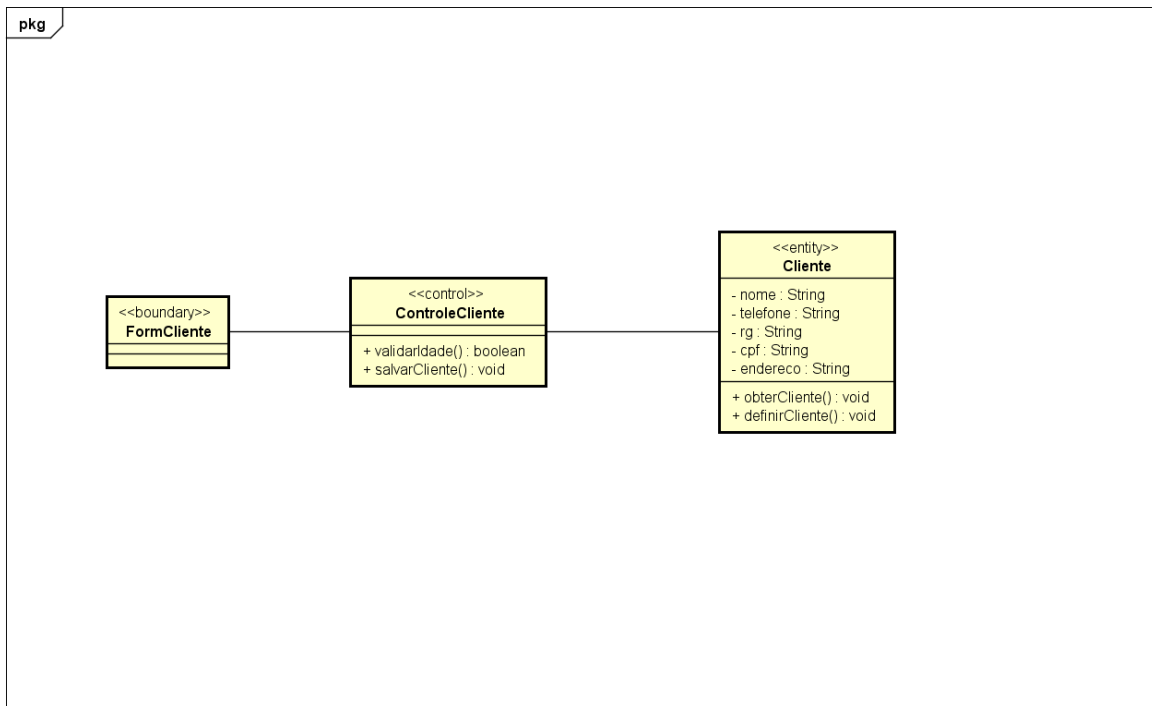


- 1- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU01. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.



