Relatório do Projeto de Laboratório de Programação

Lucas de Oliveira Ferreira da Silva

Rute Gomes de Farias

US1: Aluno possui sua matricula, nome, código do curso, e-mail e telefone respeitando o expert do GRASP. Também é responsável pelos dois tipos de retorno de sua representação textual (com e sem telefone) e pelo método que retorna um de seus atributos. No Sistema, foi implementado o método cadastrarAluno() que cria o Aluno e o guarda em uma lista para que a ordem seja preservada, por isso, foi necessário implementar um método para buscar um aluno por matricula que usa uma iteração através de um for. O sistema também é responsável pela listagem dos alunos concatenando o toString() dos alunos cadastrados na lista. E também delega para a Facade o método getInfoAluno().

US2-3:Tutor possui um Aluno, pois existem métodos para buscar um tutor tanto por e-mail quanto por matrícula, e também ao recuperar um Tutor é retornado o toString() de um aluno, logo o tutor precisa conhecer um aluno. Também possui um mapa que possui uma String contendo uma disciplina como chave e como valor a proficiência desta. Foi usado mapa porque um aluno poderá, assim, ser tutor de várias disciplinas e ter uma proficiência diferente para cada. Não foi necessária a criação de uma classe Disciplina apenas para armazenar a proficiência. Há um mapa de horários que possui como chave uma String com o horário e como valor também uma String com o dia da semana, e uma lista com Strings com os locais de atendimento. Possui sua avaliação, dinheiro e uma variável com o nível do tutor. Foi criado um enum para determinar os níveis de um Tutor. Existem métodos para cadastrar disciplina, horarios e locais de atendimento respeitando o expert do GRASP já que Tutor é responsável pela manipulação de suas informações. Sistema apenas procura o Tutor em uma lista, para manter a ordem, e usa os métodos de tutor para executar as ações.

US4: Foi criada a classe abstrata Ajuda que é extendida para as classes AjudaOnline e AjudaPresencial. Ajuda possui a matricula do Aluno que pediu a ajuda, a matricula do tutor, a disciplina e uma variável booleana que informa se a ajuda foi avaliada ou não. Existem gets para retorna informações sobre Ajuda. Sistema é o responsável pera atribuição do Tutor para a Ajuda respeitando o Creator, já que sistema possui a lista de ajudas e tutores e cria seus objetos, este deveria ser o responsável para manipular métodos que tratam das duas entidades.

US5: O cálculo da avaliação do tutor é feito em Tutor respeitando o expert. O nível também é atualizado em Tutor. Sistema apenas procura a ajuda na lista de ajuda e o tutor pela matricula na lista de tutores para executar o método.

US6: A doação ao Tutor é feita em Sistema porque parte da doação é direcionada ao próprio sistema, logo não fere o expert fazer o cálculo da doação nessa classe. Existe um método doar em Tutor que atualiza a variável que guarda o dinheiro do tutor.

US7: Foram criadas classes que implementam Comparator para ser possível as diversas formas de ordenação; Em Sistema é feito o sort para ordenar as listas de Aluno e Tutor de acordo com os atributos pedidos na chamada do método.

US8: Foi criada uma classe GerenciadorDeDados() que possui um método salvarAlunos() que gera um arquivo para salvar uma lista de alunos e outro método salvarTutores() que gera outro arquivo para salvar uma lista de tutores. Sistema usa essas classes para executar os métodos e delegar para a Facade.