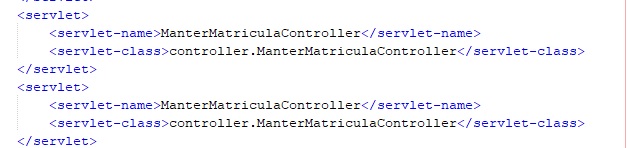
PASSO 03 – BIBLIOTECA

Depois da etapa dois o NetBeans já informa que existem importações incorretas dentro do programa, está sendo resultado da falta de uma biblioteca. Realizado a copia da biblioteca jasperreports-5.0.4 que se encontrava no diretório lib do projeto original, realizando posteriormente a adição dessa biblioteca no projeto.



PASSO 04 – DUPLICIDADE DE SERVLET

Após executar o programa tem-se como resultado erro por duplicidade de valor (ManterMatriculaController) para servlet dentro do arquivo web.xml. Removido a duplicidade para correção.



PASSO 05 – PÁGINA INICIAL

Removido o arquivo index.html do projeto, arquivo adicionado durante a criação do projeto, e alterado o atributo welcome-file no arquivo web.xml para index.jsp.

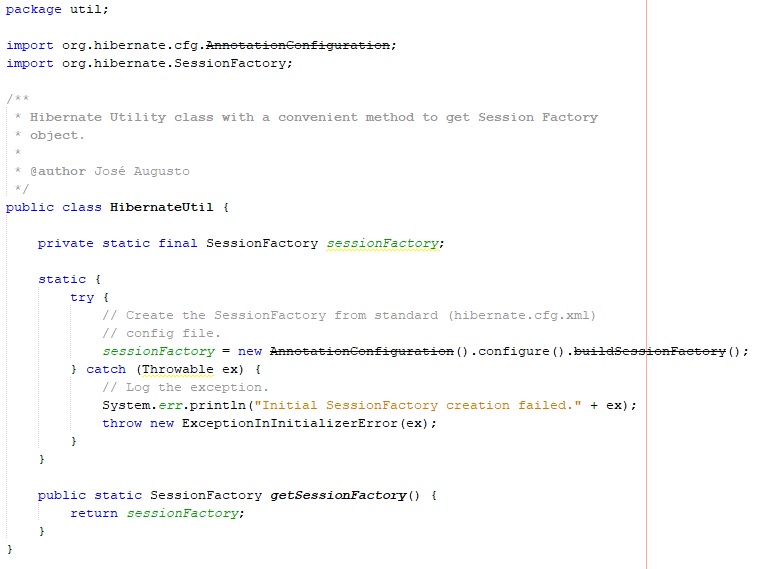


PASSO 06 – HIBERNATEUTIL

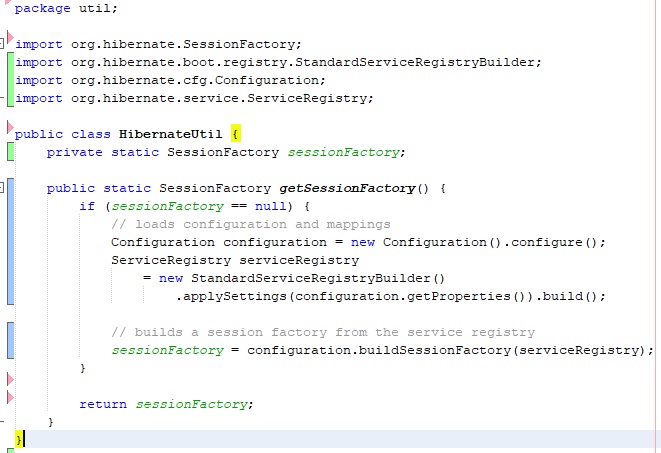
Após executar o programa e acessar algum menu que realiza busca de informações no banco de dados é apresentado mensagem de erro relacionado a inicialização do HibernateUtil. Solucionado o problema alterando a forma que esta cria a conexão.



