Universidade Federal do ABC

Centro de Matemática, Computação e Cognição

Manual dos Sistemas de Alocação Didática e Admin

Autor: Erick Alves Augusto

Santo André

2017

Sumário

1. INTRODUÇÃO

Os sistemas de Alocação Didática e Admin são sistemas desenvolvidos para auxiliar no planejamento de turmas do CMCC. O sistema de Alocação é apenas para uso dos docentes, enquanto o Admin é de uso dos coordenadores e diretores do CMCC.
O sistema de Alocação tem funções separadas em fases:

A fase de Afinidades permite que os docentes escolham as disciplinas que eles dão preferência a dar aula, atualmente essa fase se mantém sempre aberta, pois ainda não foi definido um critério para o seu período de uso.

A fase 1, ou Planejamento Anual, é a fase onde os docentes terão acesso a uma prévia do que será ofertado durante o ano letivo dividido pelos três quadrimestres. Nessa fase os docentes poderão escolher a quantidade de créditos que eles pretendem ministrar em cada quadrimestre e quais das disciplinas já planejadas eles preferem dar aula. Essas escolhas tem opções de turno e campus, além de ser possível o docente dar preferência a ministrar as aulas teóricas ou práticas se forem matérias como Processamento da Informação. Após escolher as turmas que deseja, o docente pode dar uma ordem de preferência a cada uma das turmas que ele marcou no planejamento.

A fase 2, ou Escolha de Turmas, é a fase em que os docentes irão solicitar as turmas que serão ofertadas no quadrimestre. Essa fase será aberta apenas quando estiver no período de planejamento quadrimestral. Nela é exibida uma lista com todas as turmas que serão ofertadas para que os docentes possam fazer a solicitação e os coordenadores possam analisar.

O sistema Admin tem dois níveis de acesso, o dos coordenadores e dos diretores. O nível de acesso do coordenadores exibe apenas os relatórios das fases do sistema de Alocação, enquanto o nível dos diretores dá acesso a todo o sistema, tanto aos relatórios quanto ao gerenciamento das fases do sistema de Alocação e o cadastro de dados.

Os relatório são separados da mesma forma que as fases do sistema de Alocação. O relatório de Afindades exibe uma lista com todas as disciplinas que os docentes escolheram ou uma lista de docentes que escolheram um determinada disciplina. O relatório da fase 1 mostra a lista de docentes que colocaram certa turma no planejamento ou uma lista de todas as turmas planejadas pelo docente e quantos créditos ele pretende ministrar. O relatório da fase 2 mostra quais docentes solicitaram cada turma ou uma lista de todas as turmas que cada docente solicitou e qual a ordem de prioridade ela tinha no planejamento da fase 1.

2. FICHA TÉCNICA DOS SISTEMAS

2.1. SISTEMA DE ALOCAÇÃO DIDÁTICA

O sistema de Alocação Didática foi desenvolvido usando o padrão de projeto MVC. As sessões seguintes mostram as classes componentes de cada pacote do modelo e seus métodos.

2.1.1. MODEL

Afinidade

Entidade com os atributos String estado, Date dataAcao, Pessoa pessoa, Disciplina disciplina.

Possui uma classe interna Id com os atributos Long pessoaId, Long disciplinaId.

Credito

Entidade com os atributos Long ID, double quantidade, int quadrimestre.

Disciplina

Entidade com os atributos Long ID, String nome, String eixo, String codigo, String curso, Set<Afinidade> afinidades, Set<OfertaDisciplina> ofertasDisciplinas.

Disp

Entidade com os atributos Long id, String ordemPreferencia, String tp, Pessoa pessoa, OfertaDisciplina ofertaDisciplina.

Disponibilidade

Entidade com a mesma função que Disp, mas não foi mais usada devido a problemas para adaptar os atributos que estavam contidos em uma subclasse privada.

Deixou de ser usada e foi criada a entidade Disp no lugar.

Docente

Entidade com os atributos String areaAtuacao, List<Credito> creditos, double creditoQuad.

Fase

Entidade com os atributos Long id, boolean fase1, boolean afinidades, boolean fase2_quad1, boolean fase2_quad2, boolean fase2_quad3, Date dataAtual.

Horario

Entidade com os atributos String dia, String hora, String sala, String periodicidade.

OfertaDisciplina

Entidade com os atributos Long ID, String curso, Disciplina disciplina, int t, int p, String turno, String campus, int numTurmas, String periodicidade, int quadrimestre, String funcao, Set<Disponibilidade> disponibilidades, Set<Disp> dispo.

Pessoa

Entidade com os atributos Long ID, String nome, String siape, String email, String centro, boolean adm, Set<Afinidade> afinidades, Set<Disponibilidade> disponibilidade, Set<Disp> dispo.

Turma

Entidade com os atributos Long ID, String curso, String codturma, String turno, String campus, Disciplina disciplina, List<Horario> horarios.

TurmaDocente

Entidade com os atributos Long id, Long idTurma, Long idDocente.

Preferencias

Entidade com atributos Long ID, int preferenciaHorarios, String observacoes, int quadrimestre, Long pessoa_id.

2.1.2. CONTROLLER

AfnidadeController

- Método getDisponiveis retorna a lista de disciplina disponíveis.
- Método getCursos retorna a lista de cursos para usar nos filtros.
- Método getEixos retorna a lista de eixos para usar nos filtros.
- Método filtrar aplica os filtros para busca.
- Método limparFiltro remove os filtros e retorna a lista completa de disciplinas.
- Método salvarAfinidade salva a disciplina selecionada como afinidade.
- Método removerAfinidade remove a disciplina da lista de afinidades do docente, mas a marca como uma afinidade removida e deixa salva na lista do docente.
- Método listarTodas busca todas as afinidades no banco de dados.
- Método buscar busca a afinidade por id.
- Método prepareCreate retorna a página Create.
- Método prepareAfinidades retorna a página DefinirAfindade.

CreditoController

- Método creditosQuad retorna quantos créditos o docente tem por quadrimestre.
- Método adicionaCredito soma mais créditos para o docente no quadrimestre.
- Método diminuiCredito diminui a quantidade de créditos que o docente tem por quadrimestre.

DisciplinaController

- Método init inicia os componentes da página que devem ser pré-carregados.
- Método converteEixo busca o eixo da disciplina com base na sigla.
- Método converteCurso busca o curso da disciplina com base na sigla.
- Método converteBI busca o BI da disciplina com base na sigla.
- Método cadastrarDisciplinas faz o cadastro das disciplinas usando um arquivo csv.
- Método deleteAll apaga todas as disciplinas do banco de dados.
- Método getDisciplinaLazyModel carrega um objeto lazymodel com todas as disciplinas para exibir nas páginas.
- Método recriarModelo anula o lazymodel.
- Método getDisciplinaSalvar cria uma nova disciplina para salvar.
- Método qdtAfinidades conta quantos docentes marcaram a disciplina como afinidade.

- Método preencherAfinidadesDisciplina monta uma lista com os docentes que marcaram a disciplina como afinidade.
- Método verSoAdicionadas exibe apenas as disciplinas marcadas como afinidade e não foram removidas.
- Método limparSelecao remove todo o conteúdo dos datamodels.
- Método getDisciplinaDataModel preenche o datamodel de disciplinas.
- Método filtrar aplica os filtros de busca por disciplina.
- Método limparFiltro retorna a lista completa de disciplinas.
- Método salvar salva uma nova disciplina.
- Método editar edita uma disciplina.
- Método delete deleta uma disciplina.
- Método buscar busca uma disciplina pelo id.
- Método listarTodas busca a lista completa de disciplinas.
- Método prepareCreate retorna a página Create.
- Método index retorna para a página index. Não é mais usada.
- Método prepareEdit retorna a página editDisciplina.
- Método prepareView retorna a página View.

