



Universidade Federal Rural do Semi-Árido

Campus Pau dos Ferros

Departamento de Engenharias e Tecnologia

Francisco Renan Leite da Costa,
Pedro Vinícius de Andrade Queiroz

Projeto Avaliativo da 3ª Unidade,
Gerenciador dos Horários de Aulas

Pau dos Ferros – RN

Maio - 2023

Francisco Renan Leite da Costa

Pedro Vinícius de Andrade Queiroz

Gerenciador de Horários de Aulas

Relatório apresentado à Universidade Federal Rural do Semi-Árido, na disciplina de Programação Orientada à Objetos, como requisito para avaliação na 3ª unidade, solicitado pelo professor Dr. Italo Augusto Souza de Assis

Pau dos Ferros – RN

Maio - 2023

SUMÁRIO

1. Introdução

1.1. Equipe de desenvolvimento

2. Desenvolvimento

2.1. Funcionalidades implementadas

2.2. Funcionalidades não concluídas

2.3. Cronograma das atividades realizadas

3. Conclusão

1. Introdução

Projeto Avaliativo da 3ª Unidade da disciplina Programação Orientada à Objetos. O projeto visa criar um programa capaz de auxiliar a criação de um horário de aulas semestrais do curso Bacharelado em Tecnologia da Informação(BTI) do campus Pau dos Ferros/RN.

As etapas do projeto foram projeção e desenvolvimento. Na etapa de projeção, a equipe se reuniu e discutiu a lógica geral em torno do programa, realizando pesquisas e anotações em papel, e criando diagramas com os dados reais a serem implementados no programa.

A etapa de Desenvolvimento foi a mais longa e complexa, com processos como conexão com um Banco de Dados em nuvem, criação dos menus e dos objetos professor, componente curricular, de turma e todos os seus métodos e funcionalidades. Nesse passo, muito tempo foi gasto, pois eram comuns erros dos mais diversos tipos, desde erros de execução do Java, dificuldades relacionadas ao Banco de Dados e a sua conexão, e dentre outros.

1.1. Equipe de desenvolvimento

Francisco Renan Leite da Costa - 2021010355

Pedro Vinícius de Andrade Queiroz - 2021010775

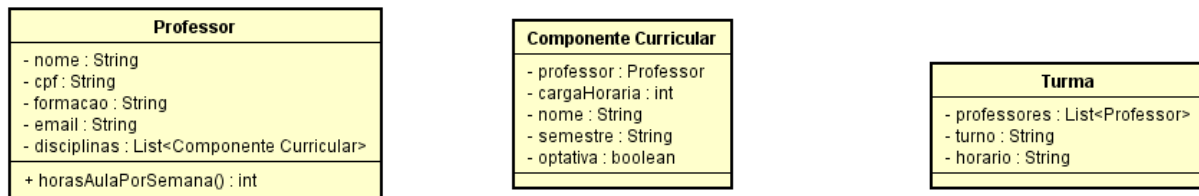
2. Desenvolvimento

2.1. Funcionalidades implementadas

Inicialmente foi criado um diagrama de classes básico para termos uma ideia geral do projeto e como seria a lógica que iríamos utilizar, com isso implementamos os objetos Professor, Componente Curricular e Turma e seus respectivos atributos iniciais, além da classe de demonstração e o seu menu. Em seguida nos deparamos com o nosso primeiro obstáculo, que foi a implementação de um servidor (o servidor foi o ElephantSQL) de um banco de dados em java, passamos alguns dias nessa etapa tentando fazê-la, nisso houve o período de 21/04 até 29/04 em que não conseguimos avançar no projeto pois, além de estarmos atarefados com outras

disciplinas estávamos, também, presos por causa da implementação do banco de dados, mas após várias tentativas e pesquisas conseguimos implementá-lo.

