

MATRICULADOR MASTER BLASTER

9000

Técnicas de Construção de Programas - INF01120 - A

Prof. Karina Kohl - Trabalho Final - Etapa 1

Grupo 07: Arthur Bridi, Bruno Mengue, João Gabriel Clementel, Mariana Burzlaff, Vicente Veiga

Facilitador: Bruno Mengue

1.2 Mudanças em relação à Etapa anterior

Sem mudanças a serem registradas.

1.3 Requisitos

Requisitos Funcionais	Requisitos Não Funcionais
<ul style="list-style-type: none">- RF-1: O sistema deve permitir o cadastramento de dados relevantes sobre as disciplinas, através de input manual e/ou importação de arquivos.- RF-2: O sistema tem gerar e exibir as opções de grade de forma gráfica/calendário, com dia e horário e com indicadores de conflito.- RF-3: O sistema tem que exibir dados de cada disciplina(ex: código, nome, professor, sala, vagas, etc).- RF-4: O usuário pode definir prioridades e parâmetros (ex: professor preferido, turno, limite de créditos.)- RF-5: O sistema deve ordenar e aplicar restrições às grades com base nos parâmetros fornecidos pelo usuário.- RF-6: As preferências do usuário devem ficar salvas localmente com opção de limpeza.	<ul style="list-style-type: none">- RNF-1: O sistema deve ser capaz de operar de forma offline (salvo a extração dos dados das disciplinas caso necessário).- RNF-2: A interface deve organizar as grades com uma indentação pré-definida, devendo ser simples e intuitiva- RNF-3: Ao gerar as grades, o sistema não deve demorar mais que poucos segundos.- RNF-4: O sistema deve ser capaz de lidar com formatos de arquivo inválidos ou dados faltantes, fornecendo uma mensagem de erro clara ao usuário.- RNF-5: O sistema deve garantir que as grades geradas são as solicitadas pelo usuário, assim como na ordem correta caso seja implementada a ordenação.- RNF-6: Cada cadeira deve possuir uma cor diferente (arbitrária), facilitando a visualização.

Justificativa de prioridades:

Os requisitos funcionais foram ordenados pensando-se num mínimo produto viável, composto pelos três primeiros requisitos. Acreditamos que somente estes serão suficientes para solucionar grande parte dos problemas reais que inspiraram o programa. Os demais requisitos são os passos para a forma final da solução, que busca tornar o processo de matrícula extremamente conveniente.

A prioridade dos requisitos não funcionais foi definida para garantir a usabilidade, desempenho e confiabilidade do MMB9K. A maior importância foi dada à operacionalidade (offline) e ao desempenho. Em seguida, focou-se na confiabilidade, com o sistema sendo capaz de tratar erros de arquivos de input e validar a precisão da grade gerada. Por fim, foca-se no aspecto visual: interface simples e uso de cores.

1.4 Projeto

Disciplina

Atributos: código: String, nome: String, créditos: int, turmas: List<Turma>

Métodos: adicionaTurma(turma:Turma) -> void; getCodigo() -> String; getNome() -> String; getCreditos() -> int; getTurmas() -> List<Turma>.

Turma

Atributos: id: String, disciplina: Disciplina, professor: Professor, sala: String, vagasOfertadas: int, horario: Horario, Cor: String.

Métodos: setHorario(horario: Horario) -> void, getHorario -> Horario, Conflito(turmas: List<Turma>) -> boolean, getId() -> String, getProfessor() -> Professor, getCor() -> String, setCor(cor: String) -> void

Professor

Atributos: nome: String

Métodos: getName() -> String

Horário

Atributos: inicio: LocalTime, fim: LocalTime, diaSemana: DiaSemana.

Métodos: getDiaSemana() -> DiaSemana, getHoralInicio() -> LocalTime.

Onde DiaSemana pode ser um enum dos dias da semana.

Grade

Atributos: turmasSelecionadas: List<Turma>, creditosTotais: int

Métodos: adicionarTurma(turma: Turma) -> boolean, removerTurma(turma: Turma) -> void, getTurmasSelecionadas() -> List<Turma>, getCreditosTotais() -> int, atualizaCreditosTotais() -> int

ExtracaoDados

Atributos: classe sem atributos

Métodos: carregarDisciplinas() -> List<Disciplina (método abstrato)

