

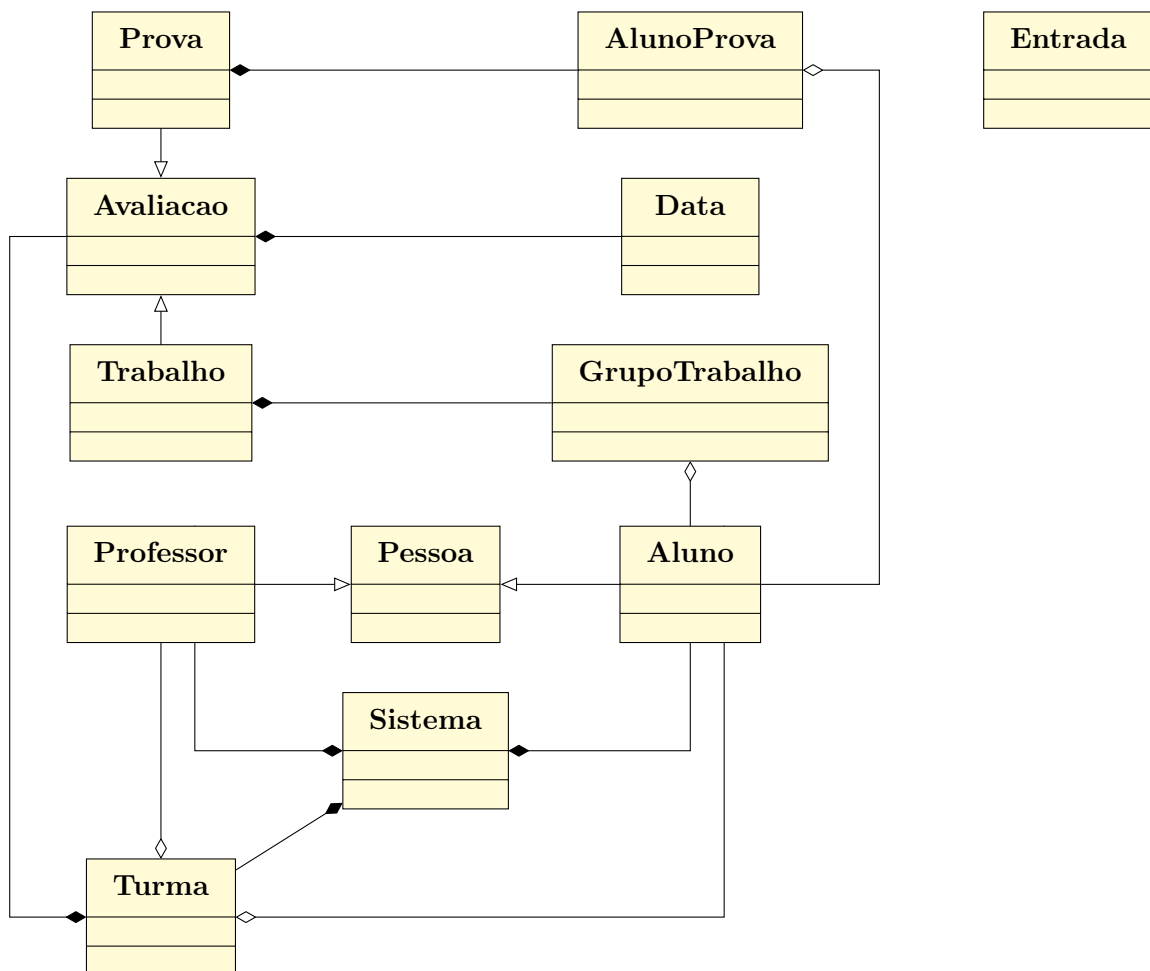
Atividade 2: Implementação

POO - BSI - Ifes Serra

14 de maio de 2025

1 Diagrama de Classes UML

Nesta atividade, você deve começar a implementar o sistema feito na Atividade 1. As classes do sistema e as relações entre elas são mostradas a seguir:



Note que há uma nova classe chamada **Entrada**, que vai cuidar de toda a leitura de dados do sistema e interação com o usuário do sistema. **Nenhuma leitura de dados pode ser feita fora da classe **Entrada****. A seguir, uma descrição detalhada dos atributos e métodos que devem ser implementados em cada classe por enquanto:

Entrada
- input : Scanner
+ menu(s : Sistema) + cadProf(s : Sistema) + cadAluno(s : Sistema) + cadTurma(s : Sistema) - lerLinha(msg : String) : String - lerInteiro(msg : String) : int - lerDouble(msg : String) : double - lerProf(s : Sistema) : Professor - lerAlunos(s : Sistema) : Aluno[] - lerAlunoProva(s : Sistema, a : Aluno, nQuestoes : int) : AlunoProva - lerProva(s : Sistema, alunos : Aluno[]) : Prova - lerGrupoTrabalho(s : Sistema) : GrupoTrabalho - lerTrabalho(s : Sistema, alunos : Aluno[]) : Trabalho - lerAvaliacoes(s : Sistema, alunos : Aluno[], nAvaliacoes : int) : Avaliacao[]

Alguns métodos já foram implementados no arquivo *Entrada.java* disponibilizado. Há um método com o *menu* do sistema:

Escolha uma opção:

- 1) Cadastrar professor:
- 2) Cadastrar aluno:
- 3) Cadastrar turma:
- 4) Listar turmas:
- 0) Sair

Os métodos de cadastro leem as informações apropriadas sobre cada novo objeto, nesta ordem:

1. Para professores:

```
Digite o nome do professor: Hilario Seibel Junior
Digite o cpf do professor: 123.456.789-00
Digite o salário do professor: R$ 1000.0
```

2. Para alunos:

```
Digite o nome do aluno: Enzo da Silva
Digite o cpf do aluno: 234.567.890-00
Digite a matrícula do aluno: A111
```

3. Para turmas:

```
Digite o nome da disciplina: Prog2
Digite o ano da disciplina: 2025
Digite o semestre da disciplina: 2
```

Digite o CPF do professor: 123.456.789-00
Digite a quantidade de alunos na disciplina: 4
Digite a matrícula do aluno: B222
Digite a matrícula do aluno: C333
Digite a matrícula do aluno: D444
Digite a matrícula do aluno: E555
Digite a quantidade de avaliações na disciplina: 3
Escolha um tipo de avaliação:
1) Prova
2) Trabalho

Se a avaliação for uma prova com 3 questões para 2 alunos:

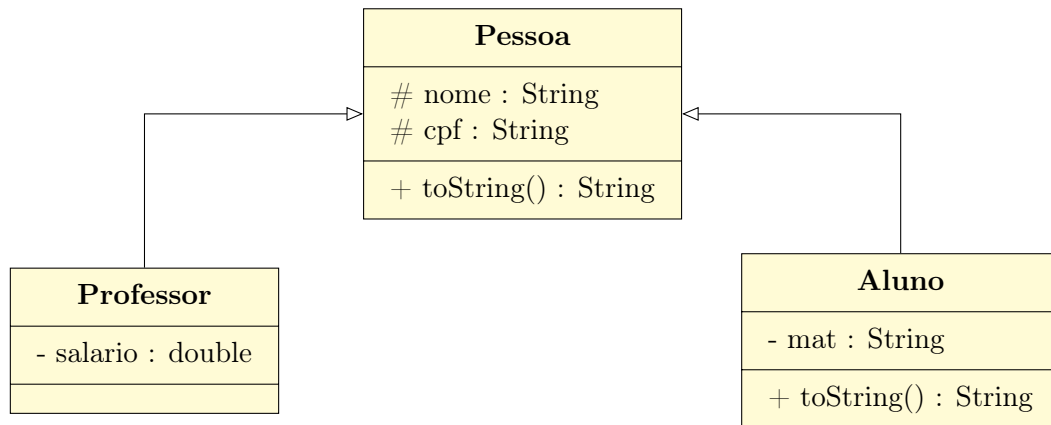
Informe o nome desta prova: Prova 1
Digite o dia da prova: 3
Digite o mês da prova: 9
Digite o ano da prova: 2025
Digite o valor máximo desta avaliação: 30
Digite o número de questões: 3
Nota de Bruno na questão 1: 10
Nota de Bruno na questão 2: 10
Nota de Bruno na questão 3: 10
Nota de Carla na questão 1: 9
Nota de Carla na questão 2: 8
Nota de Carla na questão 3: 7

