linha horizontal

**UFCG - Universidade Federal de Campina Grande**

Ciência da Computação

Projeto de Laboratório - Programação II

**13 de Março de 2018**

# O QUE FOI FEITO

Na US1 foi feita a classe Aluno, uma classe base que representa Aluno. A decisão de design para os dois tipos de toString que podem ser mostradas foi a de utilizar um método privado que complementa o toString, alterando conforme o Aluno tenha ou não telefone. Um aluno é identificado pela sua matrícula e é armazenado em um HashMap.

Na US2 criamos a classe Tutor, que seria a representação de um Aluno que também é Tutor. Devido a algumas especificações do projeto, decidimos separar as duas classes para uma melhor solução do problema. Um Tutor é um aluno que se disponibilizou para ensinar alguma disciplina. Ele é identificado pelo seu email e armazenado em um HashMap.

Na US3 adicionamos à classe Tutor métodos que armazenam em seus respectivos ArrayList os horários e dias disponíveis do tutor, e também locais de atendimento. Também foram criados métodos que checam se nessas Lists contém certo dia/horário ou local.

Na US4 criamos uma nova classe de Ajuda. Decidimos deixar Ajuda como abstrata e criar duas classes especializadas, “AjudaOnline” e “AjudaPresencial”. Criamos um controller de Ajuda para permitir o pedido de ajuda, conseguir informações da ajuda e do tutor que pegou aquela ajuda.

Na US5 foram criados os métodos para avaliar o tutor. Adicionado no controller de ajuda, passando o identificador da Ajuda e a avaliação, é alterado a nota do Tutor com base em um cálculo, mudando também seu “nível”. No controller de tutor foram adicionados os métodos para recuperar nota e nível de um tutor.

Na US6 criamos uma classe que controla as transações financeiras do sistema. Adicionamos na classe Tutor um método que calcula a taxa que ele recebe de cada doação, afinal, é baseada no nível do tutor. Essa taxa é passada para o Caixa (Controller de transações), que faz o cálculo de quanto vai para o sistema e quanto vai para o tutor.

Na US7 adicionamos classes que implementam Comparator para comparar dois objetos Alunos por: Nome, Matrícula e Email. Decidimos colocar nos controllers de Aluno e Tutor um atributo do tipo Comparator que armazena o método de comparação escolhido, iniciado sempre por Nome. o método que altera a ordem apenas inicializa o atributo para o tipo escolhido.

Na US8, adicionamos a uma classe Dados os métodos de salvar, carregar, limpar. Essa classe armazena todas as coleções do nosso sistema. Salvamos cada coleção em um arquivo, e para carregar, apenas lemos os arquivos e colocamos a coleção no sistema. Para limpar, inicializamos as coleções novamente.

# POR QUEM FOI FEITO

[Eduardo](https://github.com/EduardoMCF) - US3, US8, implementação de controllers, documentação de classes.

[Henrique](https://github.com/HenriqueCA) - US1, US7, documentação de classes, diagrama de classes, relatório, revisão do programa.

[Héricles](https://github.com/hericlesme) - US4, documentação de classes, testes Junit, WebPage, tratamento de exceções, revisão do programa.

[Rafael](https://github.com/RafaelSilvaPereira) - US2, US5, US6, documentação de classes, testes Junit.

# LINKS ÚTEIS

Diagrama:

<https://drive.google.com/file/d/1FA6zdBrQ7WXzdaCIR4yC6MhanwdefsuB/view>

WebPage:

<http://hericlesme.github.io/Projeto-LP2>

Github:

<https://github.com/hericlesme/Projeto-LP2>