

**ANÁLISE E DESENVOLVIMENTO DE SISTEMAS – 4º SEMESTRE MATUTINO – 2016**

**Lista 1 – Engenharia de Software III**

**NOMES:**

**Caio Larroza de Oliveira 1680481511006**

**Giovanni Armane 1680481511016**

**Leonardo Costa 1680481512015**

**Matheus dos Santos 1680481511044**

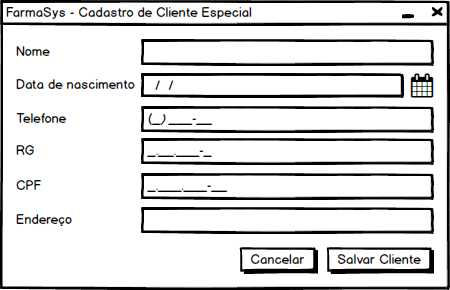
**SÃO CAETANO DO SUL**

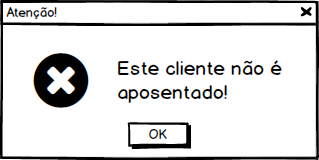
**2016**

**1 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU01. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

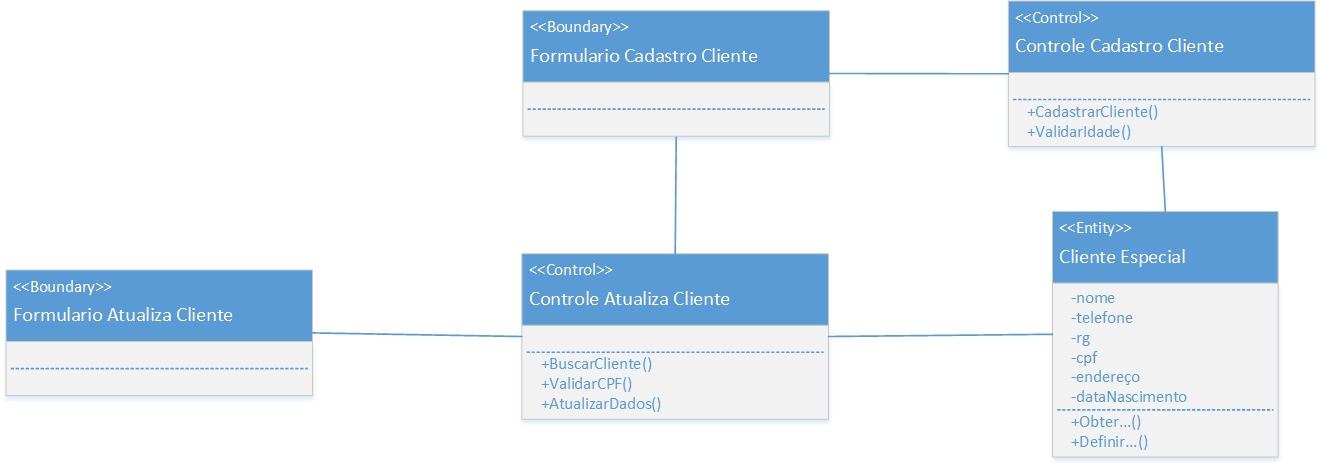


**2 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU01.

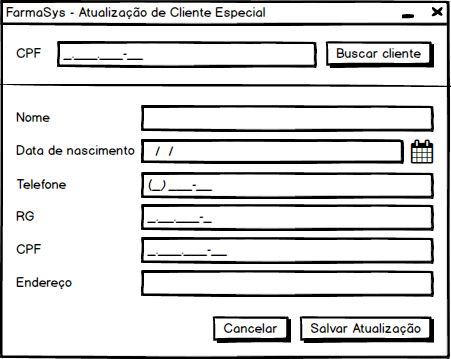


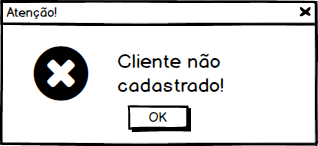


**3 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU02. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

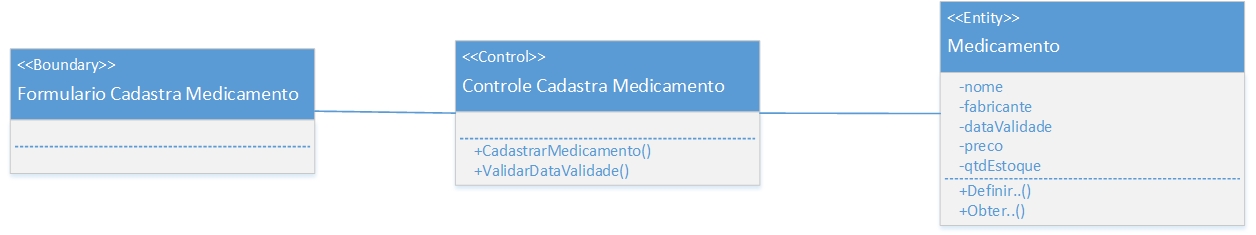


**4 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU02.

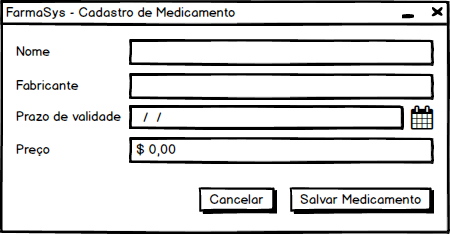


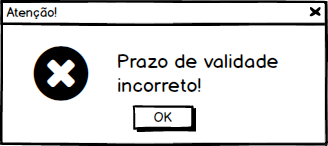


**5 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU3. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

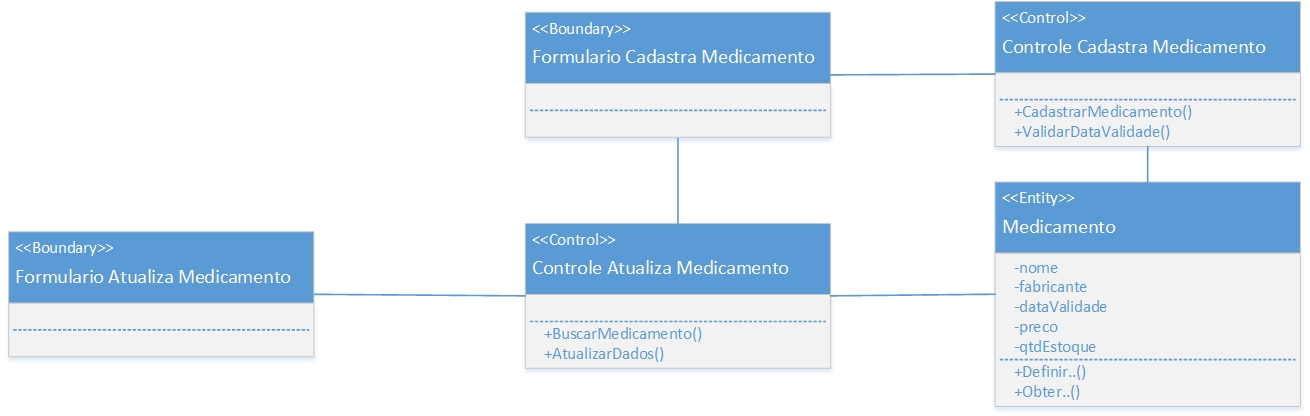


**6 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU03.

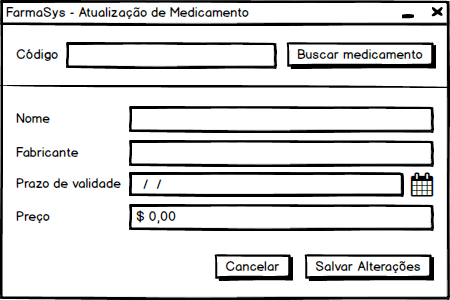


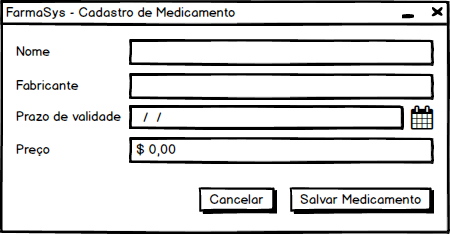


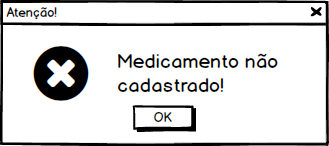
**7 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU4. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.



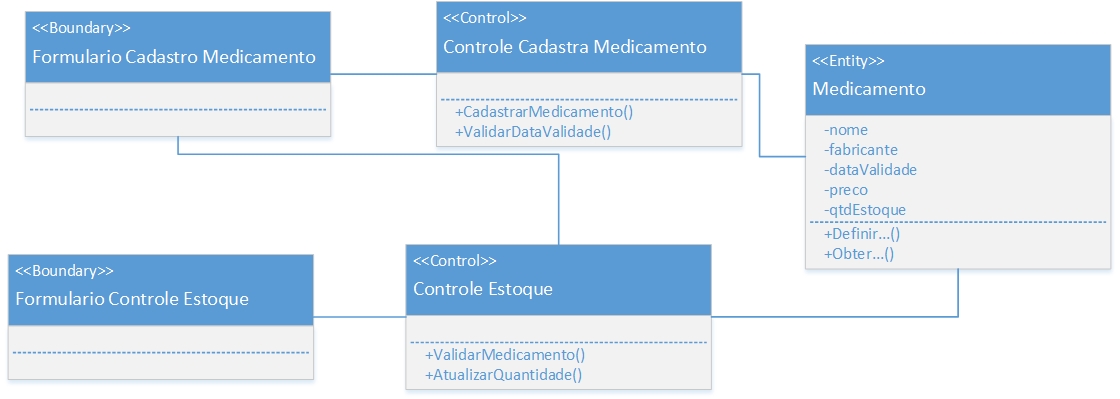
**8 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU04.



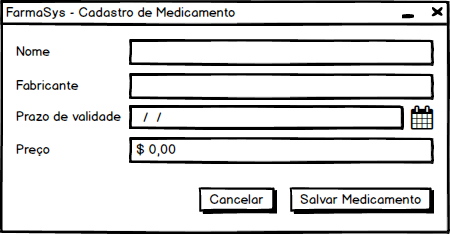


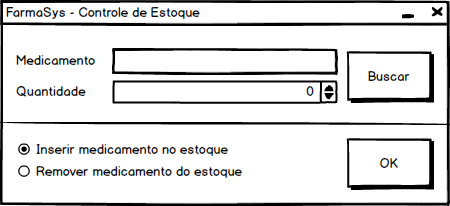


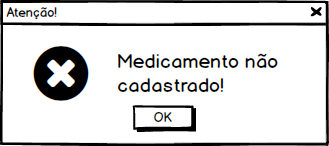
**9 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU5. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

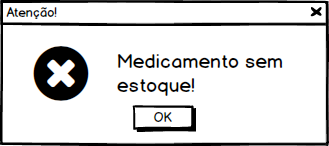


**10 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU05.

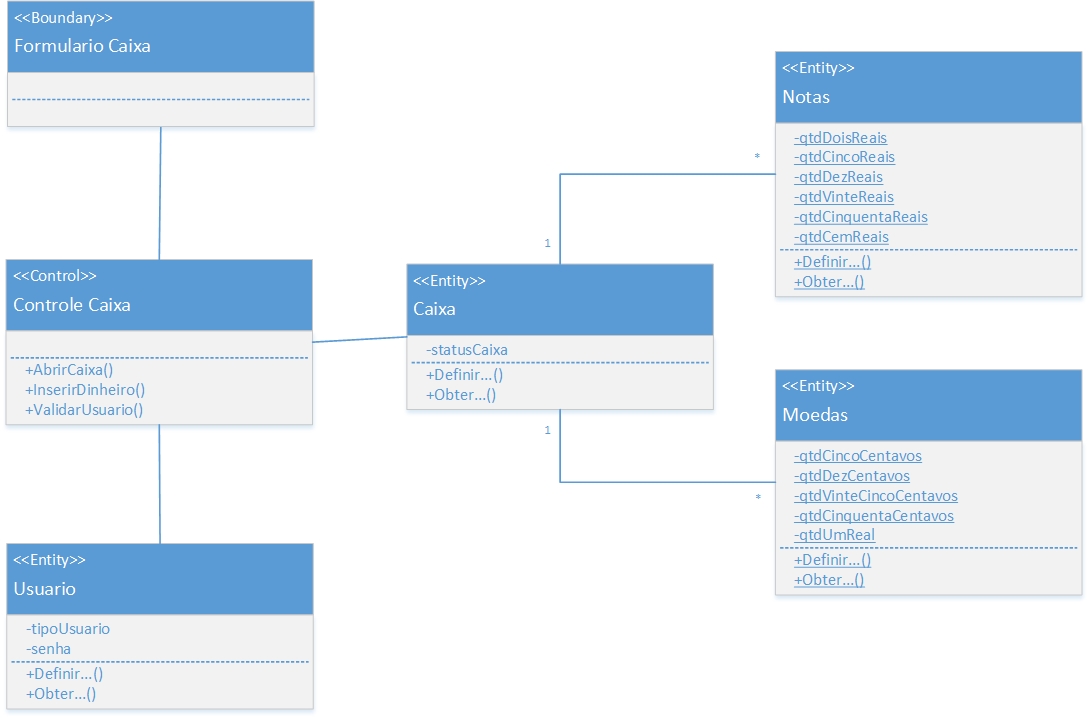




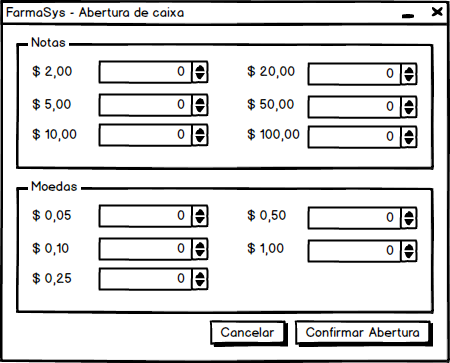


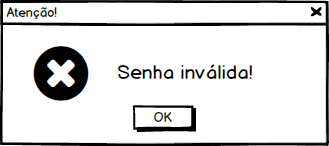


**11 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU6. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

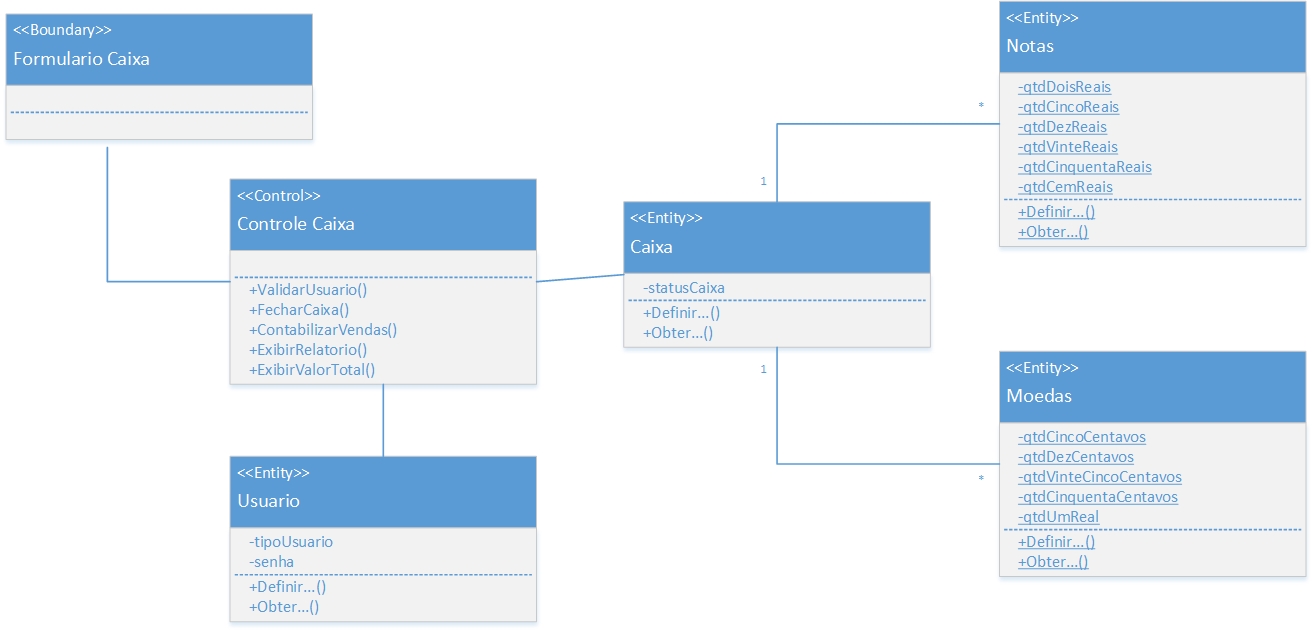


**12 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU06.

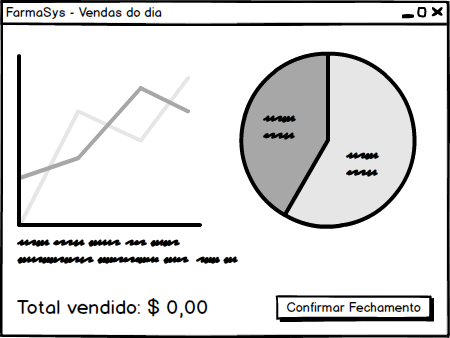


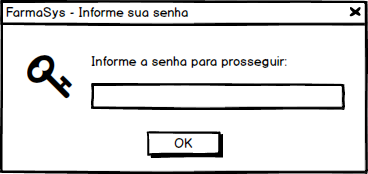


**13 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU7. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

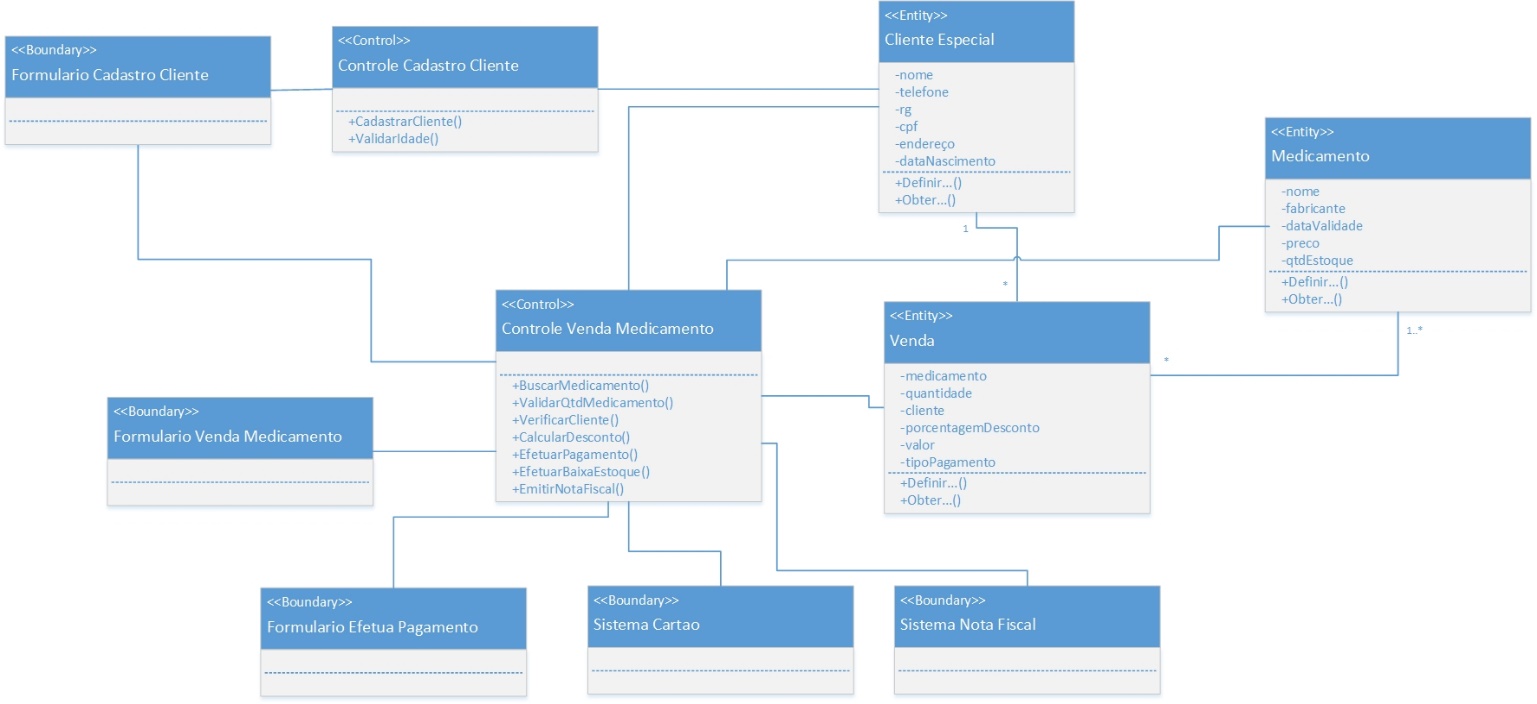


**14 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU07.

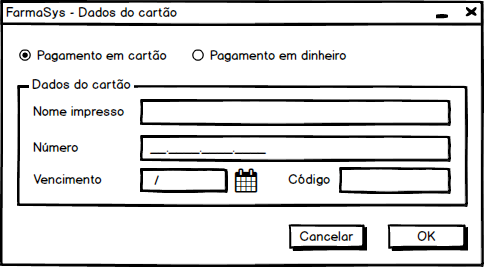


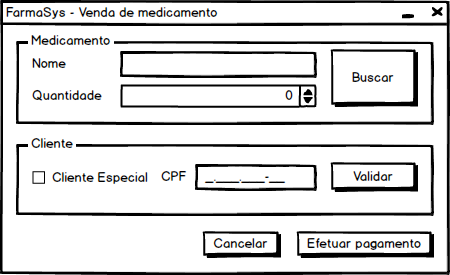


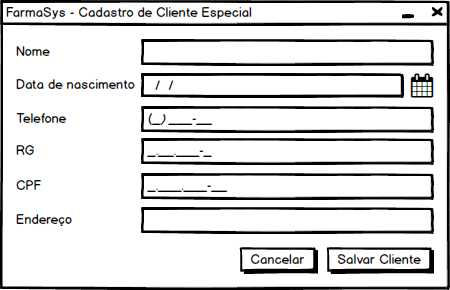
**15 -** Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU8. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

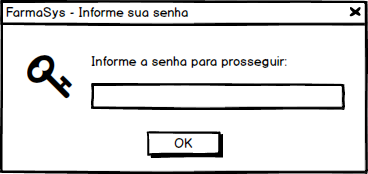


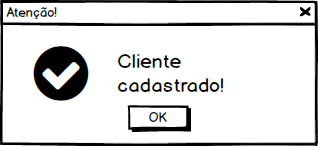
**16 -** Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU08.

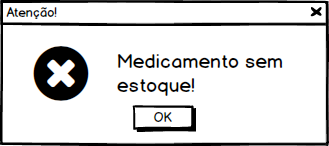








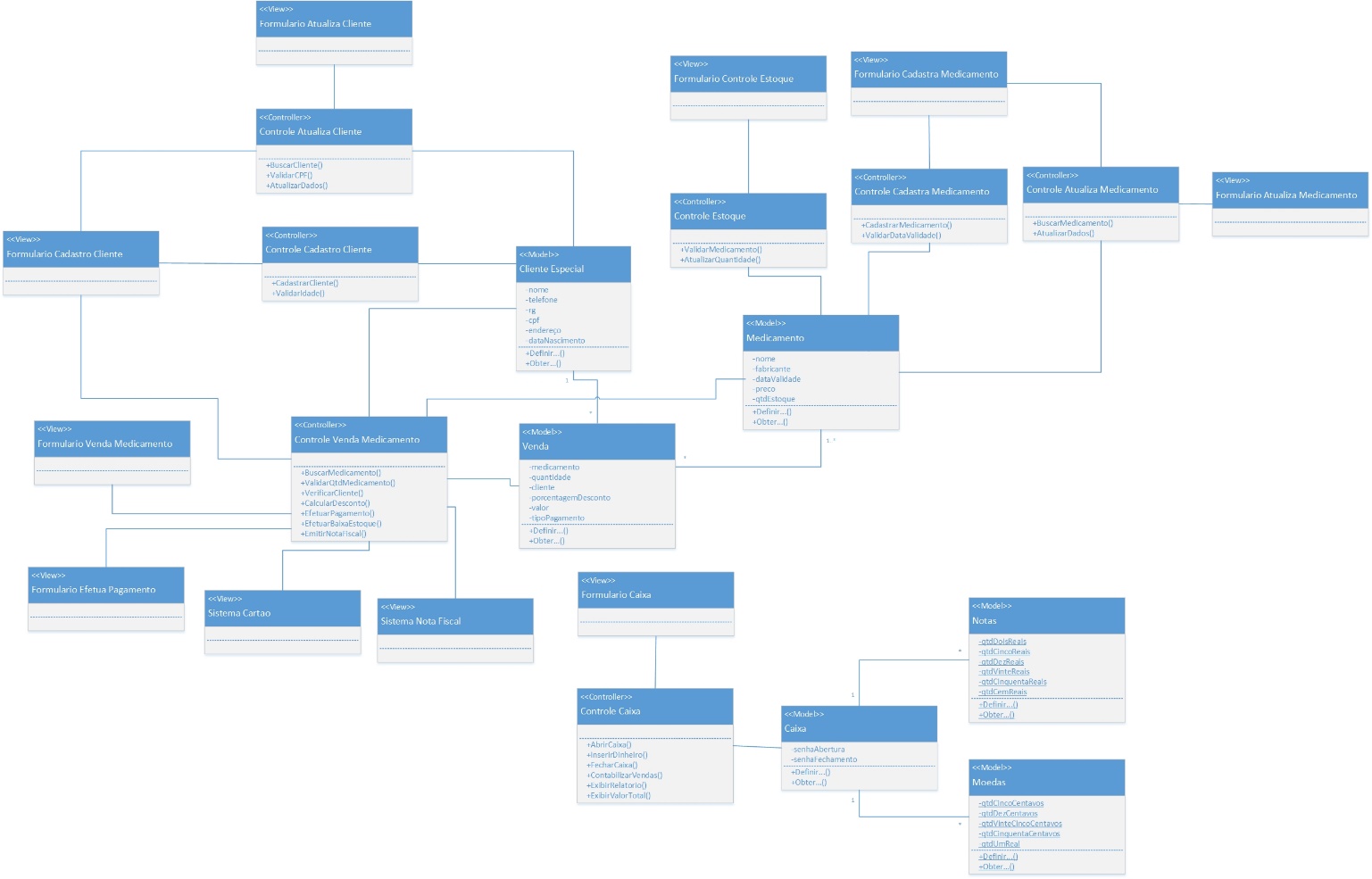




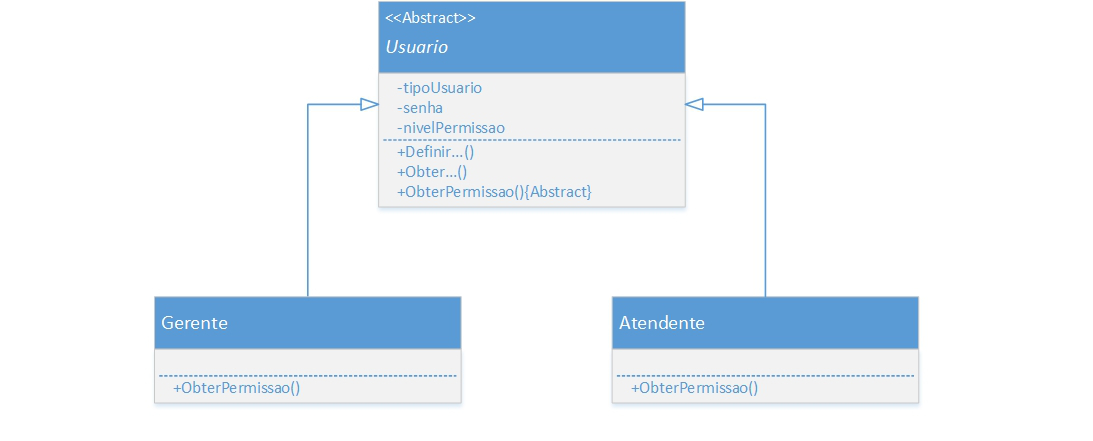
**17 -** Modele um diagrama de classes de análise a partir das VCPs modeladas e mantenha a utilização da categorização BCE. Os devidos atributos e métodos devem continuar sendo exibidos.



