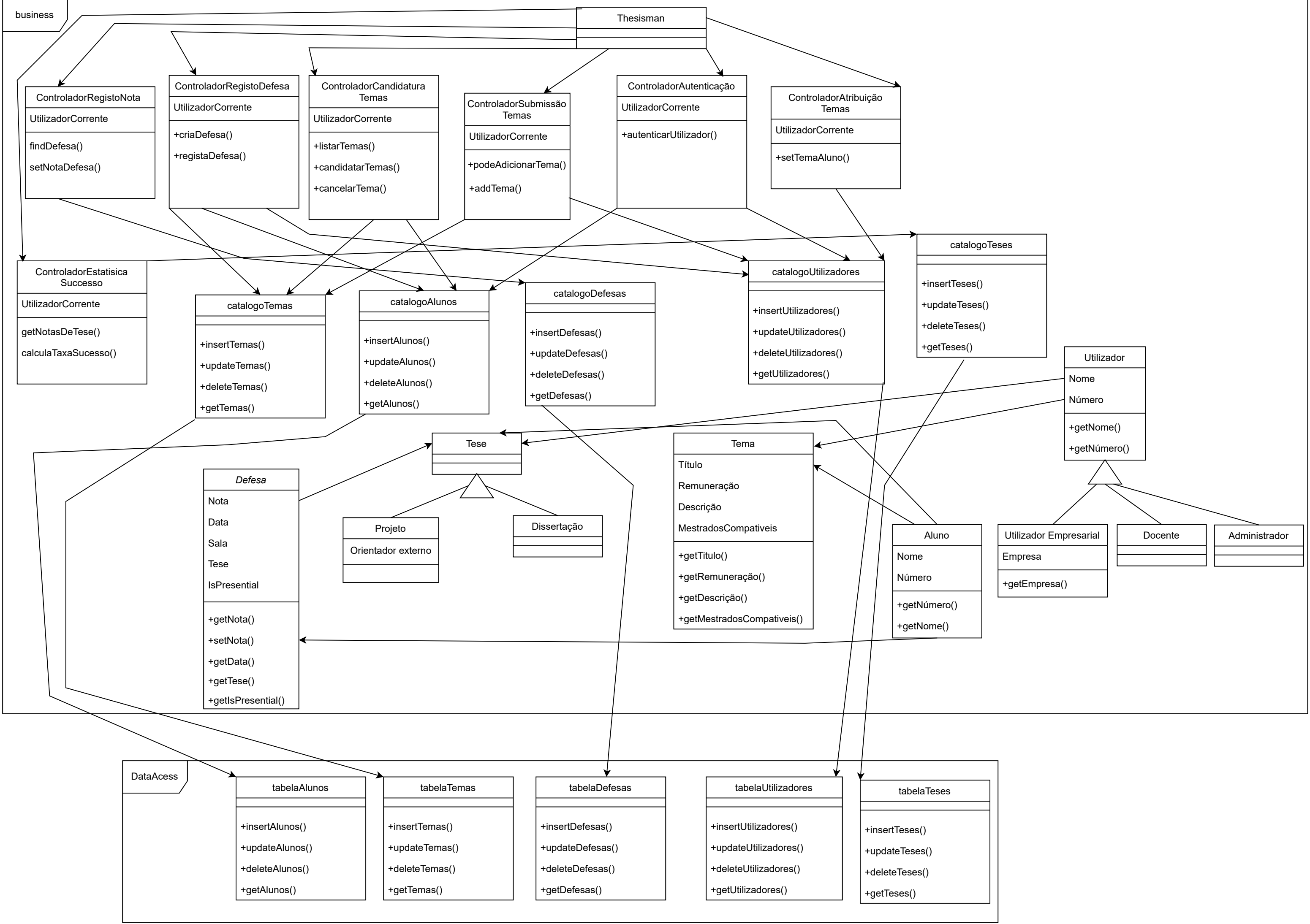


NOTAS:

A - No caso de o tipo de defesa ser presencial

B - No caso do tipo de defesa ser remoto



# Relatório Do Projeto

**@Entity** – Serve para guardar tabelas das classes, decidimos criar tabelas para todas as classes, exceto para os catálogos e os controladores, mas para as classes que sofrem de herança não foram criadas tabelas porque usamos o tipo de herança single table.

**@Inheritance(strategy = InheritanceType.SINGLE\_TABLE)** – Usamos para a classe Utilizador porque a mesma tem a classe Docente, UtilizadorEmpresarial e Administrador como seus filhos. Com o single table temos uma tabela para as 4 classes, juntando os atributos das quatro classes na mesma tabela. Para além disso adicionamos uma coluna para saber o tipo de Utilizador, ou seja, cada linha vai ter uma coluna a dizer “docente” ser for uma instância da classe docente, e o mesmo para o administrador e para o utilizadorEmpresarial. A mesma ideia foi implementada para a classe abstrata tese e seus filhos projeto e dissertação.

**@OneToOne** – Usamos para ligar duas classes quando uma instância classe tem apenas uma instância da outra, um exemplo foi a tese o aluno tem uma tese e por isso o atributo tese tem a tag @OneToOne.

**@ManyToOne/ OneToMany** – Usamos para ligar duas classes quando uma instância da classe tem várias instâncias da outra, e vice-versa. Um exemplo seria um aluno tem varias defesas, antes da final, e por tanto temos na classe aluno um atributo defesa com a tag @OneToMany.

**@Temporal (TemporalType.TIMESTAMP)** – para facilitar o armazenamento da data e hora na base de dados.