Antes:



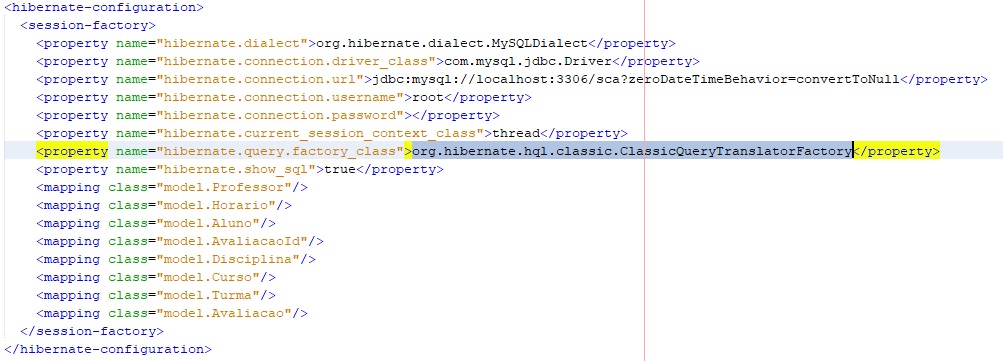
Depois:



PASSO 07 – QueryTranslatorFactory

Executando o programa é apresentado um novo problema referente a não inicialização do atributo QueryTranslatorFactory dentro do arquivo hibernate.cfg.xml. Corrigido atribuindo valor correto para o elemento.

Antes:



Depois:



PASSO 08 – CRIAÇÃO DAS TABELAS

Ao executar o programa e abrir novamente um menu que realiza consultas no banco tem-se como erro problemas ao encontrar a tabela para buscar as informações. Resolvido o problema adicionando um novo atributo nas configurações.



Atributo adicionado:



PASSO 09 – NESTED TRANSACTIONS

A principio o programa executa normalmente, mas após abrir e fechar diversos menus ele acaba por retornar erro relacionado as conexões realizadas. O problema foi resolvido ao alterar o método que estava sendo utilizado para iniciar a conexão com o banco de dados.



Antes:



Depois:





PASSO 10 – LAZZY

Após a alteração algumas chamadas obtiveram problemas, sendo este resolvido com uma nova configuração.



Atributo adicionado:



PASSO 11 – BD.CLASS

Alteração do atributo senha na conexão realizada pela classe BD.java

PASSO 12 – BIBLIOTECAS

Para gerar relatórios foram adicionadas todas as outras bibliotecas do projeto original.

PASSO 13 – MANUTENÇÃO INCREMENTAL

Adicionado ao programa um novo pedido. Anteriormente este realiza cálculo de avaliações referente a duas notas informadas e se necessário o valor de uma avaliação final. Com a modificação agora existem 3 campos para a consolidação de notas.

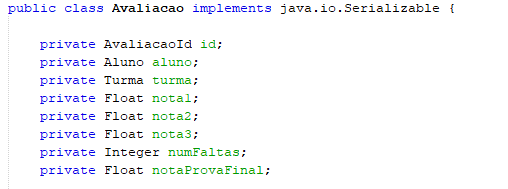
Foi necessário alterar o modelo, adicionando um novo atributo.

Não foi necessário alterar o dao, pois o hibernate já faz as alterações com base no modelo de classe.

Alterado o controller ManterAlunoNotaFrequenciaController apenas adicionando o atributo na hora de enviar as informações para a view.

Alterado 3 classes de visualização sendo elas consultarNotaFrequencia, manterAlunoNotaFrequencia e manterNotaFrequencia adicionando as colunas para apresentação do novo campo e os cálculos que são realizados dentro da própria página.

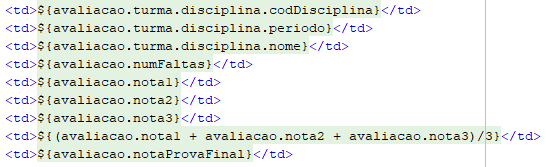
Alteração no modelo Avaliacao:



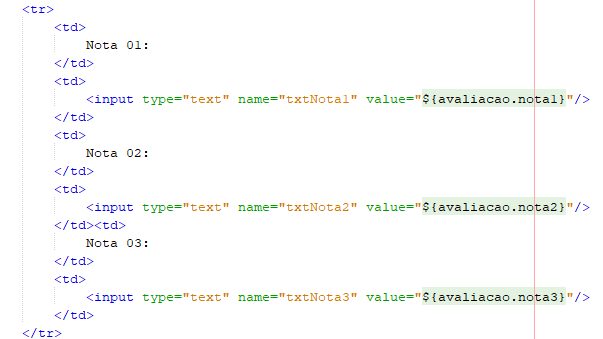
Alteração Controller ManterAlunoNotaFrequenciaController:



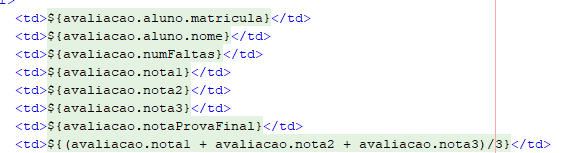
Alteração View consultarNotaFrequencia



Alteração View manterAlunoNotaFrequencia



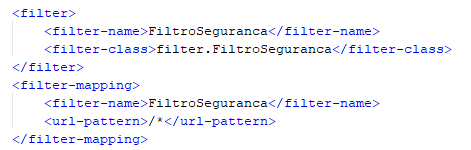
Alteração View manterNotaFrequencia



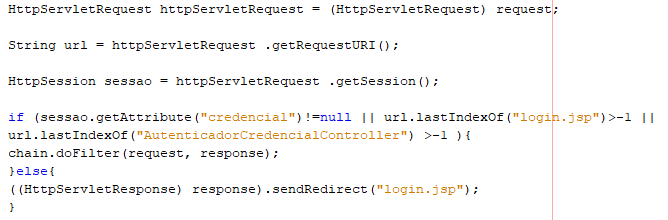
PASSO 14 – MANUTENÇÃO CORRETIVA CONTROLE DE CREDENCIAL

Adicionado ao programa o conceito de sessão e politicas de permissão para que as informações no documento de requisitos possam fazer sentido. Foi criado uma nova classe, Credenciais, e através de um de seus atributos acesso a apresentação do menu do sistema é modificado. Lembrando que só pode acessar o sistema se tiver as credenciais de acesso.

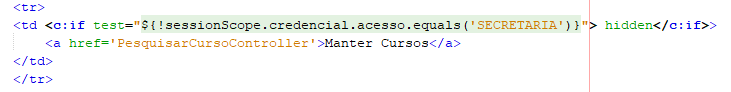
Mapeamento do Filtro:



Filtro:



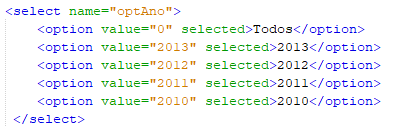
Exemplo de implementação em um menu:



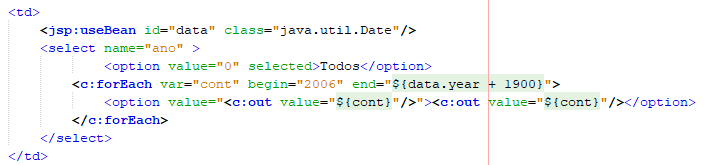
PASSO 15 – MANUTENÇÃO CORRETIVA APLICANDO OPÇÕES DE ANO

Alterado a apresentação de ano para ser automática e não com valores estáticos.

Antes:



Depois:



PASSO 16 – MANUTENÇÃO CORRETIVA IMPLEMENTAÇÃ DA FUNÇÃO FILTRO PARA TURMA

O botão pesquisar dentro do view pesquisarTurma.jsp não aplicava funcionalidade. Implementado código para que o mesmo execute a função de filtro corretamente.