DisponibilidadeController

- Método getCreditosEscolhidos busca a quantidade de créditos que o docente salvou para o quadrimestre.
- Método changeColor altera a cor da quantidade de créditos do docente.
- Método salvarCreditos adiciona créditos para a lista do docente no quadrimestre.
- Método salvaPreferencias salva a configuração de horário e as observações do docente que foram marcadas no planejamento do quadrimestre correspondente.
- Método editarPreferencias edita a preferência que o docente já salvou para o quadrimestre selecionado na fase I.
- Método getOfertasEtapa1 busca a lista de ofertas de disciplinas do primeiro quadrimestre.
- Método getDataModel preenche o datamodel de oferta de disciplinas.
- Método setFuncaoOferta atribui a escolha de teoria e/ou prática do docente para a disciplina.
- Método adicionaCredito soma a quantidade de créditos da disciplina para o docente.
- Método removeCredito subtrai a quantidade de créditos da disciplina para o docente.
- Método getDispdataModel preenche o datamodel de disponibilidades.
- Método getDispdataModel2 preenche o datamodel de disponibilidades para edição.
- Método salvarDisponibilidade salva uma nova disponibilidade para o docente.
- Método getDDataModel preenche o datamodel para exibição.
- Método onRowReorder reordena o datamodel e altera os valores das prioridades do planejamento do docente.
- Método getDModel2 preenche o datamodel para edição.
- Método getMsg cria a mensagem de aviso sobre as disciplinas do BC&T.
- Método verificarBCT conta quantas disciplinas do BC&T o docente escolheu.
- Método salva a disponibilidade do docente.
- Método onRowSelect pega o objeto da linha selecionada.
- Método deleteEscolha deleta uma disponibilidade do docente do banco de dados.
- Método deleteDisp reordena as ordens de prioridade do docente quando uma disponibilidade é removida.
- Método onCellEdit edita o campo de ordem de prioridade. Não é mais usado.
- Método getOrdem pega a ordem de prioridade da disponibilidade.
- Método getTipoOferta verifica se o docente escolheu teoria e/ou prática.
- Método getTipoDisp verifica se a disponibilidade está marcada como teoria e/ou prática.
- Método sucessoFase1 avisa que as alterações foram salvas.
- Método listarTodasQuad busca todas as ofertas do quadrimestre.
- Método prepareAfinidades retorna a página DefinirAfinidade.

- Método preparePlanejamento1, preparePlanejamento2, preparePlanejamento3 retornam a página Quadrimestre.
- Método init inicia os componentes do filtros ao carregar a página.
- Método iniciarFiltro faz um tratamento nos filtros para que eles não estejam nulos.
- Método onChangeCampus faz a mudança do filtro campus e já atualiza a lista de disciplinas.
- Método onChangeTurno faz a mudança do filtro turno e já atualiza a lista de disciplinas.
- Método onChangeAfinidades faz a mudança do filtro turno e já atualiza a lista de disciplinas.
- Método filtrarCursos busca os itens marcados da lista de cursos para fazer a filtragem de disciplinas com base neles.
- Método filtrarTurmasQuad faz a filtragem com base nos valores escolhidos pelo usuário.
- Método limparFiltroQuad remove os filtros e retorna a lista completa de disciplinas.

DocenteController

- Método getDocenteSalvar cria um novo docente no banco de dados.
- Método getDocenteLazyModel carrega o lazymodel com a lista de docentes.
- Método init inicia o lazymodel de docentes ao carregar a página.
- Método cadastrarDocentes método para cadastrar docentes usando um arquivo csv.
- Método cadastrarDocentesCMCC método para cadastrar docentes usando um arquivo que contenha apenas os docentes do CMCC.
- Método trataNome faz o tratamento dos nomes para padronizar irregularidades nos conectores que existem em alguns sobrenomes.
- Método qtdAfinidades retorn quantas afinidades o docente tem.
- Método preencherAfinidadesDoDocente busca todas as afinidades do docente.
- Método verSoAdicionadas exibe apenas as afinidades que não foram removidas pelo docente.
- Método salvarCreditos salva o planejamento de créditos do docente.
- Método qtdDisponibilidades retorna a quantidade de disponibilidades que o docente salvou.
- Método creditosQuad retorna quantos créditos o docente marcou no quadrimestre.
- Método preencherDisponibilidadesDoDocente preenche o datamodel de disponibilidades do docente.
- Método getDocenteDataModel preenche o datamodel de docentes.
- Método filtrar faz a filtragem dos docentes.
- Método limparFiltro remove os filtros e retorna a lista completa de docentes.
- Método buscar busca um docente por id.
- Método salvar salva um novo docente.
- Método editar edita um docente.
- Método delete deleta um docente do banco de dados.
- Método listarTodas busca todos os docentes.
- Método prepareEdit retorna a página editDocente.

FaseController

- Método formData padroniza o formato da data para salvar no banco de dados.
- Método optToString converte a opção selecionada para String.
- Método Verifica analisa qual opção foi selecionada.
- Método desabilitarFases atribui null à todas as opções.
- Método saveData salva as informações no banco de dados.

Filtros

- Método getFiltrosEixos retorna a lista de eixos.
- Método getFiltrosCursos retorna a lista de cursos.
- Método getFiltrosCentros retorna a lista de centros.

- Método getFiltrosAreaAtuacao retorna a lista das áreas de atuação do CMCC.
- Método filtrarDocente filtra um docente pelos filtros que foram selecionados.
- Método limparFiltroDocente remove os filtros.
- Método completeDisciplina faz o autocomplete no nome da disciplina.
- Método completeEixo faz o autocomplete no nome do eixo.
- Método completeCurso faz o autocomplete no nome do curso.
- Método completeArea faz o autocomplete no nome da área.

LoginBean

- Método doLogin faz o login do usuário verificando se ele existe no banco de dados e a senha foi encontrada no servidor LDAP da UFABC.
- Método doLogin2 faz o login do usuário verificando se ele existe no banco de dados sem procurar pela senha.
- Método page verifica qual a fase ativa no banco de dados e retorna a página correspondente.
- Método isLogado verifica se o usuário está logado.
- Método doLogout faz o logout do usuário no sistema.
- Class LDAP é uma subclasse que contêm os métodos para o acesso ao servidor LDAP.

OfertaDisciplinaController

- Método getDataModel preenche o datamodel de oferta de disciplinas.
- Método prepareQuad1, prepareQuad2, prepareQuad3 retornam a página de oferta dos quadrimestres. Não é mais usado.
- Método init inicia os lazymodels de ofertas de acordo com o quadrimestre. Não é mais usado.
- Método filtrarOfertas filtra as ofertas de disciplina.
- Método limparFiltroOfertas remove as opções de filtro.
- Método preencherDisponibilidadesOferta preenche o datamodel com as disponibilidades.
- Método listarTodas retorna todas as ofertas de disciplina.
- Método salvarNoBanco salva uma oferta no banco.
- Método buscar busca uma oferta pelo id.
- Método editar edita uma oferta.
- Método deleteAll deleta todas as ofertas do banco de dados.
- Método deleteAllQuad deleta as ofertas de disciplina pelo quadrimestre.
- Método qdtDocentesDisponibilidade retorna a quantidade de disponibilidades do docente.
- Método getDocentesPorDisciplina retorna a lista de docentes pode disciplina.
- Método cadastrarOfertasQuad1, cadastrarOfertasQuad2, cadastrarOfertasQuad3 cadastra as ofertas dos quadrimestres usando um arquivo csv para cada quadrimestre.

PessoaController

- Método getPessoaSalvar cria uma nova pessoa.
- Método prepareCreate retorna a página Create.
- Método index retorna a página index. Não é mais usado.
- Método prepareEdit retorna a página editDocente.
- Método prepareView retorna a página View.
- Método getPessoaLazyModel preenche o lazymodel de pessoa.
- Método getAdmLazyModel preenche o lazymodel de adm.
- Método init inicia os datamodel e lazymodel.
- Método listarTodas retorna todas as pessoas.
- Método salvarNoBanco salva a pessoa no banco de dados.
- Método buscar busca uma pessoa por id.

- Método salvar salva a pessoa no banco de dados.
- Método editar edita uma pessoa.
- Método delete deleta uma pessoa do banco de dados.
- Método salvarAdm salva um novo adm no banco de dados.
- Método removerAdm remove um adm do banco de dados.
- Método cadastrarPessoas cadastra pessoas usando um arquivo csv.
- Método trataNome padroniza irregularidades que possam existir nos conectivos dos nomes.
- Método completePessoa faz o autocomplete no nome das pessoas.
- Método completeCentro faz o autocomplete no nome dos centros.

