DIAGRAMA DE CLASSES

KARLLA FERREIRA LOIS

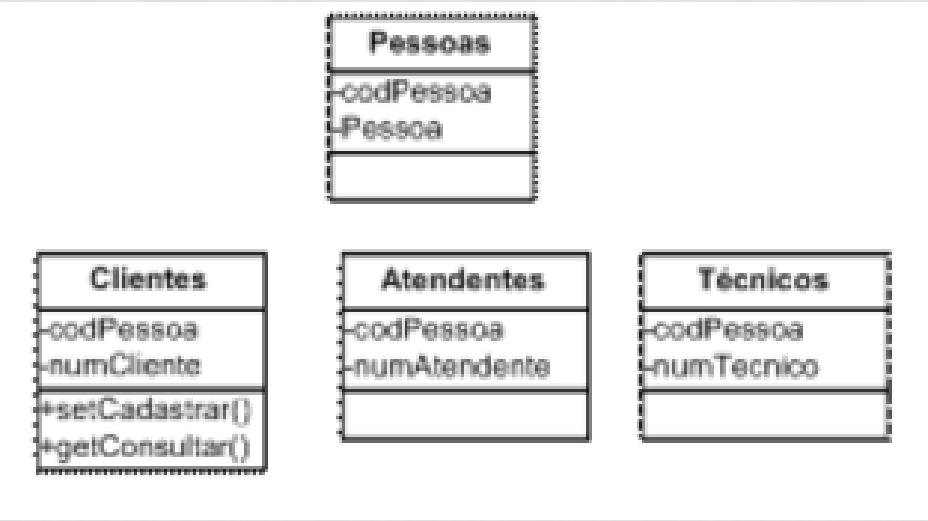
O QUE É UMA CLASSE?

- *"Grupo ou coleção de coisas que se distinguem das outras pela natureza"*.

- É uma abstração de um objeto da vida real, que agrupa dados (atributos)
  e procedimentos (operações) relacionados ao seu contexto.

CLASSE

- Uma classe é representada por um retângulo com três divisões, são elas: o nome da classe, seus atributos e os métodos.



- Cada classe do diagrama representa uma tabela do banco de dados.

POR QUE USAR UM DIAGRAMA DE CLASSES?

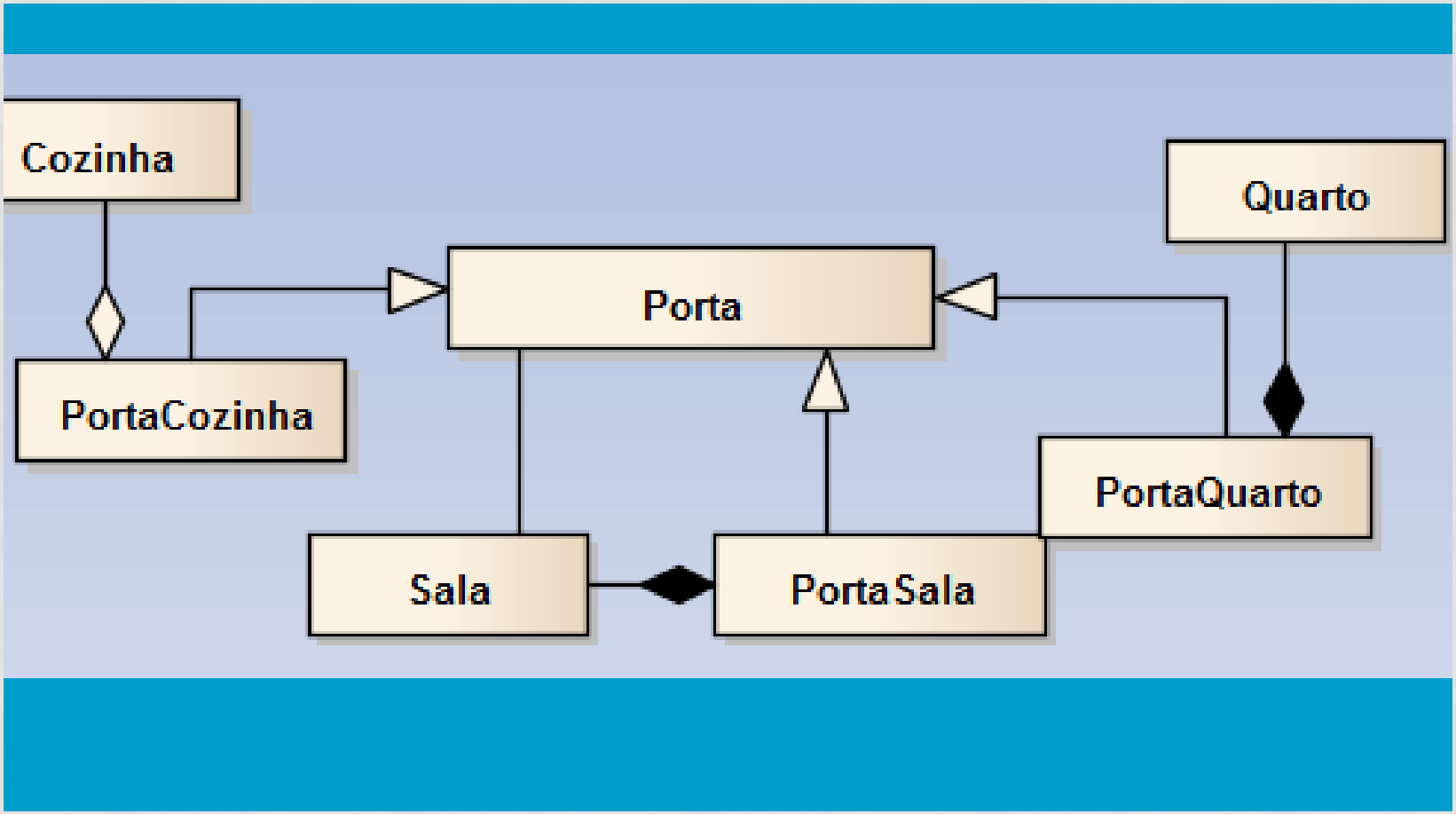
- Ilustra graficamente como será a estrutura do software, e como cada um dos componentes da sua estrutura estarão interligados.

ELEMENTOS MAIS UTILIZADOS

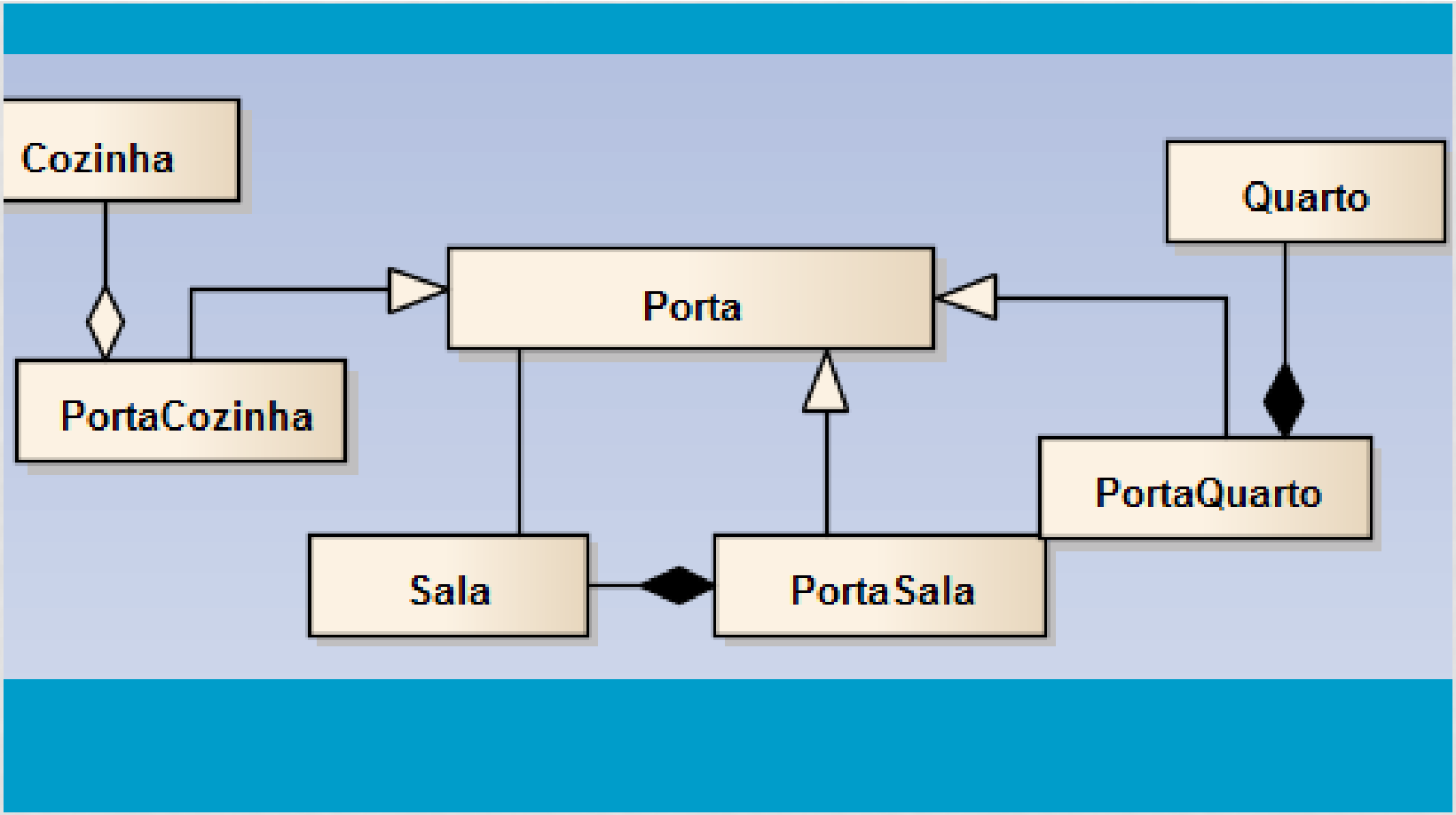
- Associação – É um tipo de relacionamento usado entre classes. Aplicável a classes que **são independentes** (vivem sem dependência umas das outras), mas que em algum momento no ciclo de vida do software (enquanto ele está em execução) podem ter alguma relação conceitual.
- Herança – É um tipo de relacionamento onde a classe generalizada (onde a "ponta da seta" do conector fica) fornece recursos para a classe especializada (herdeira). Excetuando conceitos mais avançados (como padrões de projeto, interfaces, visibilidades específicas etc.), **tudo que a classe mãe (generalizada) tem, a filha (especializada) terá**.

ELEMENTOS MAIS UTILIZADOS

- Composição – É um tipo de relacionamento onde a classe composta **depende de outras classes para "existir"**. Por exemplo, a classe "CorpoHumano" possui um composição com a classe "Coracao". Sem a classe "Coracao", a classe "CorpoHumano" não pode existir.
- Agregação – É um tipo de relacionamento onde a classe agregada **usa outra classes para "existir", mas pode viver sem ela**. Por exemplo, a classe "CorpoHumano" possui uma agregação com a classe "Mao". Sem a "Mao" a classe "CorpoHumano" pode existir.



- Cozinha **pode ter ou não** uma PortaCozinha, **podendo existir** se não tiver. (Agregação)
- PortaCozinha generaliza Porta, **possuindo todas** as características que Porta têm, além das suas específicas. (Generalização)
- Quarto **deve ter** PortaQuarto, **não podendo existir** se não tiver. (Composição)



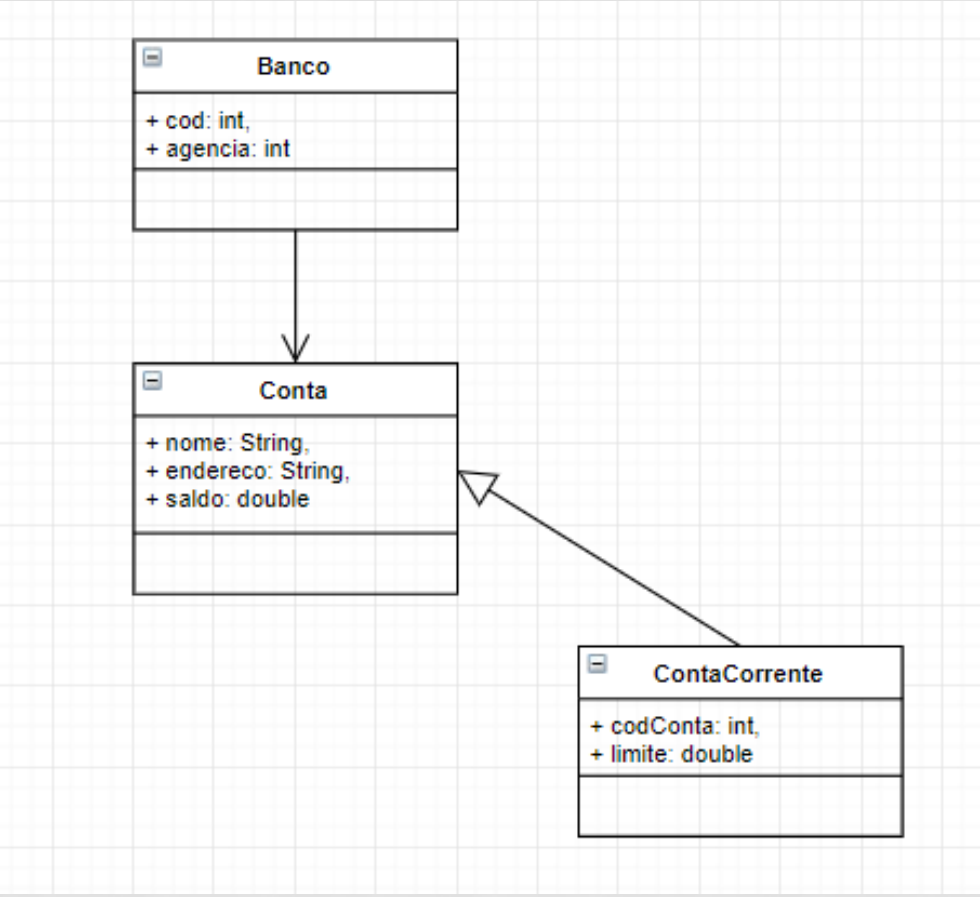
- PortaQuarto generaliza Porta, **que tem** todas as características que Porta têm, além das suas específicas. (Generalização)
- Sala **deve ter** PortaSala, **não podendo existir** se não tiver. (Composição)
- PortaSala generaliza Porta, **que tem** todas as características que Porta têm, além das suas específicas. (Generalização)
- Sala **pode ter ou não** uma Porta que não seja uma PortaSala, mas se tiver ou não isso não fará diferença, pois Porta **pode existir sem** Sala, e Sala **pode existir sem** Porta. (Associação).

QUANDO USAR?

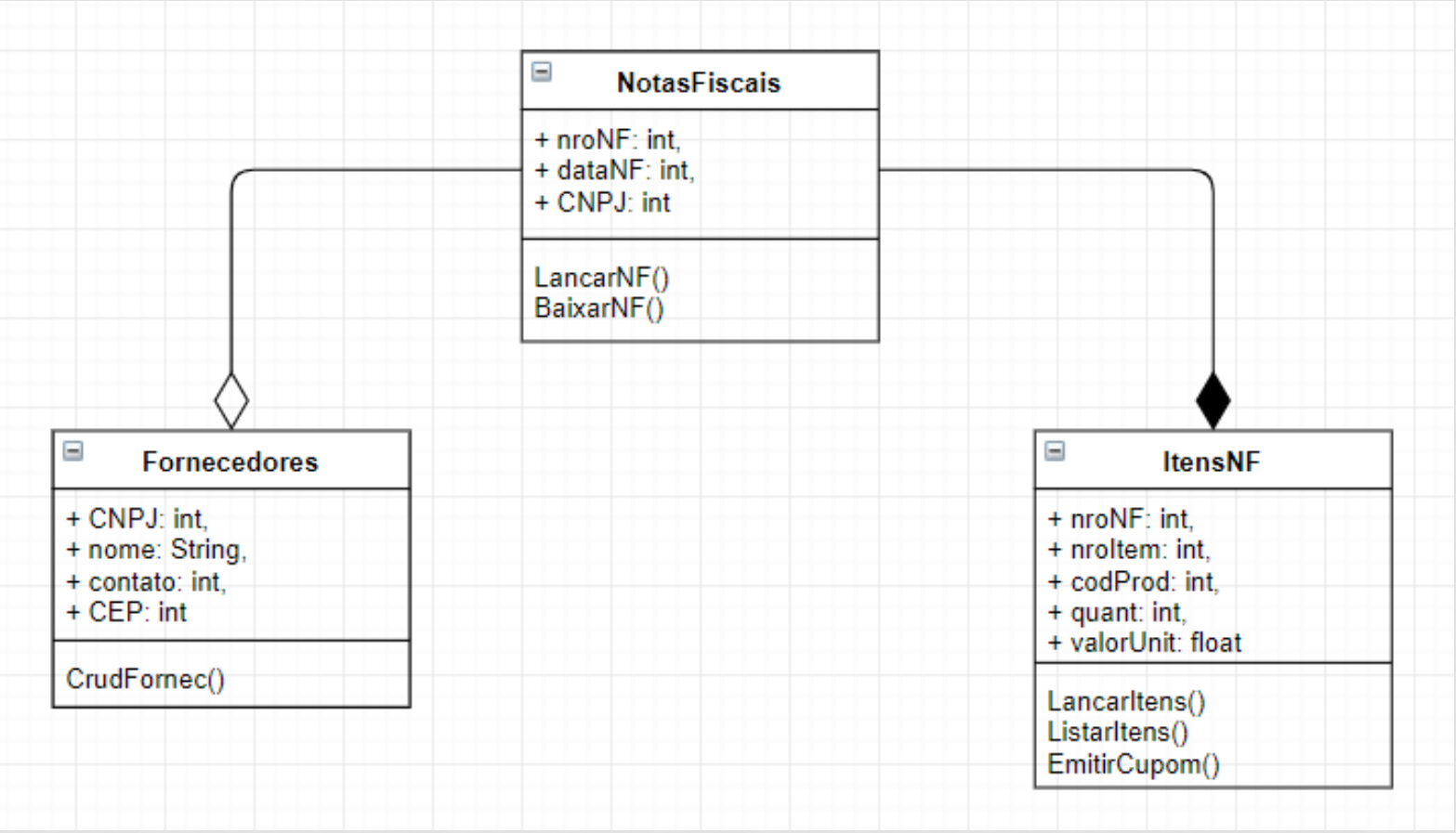
Pode ser utilizado para diferentes finalidades no contexto de produção de software, destacando:

- Design (projeto de software): definir um modelo a ser seguido para um software que ainda não existe, especificando um modelo antes de efetivamente construir software executável (diminuindo o desperdício).
- Engenharia Reversa: refletir a estrutura já existente de um software onde, a partir do código fonte realiza-se a leitura automática (ou não) das classes e suas relações e se produz o diagrama (para compreensão da estrutura do software executável).
- Esboço de ideias: desenhar modelos estruturais para troca de ideias e alinhamento entre analistas de sistemas, por exemplo.

EXEMPLO: BANCO



EXEMPLO: NOTA FISCAL (PROFESSOR)



REFERÊNCIAS

- https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224
- https://www.ateomomento.com.br/uml-diagrama-de-classes/