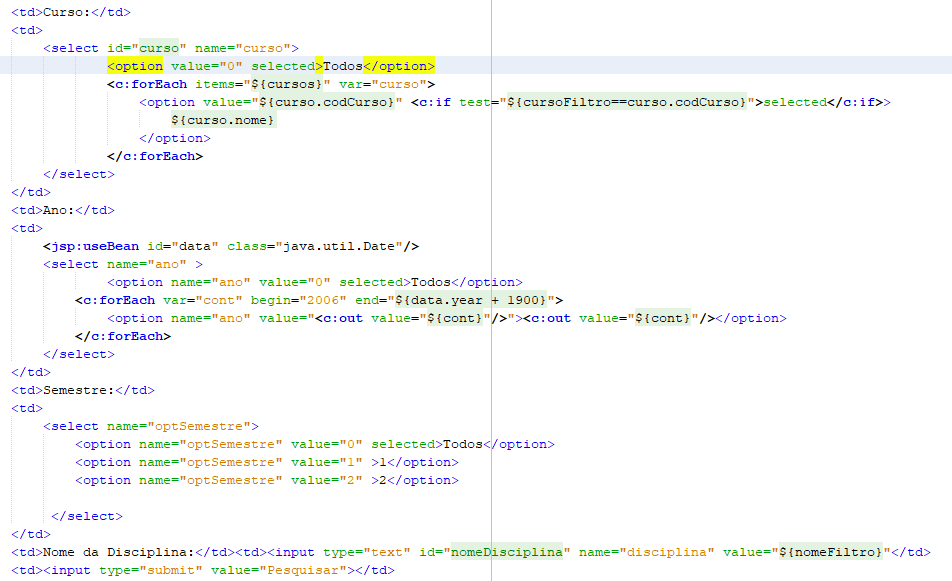
Utilizando o mesmo padrão utilizado para o filtro de disciplinas, foi modificado os seguintes arquivos: index.jsp; pesquisarTurma.jsp; Turma.java; TurmaDao.java; PesquisarTurmaController.java.

Index: Atribuindo referência inicial de que não terá filtro.

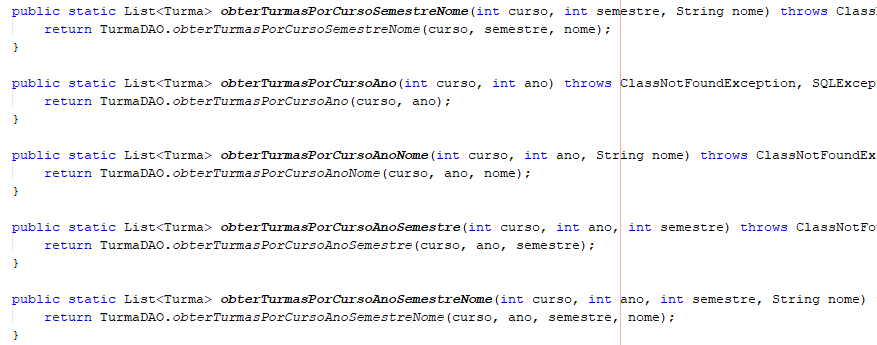


PesquisarTurma: Adicionando funcionalidade para o botão pesquisar e nomeando o campos para parse posterior.

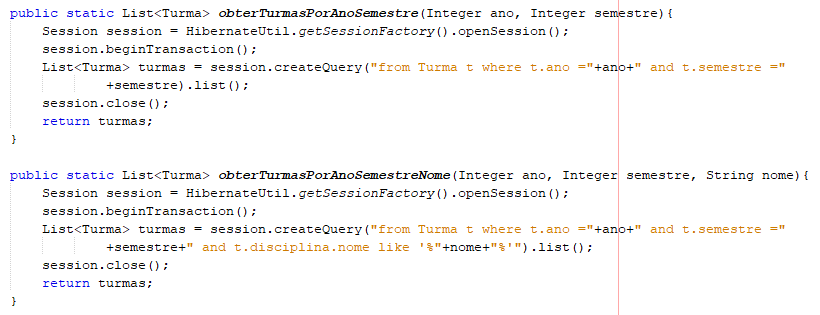




Turma: Adicionando os métodos de busca



TurmaDao: Implementando as queries que irão realizar a busca de informações filtradas no banco.



PesquisarTurmaController: Implementando a arvore de decisão para cada caso possível entre os filtros disponíveis.