Figura 01



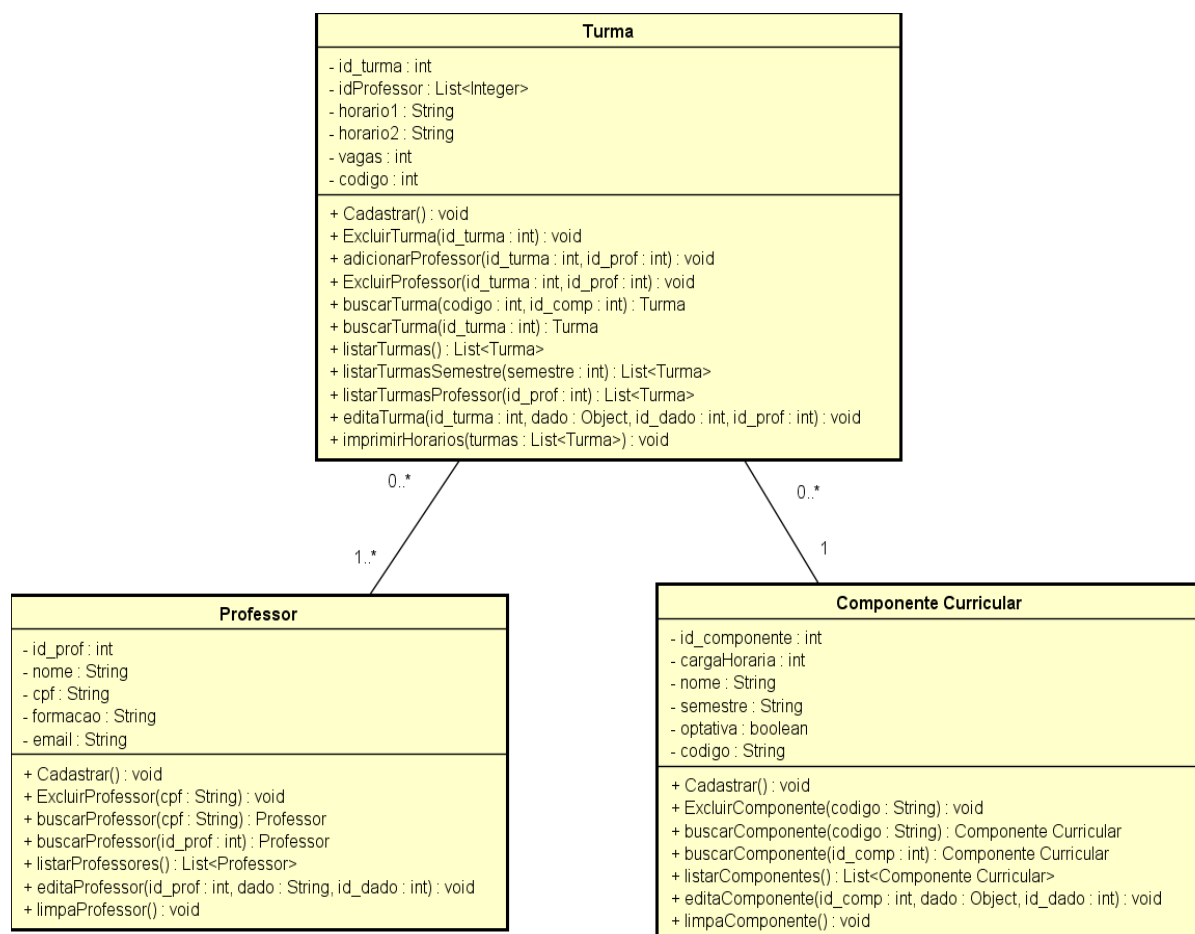
Com a conexão com o banco de dados estabelecida através da classe “PostgreSQLConnection”, voltamos a focar na implementação dos métodos das classes. Primeiramente foi feito os métodos de cadastro, busca, lista e excluir da classe professor e componente curricular, juntamente com as suas respectivas tabelas no banco de dados, inicialmente não houve muitos problemas, o maior processo foi no aprendizado da implementação dos códigos do SQL no java, no decorrer da criação dos outros métodos todos sofreram pequenas mudanças para tornar o código melhor, como por exemplo, foi adicionado a todas as classes um atributo do id.