Preferencias

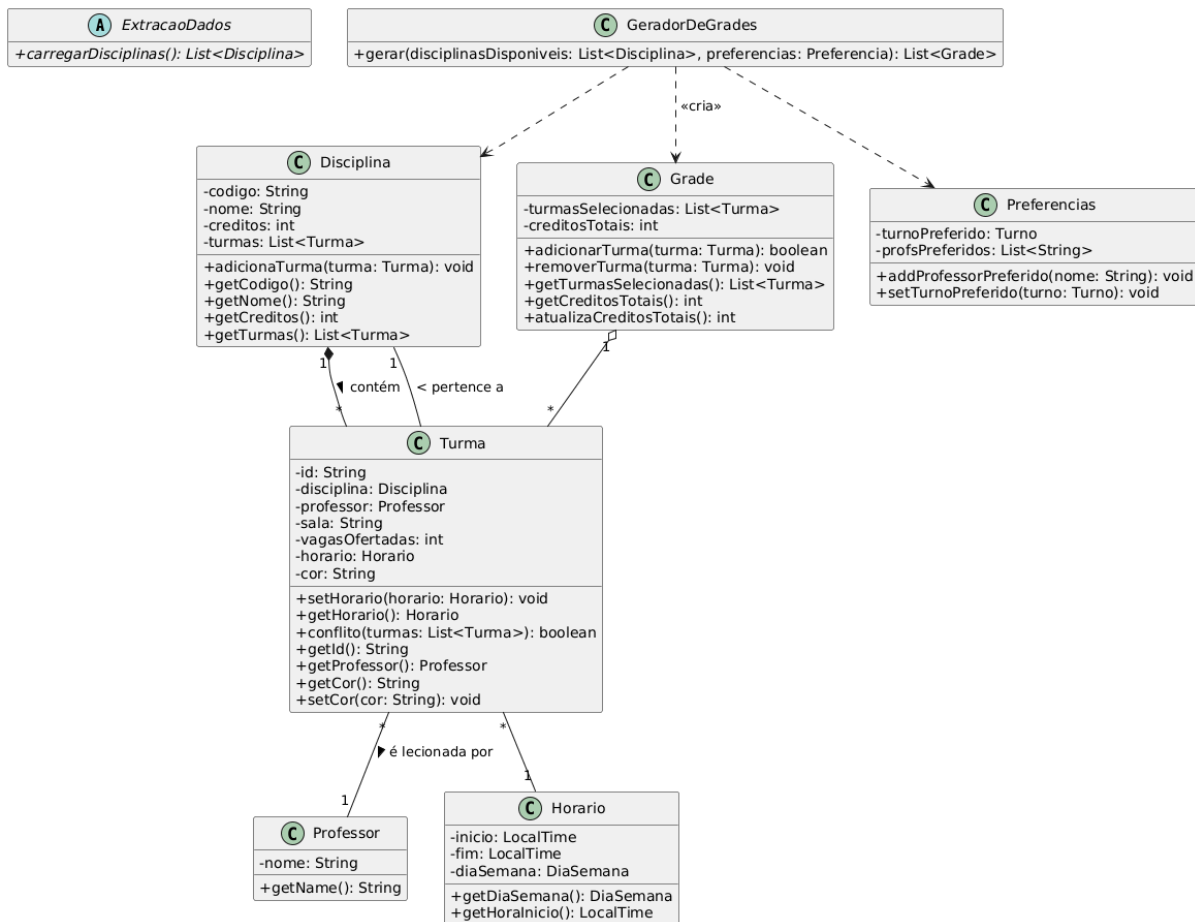
Atributos: TurnoPreferido: Turno, profsPreferidos: List<String>.

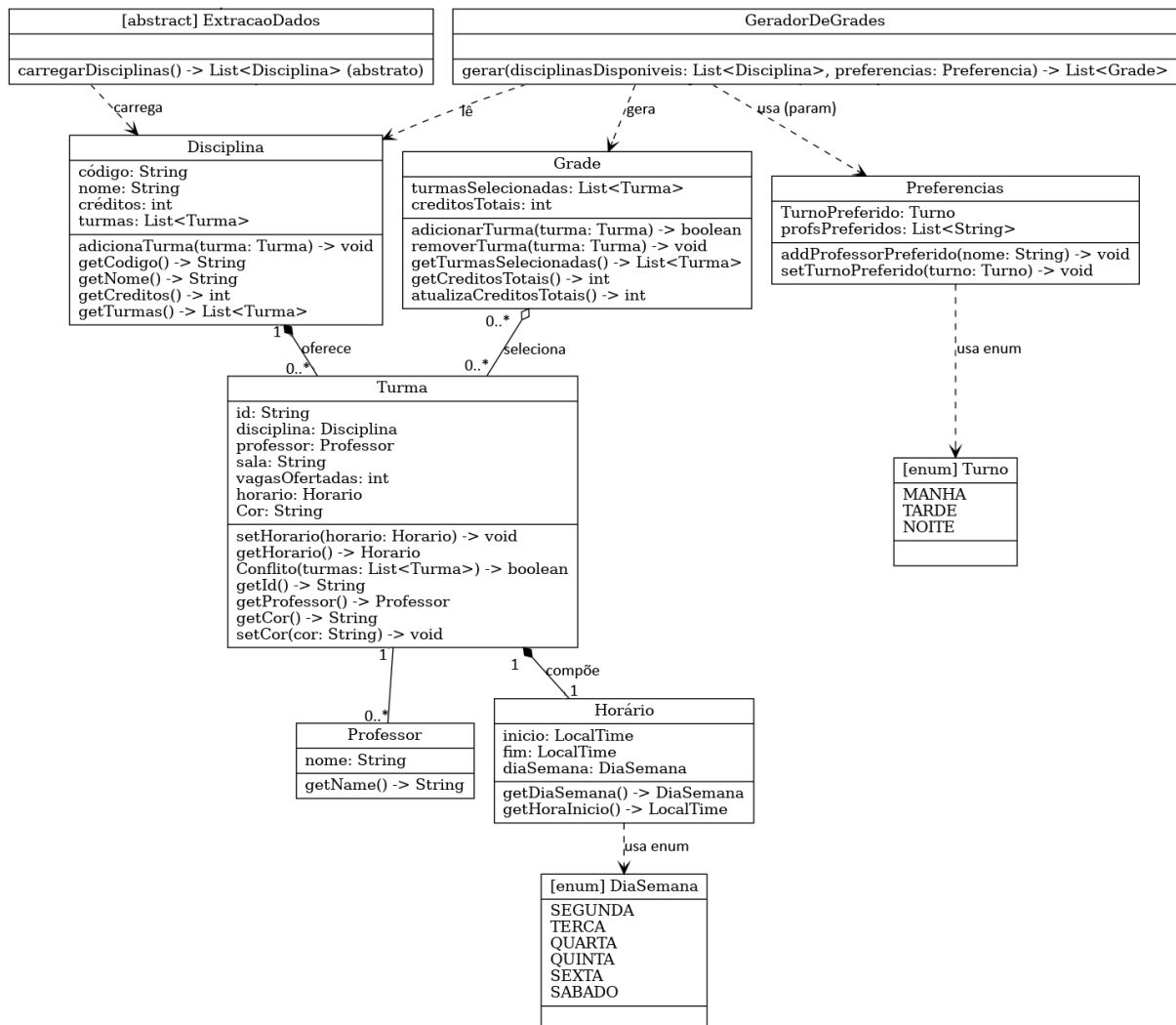
Métodos: addProfessorPreferido(nome: String) -> void, setTurnoPreferido(turno: Turno) -> void.

