

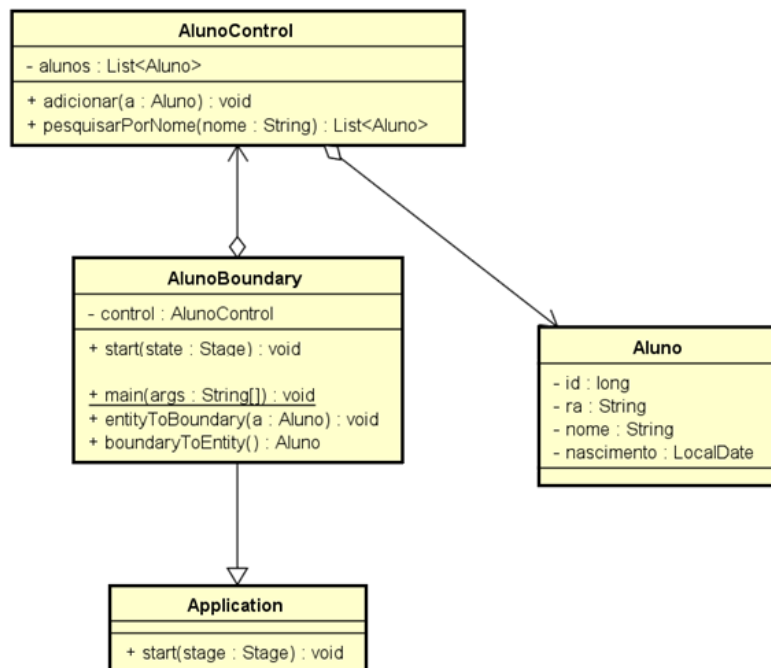
## Exercício - Aluno (Boundary, Control e Entity) - Usando Java FX

Crie um CRUD para fazer a manutenção de aluno, contendo o Boundary, o Control e a Entidade (BCE), vinculando os componentes gráficos do tipo TextField com as propriedades do aluno, e os dados da tabela devem mostrar todos os alunos que estão cadastrados, conforme layout abaixo:

Aluno:

Id  
Ra  
Nome  
Nascimento (pode ser tipo Date ou LocalDate)

Id	Ra	Nome	Nascimento
Não há conteúdo na tabela			



Para isto faça:

1. Crie a classe da Entidade com os atributos todos como privado, e os respectivos **gets** e **sets**
2. Crie a classe **Control** contendo um ArrayList do tipo da Entidade, escrevendo os métodos abaixo:
  - a. **void adicionar(<Entidade> e)**, deverá adicionar o objeto do tipo Entidade recebido como parâmetro no ArrayList
  - b. **<Entidade> pesquisarPorNome(String nome)**, deverá pesquisar no ArrayList para localizar se há um objeto do tipo da Entidade com o campo Nome contendo o texto que foi recebido como o parâmetro, em caso positivo deve retornar o objeto encontrado. Caso não encontre nenhum objeto, deverá retornar null.
3. Crie a classe **Boundary** conforme o diagrama de classe:
  - a. Declare os componentes gráficos necessários conforme o layout abaixo, como sendo variáveis de instância, lembre-se que você não precisa declarar os componentes do tipo Label, desde que os crie diretamente quando forem ser adicionados no GridPane.
  - b. Escreva o código do método **start()** para que ele crie um objeto do tipo **GridPane** e posicione os componentes gráficos conforme o layout abaixo.
  - c. Escreva o código do método **<Entidade> boundaryToEntity()** para que gere um novo objeto do tipo da Entidade e popule suas propriedades com as informações provenientes dos TextFields
  - d. Escreva o código do método **entityToBoundary (<Entidade> e)** para que leia as propriedades do objeto do tipo da Entidade recebido como parâmetro e popule os TextFields com estas informações.
  - e. Dentro da função **start()** registre uma função lambda/arrow no botão Adicionar para que execute a função **adicionar()** do **Control** quando for pressionado, e uma função lambda/arrow no botão Pesquisar para que execute a função **pesquisar()** do **Control** quando este botão for pressionado