Engenharia de Software
Professor: Eduardo Manuel de Freitas Jorge
(https://simcc.uesc.br/researcher?researcher_name=Eduardo%20Manuel%20de%20Freitas%20Jorge&search_type=name&terms=)

Estudo de Caso I [Agenda/Contato]

O objetivo deste projeto é a construção de um conjunto de classes que permitam a criação de uma agenda. A agenda por definição neste exemplo possui uma lista de contatos. Através da agenda será possível efetuar operações para adicionar, localizar, remover e listar os contatos. A unicidade dos contatos é controlada pelo seu telefone, ou seja, não deverão existir dois contatos com o mesmo telefone. A figura 1 apresenta o projeto do diagrama de classe da solução.

RF1 – Adicionar Contato

- **Objetivo:** Permitir a inclusão de um novo contato na agenda.
- **Entradas:** dados do contato (nome e telefone)
- **Validações:**
  - O telefone do contato deve ser único na agenda.
  - O contato não pode ter campos obrigatórios vazios (ex.: telefone, nome).
- **Saída/Resultado:**
  - Confirmação de sucesso com referência ao contato adicionado.
  - Em caso de duplicidade de telefone, rejeitar a operação com mensagem de erro adequada.

RF2 – Localizar Contato

- **Objetivo:** Permitir localizar contatos na agenda por telefone.
- **Entradas:** critérios de busca por telefone

- **Saída/Resultado:**
  - Caso encontrado: retornar o(s) contato(s) correspondente(s).

RF3 – Remover Contato

- **Objetivo:** Permitir remover um contato da agenda.
- **Entradas:** identificador único do contato por telefone
- **Validações/Comportamento:**
  - Confirmar a existência do contato antes de remover.
  - Remoção apenas do contato correspondente ao identificador fornecido.
- **Saída:**
  - Confirmação de remoção.
  - Em caso de não encontrado: mensagem de erro apropriada.

RF4 – Listar Contatos

- **Objetivo:** Fornecer uma visão completa ou filtrada dos contatos da agenda.
- **Entradas:** Nenhuma
- **Saída:**

- Lista de contatos.

A solução é composta da classe **AgendaMap** que implementa a lista de contatos com um MAP. A classe **AgendaList** faz uso de um LIST na sua implementação. Apesar da implementação para cada tipo de agenda ser diferente a interface realizada é a mesma (**IF_Agenda**). A realização da interface IF_Agenda garante a boa prática *"Program to an interface, not an implementation"*. A clsse **FabricaAgenda** é responsável pela criação das agendas e segue os padrões de projeto *Factory Method* e *Singleton*.

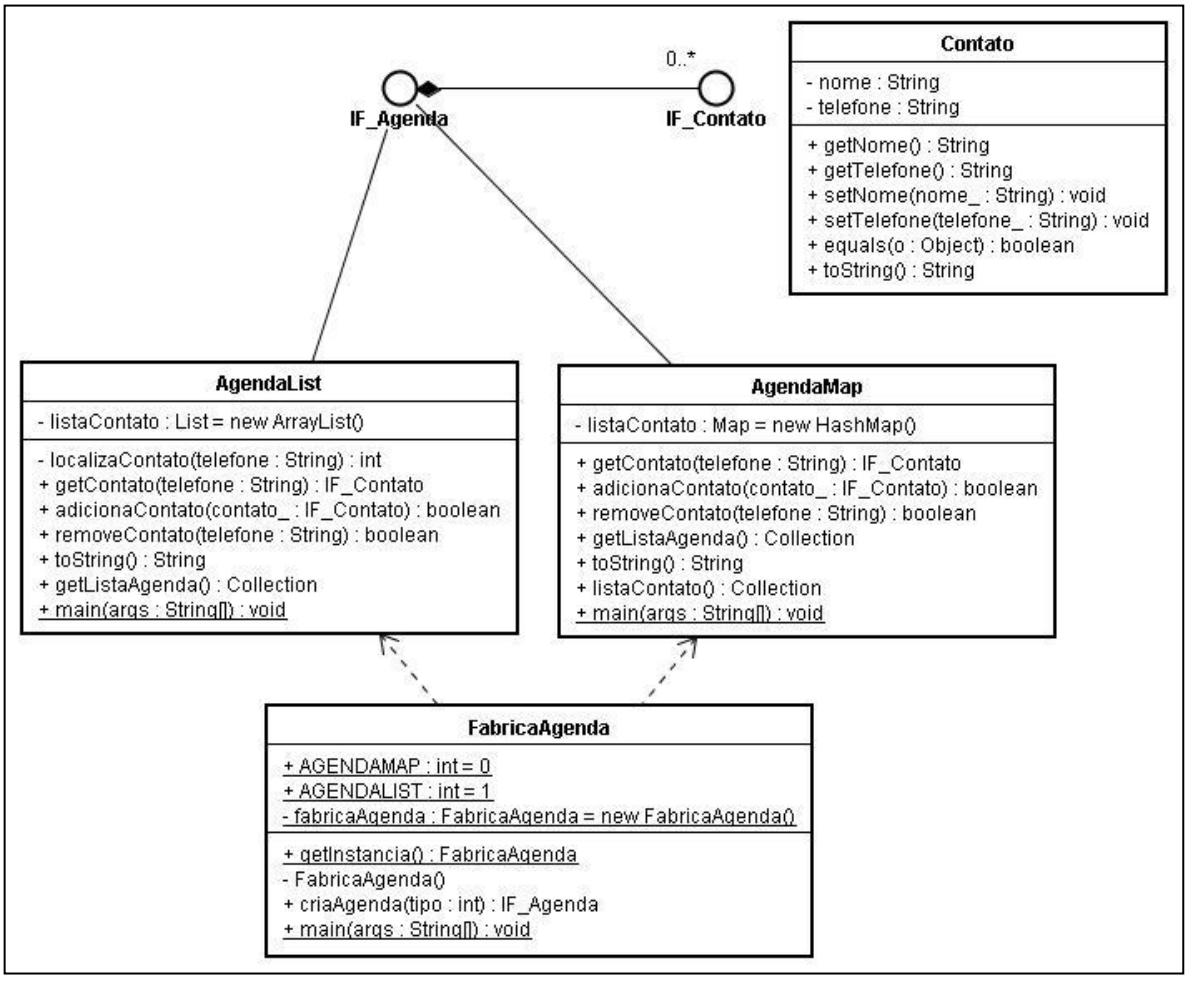

Figura 1: Diagrama de Classe

Apresentação em todas as quartas feiras para o professor e uma apresentação final no dia 17/09 para a turma

Entregáveis	Valor	Prazo (dias)
Protótipo de Interface e GitHub	1	2
Diagrama (Caso de Uso, Classe, Sequência, Entidade Relacionamento) e Matriz de Rastreabilidade	1	2

Quadro Kanban	1	Continuo
Camada de Aplicação e Persistência com testes unitários	2	5
Camada Interface Web	2	5
Mudança de Requisito	1	Na aula
Apresentação do Projeto	1	Na aula
Total=	10	10 (dias)