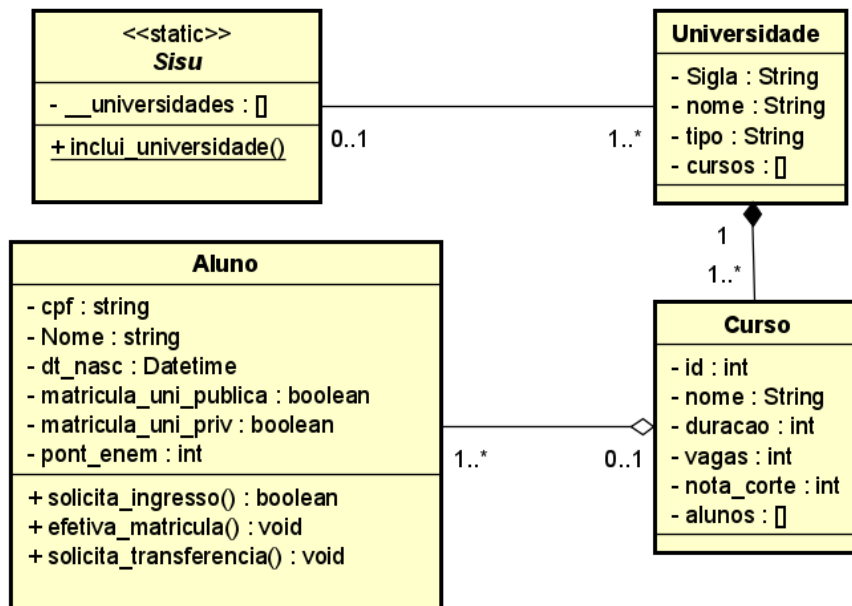


## EXERCICIO – ASSOCIAÇÃO DE CLASSES 1:N



Codifique a associação de classes representada no diagrama acima:

**Classe Aluno:**

Métodos:

- solicita\_entrada(curso, universidade)

O Objetivo deste método é retornar um valor True caso a pontuação do Aluno seja maior ou igual à pontuação do curso pretendido. Caso contrário, retornará um valor False.

-efetivar\_matricula(curso, universidade)

O objetivo deste método é efetivar a matricula no curso pretendido de certa universidade. Depende do retorno do método: solicita\_entrada. Em caso positivo, o numero de vagas restantes do curso é decrementado pois o aluno vai ocupar uma das vagas. Fazer a validação no caso de não haver mais vagas disponíveis.

OBS: O aluno só pode estar matriculado em apenas um curso de universidade pública.

- solicita\_transferencia(self, univ\_origem, curso\_origem, univ\_destino):

O objetivo deste método é transferir o aluno para um mesmo curso em outra universidade. Condição: Verificar se o aluno está matriculado no curso e respectiva universidade. A universidade destino tem que ter o curso e ter vaga. Não precisa checar a pontuação do enem.

### **Classe Curso:**

#### **Métodos:**

- cadastrar\_aluno(aluno)

O Objetivo deste método é incluir o aluno que solicitou matrícula e cumpre todos os requisitos (tem pontuação superior à nota de corte e o curso ter vaga disponível)

- \_\_str\_\_(self)

O objetivo deste método é imprimir um relatório com a relação de alunos matriculados.

### **Classe Universidade:**

#### **Métodos:**

- Cadastrar\_curso(self, curso):

O objetivo deste método é cadastrar uma instância que represente um curso. Verificar se a instância é realmente do tipo Curso.

- \_\_str\_\_(self)

O objetivo deste método é imprimir um relatório com a relação de cursos existentes na Universidade.

### **EXECUÇÃO**

Criar instancias de alunos, cursos e universidades e testar os métodos implementados.