

FACULDADE DE COMPUTAÇÃO E INFORMÁTICA
Ciência da Computação, Sistemas de Informação e Análise e Desenvolvimento de Sistemas
Linguagem de Programação I – PROJETO
1º semestre de 2016

PROJETO: SISTEMA DE GESTÃO ACADÊMICA



Nosso **objetivo** é:

- Estender as funcionalidades da versão 1.0 de forma a gerenciar também estudantes de pós-graduação.

Concepção

O usuário do sistema de gestão acadêmica deseja poder gerenciar também os estudantes de pós-graduação. Um estudante de pós-graduação possui os mesmos atributos que um estudante de graduação acrescidos dos atributos **tema** e **orientador**.

Além disso, a faculdade agora precisa controlar o número de horas de atividades complementares realizadas pelos estudantes de graduação.

Por fim, também será necessário obter o total de créditos realizados pelos estudantes, sendo que o cálculo do número de créditos é diferente para os estudantes de graduação e os de pós-graduação.

VERSÃO 2.0

Requisitos

Para esta nova versão, todas as informações (estudantes, disciplinas e matrículas) continuarão sendo gravadas em arquivos no formato texto, sendo que somente o arquivo com os dados dos estudantes será modificado.

O arquivo texto com os dados dos estudantes deverá ter os dados de cada estudante em uma linha separada do arquivo, onde a linha terá o seguinte formato para o estudante de graduação:

[número]:[nome]:[email]:GRAD:[horas de atividades complementares]

e o seguinte formato para o estudante de pós-graduação:

[número]:[nome]:[email]:POS:[tema]:[orientador]

Exemplo do conteúdo do arquivo com os dados dos estudantes:

```
1000:Marco Antônio:marco@gmail.com:GRAD:30
1001:Ângela Maria:amaria@yahoo.com.br:POS:Controle de Temperatura:Carlos Torres
1002:Fábio Teixeira:fabiot@hotmail.com:POS:Energia Solar:Maria Marcondes
1003:Ana Cristina:anacris@mackenzie.br:GRAD:40
```

Os arquivos de disciplinas e matrículas continuam com o mesmo formato da versão 1.0.

O número de créditos de cada estudante será calculado da seguinte forma:

- Para os estudantes de graduação, o número de créditos será o número de horas de atividades complementares mais a soma dos créditos das disciplinas em que ele está matriculado.
- Para os estudantes de pós-graduação, o número de créditos será somente a soma dos créditos das disciplinas em que ele está matriculado.

A sua aplicação deverá ler as informações armazenadas nestes arquivos e permitir que o usuário realize as seguintes consultas:

- Listar os números, nomes e **créditos** de todos os estudantes. Se o estudante for de pós-graduação, deverão ser mostrados também o seu tema e o seu orientador. **Se o estudante for de graduação, deverá ser apresentada a sua quantidade de horas de atividades complementares.**
- Listar os códigos de todas as disciplinas.
- Listar todas as informações dos estudantes matriculados em uma determinada disciplina. Nesta opção, o usuário deverá fornecer o código da disciplina e a aplicação deverá apresentar uma lista dos estudantes matriculados nesta disciplina (com o número

identificador, o nome, o e-mail do aluno **e o número de créditos**) e o número total de alunos matriculados nesta disciplina. Se o estudante for de pós-graduação, deverão ser mostrados também o seu tema e o seu orientador. **Se o estudante for de graduação, deverá ser apresentada a sua quantidade de horas de atividades complementares.**

- Listar todas as informações das disciplinas em que um determinado estudante está matriculado. Nesta opção, o usuário deverá fornecer o número identificador do estudante e a aplicação deverá apresentar uma lista das disciplinas em que ele está matriculado (com o código e número de créditos da disciplina) e o **total de créditos do aluno**.

VERSÃO 2.0

Especificação

As classes Disciplina e Matrícula que foram desenvolvidas na versão 1.0 não deverão ser modificadas.

Serão desenvolvidas duas novas classes, **EstudanteGrad** e **EstudantePos**, e a classe **Estudante** sofrerá algumas modificações, conforme mostra a Figura 1.

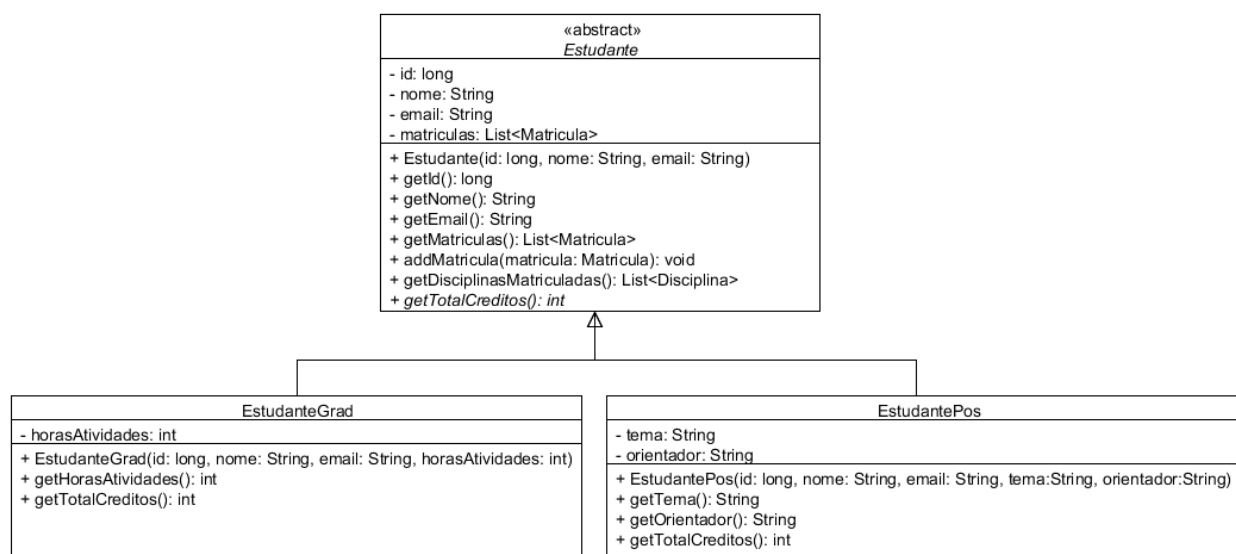


Figura 1: Atributos e métodos da classe EstudantePos.

Note que agora a classe **Estudante** é abstrata e possui o método abstrato **getTotalCreditos**, uma vez que cada tipo de estudante tem uma forma diferente para calcular o seu total de créditos.

A classe **Universidade** (Figura 2) continuará tendo os mesmos atributos e métodos, mas a implementação em Java do método **carregarDados** deverá ser modificada para poder lidar com as modificações no formato do arquivo de estudantes.

Universidade
- nome: String - estudantes: List<Estudante> - disciplinas: List<Disciplina>
+ Universidade(nome: String) + carregarDados(arquivoDisciplinas: String, arquivoEstudantes: String, arquivoMatriculas: String): void + getNome(): String + getEstudantes(): List<Estudante> + getDisciplinas(): List<Disciplina>

Figura 2 - Atributos e métodos da classe Universidade.

EXERCÍCIO: Implemente em Java a classe **GestaoAcademicaApp2**, que deverá ser a versão anterior (**GestaoAcademicaApp1**) alterada para atender os requisitos desta versão 2.0 do sistema.