TurmaController

- Método cadastrarTurmas2 cadastra as turmas usando um arquivo csv. O original não estava funcionando direito, então foi feito outro método.
- Método criaTurma cria o objeto turma separando as informações do arquivo.
- Método deleteTurmas deleta todas as turmas.
- Método getTurmalazymodel preenche o lazymodel de turmas.
- Método getListaRequisicoes preenche o lazymodel com as turmas que o docente solicitou.
- Método solicitacoes retorna a quantidade de solicitações por turma.
- Método horariosTurma monta uma String com os horários da turma.
- Método onChangeCampus aplica o filtro ao selecionar o campus.
- Método onChangeTurno aplica o filtro ao selecionar o turno.
- Método onChangeAfinidades aplica o filtro ao selelcionar as afinidades.
- Método onChangePlanejamento aplica o filtro ao selecionar o planejamento.
- Método filtrar aplica os filtros escolhidos.
- Método limparFiltros remove os filtros e retorna a lista completa.
- Método onRowSelect retorna o objeto da linha selecionada.
- Método onRowUnselect remove a seleção da linha.
- Método getDocenteSchedule retorna o conteúdo do schedule do docente.
- Método incluirTurma salva uma turma nas escolhas do docente.
- Método verificar analisa se existe um conflito de horários entre as turmas já salvas pelo docente e a que ele quer salvar.
- Método excluirTurma exclui a turma selecionada das salvas pelo docente.
- Método preecherSchedule preenche o schedule do docente com as turmas que ele salvou.
- Método init inicia os valores do filtros ao carregar a página.
- Método preencheDocente converte os objetos turma para preencher os schedule com as informações das turmas.
- Método conversorDia converte a informação do dia para preencher o schedule.
- Método converterSigla converte o nome da disciplina em uma sigla para preencher o schedule.

2.1.3. FACADE

AbstractFacade

- Método save é um método padrão herdado por todas as classes filhas que salva um objeto na entidade correspondente.
- Método edit é um método padrão herdado por todas as classes filhas que edita um objeto na entidade correspondente.
- Método merge é um método padrão herdado por todas as classes filhas que faz um merge em um objeto na entidade correspondente.
- Método remove é um método padrão herdado por todas as classes filhas que remove um objeto na entidade correspondente.
- Método find é um método padrão herdado por todas as classes filhas que busca um objeto na entidade correspondente.

- Método findAll é um método padrão herdado por todas as classes filhas que busca todos os objetos na entidade correspondente.
- Método findRange é um método padrão herdado por todas as classes filhas que busca um objeto dentro de um arranjo na entidade correspondente.
- Método count é um método padrão herdado por todas as classes filhas que conta a quantidade de objetos na entidade correspondente.

AfinidadeFacade

- Método remove sobrescrito do abstract.

CreditoFacade

- Método creditoQuadrimestre retorna a quantidade de créditos que o docente tem no quadrimestre.

DisciplinaFacade

- Método inicializarColecaoAfinidades busca a lista de disciplinas no banco de dados.
- Método findByName faz a busca pelo nome da disciplina.
- Método findByCodOrName faz a busca pelo nome ou código da disciplina.
- Método findByEixoCurso faz a busca pelo curso ou eixo.

DispFacade

- Método findByDocente busca a lista de disponibilidades pelo docente.
- Método findByDocenteQuad busca a lista de disponibilidades pelo docente e o quadrimestre.
- Método findByAreaQuad busca a lista de disponibilidades pela área e o quadrimestre.
- Método findByDiscTurCamQuad busca a lista de disponibilidades pela disciplina, turma, campus e quadrimestre.
- A classe DisponibilidadeFacade possui os mesmos métodos que o DispFacade, mas já não é mais usado.

DocenteFacade

- Método findByName faz a busca do docente pelo nome.
- Método findByCentroArea faz a busca do docente pelo centro e área.
- Método findByArea faz a busca do docente pela área.

FaseFacade

- Método achaMax retorna o último registro na tabela do banco de dados.