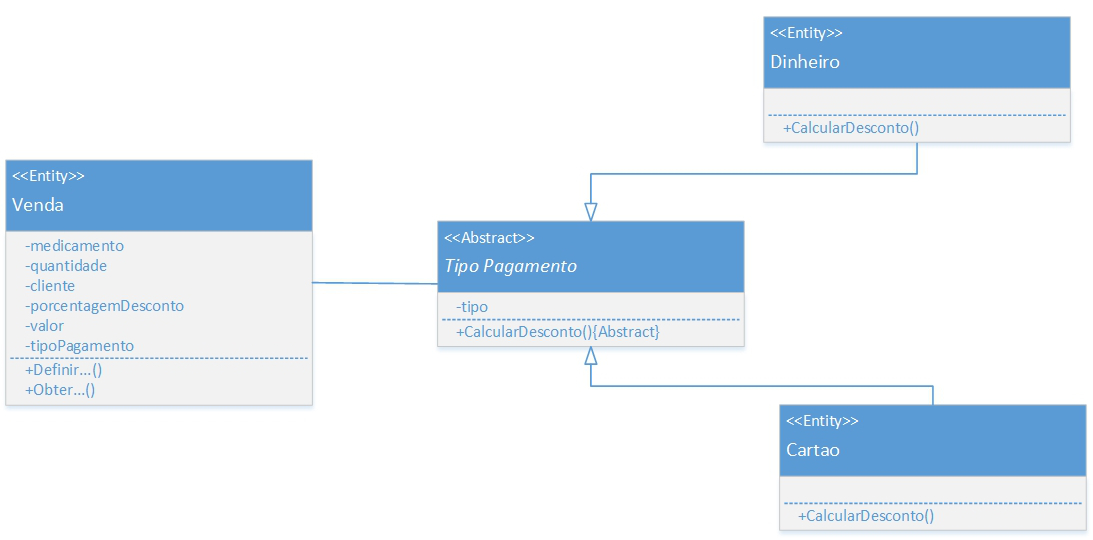
**18 -** Modele um diagrama de classes de projeto com padrão de projeto MVC a partir do diagrama de classes de análise, complementando as classes com atributos e métodos, se necessário. A notação de classe com seu devido <<estereótipo>> deve continuar sendo mantida.



**19 -** Modele duas relações de gen/espec e ative o princípio de polimorfismo universal de inclusão em cada uma delas. Justifique a razão de existência de cada gen/espec e das operações polimórficas.

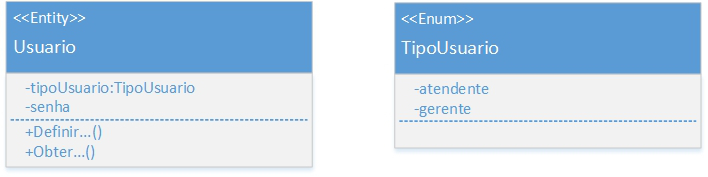


**Justificativa:** Os dois usuários do sistema, atendente e gerente, tem níveis de permissão diferentes. O método obterPermissao() serve para que, de acordo com o tipo de usuário, o sistema adeque as funções.



**Justificativa:** Dependendo do tipo de pagamento, o sistema deve realizar diferentes operações, no caso, o cálculo de desconto concedido, usando o mesmo método. A herança é cabível neste caso.

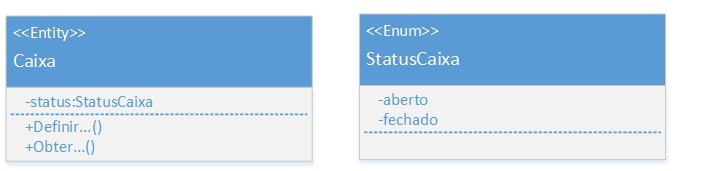
**20 -** Modele três classes enumeradas e utilize as mesmas como tipos de atributos. Justifique a existência de cada uma das classes enumeradas modeladas.



**Justificativa:** O sistema deve manter 2 tipos de usuários: atendente e gerente, onde cada um tem suas devidas permissões.

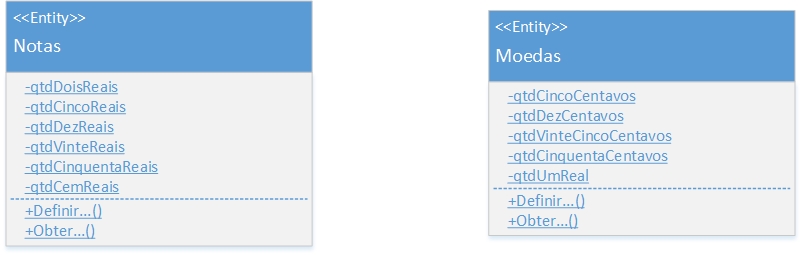


**Justificativa:** O sistema deve permitir apenas as formas de pagamento em dinheiro e em cartão, onde cada um possui uma porcentagem de desconto na hora da compra.



**Justificativa:** O caixa que será utilizado no sistema possui apenas 2 status, ou está aberto ou está fechado. O Enum permite listar essas opções facilmente.

**21 -** Modele seis membros estáticos, sendo três atributos e três métodos. Justifique a criação de existência de cada um dos membros estáticos modelados.



**Justificativa:** As classes notas e moedas funcionam como contadores do total do caixa. Não havendo sentido em instanciá-las, o sistema mantém as classes estáticas e funcionando como apenas um contador geral.