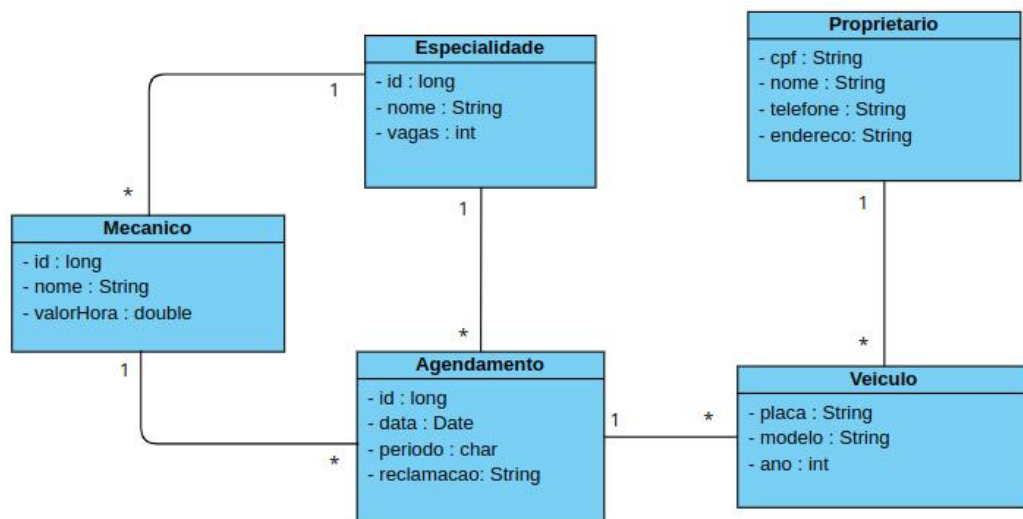


### Prova 3 - 2023/1

Crie um banco de dados no **Postgresql**, Utilize o script disponível no Moodle junto a esta prova para criar as tabelas do banco de dados, com dados necessários para os testes. Este script criará as tabelas **especialidade**, **mecanico**, **proprietario**, **veiculo** e **agendamento** e fará a inserção de alguns registros. No Netbeans, crie um projeto Java EE (com JSF e PrimeFaces) e execute os passos:

- a) No servidor Payara, crie um Connection Pool para o banco, e um Resource JDBC com o nome **jdbc/mecanica**
- b) No módulo EJB, crie uma Unidade de persistência com as entidades a partir do banco de dados, usando recurso criado acima. Use EclipseLink.
- c) Crie EJBs no módulo EJB, com métodos para realizar a marcação e o cancelamento de agendamentos, bem como para listar especialidades, mecanicos (geral e por especialidade), e métodos para listar agendamentos (por especialidade e por mecanico), inserir e remover (desmarcar) agendamentos, com a inclusão de veículos e proprietários. Regras a serem implementadas:
  - Os agendamentos são para um dia, podendo o período ser 'M' (matutino), 'V' (vespertino) ou 'I' (integral). Considere que os serviços sempre são terminados no mesmo dia.
  - Cada especialidade tem um número de vagas simultaneas (capacidade de consertar veículos ao mesmo tempo). Ao marcar um agendamento, é necessário contar quantos agendamentos existem naquela data no mesmo período ou integral. Impedir o excesso de agendamentos.
  - Ao cadastrar um agendamento, deve ser possível:
    - Pesquisar se o cliente (proprietário) já está cadastrado, pelo CPF. Se não estiver, cadastrar.
    - Pesquisar se o veículo já está cadastrado, pela placa. Se não estiver, cadastrar. LEMBRE-SE que os veículos podem mudar de dono (venda).
- d) No módulo WEB, crie conversores para alimentar os **select** de especialidade e de mecanicos da interface web.
- e) Crie um Bean Gerenciado para fazer/alterar/remover agendamentos, usando os componentes do PrimeFaces. Antes de gravar (persistir) um novo agendamento, verifique se a data é maior que a data atual, gerando um erro caso contrário. Verificar se há vaga na especialidade nesta data (ver acima). Implementar a marcação de novo agendamento, e para cada agendamento, um botão (método) para cancer. NÃO é necessário editar os agendamentos.
- f) Ao selecionar uma especialidade, filtrar os mecânicos daquela especialidade. Ao selecionar um mecânico, listar os agendamentos do mesmo.
- g) O acesso à tela de marcação deve ser restrito por um filtro, que verifique se o usuário está logado no sistema. Pode usar um usuário/senha fixos, ou uma tabela no banco de dados (acessada por um EJB)
- h) Crie webservices REST para buscar os agendamentos em determinadas datas, recebendo os dados como parâmetros na URL e retornando os dados em JSON



O projeto deverá ser compactado e postado no Moodle até o prazo indicado na tarefa.