

LP2 – 2024.2

Curso de Ciência da Computação - UFMA

Segunda avaliação

ATENÇÃO: Todas as respostas devem obrigatoriamente utilizar fortemente orientação a objetos e Java. Sempre que existir necessidade use coleções. Procure a melhor coleção para o problema, você está sendo avaliado por isto também. Finalmente, a prova é individual.

Boa prova.

No CCET, todos os períodos, se faz necessário alocar salas (espaços físicos para aulas) para as disciplinas dos diferentes cursos de graduação e pós-graduação associados à unidade. Esse seria um trabalho simples se houvessem espaços físicos sobrando. O trabalho se torna complicado no CCET pela ausência de espaços suficientes, pela capacidade das salas que muitas vezes não atendem a demanda e pela ausência de colaboração entre cursos para ajustar horários que quando combinados maximizem a utilização dos espaços. Assim, o trabalho do gestor todos os períodos é de organizar todas as demandas de forma a atendê-las integralmente sem conflitos.

O processo funciona nas seguintes etapas:

- O coordenador de curso cadastra a turma no sigaa, composta por curso, disciplina (código e nome), departamento responsável, horário e vagas. Isso gera uma solicitação ao departamento responsável;
- O departamento recebe o pedido de turma e aloca um professor para lecioná-la. Ao fazer isto, o pedido segue para o CCET alocar uma sala de aula;
- O CCET aguarda o período de matrículas dos alunos. Após esse período, ele analisa o pedido de sala baseado na quantidade de alunos matriculados na turma. A análise consiste em: alocar uma sala disponível no horário especificado na turma, que caiba a quantidade de alunos matriculados.

Informativo:

Os horários cadastrados na turma estão no formato: DIAS TURNO HORÁRIO. Considere que as turmas ficam alocadas em somente uma sala e que os horários sempre ocorrem em apenas um turno.

Exs: 234T12 - 24T23 - 35M12

Possíveis valores para dias: 1 2 3 4 5 6 7

Possíveis valores para turno: M T N

Possíveis valores para horário: 1 2 3 4 5 6

Considerando a descrição acima, implemente um pequeno alocador de salas para o CCET.

Questões:

- 1) [2pt] Implemente as classes de model, que representam dados de turma e classes associadas. ✓
*não é necessário implementar get/set, mas no código pode considerar que elas existem
- 2) [3pt] [Necessita da questão 1] Implemente as classes dos services
→ Lembre-se de armazenar os dados em coleções adequadas para a finalidade
→ Lembre-se de declarar as classes, construtores, e tudo que for necessário
 - a. [Coordenação] cadastrar turmas por período ✓
 - b. [Coordenação] remover turmas pelo código da disciplina e período ✓
 - c. [Coordenação] consultar turmas por período ✓
 - d. [Departamento] associar turma a professor ✓
 - e. [Departamento] consultar turmas por período e professor ✓
- 3) [Necessita da questão 2] Implemente as funcionalidades dos services (cont.)
→ Lembre-se de armazenar os dados em coleções adequadas para a finalidade
→ Lembre-se de declarar as classes, construtores, e tudo que for necessário
 - a. [4pt] [CCET] funcionalidade para alocar a turma a uma sala, sem conflitos → não pode haver duas ou mais turmas no mesmo horário, mesmo que a colisão seja parcial
 - b. [1pt] [Necessita da letra a] [CCET] funcionalidade para listar todas as turmas com suas salas por curso e período ✓
 - c. [Extra][2pt] [Necessita da letra a] [CCET] funcionalidade para listar todas as turmas com seus horários por sala e período

**Não é necessário a implementação da camada view, e tudo de entrada e saída por terminal.

```
public void listarPor Sala Período (String sala, String período) {  
    for (Sala sala : salas.entrySet()) {  
        for (String key : coordenacao.getTurmas().entrySet()) {  
            if (sala.equals(sala) && key.equals(período)) {  
                System.out.println(sala + " " + key);  
            }  
        }  
    }  
}
```