

Universidade Federal Rural do Semi-Árido
Campus Pau dos Ferros
Departamento de Engenharias e Tecnologia

LUIS DAVI DA SILVA SARMENTO
MARCOS MIKAEL LIMA VIDAL

Sistema para gerenciar os horários de um semestre das
turmas do curso Bacharelado em Tecnologia da Informação
(BTI) do Campus Pau dos Ferros (CMPF) da Universidade
Federal Rural do Semi-Árido (UFERSA)

Pau dos Ferros – RN
Maio - 2023

LUIS DAVI DA SILVA SARMENTO
MARCOS MIKAEL LIMA VIDAL

Sistema para gerenciar os horários de um semestre das turmas
do curso do Bacharelado em Tecnologia da Informação (BTI)
do Campus Pau dos Ferros (CMPF) da Universidade Federal
Rural do Semi-Árido (UFERSA)

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

O projeto consiste na elaboração de um programa que tem como objetivo gerenciar os horários de um semestre do curso de Bacharelado em Tecnologia da Informação (BTI) - UFRSA do Campus Pau dos Ferros. A implementação desse programa, por sua vez, será uma ferramenta útil para para coordenadores e professores, onde será possível identificar os horários da turma, de forma a evitar conflitos de horários entre disciplinas e garantir que cada disciplina seja ministrada pelos professores designados. O projeto foi construído por etapas de implementação de classes como: Professor, Componente Curricular, Menu, Main, Turma e suas respectivas funcionalidades.

- Etapa 1 - Classe Professor: classe onde será instanciado objetos de mesmo nome da classe (Professor), utilizadas para armazenar dados, exibir, inserir e excluir dados em nosso banco de dados.
- Etapa 2 - Classe Componente curricular: classe onde será instanciado objetos de mesmo nome da classe (ComponenteCurricular), utilizada para armazenar dados, exibir, inserir e excluir dados em nosso banco de dados.
- Etapa 3 - Classe Turma: classe onde será instanciado objetos de mesmo nome da classe (Turma), utilizada para armazenar dados, exibir, inserir e excluir dados em nosso banco de dados.
- Etapa 4 - Classe Menu: classe onde será instanciado objetos de mesmo nome da classe (Menu), utilizada para armazenar dados, exibir, inserir e excluir dados em nosso banco de dados. Esta classe utiliza todas as classes anteriores, sendo nossa “central de controle” geral, pois, a partir dela podemos colocar em vigor todas as utilidades das classes anteriormente criadas.
- Etapa 5 - Classe Main: para execução do programa com o programa apresentando. Utilizou-se um menu que usa os métodos da classe Menu para realizar as manipulações desejadas no programa.

O objetivo principal do programa foi criar um sistema onde o usuário consiga realizar manipulações de dados relacionados a um sistema de uma universidade (turmas, componentes curriculares e professores vinculados a universidade) de modo que facilite o gerenciamento de todos estes dados.

1.1. Equipe de desenvolvimento

Luis Davi da Silva Sarmento - Matrícula: 2019022930

Marcos Mikael Lima Vidal - Matrícula: 2021010988

2. Desenvolvimento

2.1. Funcionalidades implementadas

Primeiramente, foi implementado diagramas de classes que serviram de base para modelar as classes e os respectivos relacionamentos do sistema. Pois com seu uso foi possível fornecer uma representação visual clara da estrutura do comportamento do nosso sistema, o que ajudou bastante a entender e se comunicar sobre funcionalidades e características de uma forma mais eficiente. Com essa implementação na prática, nos ajudou na identificação de problemas e erros, além de contribuir com o desenvolvimento do sistema elaborado pela equipe. Abaixo segue as representações de diagramas de classes.

ComponenteCurricular
<ul style="list-style-type: none">- cargaHoraria : int- nome : String- ID : int- idProfessor : int- idKey : int- turmaDaDisciplina : LinkedList
<ul style="list-style-type: none">+ addTurmaParaOComponente() : void

Professor
<ul style="list-style-type: none">- id : int- idKey : int- titulo : String- nome : String- cargaHoraria : int- componentes : LinkedList
<ul style="list-style-type: none">+ adicionaComponenteCurricular() : void+ removerComponenteCurricular(): void+ adcionaTurmaAoComponenteDoProfessor() : void

Menu
<ul style="list-style-type: none"> + static cadastrarProfessor() : void + static editarProfessor() : void + static verDadosDoProfessor() : void + static listarProfessores() : void + static excluirProfessor() : void + static cadastrarComponenteCurricular() : void + static editarComponenteCurricular() : void + static verComponenteCurricular() : void + static listarComponentesCurriculares() : void + static excluirComponenteCurricular() : void + static cadastrarTurma() : void + static verDadosDaTurma() : void + static listarTurmas() : void + static listarTurmasPorSemestre() : void + static listarTurmasPorProfessor() : void + static excluirTurma() : void

Turma
<ul style="list-style-type: none"> - iD : String - idKey : int - idComp : int - idProf : int - semestre : int - nomeDaTurma: String

Diante dos modelos de diagrama de classes apresentados com sua entidade envolvidas como turma, professor, componente curricular, menu foi possível obter com êxito as implementações das classes criadas com seus respectivos métodos. As funcionalidades implementadas foram: cadastrar professor, editar professor, ver dados do professor, listar professores, excluir professor, cadastrar, editar, ver, listar e excluir componente curricular, cadastrar turma, listar turmas. Além do mais, foi utilizado um banco de dados, que foi de fundamental importância para o armazenamento dos dados do professor, componente curricular, menu e a turma.

Para isso, foi utilizado o postgresql para a execução do banco de dados com o código em java.

Uma das principais maneiras que lidamos com os obstáculos foi manter a comunicação clara e eficaz para o desenvolvimento do projeto com reuniões e atualizações das atividades.

A evolução do projeto se manteve constante, estando sempre em mudança, implementando seus métodos necessários, corrigindo bugs e problemas com o decorrer do que era implementado no nosso código. As maiores dificuldades estavam relacionadas ao tratamento de dados, diante de que sempre que implementamos um novo método, tivemos que modificar métodos anteriores de modo que funcionassem com métodos criados posteriormente.

2.2. Funcionalidades não concluídas

As funcionalidades que acabaram por não serem concluídas:

- Método editar turmas
- Exibição de horários em tabela por semestre.

Estas funcionalidades acabaram por não serem concluídas devido à grande quantidade de tempo investido em outras funcionalidades e ao tratamento de dados das mesmas.

2.3. Cronograma das atividades realizadas

Data	Autor	Descrição da atividade
16/04	Mikael Vidal	Implementação da classe Professor e métodos, implementação da interface MenuSistemaHorarios, implementação da classe ComponenteCurricular.
18/04	Mikael Vidal	Adição de métodos e quals, hashCode e construtor2.
18/04	Mikael Vidal	Adição e resolução de problemas em métodos da classe professor.
20/04	Luis Davi	Adição de algumas funcionalidades do métodos da classe professor.

20/04	Mikael Vidal	Modificações em atributos, métodos e construtor da classe professor.
20/04	Mikael Vidal	Adição e modificação em métodos e atributos da classe ComponenteCurricular.
23/04	Luis Davi	Adição de algumas funcionalidades do método ComponenteCurricular.
20/04	Mikael Vidal	Adição da classe Turma e alguns métodos e atributos necessários da classe.
20/04	Mikael Vidal	Adição do método adicionarTurmaAoComponenteDoProf na classe professor.
22/04	Luis Davi	Iniciativa de desenvolvimento no método adicionarTurmaAoComponenteDoProf da classe professor.
20/04	Mikael Vidal	Ajuste de implementações erradas nas classes professor e componente curricular.
24/04	Mikael Vidal	Adição de escopos de métodos estáticos na classe Menu.
26/04	Luis Davi	Desenvolvimento no método da classe menu.
25/04	Mikael Vidal	Implementação de alguns métodos da classe Menu.
25/04	Mikael Vidal	Upload de arquivo JDBC necessário para se conectar ao banco de dados.
25/04	Mikael Vidal	Integração do banco de dados ao nosso sistema.
27/04	Luis Davi	Verificação da conexão do banco de dados.
25/04	Mikael Vidal	Adição de bibliotecas necessárias para utilizar comandos do banco de dados (consultas) no java.
25/04	Mikael Vidal	Implementação de exceções.
23/04	Luis Davi	Criação de diagramas de classes.
26/04	Mikael Vidal	Implementações necessárias para exceções, edits.
27/04	Mikael Vidal	Implementação de mais métodos na classe Menu, exceções, criação de exceções para situações específicas.
28/04	Mikael Vidal	Implementei mais métodos na classe Menu, exclui uma exceção desnecessária, exclui uma classe desnecessária, ajustei linhas atributos e métodos de outras classes.

03/05	Mikael Vidal	Implementação de mais métodos na classe Menu, adição de atributo semestral na classe turma
05/05	Luis Davi	. Ajuste na classe menu no método excluirProfessor. Finalização de alguns métodos da classe Menu.
17/04	Luis Davi	Introdução ao relatório, Capa, Contra capa e apresentação da introdução.
22/04	Luis Davi	Desenvolvendo a introdução e as funcionalidades.
08/05	Luis Davi	Elaboração da conclusão.
10/05	Mikael Vidal	Implementação de métodos para turma, tratamento de dados dos métodos posteriores e correções de erros.

3. Conclusão

Concluimos que o projeto, por sua vez, foi desenvolvido com êxito, e que a respeito do sistema, ficou estabelecido que tal gerenciamento de horários das turmas do curso BTI, seria uma ferramenta bastante útil e eficiente para a administração acadêmica. Tal sistema desenvolvido apresentou diversos recursos que permitiram o cadastro de disciplinas, turmas, professores bem como a geração de horários de aulas para cada turma. Além do mais, o sistema apresentou funcionalidades de resolução das turmas de um mesmo semestre não ter conflito quanto aos horários, como também a carga horária estabelecida de cada disciplina. No mais, é válido ressaltar que o banco de dados implementado foi bastante útil para o armazenamento de dados do sistema.

De maneira geral, o grau de dificuldade foi meio relevante devido algumas funcionalidades mais complexas e pelo fato de muitas requisitarem uma demanda maior de tempo. De fato, o projeto nos proporcionou um conhecimento bastante eficaz além da utilização de métodos avançados do java, além de podermos lidar da conexão do banco de dados com o código, onde foi possível adquirir um conhecimento a mais diante da implementação. Do mesmo modo, o acompanhamento do professor em sala de aula foi muito eficiente a respeito das

dúvidas esclarecidas, bem como das sugestões apresentadas para o bom desenvolvimento do projeto.