OfertaDisciplinaFacade

- Método inicializarColecaoDisponibilidades retorna a lista de oferta de disciplinas.
- Método filtrarAfinidTurnCampQuad faz a busca de oferta de disciplinas pela afinidade, turma, campus e quadrimestre.
- Método filtrarEixoCursoTurnoCampusQuad faz a busca de oferta de disciplinas pelo eixo, curso, turno, campus e quadrimestre.
- Método findAllQuad faz a busca de todas as ofertas de disciplinas do quadrimestre.

PessoaFacade

- Método findByName busca uma pessoa pelo nome.
- Método findByUsername busca uma pessoa pelo nome de usuário.

- Método listAdms retorna a lista de adms.
- Método listDocentes retorna a lista de docentes.

PreferenciasFacade

- Método verificaPreferencia busca no banco de dados se existe uma preferência salva pelo docente no quadrimestre selecionado na fase I.

TurmaDocenteFacade

- Método listTurmas retorna a lista de TurmaDocente pelo id do docente.
- Método TurmaSelecionada retorna o objeto TurmaDocente pelo id do docente e da turma.
- Método totalSolicitacoes retorna o total de solicitações da turma.

TurmaFacade

- Método listByID busca turma pelo id.
- Método filtrarTurmas retorna a lista de turmas com base nos filtros.

2.1.4. ÚTIL E CONTROLLERCONVERTER

O pacote Útil contêm as classes de configuração dos datamodel e lazymodel. Sempre que for necessário criar um novo objeto desse tipo que não possuir uma classe para ele nesse pacote é só criar uma nova seguindo as mesmas configurações das classes já existentes.

O pacote Converter possui alguns métodos usados para fazer o autocomplete das listas de filtros. Nunca foi necessário acrescentar ou modificar nada nesse pacote, apenas os outros foram usados até o momento.

2.2. SISTEMA ADMIN

O sistema Admin inicialmente estava integrado dentro do sistema de Alocação Didática, apenas as funções dele teriam um controle de acesso. Após algumas reuniões foi decidido que o sistema devia ser dividido em dois e o sistema Admin ficaria responsável por todo o gerenciamento de dados do sistema de Alocação Didática.

Para fazer o sistema Admin foi feita uma cópia de todas as classes e pacotes do sistema de Alocação Didática. A única diferença ficou no pacote View, onde foram deixadas apenas as páginas que seriam usadas no sistema Admin. O método de login foi o único alterado para poder direcionar o usuário para a página inicial criada para o sistema Admin.

Além das mudanças nas páginas foram criados novos métodos e classes nos pacotes, mas apenas acrescentando às já existentes. Nas próximas sessões serão mostradas apenas as classes e métodos que diferem do sistema de Alocação Didática.

2.2.1. MODEL

Possui os mesmos componentes na camada Model que o sistema de Alocação Didática, apenas uma classe foi criada diretamente para ele.

NivelAcesso

Entidade com os atributos Long ID, int nivel, Long pessoa_id.

2.2.2. CONTROLLER

Apenas algumas alterações foram feitas nos métodos de algumas classes do pacote Controller do sistema Admin ou novos métodos foram acrescentados.

DisponibilidadeController

- Método preferenciaHorarios converte o valor numérico da configuração de horários escolhida pelo docente em texto para ser exibo no resumo da fase I.
- Método retornaObservacoes exibe no resumo da fase I o texto encontrado no banco de dados contendo as observações do docente.

DocenteController

- Método totalTurmas busca a quantidade de turmas que o docente escolheu na fase 2.
- Método buscarSelecionadas busca todas as turmas que o docente escolheu na fase 2 do sistema de Alocação Didática.
- Método prioridadeTurma verifica se a disciplina e o turno que ela é ofertada estavam no planejamento do docente e retorna a ordem de prioridade que ele marcou.

LoginBean

- Método page foi alterado para que o usuário seja direcionada para a página de index do sistema Admin.