O nosso segundo obstáculo foi na criação dos método da classe turma, pois, como uma turma pode ter vários professores, o processo da criação das tabelas no banco de dados é diferente, e para implementar isso no java demandou muitos testes e bastante tempo, no começo adicionamos um atributo de um array(int) à tabela “turma” do banco para armazenar os ids dos professores, mas essa é uma maneira errada de armazenar os dados porque, por ter sido armazenado como array, não é possível definir esse atributo como chave estrangeira da tabela professor, mas continuamos com essa ideia e fizemos alguns métodos da classe turma dessa forma. Entretanto, um tempo depois, tivemos que corrigir isso, o atributo de array foi excluído da tabela “turma” e foi criada uma nova tabela “turma_professor” para fazer essa conexão entre as tabelas da turma e do professor, essa tabela possui os atributos “id_turma” e “id_prof”, e com isso foi corrigido todos os métodos que foram feitos anteriormente também.

O próximo passo foi implementar os métodos de editar para todas as classes, do Professor, do Componente Curricular e da Turma, conseguimos fazer com que o usuário edite qualquer uma das informações da classe em um único método, fazendo com que esse método recebesse a nova informação que será editada como “Object” e convertesse dentro do método, dessa forma seria possível editar atributos que possuem tipos diferentes (int e string, por exemplo), em um mesmo método.

Com isso, falta poucos passos para finalizar o projeto, em seguida começamos a trabalhar nas exceções impostas pelo projeto, no qual os professores não podem ultrapassar o limite de horas-aula de 20 horas por semana, não podem haver turmas do mesmo semestre no mesmo horário e não pode haver turmas com o mesmo horário com o mesmo professor. Para fazer essa implementação foi preciso adicionar e modificar os métodos de cadastro e edição de turmas.

Figura 02



O nosso próximo obstáculo foi na lógica para a impressão das tabelas dos horários por semestre e por professor, para resolver isso criamos 3 classes: `listarTurmasSemestre()`, `listarTurmasProfessor()` e `imprimirHorarios()`. Na classe `listarTurmasSemestre()` é feito uma consulta no banco de dados onde irá retornar uma lista de todas as turmas que correspondem ao semestre que foi passado como parâmetro para essa função, na classe `listarTurmasProfessor()` é feito também uma consulta no banco de dados mas nessa classe irá retornar uma lista com todas as turmas que correspondem ao professor que foi passado como parâmetro para essa função. Já na classe `imprimirHorarios()`, que vai receber como parâmetro uma lista de turmas, irá imprimir um horário preenchido com todos os horários de cada turma em formato de tabela.

Posteriormente, após ter finalizado todas as exigências do projeto, resolvemos implementar outros métodos para a edição da turma, um método para adicionar professor a uma turma e um método para excluir professor de uma turma.

Assim, foi finalizado as classes de Professor, Componente Curricular e Turma e seus respectivos métodos, após isso restou apenas organizar a classe main para a demonstração e a interação com o usuário, além de algumas pequenas correções no código.

2.2. Funcionalidades não concluídas

Uma funcionalidade que não foi concluída foi o tratamento de dados de alguns atributos das classes, o usuário consegue passar dados que não deveria ser possível, não conseguimos terminar a tempo.

