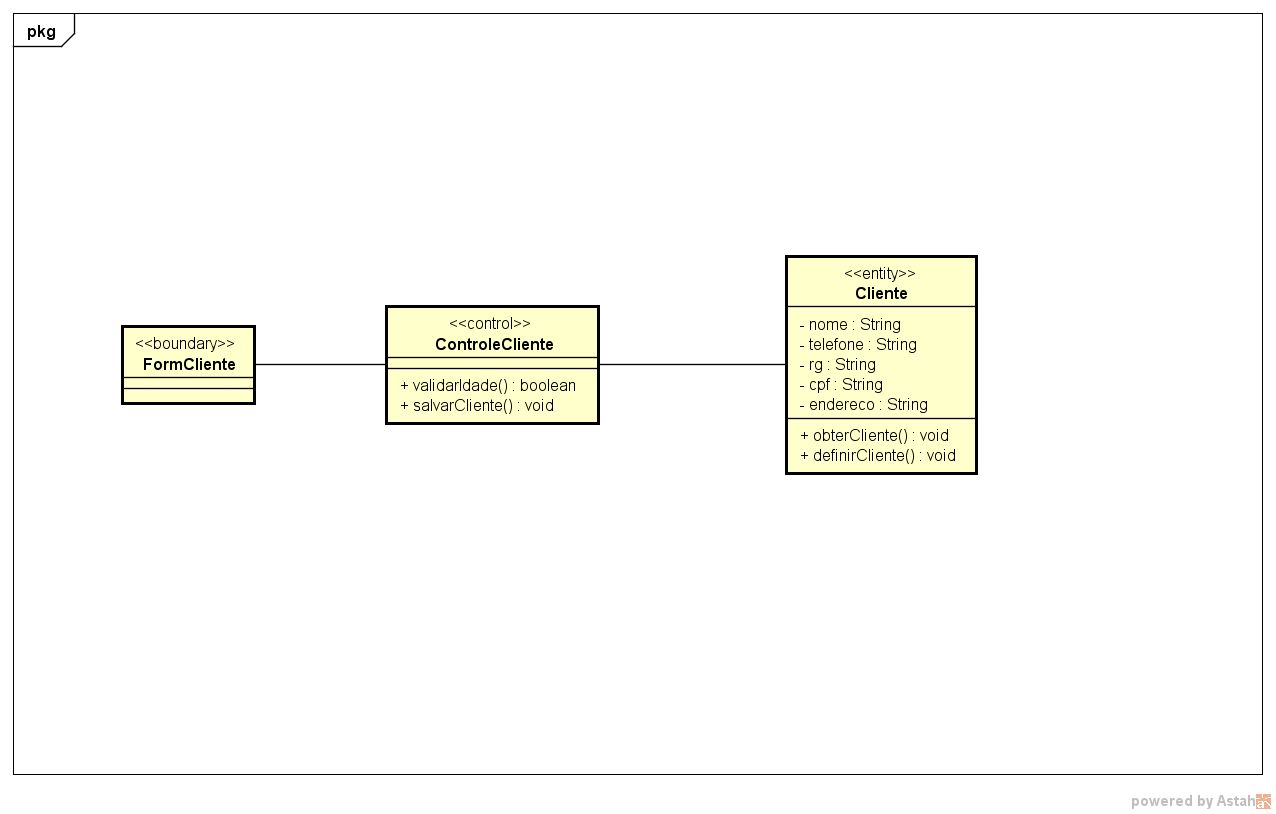
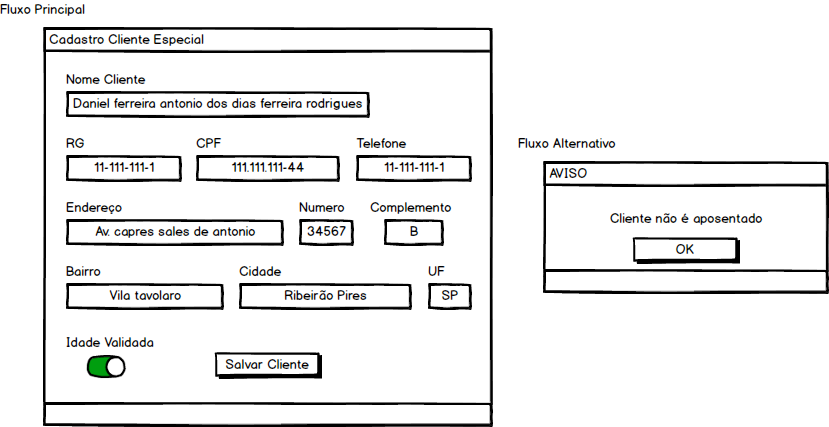
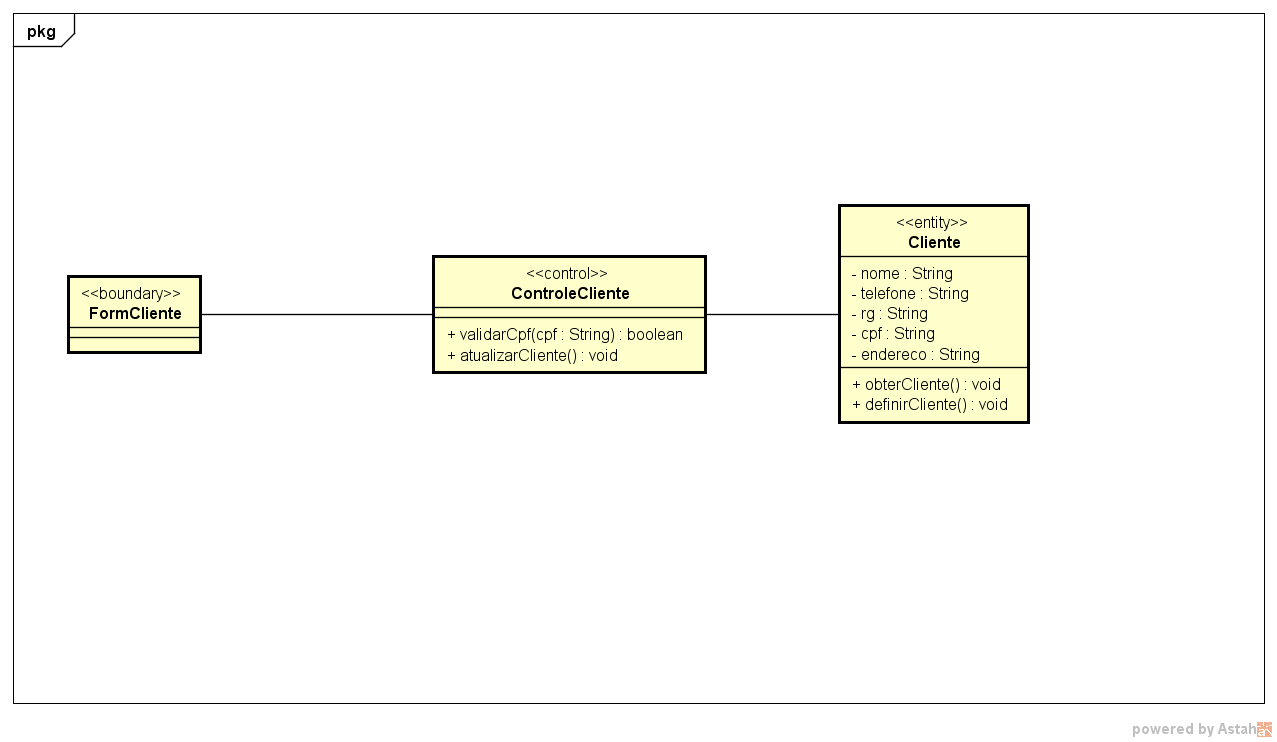
1. Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU01. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.



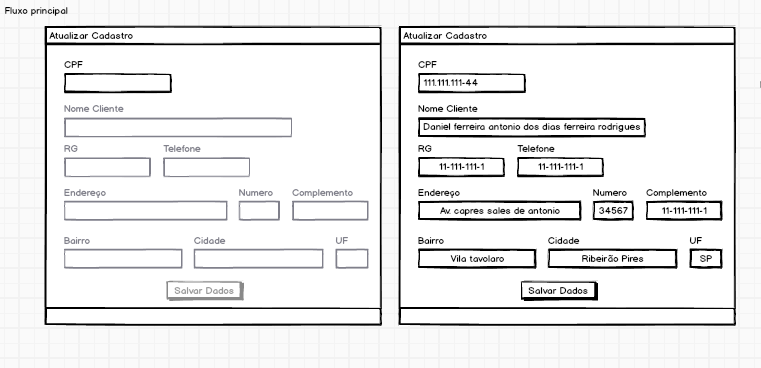
2- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU01.

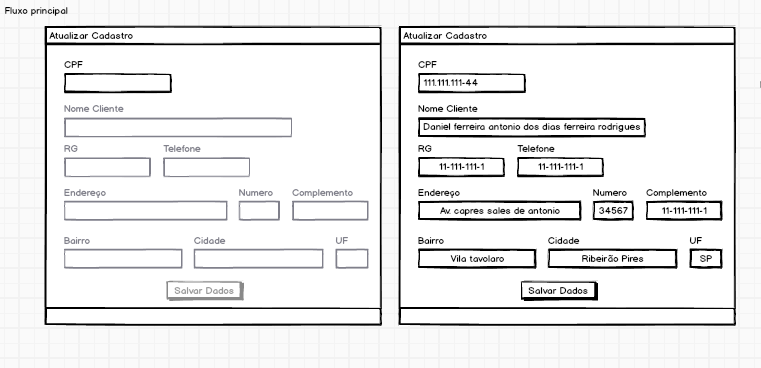


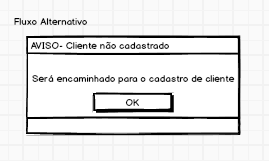
3- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU02. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.



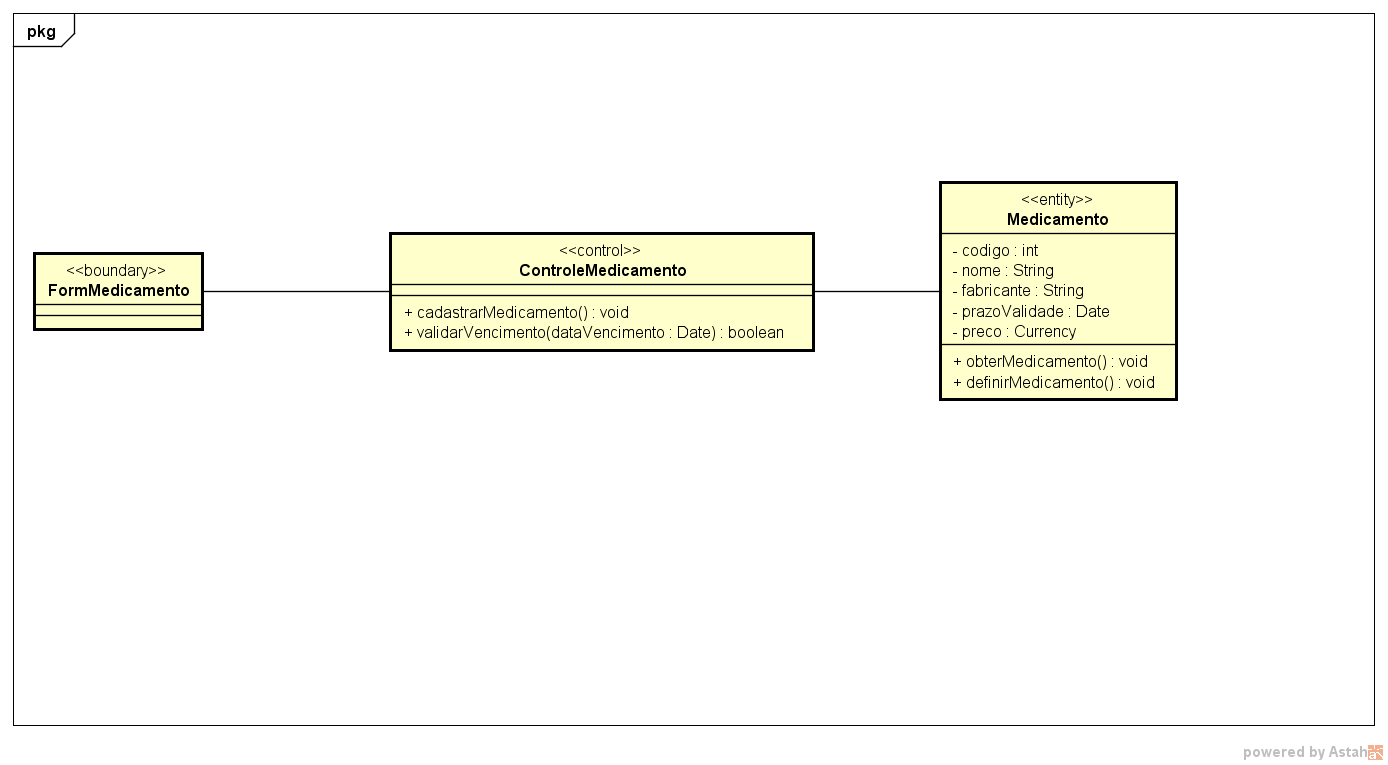
4- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU02.



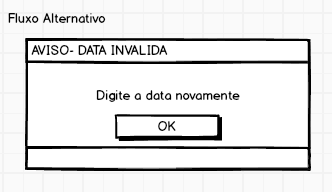
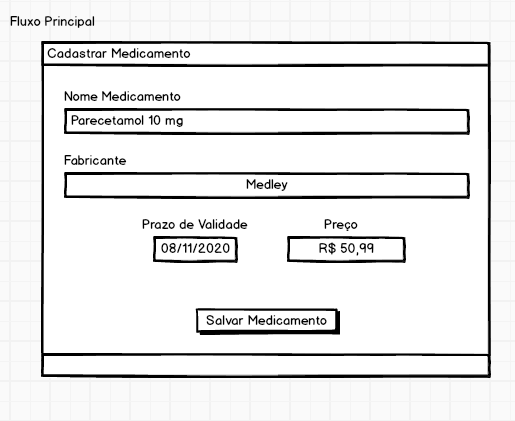




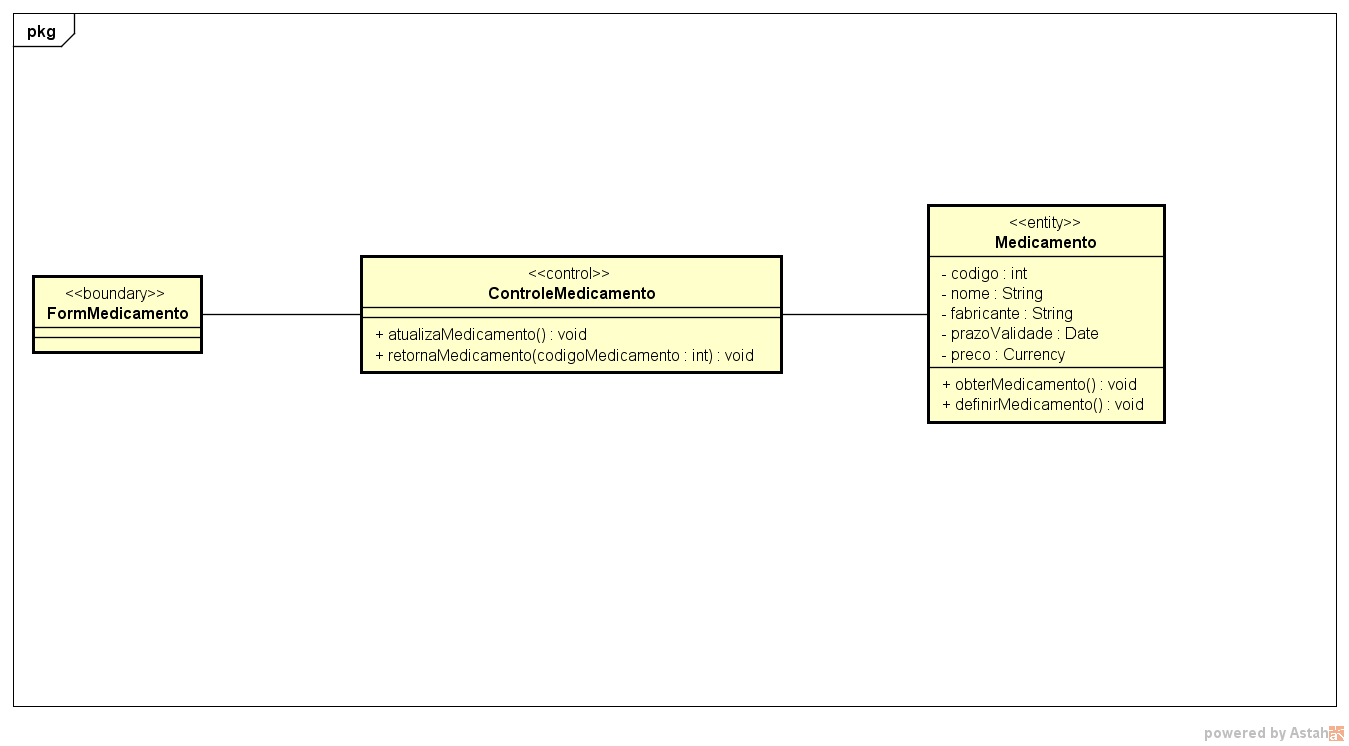
5- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU3. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.



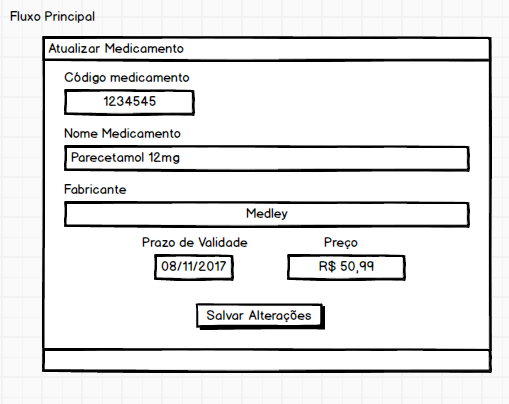
6- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU03.

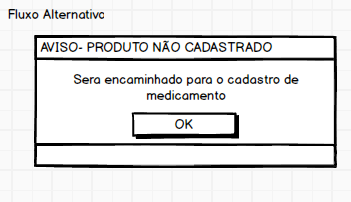


7- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU4. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

