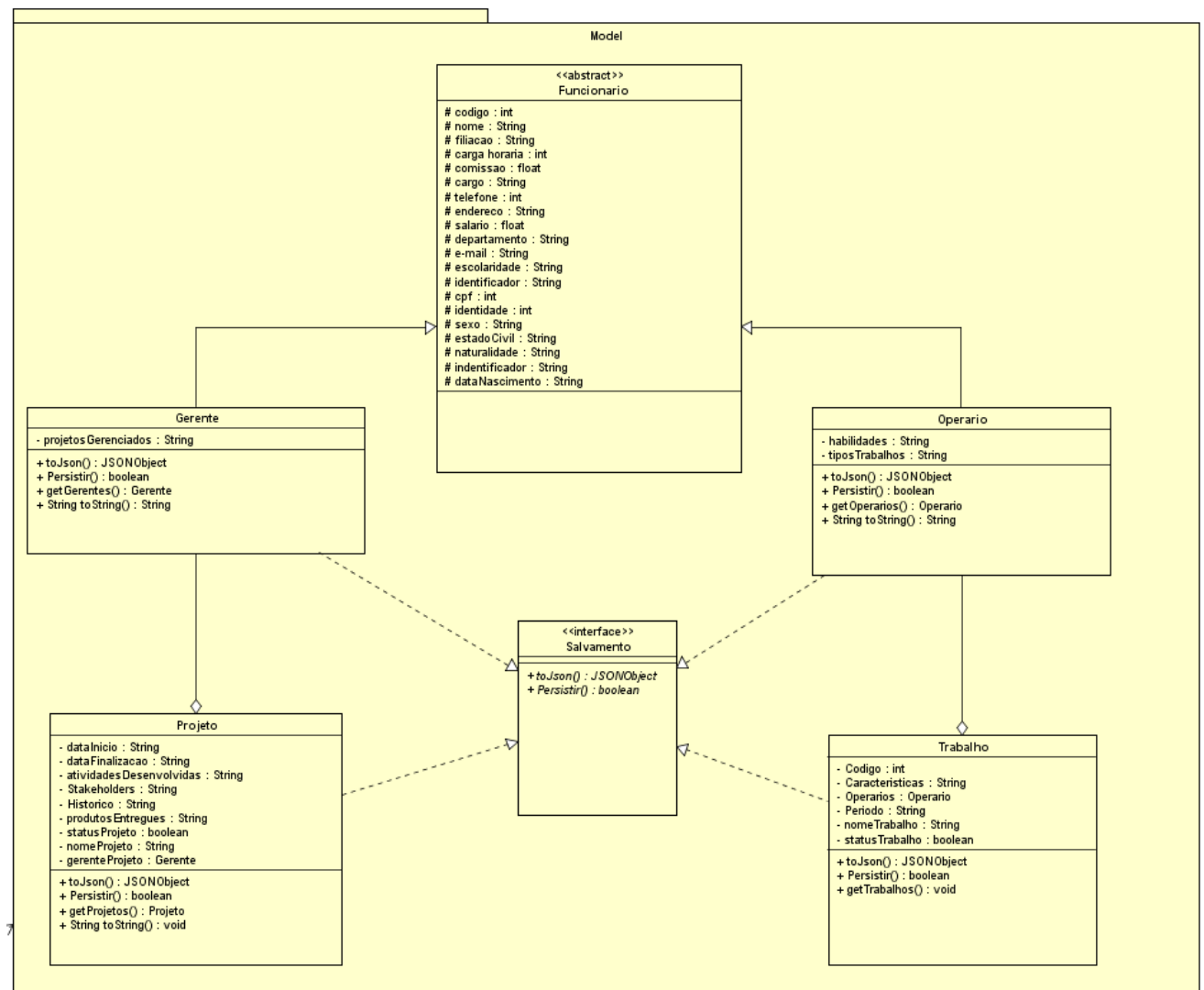
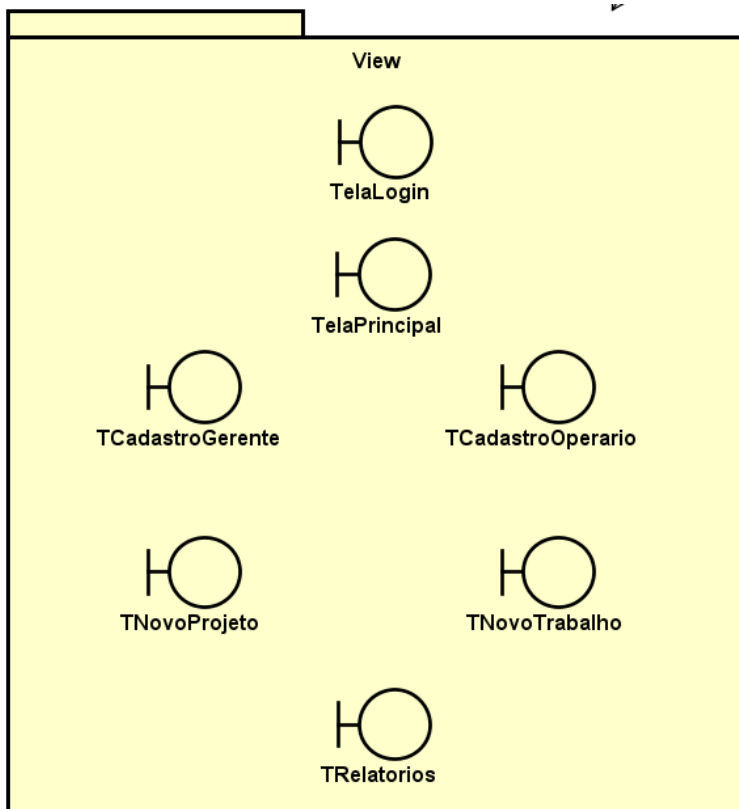


