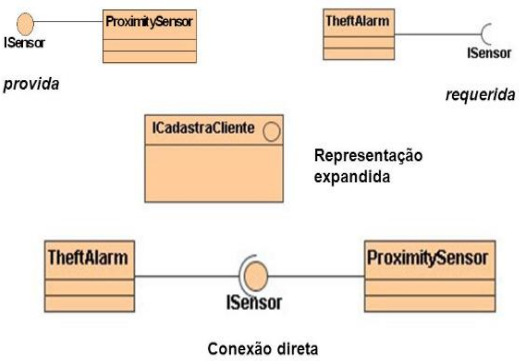
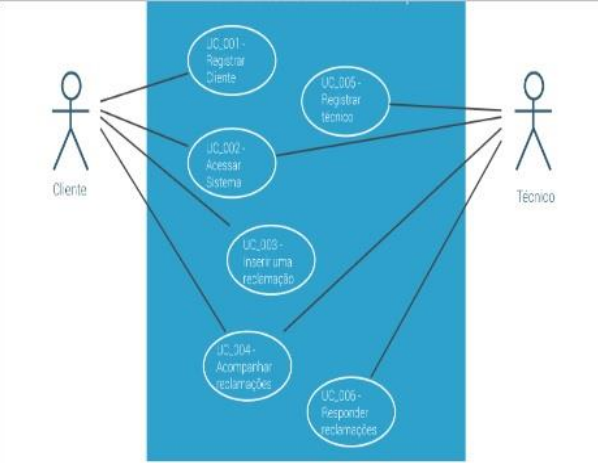


DIAGRAMA DE CLASSES - CONCEITOS
O que é o <b>Diagrama de Classes</b> ?
O que são as <b>Interfaces</b> de um Sistema?
Qual a diferença entre as <b>Interfaces Providas</b> e as <b>Requeridas</b> ?
Demonstre como representamos as Interfaces Requeridas e Providas em UML:
- reservada para a questão acima -
Como o <b>Diagrama de Casos de Uso</b> têm um impacto direto sobre a criação de um Diagrama de Classes?
- reservada para a questão acima -
- reservada para a questão acima -
- reservada para a questão acima -
Que <b>passo a passo</b> podemos seguir á partir do Diagrama de Casos de Uso para gerar o nosso Diagrama de Classes?
DIAGRAMA DE CLASSES - CONCEITOS

DIAGRAMA DE CLASSES - CONCEITOS
<p><b>Diagrama de Classes:</b> é mais uma das ferramentas UML (Unified Modeling Language), mas diferente do Diagrama de Casos de Uso, que tem como foco mostrar somente o lado funcional de um sistema, o Diagrama de Classes já foca em algo mais profundo, que é a estrutura do sistema. É por através dele que nós definimos como serão as nossas Classes, Interfaces e os relacionamentos entre elas. Para isso o Diagrama de Classes utiliza ferramentas UML que nós já vimos, como as Classes, Atributos, Métodos e Associações.</p>
<p><b>Interface:</b> é a coleção de operações das classes que nós desejamos deixar disponíveis para os usuários. As interfaces fazem a interação entre o usuário externo, a aplicação e os usuários internos de um sistema. Como o Sistema de Classes são o que nós chamamos de Sistema de Baixo Acoplamento (Sistemas que não estão fortemente interligados), as interfaces são responsáveis por fazer a ligação entre as classes. Entre elas temos as Interfaces Providas e as Requeridas.</p>
<p><b>Interface Provida:</b> é uma interface que é provida pela própria classe e é usada como forma de acessar os elementos da classe provedora. Por exemplo, o método de uma classe que fica público para ser acessado de fora da classe;</p> <p><b>Interface Requerida:</b> é uma interface que é requisitada por uma classe externa para a classe interna que pode fornecer àquele elemento. Por exemplo, o método de uma classe que é acessado diretamente por outra classe. A Interface Requerida não fica exposta publicamente, ela é usada somente entre as classes fornecedora e requisitora;</p>

<p>Por que o Diagrama de Casos de Uso já irá nos entregar uma ideia geral das funcionalidades de um sistema, facilitando a criação das classes, associações e interfaces que irão compor o nosso projeto usando POO. Por exemplo, veja como o Diagrama de Casos de Uso pode facilitar a nossa visão sobre um Sistema de Gerenciamento de Reclamações...</p>

<p>Percebe que o Diagrama de Casos de Uso praticamente já entrega de mão beijada a Interface do Sistema, entre as operações que estarão disponíveis para os Clientes e os Funcionários que utilizarão o Sistema.</p>
<p>1º Definir as Classes do Sistema; 2º Definir os Atributos das Classes; 3º Definir os Relacionamentos entre os atributos; 4º Definir os Métodos para a Classes; 5º Procurar semelhanças entre as Classes que poderiam ser melhor interpretadas por através de Heranças; 6º Definir as Associações entre as Classes e Atributos; 7º Definir Níveis de Encapsulamento entre Atributos e Métodos, para gerar a Interface;</p>
DIAGRAMA DE CLASSES - CONCEITOS

Como podemos seguir o **1º Passo** da Diagramação de Classe: Definição de Classes?

Como podemos seguir o **2º Passo** da Diagramação de Classe: Definição de Atributos?

- reservada para a questão acima -

Como podemos seguir o **3º Passo** da Diagramação de Classe: Relacionamento entre os Atributos?

- reservada para a questão acima -

- reservada para a questão acima -

Como podemos seguir o **4º Passo** da Diagramação de Classe: Definição de Métodos?

- reservada para a questão acima -

- reservada para a questão acima -

Como podemos seguir o **5º Passo** da Diagramação de Classe: Busca por Heranças?

DIAGRAMA DE CLASSES - CONCEITOS

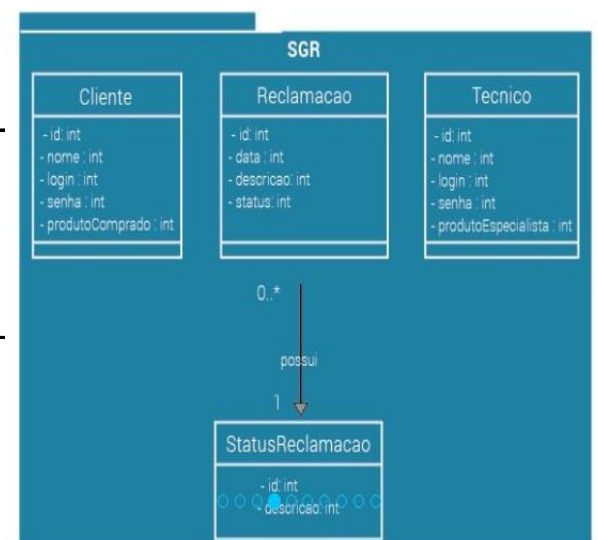
Por definir os elementos do Diagrama de Casos de Uso que poderão gerar **Classes**...



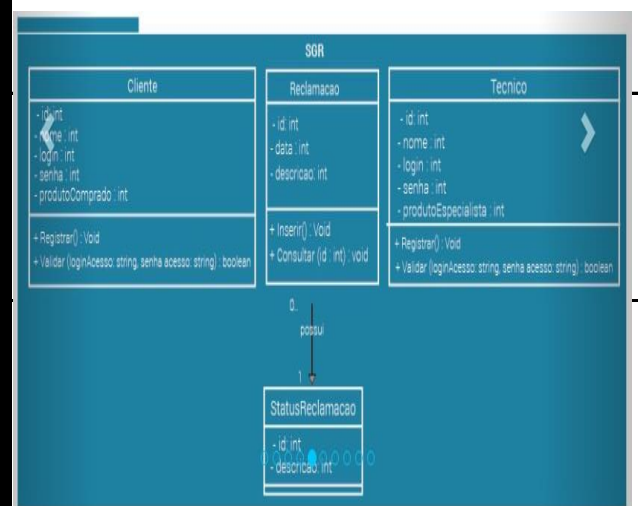
Esse passo vai depender da abstração que nós desejamos gerar, é nesse passo que nós **analisamos criticamente as classes em busca dos atributos que realmente serão relevantes para o nosso sistema** e para os relacionamentos entre as Classes.



Nesse passo nós analisamos os atributos para verificar se possuem entre eles algum tipo de **agregação, composição ou generalização**...



Nesse passo já podemos implementar as **operações que identificamos até esse ponto**, porém, temos que ter a consciência de que mais operações poderão ser identificadas no decorrer da diagramação...



Podemos **buscar por semelhanças entre os atributos e métodos** das classes existentes na busca pela geração de Super Classes que poderiam derivar esses elementos sobre as classes existentes...

DIAGRAMA DE CLASSES - CONCEITOS

Ilustre como Podemos implementar a Herança no nosso Diagrama de Classes:

- reservada para a questão acima -

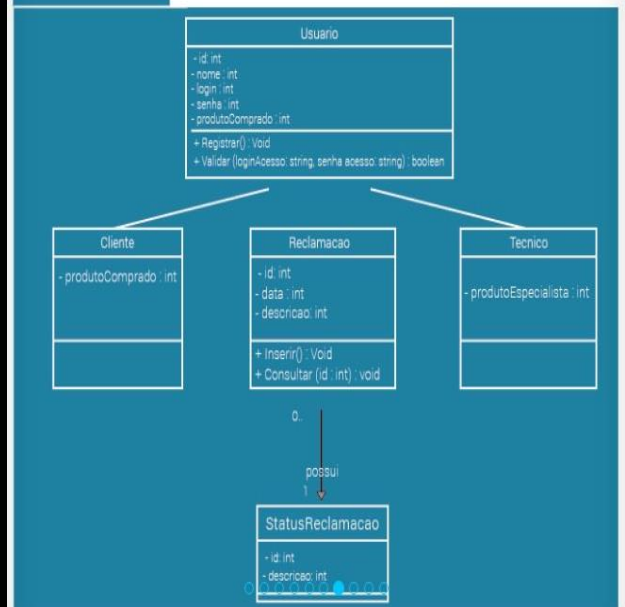
- reservada para a questão acima -

Como podemos seguir o 6º Passo da Diagramação de Classe: Definição de Associações?

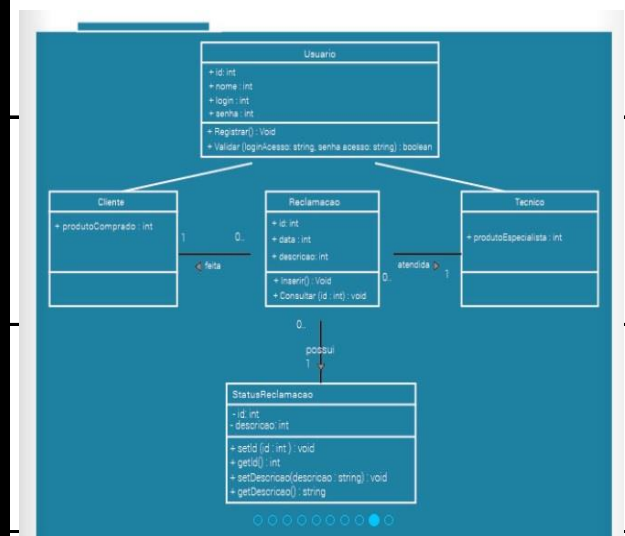
- reservada para a questão acima -

- reservada para a questão acima -

Como podemos seguir o 7º Passo da Diagramação de Classe: Níveis de Encapsulamento?



Por definir as associações entre as classes e os atributos de forma definitiva, estabelecendo seus **graus de multiplicidade** e **formas de relacionamento**.



Definindo que elementos ficarão **privados** e quais serão **públicos**, geralmente pelo uso de **getters** e **setters** - como no Java por exemplo. Dessa forma iremos definir a nossa interface, deixando visível ao usuário somente aquilo que nós desejarmos, enquanto os elementos que exigirem uma segurança maior, permanecerão encapsulados.

[illegible]

[illegible]

[illegible]