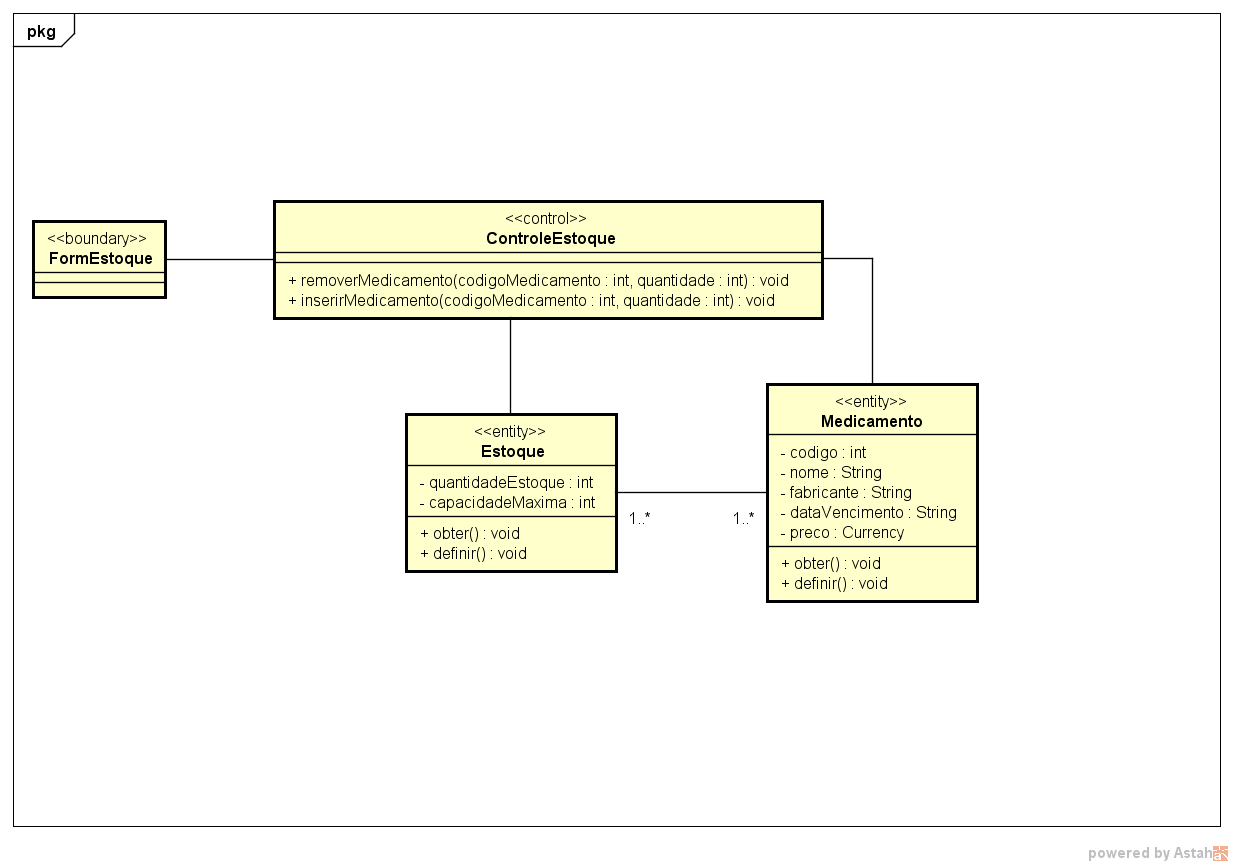


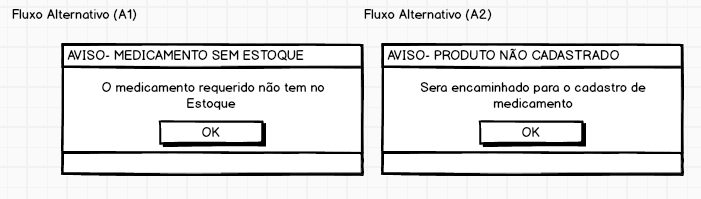
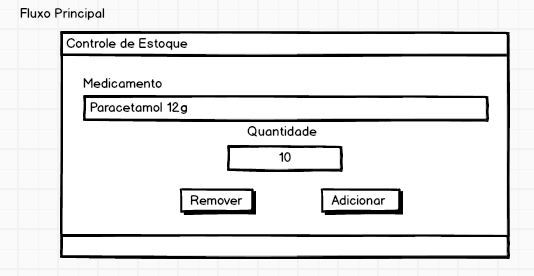
8- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU04.



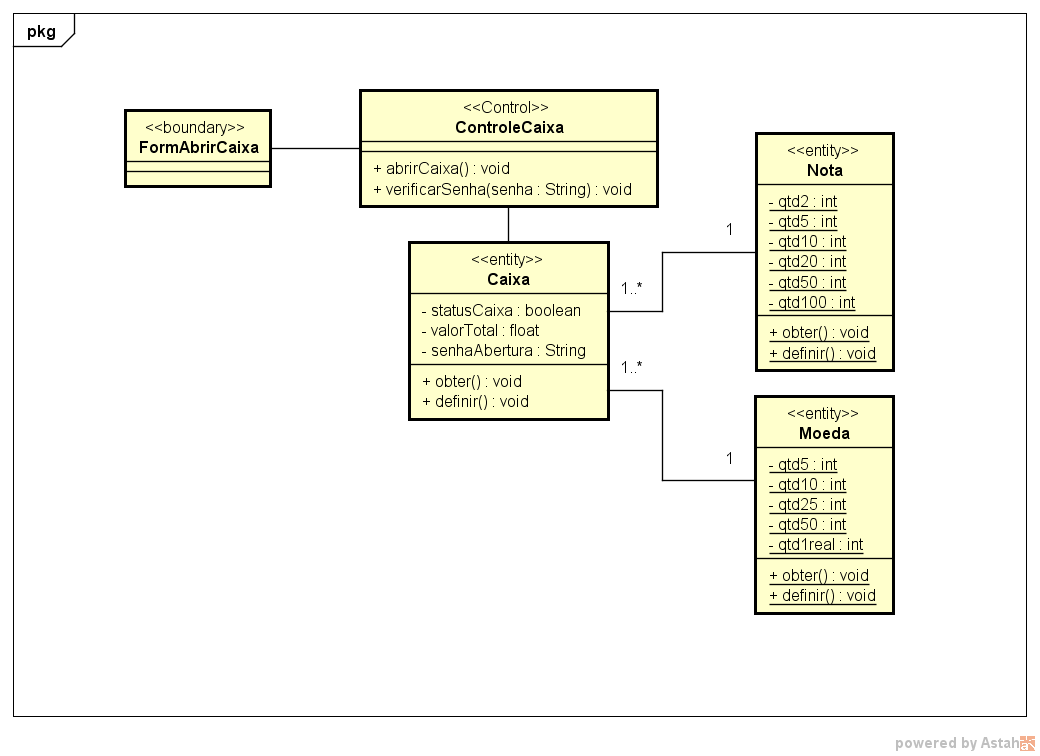


9- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU5. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

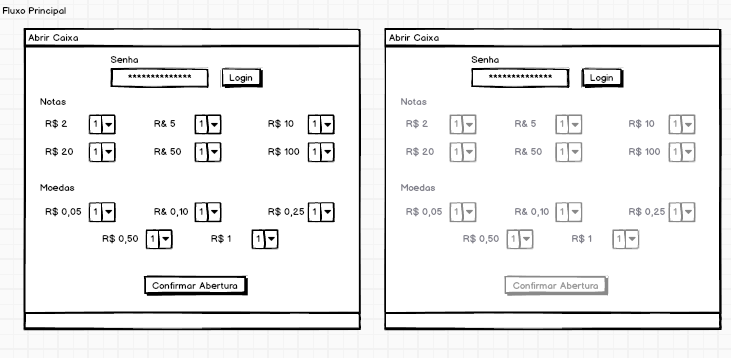


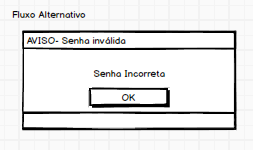
10- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU05. ]

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.

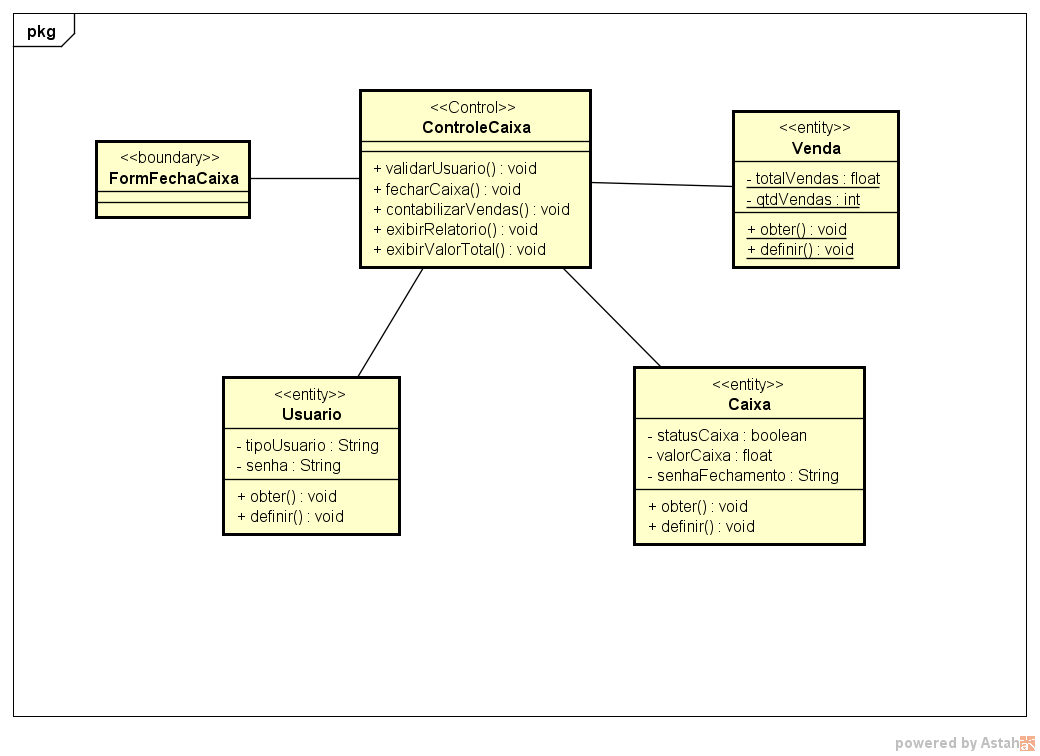


12- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU06.

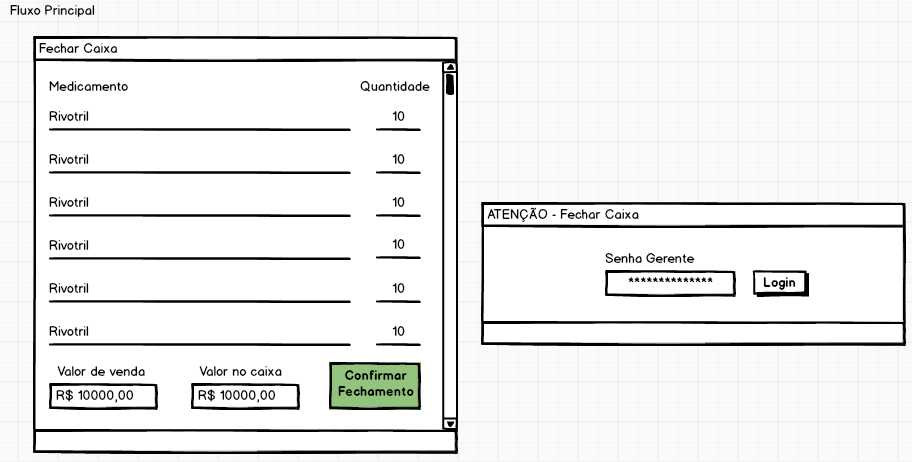


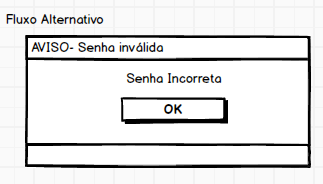


13- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU7. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

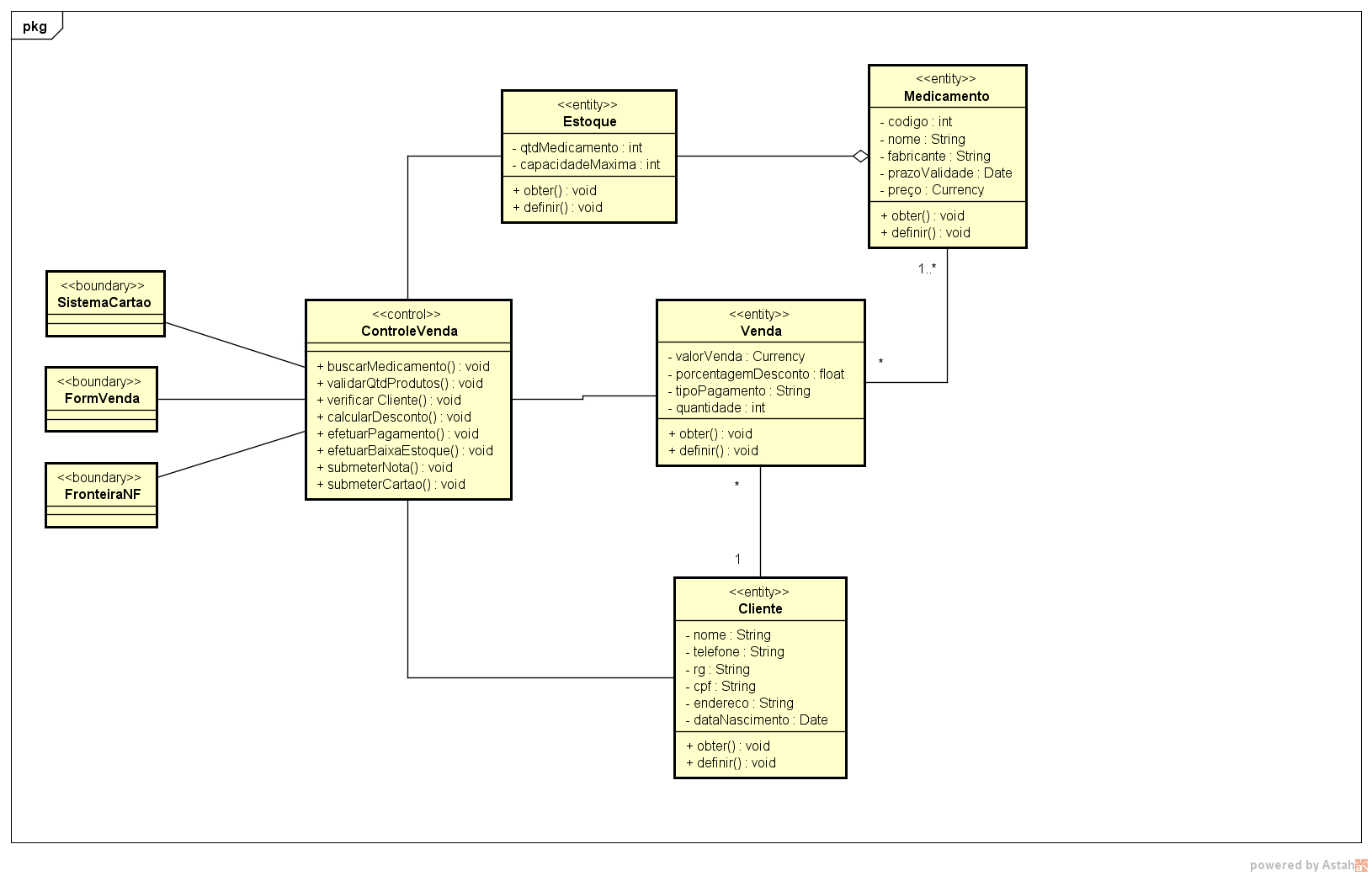


14- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU07.

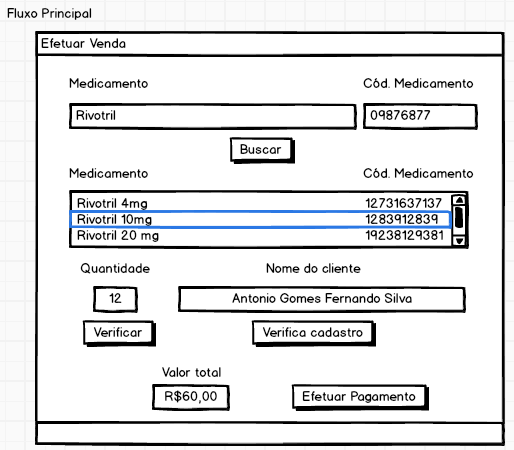


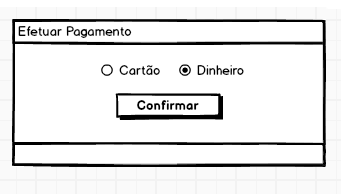


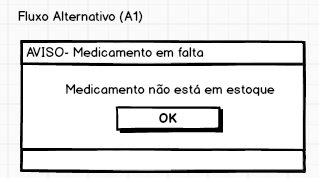
15- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU8. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.

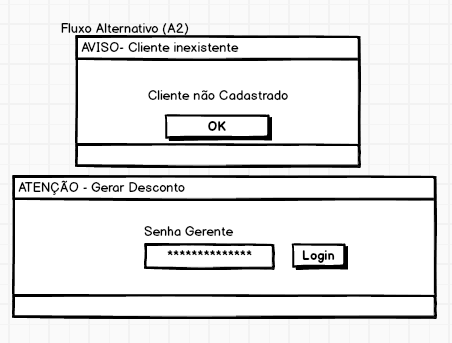


16- Faça o protótipo da interface de usuário para a classe <<boundary>> do CSU08.

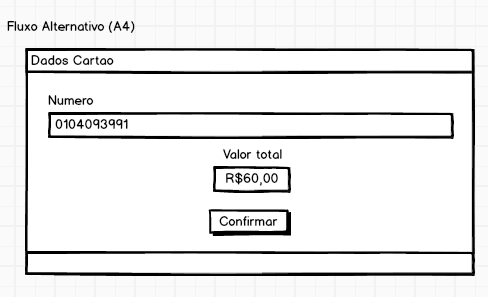




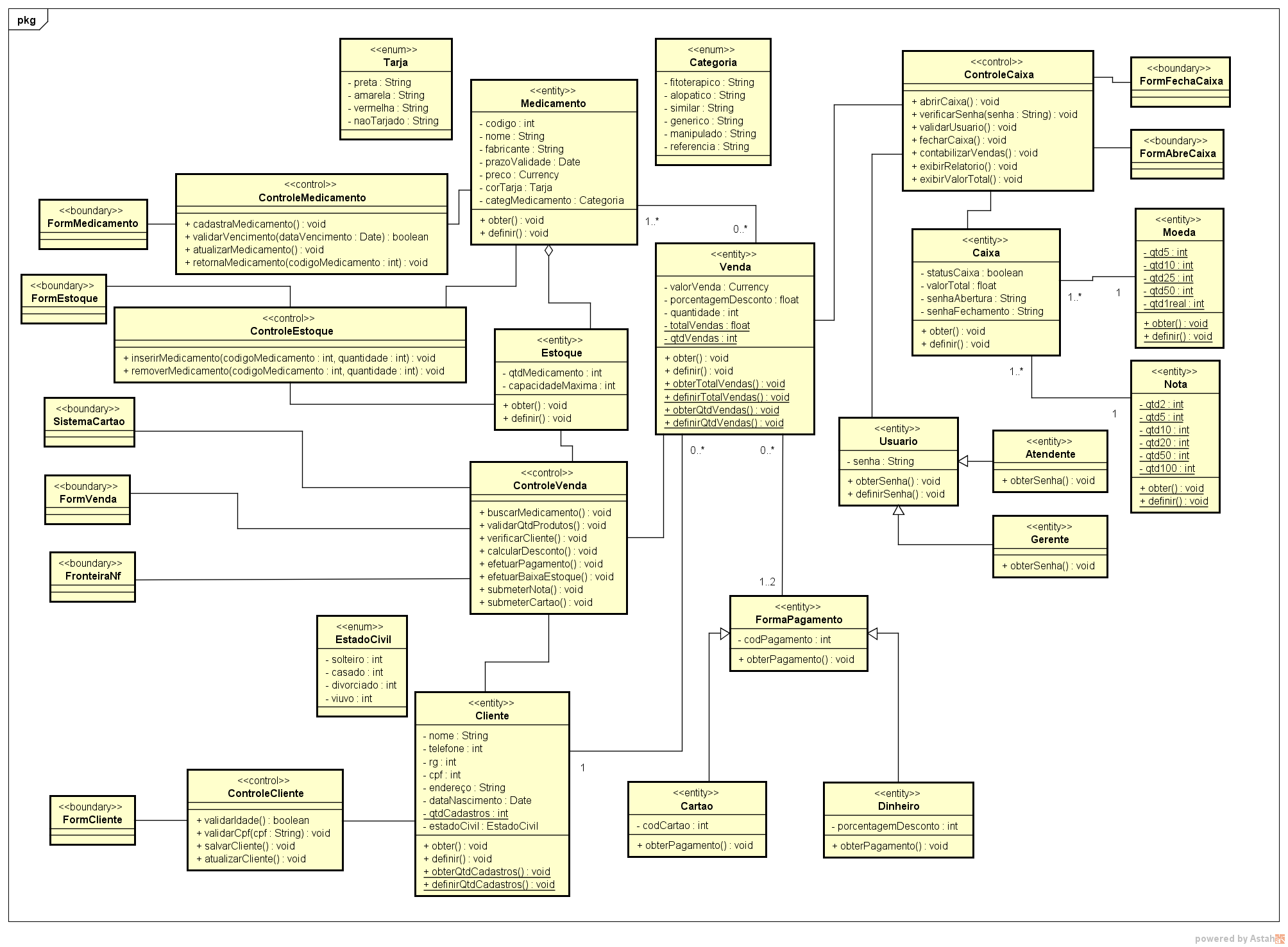




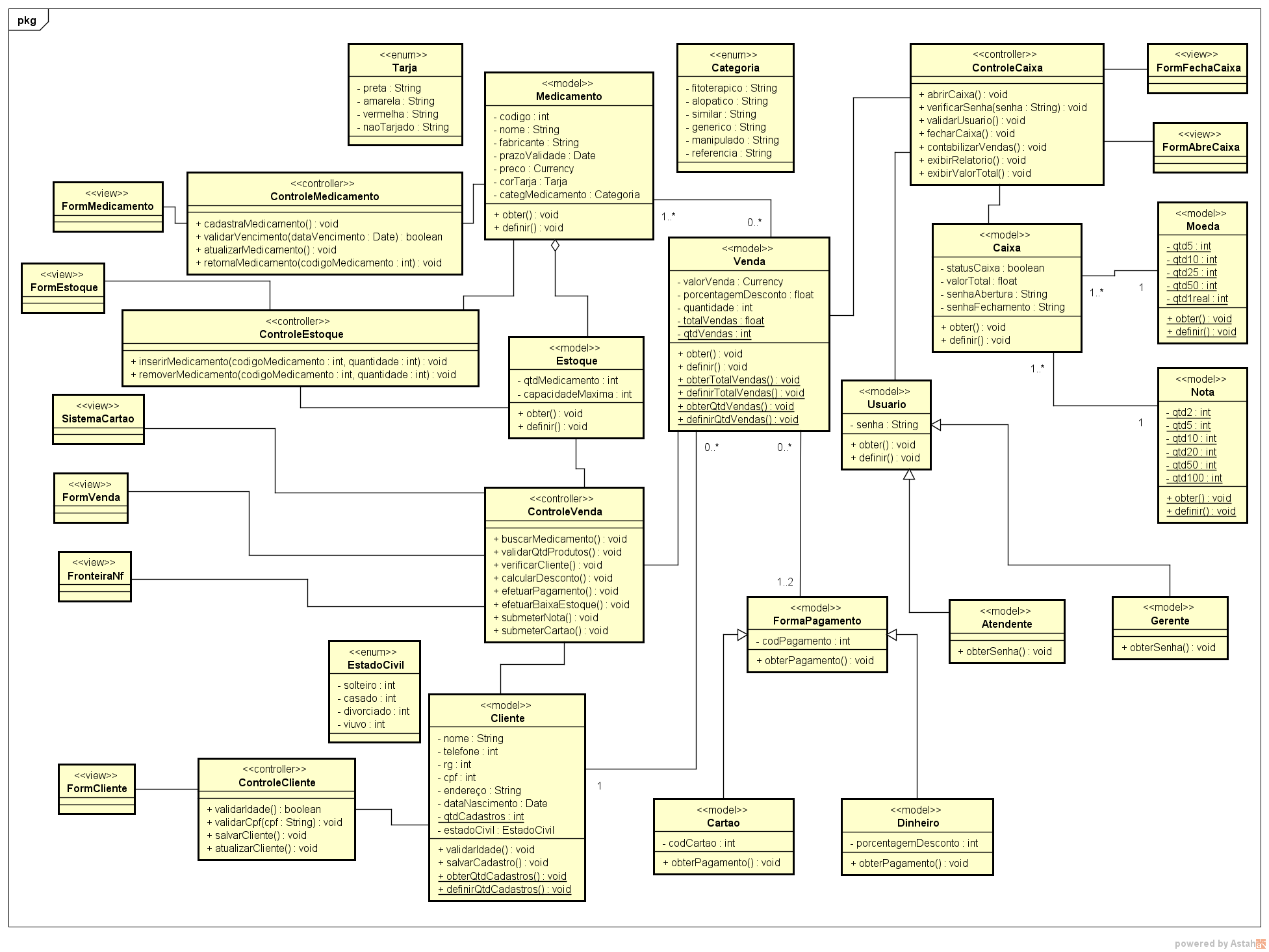




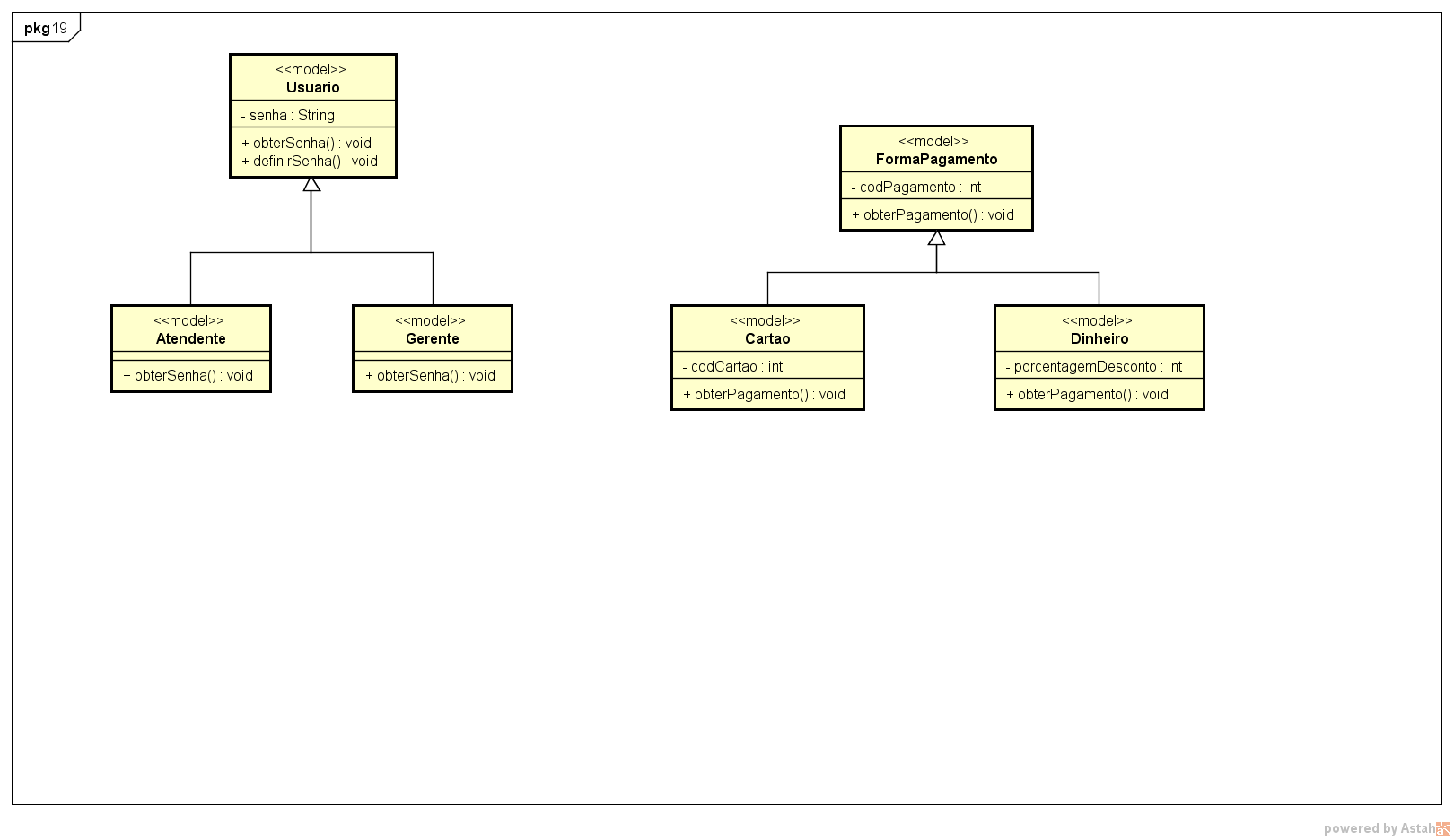
17- Modele um diagrama de classes de análise a partir das VCPs modeladas e mantenha a utilização da categorização BCE. Os devidos atributos e métodos devem continuar sendo exibidos.



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.



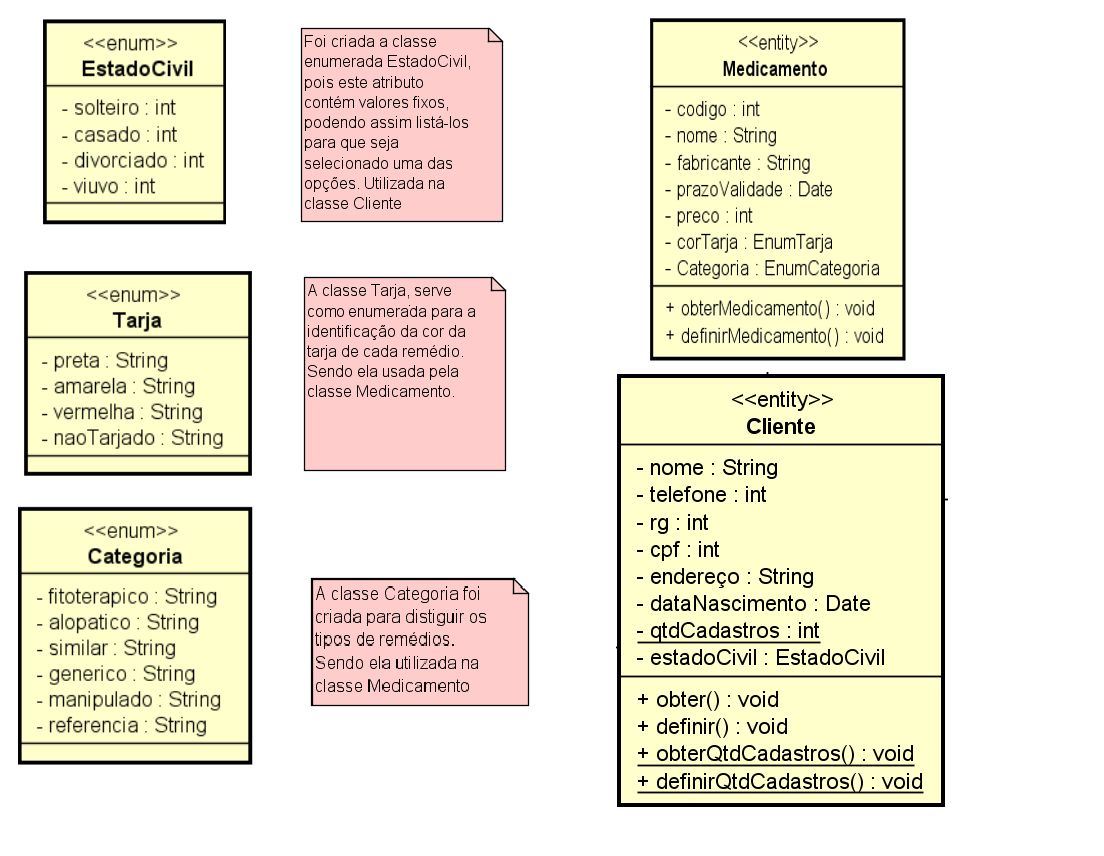
19- Modele duas relações de gen/espec e ative o princípio de polimorfismo universal de inclusão em cada uma delas. Justifique a razão de existência de cada gen/espec e das operações polimórficas.



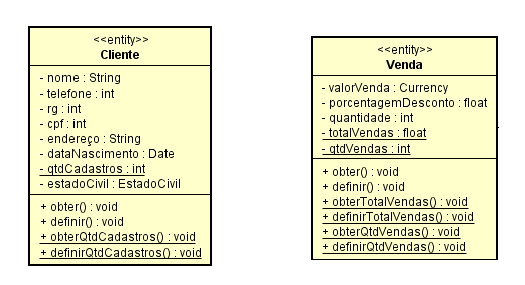
Para a gen/espec de usuário tanto o gerente quanto a atendente precisam de senhas para o sistema. Porém, as senhas podem ser obtidas de modo diferente. Então o método obterSenha() sofre polimorfismo.

A gen/espec de FormaPagamento é criada devido ao desconto que irá variar de acordo o modo de pagamento. Então o método obterPagamento() sofre polimorfismo.

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



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



Na classe Cliente, qtdCadastros mostrará a quantidade total de clientes cadastrados na farmacia, já o método definirQtdCadastros() atualizaria o valor de qtdCadastros e o método obterQtdCadastros() o exibiria.

Na classe Venda temos dois atributos estáticos, uma que armazenaria a soma do valor de vendas do dia, totalVendas. Bem como os metódos estáticos para exibir e modificar o valor, que são respectivamente obterTotalVendas() e definirTotalVendas(). E Tambem o outro atributo qtdVendas que armazena a quantidade de vendas feitas durante o dia. E os dois atributos para exibição e atualização dos dados, obterQtdVendas() e definirQtdVendas().