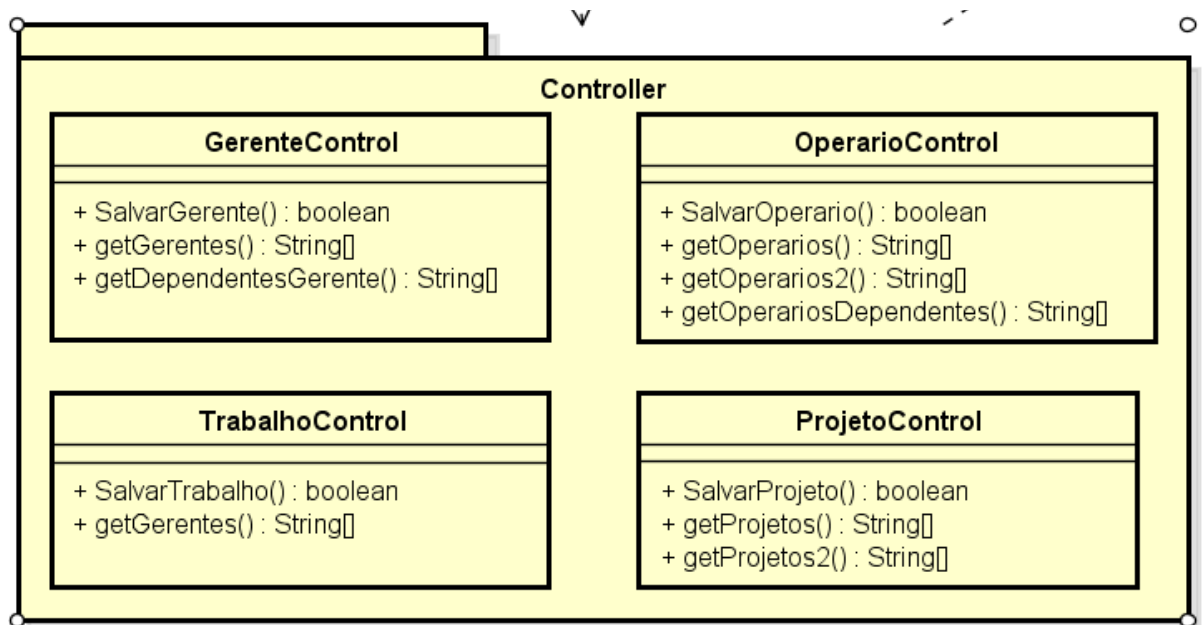
O projeto foi basicamente construído a partir do padrão MVC (Model- View- Controller). A camada Model, que serve como ligação entre as duas outras camadas, funciona como modelador, atuando apenas com armazenamento, manipulação e geração de dados. Abaixo segue a imagem do diagrama de classes relacionado à camada Model:



Seguindo o mesmo intuito, a camada View é formada por todas as classes que apresentam interface gráfica com interação com usuário, é onde os dados são cadastrados pelo usuário ou apresentados. Abaixo segue o diagrama de classe com uma representação do pacote View:

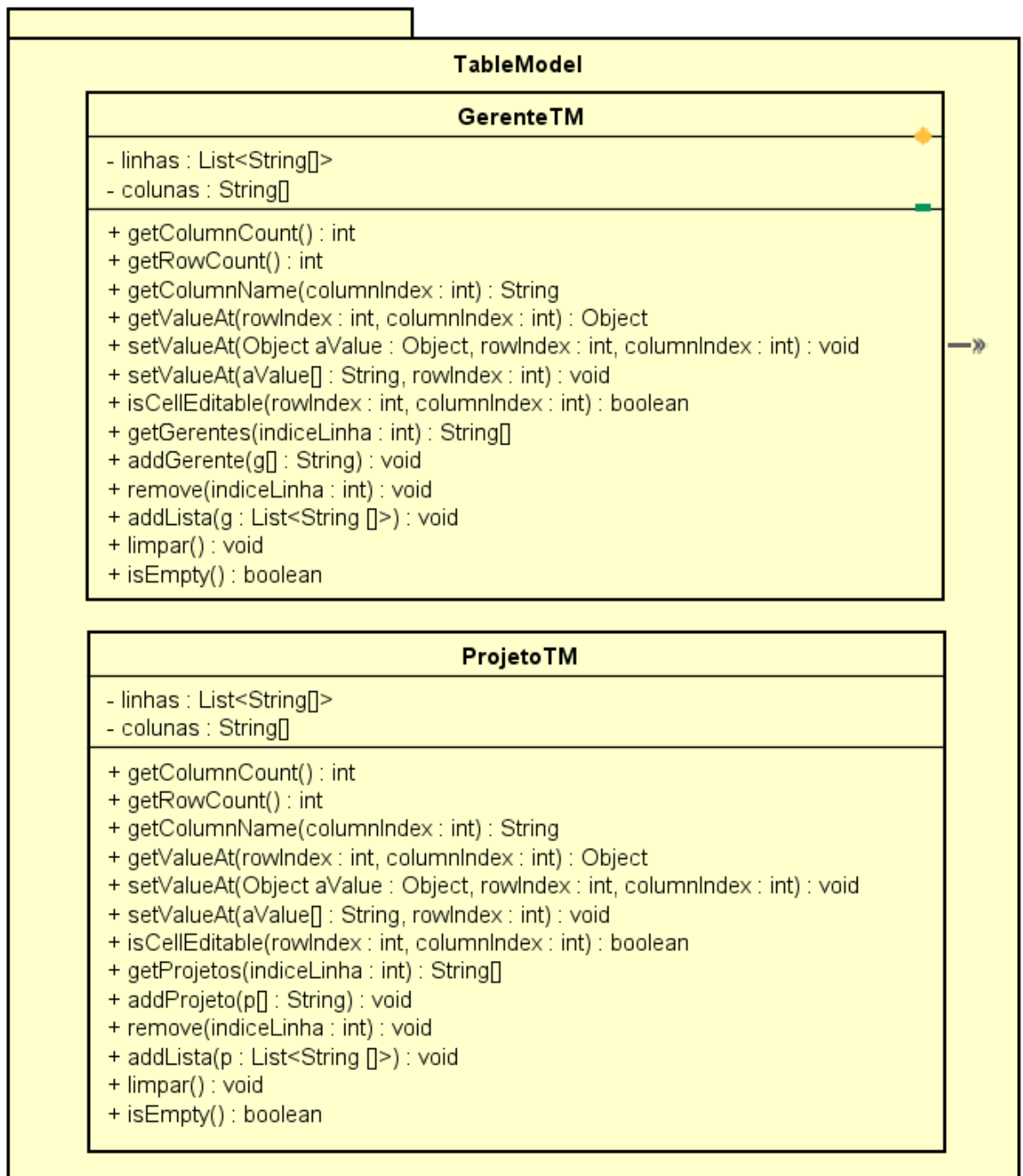


A camada Controller é responsável por fazer a mediação entre a entrada e a saída, comandando como deve ser o View e o Model para que sejam alterados da melhor forma possível. A classes dentro dessa camada, basicamente, controlam o fluxo e as regras de negócio. Segue abaixo a organização do pacote Controller:



Para complementar o projeto e ter maior organização, foram adicionados alguns outros pacotes.

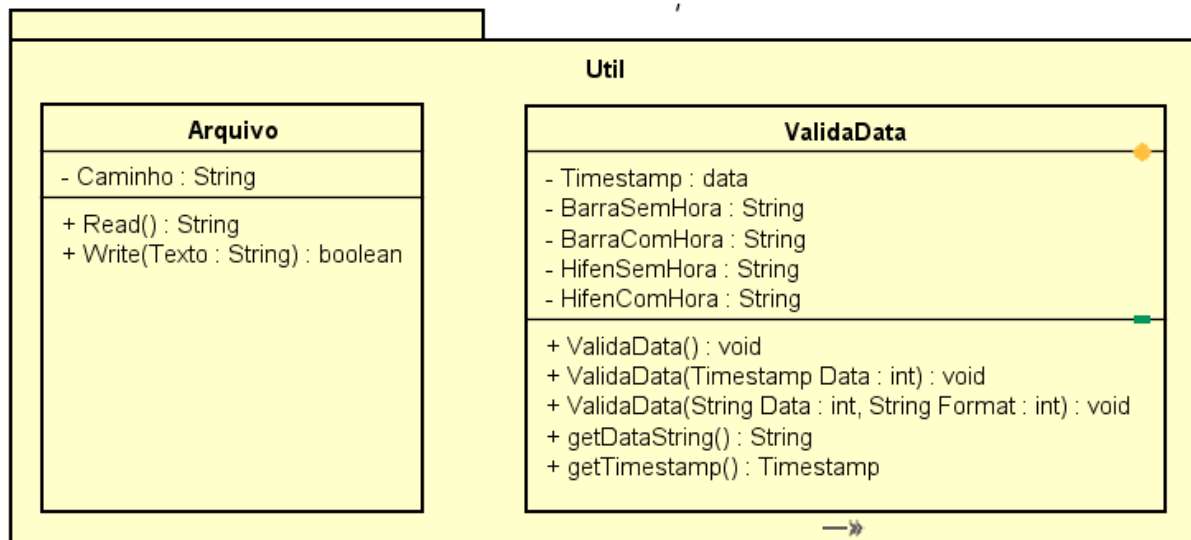
O pacote TableModel contempla todas as classes responsáveis por dar uma modelagem as tabelas criadas na camada View, tendo assim uma maior facilidade para extrair e modificar dados das tabelas, essas classes herdam os métodos e atributos da classe AbstractTableModel. Abaixo é apresentado uma parte do diagrama de classes para o pacote TableModel, as outras classes seguem praticamente o mesmo padrão.



O pacote JSON é responsável por mapear objetos para texto puro, esse pacote foi adicionado por uma biblioteca exterior, previamente pronta e foi utilizada para salvar as informações no arquivo de maneira padronizada.

O pacote “Images” foi utilizado para adicionar todas as imagens que foram utilizadas no pacote View.

O pacote Util, que é responsável por conter todas as classes que funcionam como utilidades extras para o projeto. No caso, a classe Arquivo é responsável por fazer a leitura do arquivo correspondente ou preencher o arquivo conforme o que foi passado pelo usuário e a classe ValidaData é responsável por facilitar o trabalho com manipulação de datas e casos de testes que necessitem de comparação entre duas datas.



Segue abaixo o diagrama de classes correspondente:

