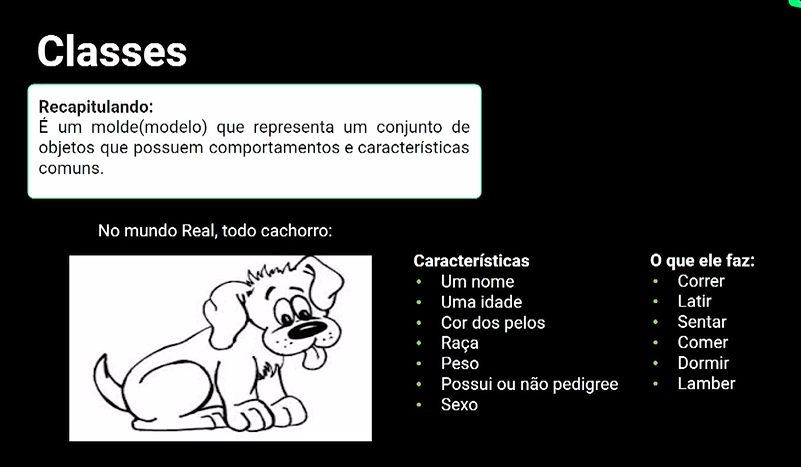
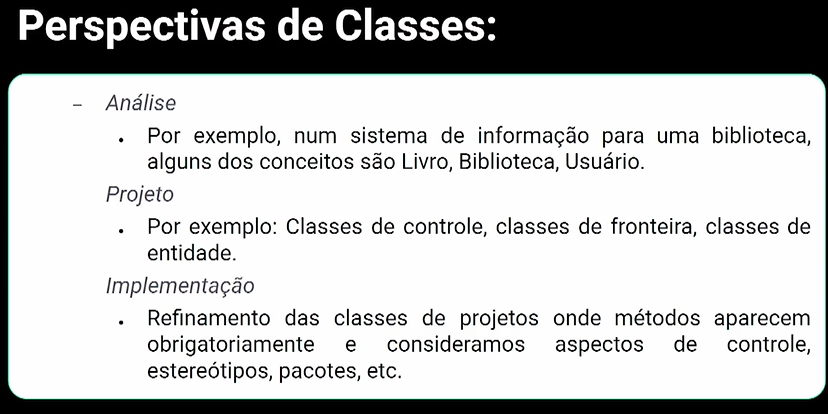


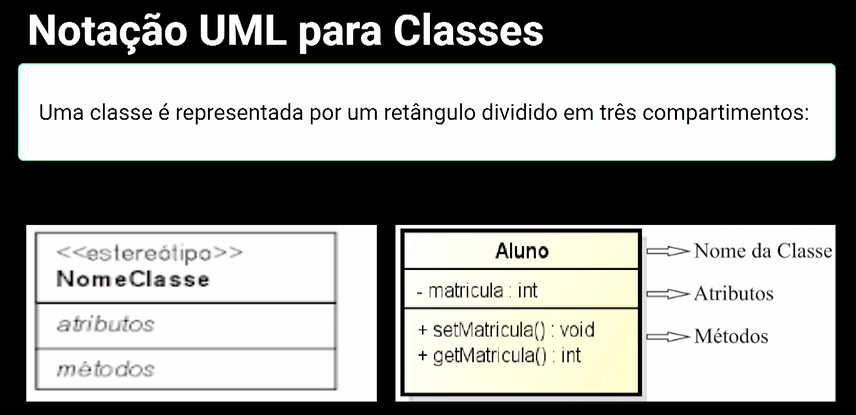
Diagrama de classe modelo que vai definir a estrutura que os dados terão dentro do contexto do negocio para poder da suporte as funcionalidades.