GeradorDeGrades

Atributos: Não possui.

Métodos: gerar(disciplinasDisponiveis: List<Disciplina>, preferencias: Preferencia) -> List<Grade>.





A arquitetura do projeto foi construída em função da divisão de tarefas de cada classe, separando o sistema em lógicas distintas. Isso separa as classes (como Disciplina e Turma) da lógica do sistema (GeradorDeGrades) e da extração de dados. As relações entre as classes, como Composição e Agregação, foram escolhidas de forma que faça sentido lógico para o projeto. O resultado é uma organização modular, sem complexidade de compreensão que garante a fácil manutenção e futura expansão do sistema caso desejado.

1.5 Interface com o Usuário

1. Tela de exibição dos resultados da montagem de grades, com as características definidas

Grade 1			Grade 2			Grade 3			Preferências		+					
SEG			TER			QUA			QUI			SEX				
8:30	Fundamento de Banco de Dados			Organização de Computadores B			Fundamento de Banco de Dados			Organização de Computadores B						
				Fundamento de Computação Gráfica						Fundamento de Computação Gráfica						
	Técnicas para Construção de Programas						Técnicas para Construção de Programas									
10:30																
13:30																
15:30																

2. Essa tela mostra como deve ser a resposta do site à passagem do hover por uma das disciplinas, exibindo as informações mais detalhadas

Grade 1					
Grade 2					
Grade 3					
	SEG	TER	QUA	QUI	SEX
8:30	Fundamento de Banco de Dados	Organização de Computadores B	Fundamento de Banco de Dados	Organização de Computadores B	
10:30		Fundamento de Computação Gráfica		Fundamento de Computação Gráfica	
13:30	Técnicas para Construção de Programas		Técnicas para Construção de Programas		
15:30					

Professor:
Karina Kohl
Turma A
Sala 108
Vagas: 30
Créditos: 6

3. Essa tela mostra a inserção das entradas, que pode se dar por um arquivo csv ou por entradas manuais das disciplinas e das turmas, sendo exibidas abaixo as turmas já inseridas.

Preferências

+

Importar arquivo CSV

Nome da Disciplina

Código da Disciplina

Número de Créditos

Nome da Disciplina

Código da Turma

Vagas

Dias da semana

Horário

Nome da Professor

Prédio

Sala

Turmas Criadas

- A
- B

- Fundamentos de Banco de Dados
- Organização de Computadores B
- Técnicas de Construção de Programas
- Fundamentos de Computação Gráfica

4. Essa tela apresenta a forma como são inseridas as restrições, com o usuário podendo clicar nos horários em que não deseja ter aula, inserir seu número máximo de créditos e marcar com quais professores não deseja ter aula.

Preferências

+

Marque os horários em que não deseja ter aula

	SEG	TER	QUA	QUI	SEX
8:30					X
10:30					
13:30				X	
15:30	X		X		

Número Máximo de Créditos

Marque os professores com quem não deseja ter aula

☐

Karina Kohl
Técnicas de Construção de Programas

☐

Marcelo Pimenta
Técnicas de Construção de Programas

☐

Karin Becker
Fundamentos de Banco de Dados

☐

Renata Galante
Fundamentos de Banco de Dados

☐

Luigi Carro
Organização de Computadores B

☐

Gabriel Nazar
Fundamentos de Banco de Dados