Se a avaliação for um trabalho com 2 grupos:

Informe o nome desta avaliação: Trabalho 1
Digite o dia do trabalho: 10
Digite o mês do trabalho: 10
Digite o ano do trabalho: 2025
Digite o valor máximo desta avaliação: 40
Digite o número máximo de integrantes: 2
Digite o número de grupos: 2
Digite o número de alunos neste grupo: 2
Digite a matrícula do aluno: A111
Digite a matrícula do aluno: B222
Nota do grupo: 35
Digite o número de alunos neste grupo:
Digite a matrícula do aluno: C333
Digite a matrícula do aluno: D444
Nota do grupo: 42

Note que o professor deu bônus para um dos grupos, que ficou com mais pontos que a nota máximo do trabalho (42 de 40). Neste caso, quando for calcular a nota final do aluno na disciplina, é preciso verificar se o total do semestre ultrapassou 100 pontos. Se tiver ultrapassado, a nota final deve ser 100.

Os outros métodos **sugeridos** no diagrama são auxiliares para que os cadastros funcionem. O diagrama abaixo mostra como os alunos e professores serão modelados:



O método **toString** em uma classe qualquer indica como um objeto desta classe será representado quando for impresso. Na classe **Pessoa**, por exemplo, caso o nome seja “João da Silva” e seu CPF seja “123.456.789-00”, o método deve retornar a String “João da Silva (CPF: 123.456.789-00)”. Fazendo isso, supondo que o nome deste objeto é p1, isso significa que a impressão de p1:

```
System.out.println(p1);
```

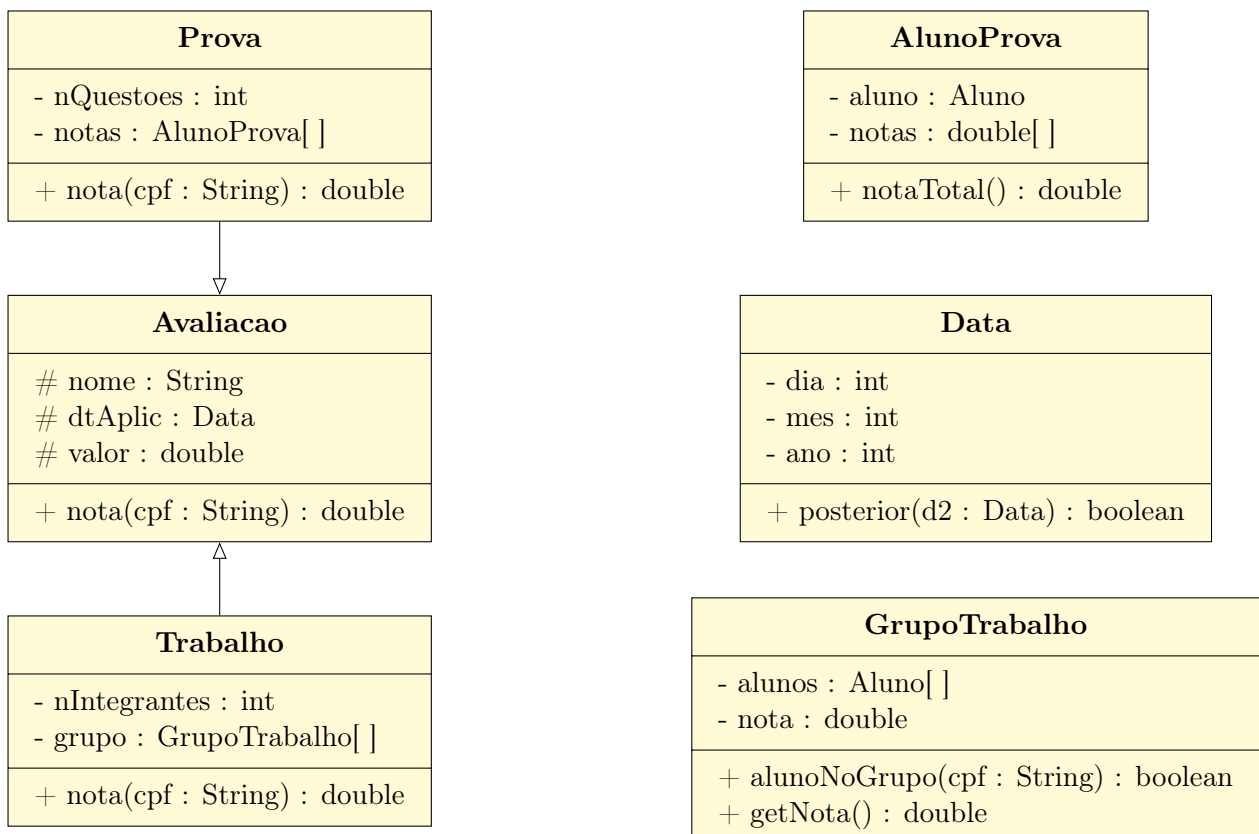
irá exibir na tela:

João da Silva (CPF: 123.456.789-00)

Na classe **Aluno**, o método **toString** deverá ser *reescrito* para exibir a matrícula do aluno ao invés do CPF:

João da Silva (Matrícula: 1234-5)

Já as avaliações serão representadas a partir do diagrama a seguir:



O método **posterior** da classe **Data** recebe outra Data d2 como parâmetro, e retorna verdadeiro apenas quando a data em questão é posterior a d2.

Toda **Avaliação** possui uma data de aplicação, um nome (exemplo: “Prova 1”), e um valor máximo. Mas existem dois tipos de avaliações:

- Uma **Prova** possui um número de questões e um array que salva um objeto da classe **AlunoProva** para cada aluno que fez a prova. Por sua vez, a classe **AlunoProva** armazena o aluno e um array com as notas que ele tirou em cada questão da prova. O método **nota** da classe **Prova** recebe o cpf de um aluno no array, acessa o objeto **AlunoProva** do aluno com este cpf, e retorna a nota total que ele tirou na prova. O método **notaTotal** da classe **AlunoProva** simplesmente soma as notas do aluno em cada questão.
- Todo **Trabalho** possui um número máximo de integrantes no grupo e um array com objetos da classe **GrupoTrabalho** para cada grupo que fez este trabalho. A classe **GrupoTrabalho** armazena o array de alunos do grupo e a nota do grupo. Há um método para verificar se um aluno pertence ao grupo, dado seu cpf.

Não deve ser criado um objeto da classe **Avaliacao** que não seja uma **Prova** ou um **Trabalho**. Por isso, o método **nota** da **Avaliação** pode apenas retornar um valor qualquer. Quando ele for executado, na verdade, o método utilizado será o método **nota** que está sendo reescrito nas subclasses **Prova** e **Trabalho**.

Cada turma no Sistema Acadêmico será representada pelo diagrama a seguir:

Turma
- nome : String - ano : int - sem : int - prof : Professor - alunos : Aluno[] - avs : Avaliacao[]
+ medias()

Uma turma possui o nome da disciplina (por exemplo, “Programação 2”), o ano e semestre em que ocorreu (por exemplo, semestre 2 do ano de 2023), o professor responsável, um array com todos os alunos matriculados na turma e um array com as avaliações que o professor aplicou.

O método **medias** exibe na tela a média de cada aluno e, ao final, a média da turma. Exemplo:

Médias da Turma Prog2 (2023/2):

Bruno (Matrícula: B222): 30.0 32.0 0.0 = 62.0

Carla (Matrícula: C333): 24.0 40.0 3.0 = 67.0

Daniel (Matrícula: D444): 30.0 42.0 30.0 = 100.0

Enzo (Matrícula: E555): 25.0 30.0 9.0 = 64.0

Média da turma: 73.25

No exemplo acima, o nome da turma era “Prog2” e ela ocorreu no 2º semestre de 2023. O aluno Daniel tirou 30 na primeira avaliação, 42 na segunda, e 30 na terceira. Porém, sua média final foi 100 (ao invés de 102) porque a soma da nota máxima de cada avaliação era 100. Isso pode ter acontecido porque o aluno teve bônus de 2 pontos no trabalho. Mas você deve se

certificar que a soma total de todas as avaliações do aluno (mesmo com possíveis bônus) não exceda a somatória total das notas máximas de cada avaliação.