PessoaController

- Método prepareFase foi alterado para verificar o nível de acesso do usuário antes de direcioná-lo para a página de gerenciamento de fases, se o usuário não tiver permissão para acessar essa página ela será direcionado novamente para o index.
- Método prepareAfinidades foi alterado para verificar o nível de acesso do usuário antes de direcioná-lo para a página de gerenciamento de fases, se o usuário não tiver permissão para acessar essa página ela será direcionado novamente para o index.
- Método prepareFaseI foi alterado para verificar o nível de acesso do usuário antes de direcioná-lo para a página de gerenciamento de fases, se o usuário não tiver permissão para acessar essa página ela será direcionado novamente para o index.
- Método prepareFaseII foi alterado para verificar o nível de acesso do usuário antes de direcioná-lo para a página de gerenciamento de fases, se o usuário não tiver permissão para acessar essa página ela será direcionado novamente para o index.
- Método prepareCadastro foi alterado para verificar o nível de acesso do usuário antes de direcioná-lo para a página de gerenciamento de fases, se o usuário não tiver permissão para acessar essa página ela será direcionado novamente para o index.
- Método salvarAdm foi alterado para salvar o nível de acesso do novo usuário do sistema Admin no banco de dados para consulta quando o usuário entrar no sistema e for acessar as páginas de resumo e gerenciamento.
- Método removerAdm foi alterado para remover o registro do usuário da tabela de nível de acesso no banco de dados.

TurmaController

- Método novaTurma permite que seja adicionada uma nova turma no banco de dados caso seja criada uma turma que não estava no arquivo do planejamento do quadrimestre.
- Método preencherResumoTurmas busca as turmas do quadrimestre e preenche o datamodel para ser visualizado no resumo da fase 2.

2.2.3. FACADE

NivelAcessoFacade

- Método achaAdm verifica se o docente é adm.

OfertaDisciplinaFacade

- Método ofertaPorId busca uma oferta de disciplina pelo id.
- Método buscarDiscplina busca uma oferta de disciplina pelo id da disciplina.

PreferenciasFacade

- Método retornaPreferencia retorna o valor númerico da opção de configuração de horários escolhida pelo docente na fase I.
- Método docenteObservacoes retorna as observações escritas pelo docente na fase I.

TurmaFacade

- Método achaTurma retorna uma turma buscando pelo id.

2.2.4. ÚTIL E CONTROLLERCONVERTER

O pacote Útil contêm as classes de configuração dos datamodel e lazymodel. Sempre que for necessário criar um novo objeto desse tipo que não possuir uma classe para ele nesse pacote é só criar uma nova seguindo as mesmas configurações das classes já existentes.

O pacote Converter possui alguns métodos usados para fazer o autocomplete das listas de filtros. Nunca foi necessário acrescentar ou modificar nada nesse pacote, apenas os outros foram usados até o momento.

2.3. PRÓXIMAS FUNÇÕES A SEREM ACRESCENTADAS AOS SISTEMAS

Criar as tabelas de histórico das fases I e II.

Alterar o método deleteAllQuad() para salvar as informações da tabela Disp no histórico, de acordo com cada quadrimestre, para depois apagar os registros das tabelas Disp referentes aos registros da tabela OfertaDisciplina e depois apagar os registros da tabela OfertaDisciplina referentes ao quadrimestre selecionado.

Alterar o método deleteTurmas() para salvar as informações da tabela TurmaDocente no histórico, fazendo uma referência ao quadrimestre e ao ano, depois apagar os registros da tabela TurmaDocente para poder apagar os registros da tabela Turma.

Alterar a fase II para que apenas os coordenadores tenham acesso, fazendo a verificação na tabela de administradores para poder acessar o sistema quando a fase II estiver aberta e adicionar a lista de docentes para que os coordenadores possam selecionar qual docente eles irão montar a grade para poder salvar no banco de dados e poder exibir os registros no componente schedule.

Nos métodos da fase II será preciso alterar as informações sobre horários para que todas elas passem a ser apenas referentes ao turno, sem dias e horários definidos, mas tendo apenas valores fictícios devido a indisponibilidade das informações, por parte da PROGRAD, no momento em que o sistema estiver aberto. Também será necessário desabilitar a verificação de conflitos de horários, visto que não haverá mais horários precisos para cada turma e a grade montada será apenas uma ideia de como ele irá ficar, visto que os horários das turmas só serão definidos depois do fim da alocação.