2.3. Cronograma das atividades realizadas

Data	Autor	Descrição da atividade
18/04	Pedro Queiroz/ Renan Costa	Implementação do diagrama de classes do projeto

19/04	Pedro Queiroz	Implementação das primeiras instâncias das classes Professor, ComponenteCurricular e Turma
20/04	Renan Costa	Criação do servidor do banco de dados que será utilizado
20/04	Pedro Queiroz	Prototipagem e organização as classes e criação do inicio da classe Demonstração
21/04	Pedro Queiroz	Criação de um menu
29/04	Renan Costa	Updates na conexão com o banco de dados e criação dos métodos de cadastrar e buscar professores
30/04	Renan Costa	Método de listar os professores implementado Método de excluir um professor implementado
01/05	Renan Costa	Métodos cadastrar, buscar, excluir e listar componentes implementados
02/05	Renan Costa	Protótipos dos métodos de cadastrar, buscar e listar turmas
03/05	Renan Costa	Implementados os métodos de turma

		Métodos de edição para todas as classes
05/05	Renan Costa	<p>Tratamento de dados:</p> <p>Professores não tem um número de turmas que ultrapasse o seu limite de horas-aula por semana</p> <p>Não é possível cadastrar uma turma que tenha o mesmo horário de outra do mesmo semestre</p> <p>Não é possível cadastrar uma turma que tenha um professor que já tenha uma turma no horário que foi passado no cadastro dessa nova turma</p>
05/05	Pedro Queiroz	Implementação na Main dos métodos Inserir, Ver e Remover Professor
06/05	Renan Costa	Protótipo da impressão das tabelas das turmas por semestre e por professor
06/05	Pedro Queiroz	Implementação na Main dos métodos Cadastrar, Ver, e Excluir Componente Curricular
06/05	Pedro Queiroz	Implementação do método Listar Professor

07/05	Renan Costa	Método de excluir professor de uma turma foi implementado Método para imprimir os horários em tabelas das turmas por semestre e por professor
Resto dos dias	Renan Costa Pedro Queiroz	Implementação do tratamento de dados de alguns atributos Correções

3. Conclusão

O projeto avaliativo da terceira unidade da disciplina Programação Orientada a Objetos - PEX0130 foi uma importante maneira de testar todos os conhecimentos adquiridos durante o decorrer do período. Alguns pontos positivos podem ser destacados, como o bom emprego da máxima de programação orientada a objetos, o desenvolvimento de habilidades de trabalho em grupo, e a interdisciplinaridade com outras áreas da programação. Dentre os pontos negativos, podemos citar a difícil compreensão do projeto e o tempo que não foi o suficiente para conseguirmos finalizar do jeito que queríamos.

O acompanhamento do professor foi suficiente para esclarecer dúvidas mais gerais, relacionadas principalmente ao próprio Java, mas deixou a desejar em pontos e erros mais complexos, fazendo com que os grupos aprendessem a resolver ou procurar ajuda em outras fontes.

Autoavaliação - Renan Costa

O projeto não era essencialmente difícil mas havia muitas coisas novas a serem aprendidas, principalmente no quesito de conexão com banco de dados, inserção, verificação e remoção de dados, além de que o projeto era relativamente

extenso pro tempo disponível, porque não é possível dedicar o mês inteiro para esse projeto por causa de outras disciplinas.

Os conhecimentos que adquiri foram bem relevantes, aprendi bastante coisa sobre ligação com o banco de dados principalmente, e deu pra pôr em prática a teoria dos conteúdos de Programação Orientada a Objetos. Acredito que a aplicação de um projeto em relação a uma prova convencional é que o projeto é muito melhor de aprender e é mais divertido, só dá mais trabalho do que simplesmente fazer uma prova.

Nós precisamos de um acompanhamento do professor em alguns tópicos para tirar dúvidas que não estávamos entendendo acerca do projeto.

Autoavaliação - Pedro Queiroz

O projeto avaliativo foi um bom teste de conhecimento da disciplina, uma vez que me fez praticar e aprender novos métodos e funcionalidades da linguagem Java e da máxima de programação orientada a objetos.

A escolha do Banco de Dados como fonte dos dados do programa foi um fator que abalou minha produtividade, pela falta de familiaridade com a área, fazendo com que produzisse mais da parte prática relacionada à usabilidade e execução. Também foi um desafio, pois foi um projeto grande que envolveu muita pesquisa e estudos para funcionar, e chegar ao fim depois do processo de produção cheio de problemas e erros foi muito proveitoso e gratificante.