

EXERCICIO

Minimundo tema:

Imagine o contexto de um restaurante universitário onde alunos e funcionários comprem refeições diariamente. O sistema a ser implementado permitirá o controle de alunos, funcionários, departamentos e cursos assim como controlará as refeições disponíveis no restaurante e quais refeições foram consumidas pelos alunos e funcionários.

Um Consumidor apresenta:

- um nome
- uma matrícula (por exemplo: 23)
- um ano de ingresso (por exemplo: 2014).
- Um sexo onde o Sexo pode assumir os valores MASCULINO e FEMININO
- Um título que pode assumir os valores ESPECIALIZACAO, MESTRADO e DOUTORADO.
- Um CPF apresenta dígitos verificadores que devem ser verificados durante a criação do professor. Olhem : http://pt.wikipedia.org/wiki/Cadastro_de_Pessoas_Físicas
- Não podem existir dois consumidores com o mesmo CPF.
- O sistema listará todos consumidores disponíveis até o momento apresentando o nome, a matrícula, o ano de ingresso, o sexo, o código do departamento (caso seja funcionário), o código do curso(caso seja aluno) e o CPF.
- O sistema permitirá alterar o nome, a matrícula, o sexo e o ano de ingresso de um consumidor.

O nome, a matrícula e o ano de um novo consumidor devem ser informados no momento de sua criação. Um Consumidor ou é um Aluno ou é um Funcionário. Um Aluno apresenta o Curso em que está inscrito e cada Funcionário apresenta o Departamento em que está lotado.

O Curso e o Departamento de um novo Aluno e Funcionario, respectivamente, devem ser informados no momento da sua criação (caso não existam o Aluno ou o Funcionário não podem ser criados, isto é, o Curso e o departamento deve ser criados antes).

Um Departamento:

- Apresenta um nome (por exemplo: Departamento de ciência da computação) e uma sigla (por exemplo: DCC).
- Tem vários Funcionários lotados nele.
- Tem vários Cursos.
- Não podem existir dois departamentos com a mesma sigla.
- O sistema permitirá a criação de novos departamentos

Um Curso:

- Apresenta um nome (por exemplo: Ciência da computação) e uma sigla (por exemplo: CComp).
- Apresenta um Departamento onde, não existe curso sem o seu respectivo Departamento.
- Tem vários Alunos inscritos
- Não podem existir dois cursos com a mesma sigla.
- O sistema permitirá a criação de novos cursos.

Os preços que deverão ser pagos pelo Funcionario ou pelo Aluno estão associados ao Turno da refeição. Por exemplo: de Manhã o aluno pagar 0.5 enquanto o Funcionario paga 3.0

O Turno pode apresentar três valores:

- Manhã: Onde o preço do Funcionario é 3.0 e o preço do Aluno é 0.5
- Tarde: Onde o preço do Funcionario é 6.0 e o preço do Aluno é 1.0
- Noite: Onde o preço do Funcionario é 6.0 e o preço do Aluno é 1.0

Uma Refeição apresenta um Turno uma descrição e uma opção vegetariana. Um exemplo de refeição apresenta uma descrição “Feijão, arroz, bife e batata-frita”, um Turno tarde e uma opção vegetariana “Bolinho de arroz com macarrão integral e soja”.

Uma Refeicao ou é um Dejejum ou é um Almoco ou é um Jantar. Onde:

- Um Dejejum é sempre do turmo Manhã.
- Um Almoco é sempre do Turno Tarde
- Um Jantar é sempre do Turno Noite.
- Após a criação da refeição o seu turno não pode ser alterado!
- O sistema permitirá a criação de novas refeições.

- O sistema listará todas as refeições disponíveis até o momento apresentando o turno, a descrição e a opção vegetariana.
- O sistema permitirá alterar o turno, a descrição e a opção vegetariana de qualquer refeição.

Um Ticket:

- Armazena a informação de que uma Refeição foi consumida por um Consumidor
- Isto é : Pertence a um Consumidor e Um consumidor tem vários Tickets.
- Apresenta um valor (Por exemplo : 0.5, 10.0, etc ...), se foi pago ou não e a Refeicao que foi consumida.
- O valor do ticket está associado com o Turno da refeição e qual é o consumidor (aluno ou funcionário).
- valor e a Refeição devem ser informados no momento da criação do Ticket e não podem ser alterados.
- O sistema listará todos os tickets apresentando o a matrícula do consumidor, o valor, se foi pago, O turno e a descrição da refeição.
- O sistema permitirá alterar no ticket apenas a informação de pagamento (isto é, se ele foi pago ou não).

1) Enumere os possíveis CRUD (Create, Read, Update e Delete) do minimundo.

2) Enumere os requisitos funcionais do sistema apresentado

3) modele as Fronteiras, os controladores e as entidades de (1) e (2)

4) Implemente Utilizando jsps e servlets (1), (2) e (3).

Dica : Armazene as novas instâncias de Entidades na sessão (até aprendermos JDBC).