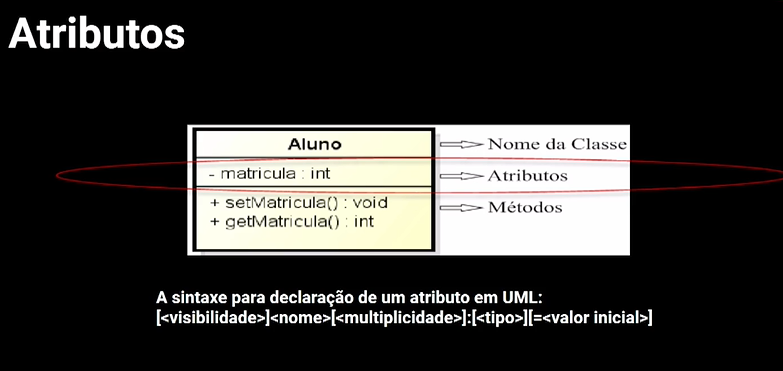
Diagrama de classe é um diagrama estrutural.

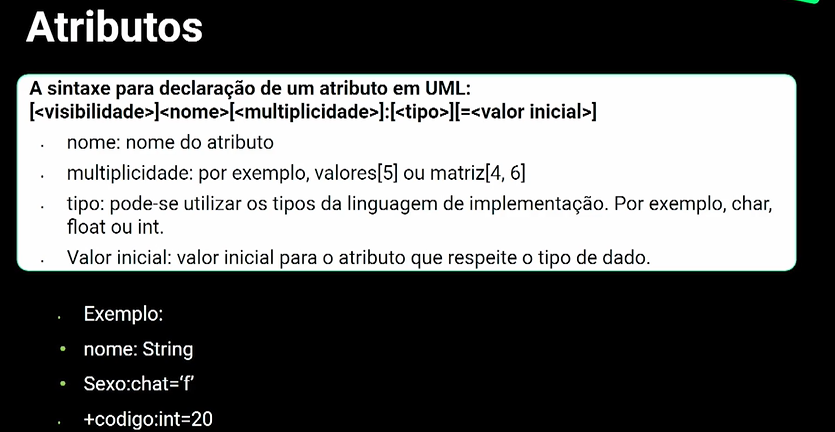


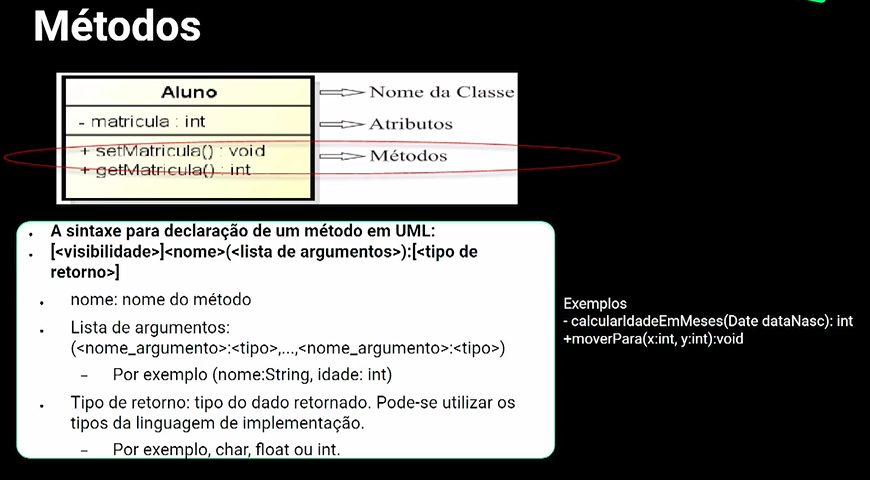




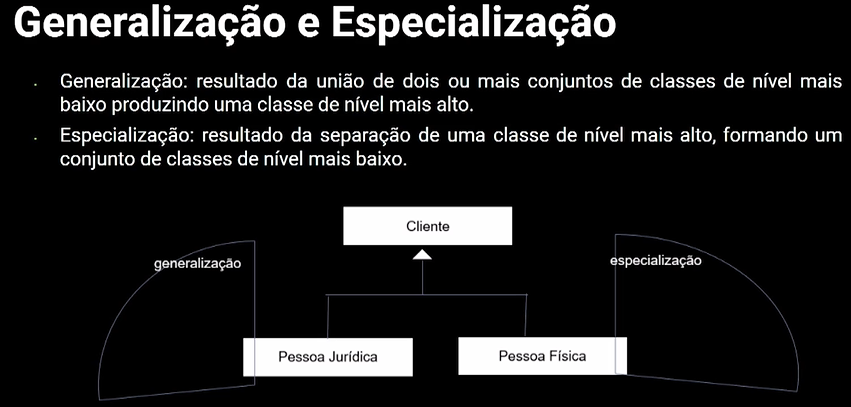


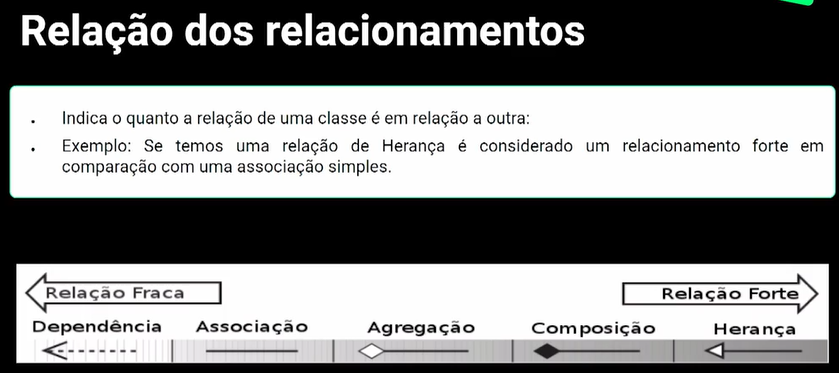






Notação:





**Dependência:**

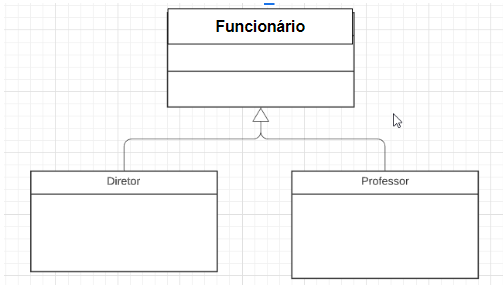
Uma dependência significa a relação entre duas ou mais classes em que uma mudança em uma, pode forçar mudanças na outra. No entanto, sempre criará um relacionamento mais fraco. A Dependência indica que uma classe depende de outra.

**Generalização / Especialização:**

A generalização é a representação que se dá a conjuntos de objetos que podem ser classificados em tipos diferentes. Geralmente existem semelhanças entre essas diferentes classes, o que faz com que criemos especializações.

Dentre as especializações, duas ou mais classes compartilham os mesmos atributos e / ou os mesmos métodos. Como você não precisa escrever o mesmo código repetidamente, deseja um mecanismo que aproveite essas semelhanças. Quando A herda (usa) de B, dizemos que A é a subclasse de B e B é a superclasse de A. Além disso, dizemos que temos “herança pura” quando A herda todos os atributos e métodos de B.

A notação de modelagem UML para herança é uma linha com uma ponta de seta fechada apontando da subclasse para a superclasses.

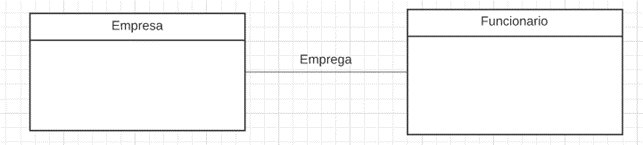


**Associação:**

Esse tipo de relacionamento representa relacionamentos estáticos entre as classes A e B. Por exemplo: um funcionário trabalha para uma organização.

Aqui estão algumas regras para associação:

* Associação é principalmente verbo ou frase verbal ou substantivo ou frase substantivo.
* Deve ser nomeado para indicar o papel desempenhado pela classe anexada no final do caminho da associação.
* Obrigatório para associações reflexivas

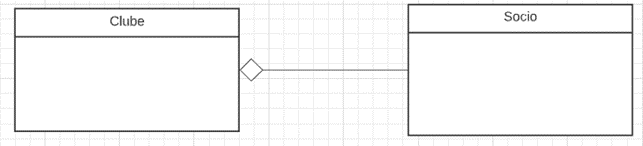


**Associação – Multiplicidade:**

Uma multiplicidade é um fator associado a um atributo. Ele especifica quantas instâncias de atributos são criadas quando uma classe é inicializada. Se uma multiplicidade não for especificada, por padrão, uma será considerada como uma multiplicidade padrão.

**Agregação**

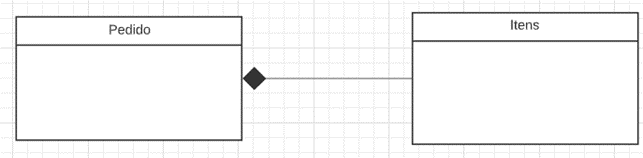
A agregação é um tipo especial de associação que modela um relacionamento de parte inteira entre agregado e suas partes.



**Composição**

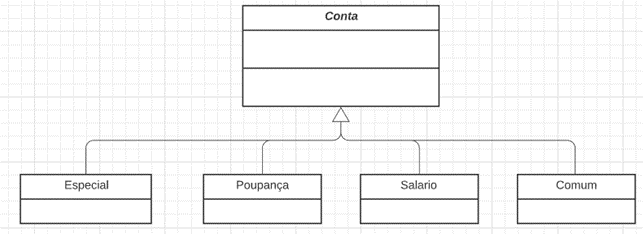
A composição é um tipo especial de agregação que denota forte propriedade entre duas classes quando uma classe faz parte de outra classe.

Por exemplo, se um pedido é composto de itens de pedidos. O pedido pode conter muitos itens. Portanto, se não tiver pedido, todos os itens de pedido também serão removidos.

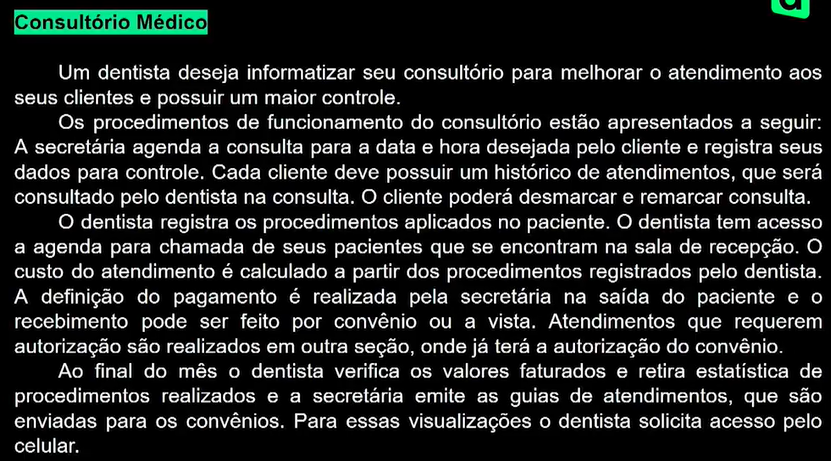


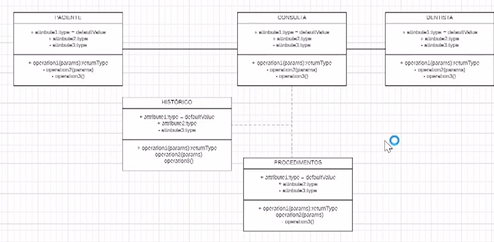
**Classes abstratas**

É uma classe com um protótipo de operação, mas não a implementação. Também é possível ter uma classe abstrata sem operações declaradas dentro dela. Um resumo é útil para identificar as funcionalidades entre as classes.



Ex de diagrama de classes:





Outro Ex:

