Desenvolvimento

O projeto foi desenvolvido usando a arquitetura MVC, a solução do projeto foi dívida em três projetos diferentes, sendo eles:

* ToDo.Web: projeto que contem a parte web, onde se encontra os controllers, views e os arquivos Java Script.
* ToDo.Domain: projeto que contém todos os models da aplicação.
* ToDo.Data: projeto onde é encontrado toda a parte de manipulação de dados da aplicação, nesse projeto se encontra a o contexto do projeto assim como as Classes DAL da aplicação.

Funções

* Principal.js
  + convertDate: recebe a data passa em json e converte para o formate de data do Jquery.
  + loadListas: carrega com combo de listas no cadastro de tarefa.
  + loadData: preenche a tabela com as listas cadastradas.
  + loadTarefa: recebe o id da lista e retorno todas as tarefas que pertencem aquela lista.
  + addLista: adiciona nova lista no banco.
  + addTarefa: adiciona nova tarefa no banco.
  + getByIDLista: recebe o Id da lista como parâmetro e recuperar os dados a serem editados e abre o modal de edição de lista.
  + getTarefa: recebe o id da tarefa como parâmetro recupera os dados a serem editados e abre o modal de edição de tarefas.
  + UpdateLista: grava as atualizações feitas nas listas.
  + UpdateTarefa: grava as atualizações feitas nas tarefas.
  + DeleteLista: deleta a lista com o Id passado.
  + DeleteTarefa: deleta a t arefa com o Id passado.
  + LimpaCamposLista: limpas os campos de texto do modal de cadastro/edição de lista.
  + LimpaCamposTarefa: limpas os campos do modal de cadastro/edição de tarefa.
  + validate: valida se os campos de cadastro/edição da lista estão preenchidos.
  + validateTarefa: valida se os campos de cadastro/edição da tarefa estão preenchidos.
* TarefasController
  + ObterTarefas: retorna um json com todas as tarefas cadastradas
  + ObterTarefasLista: recebe como parâmetro o Id da lista, e retorna um json com todas as tarefas daquela lista
  + ObterListas: retorna um json com todas as listas cadastradas para preencher o combobox de lista
  + Create: recebe como parâmetro um objeto do tipo tarefa, e grava no banco de dados
  + Edit: recebe como parâmetro um objeto do tipo tarefa, e atualiza no banco
  + Delete: recebe o Id da tarefa como parâmetro e deleta ela do banco
  + ObterTarefaPorId: recebe o id da tarefa por parâmetro e retorna o json com os dados
* ListasController
  + ObterListas: retorna um json com todas as listas cadastradas
  + Create: recebe um objeto do tipo Lista como parâmetro e grava no banco
  + Edit: recebe um objeto do tipo Lista como parâmetro e atualiza os dados no banco
  + Delete: recebe o id da lista como parâmetro e deleta a lista do banco
  + ObterListaPorId: recebe o id da lista como parâmetro e retorna um json com os dados
* ListaDAL
  + ObterLista: retorna todas as listas gravadas.
  + ObterListasPorNome: retorna todas as listas gravadas ordenadas por nome.
  + ObterListaPorId: recebe o id da lista e retorna os dados.
  + GravarLista: recebe um objeto lista e grava uma nova caso o Id seja nulo, ou atualiza a lista caso Id esteja preenchido.
  + Deletar: deleta a lista com o id passado.
* TarefaDAL
  + ObterTarefas: retorna todas as tarefas gravadas.
  + ObterTarefaPorLista: retorna todas as tarefas que pertencem a uma lista.
  + ObterTarefaPorId: retorna os dados da tarefa que possui o Id passado.
  + GravarTarefa: recebe um objeto tarefa e grava uma nova caso o Id seja nulo, ou atualiza a tarefa caso Id esteja preenchido.
  + Deletar: deleta a tarefa com o id passado.