Por fim, a classe Sistema armazena todos os alunos, professores e turmas cadastrados no Sistema Acadêmico:

Sistema
- profs : Professor[] - alunos : Aluno[] - turmas : Turma[]
+ novoProf(p : Professor) + novoAluno(a : Aluno) + encontrarProf(cpf) : Professor + encontrarAluno(mat) : Aluno + listarProfs() + listarAlunos() + listarTurmas()

Os professores, alunos e turmas serão salvos utilizando ArrayList.

O método **novoProf** recebe um objeto da classe **Professor** já instanciado, e insere-o na primeira posição livre do array de professores. O método **novoAluno** e **novaTurma** fazem o análogo para novos alunos e novas turmas, respectivamente.

O método **encontrarProfessor** recebe uma String contendo um CPF e retorna o professor que tiver sido cadastrado com aquele CPF (ou **null**, caso não exista). O método **encontrarAluno** faz o análogo para localizar um aluno pela matrícula.

O método **listarAlunos** imprime os dados de cada aluno (utilizando a formatação definida pelo **toString** desta classe). Ex:

Alunos cadastrados:

```
* Ana (Matrícula: A111)
* Bruno (Matrícula: B222)
* Carla (Matrícula: C333)
* Daniel (Matrícula: D444)
* Enzo (Matrícula: E555)
```

O método **listarProfs** faz o análogo para listar os professores cadastrados até agora. Em ambos, imprima uma mensagem no caso de ainda não haver nenhum objeto cadastrado.

Por fim, o método **listar turmas** lista todas as turmas salvas até o momento, bem como as médias de cada uma. É este método que será chamado na opção 4 do menu. Ex:

Médias da Turma Prog2 (2025/2):

```
Bruno (Matrícula: B222): 30.0 35.0 0.0 = 65.0
Carla (Matrícula: C333): 24.0 35.0 3.0 = 62.0
Daniel (Matrícula: D444): 30.0 42.0 30.0 = 100.0
Enzo (Matrícula: E555): 25.0 42.0 9.0 = 76.0
Média da turma: 75.75
```

Médias da Turma Estrutura de Dados (2025/2):

```
Ana (Matrícula: A111): 30.0 0.0 30.0 = 60.0
Fred (Matrícula: F666): 17.0 0.0 20.0 = 37.0
Média da turma: 48.5
```

Se necessário, crie métodos **getters** para acessar o valor de algum atributo que esteja encapsulado nas classes.

É importante ler os dados de cadastros e opções do menu na ordem em que foram descrito neste documento, pois a correção será automatizada. Na página da disciplina, há um arquivo “input.txt” com um exemplo de entrada do arquivo. Basta rodar seu programa com esse arquivo na pasta onde o programa está sendo executado, e o arquivo será lido automaticamente (ao invés do programa ler a entrada do teclado).

2 Observações

1. A atividade vale 40 pontos e deve ser entregue até 27/05.
2. O código deve ser feito em Java, considerando as boas práticas de POO (como reuso de código, por exemplo, sempre que possível).
3. A atividade pode ser feita em grupos de até dois integrantes.
4. Atividades entregues após o prazo serão automaticamente rejeitados.
5. Atividades com erro de execução, com formato de saída incorreto, ou que não compilarem terão nota 0.
6. Atividades considerados plágio terão nota 0 para quem copiou e para quem forneceu o exercício, e serão enviados ao Conselho de Ética.
7. A Atividade deve ser enviado na sala da disciplina do AVA.
8. Em caso de dúvidas na especificação da atividade ou na própria atividade, contate-me em sala de aula.