

---

Projeto de Programação Orientada a Objetos

Data Entrega e Defesa: **18/11/2013 (no horário da aula)**

Pontuação: **4,0 pontos** (a prova vale **6,0 pontos** inclusive para quem não entregar este projeto – este projeto não vale para a 1ª chamada de final)

Equipe: até 2 pessoas

Enviar e-mail: Deve obedecer ao seguinte padrão:

- Assunto: Projeto POO
- Corpo: Incluir os nomes completos dos integrantes da equipe.
- Anexo: Um único arquivo .ZIP contendo a pasta com todos os arquivos do projeto do **Netbeans**.

**OBS1:** e-mails fora deste padrão serão desconsiderados

**OBS2:** renomeie de .ZIP para .ZIP.BAK (evita rejeição de email)

---

**Controle Acadêmico:**

1. O sistema irá controlar as entregas de exercícios e notas dos alunos de diversas disciplinas. Será considerado apenas o semestre atual, portando não será necessário compor um histórico de turmas / alunos que um professor teve. Um professor pode lecionar várias turmas dentro de um semestre. Cada disciplina, pode ter várias turmas num determinado semestre (ou até mesmo nenhuma). Cada turma é cursada por vários alunos, onde cada aluno desta turma apresentará um rendimento escolar.
2. O sistema deverá ser implementado utilizando desenvolvimento em camadas, e utilizando os padrões de projeto *Singleton* e *Façade*. Uma camada só pode acessar funcionalidades da camada imediatamente inferior e apenas através de métodos especificados em uma *interface*.
3. As classes básicas de negócio do controle acadêmico são: professor, disciplina, turma, aluno e rendimento escolar. A classe professor possui os atributos: id (int), nome (String), cargo (String), data de nascimento (java.util.Date), nome de usuário (String) e senha (String). A classe disciplina possui os atributos: id (int), nome (String) e ementa (String). A classe turma possui os atributos id (int), disciplina (Disciplina), professor (Professor) e capacidade da turma. A classe aluno possui os atributos: id (int), nome (String), data de nascimento (java.util.Date), período (int), nome de usuário (String) e senha (String). A classe rendimento escolar possui os atributos turma (Turma), aluno (Aluno), nota 1ª prova (int), nota 2ª prova (int), trabalhos (array de String com 4 posições) e notas dos trabalhos (array de int com 4 posições).
4. O sistema possuirá três interfaces gráficas: uma para professores, uma para alunos e uma para o administrador:
  - a. A interface para os professores deverá disponibilizar as seguintes opções:

- i. Logar no sistema (caso já esteja cadastrado) e cadastrar novo usuário (caso ainda não esteja cadastrado);
    - ii. Assim que entrar no sistema exibir as suas turmas do semestre;
    - iii. Exibir detalhes da turma que o usuário selecionar, nesta tela o professor poderá visualizar os trabalhos entregues pelos alunos e dar uma nota de 0 a 10 (com uma casa decimal de precisão). Também será possível digitar as notas do 1º e 2º GQ dos alunos daquela turma;
    - iv. Exibir uma listagem da turma com as médias de todos os alunos, destacando quem já foi reprovado (em vermelho), quem precisa ir pra final e a nota necessária (em laranja) e os aprovados (em verde). Para isto a média é calculada da seguinte maneira: 10% da média aritmética das notas dos primeiros dois trabalhos será o ponto extra a ser adicionado a nota do 1º GQ e da mesma maneira os outros dois trabalhos irão contar para o 2º GQ (lembre-se que 10 é a nota máxima). Depois disto, é calculada a média ponderada com 40% de peso para o 1º GQ e 60% de peso para o 2º GQ. No final da listagem deve aparecer um resumo com as quantidades e percentuais das seguintes informações: alunos aprovados, alunos na final e alunos reprovados;
    - v. Exibir listagem de todas as turmas disponíveis (que o professor ainda não esteja associado) e o professor poder indicar que ele é o professor daquela turma;
  - b. A interface para os alunos deverá disponibilizar as seguintes opções:
    - i. Logar no sistema (caso já esteja cadastrado) e cadastrar novo usuário (caso ainda não esteja cadastrado);
    - ii. Assim que entrar no sistema exibir as suas turmas do semestre;
    - iii. Exibir detalhes da turma que o usuário selecionar, nesta tela o aluno poderá visualizar os trabalhos entregues, e as notas atribuídas pelo professor, além da possibilidade de entregar os trabalhos;
    - iv. Exibir listagem de todas as turmas disponíveis (que o aluno ainda não está matriculado) e o aluno poder se matricular nas turmas;
  - c. A interface para administração deverá disponibilizar as seguintes opções:
    - i. Cadastrar / Remover / Consultar / Listar disciplinas;
    - ii. Cadastrar / Remover / Consultar / Listar turmas;
    - iii. Cadastrar / Remover / Consultar / Listar disciplinas;
    - iv. Remover / Consultar / Listar professores;
    - v. Remover / Consultar / Listar turmas;
5. Todas as exceções deverão ser capturadas na interface gráfica. Para cada exceção em particular deverá ser criada um classe que herda de Exception. Então, ao tentar se matricular numa turma já lotada deverá ser gerada uma exceção, ao tentar logar com um nome de usuário não cadastrado ou senha inválida será gerada uma exceção, etc. Todos os erros deverão ser cobertos com exceções.
6. Deverá ser utilizado um banco de dados ou arrays para armazenar os dados (facilmente intercambiáveis) e você deverá ter pelo menos os seguintes dados já cadastrados:
  - a. 3 disciplinas;
  - b. 7 turmas;
  - c. 2 professores com 3 turmas cada
  - d. 2 alunos com 3 turmas cada
  - e. os 4 trabalhos de 1 dos alunos já enviados e com notas dos trabalhos e dos GQs