powered by Astah

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

Fluxo Principal

Cadastro Cliente Especial

Nome Cliente

RG

CPF

Telefone

Endereço

Numero

Complemento

Bairro

Cidade

UF

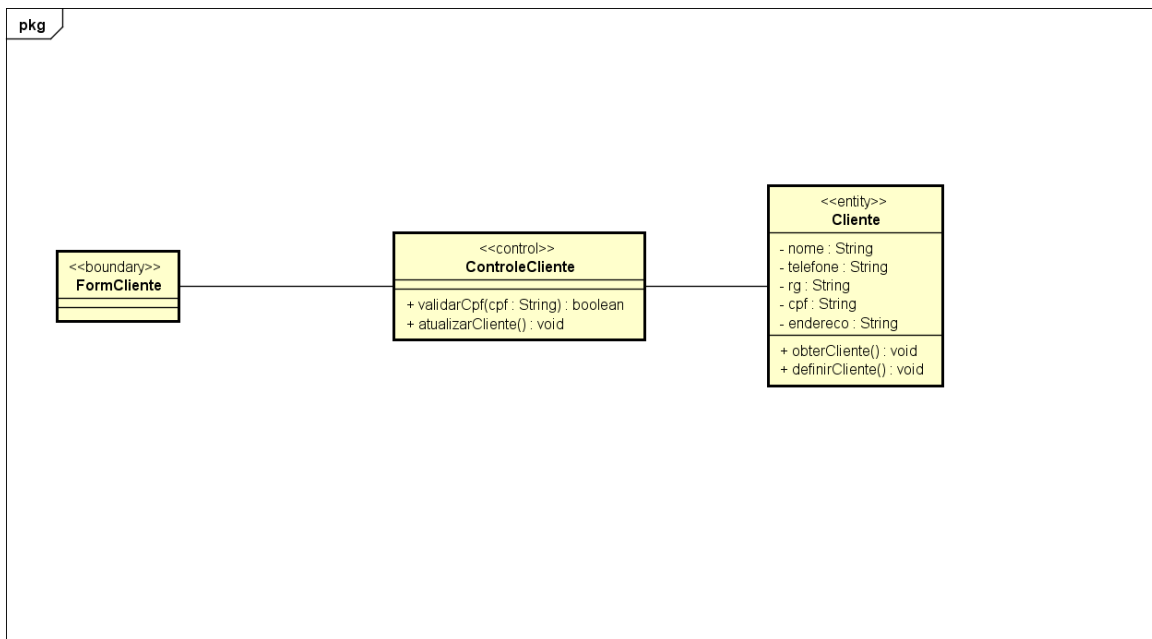
Idade Validada
☒

Fluxo Alternativo

AVISO

Cliente não é aposentado

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.



powered by Astah

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

Fluxo principal

Atualizar Cadastro

CPF

Nome Cliente

RG Telefone

Endereço Número Complemento

Bairro Cidade UF

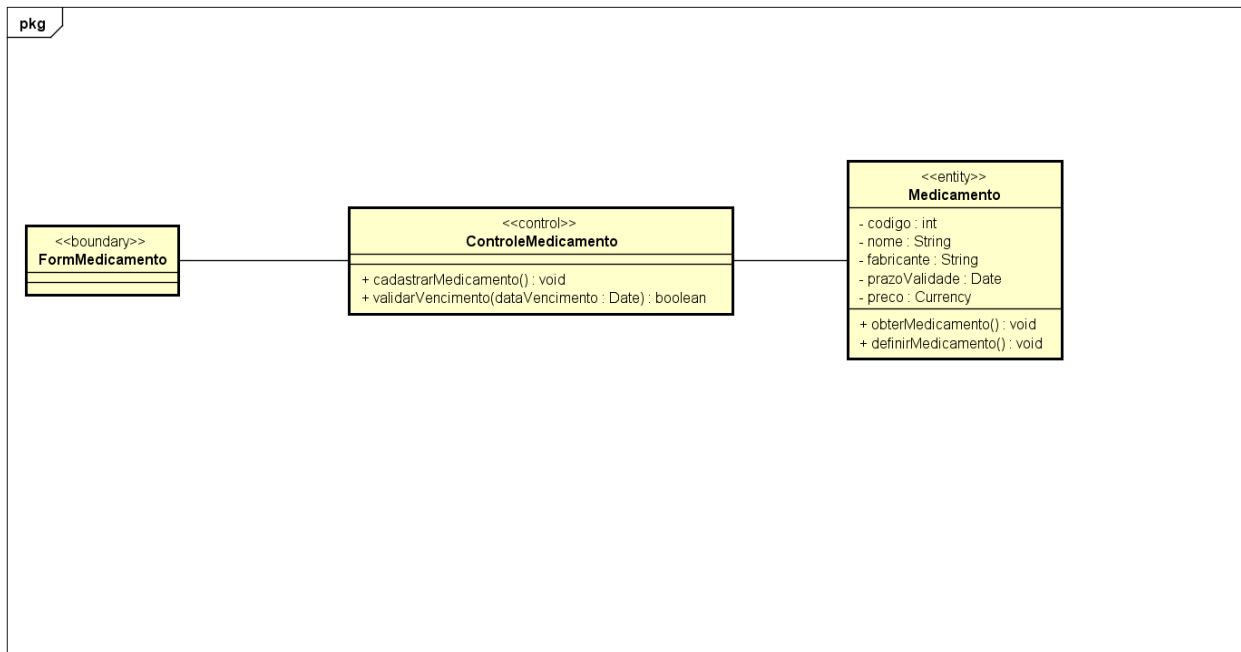
Salvar Dados

Atualizar Cadastro		
CPF		
<input type="text" value="111.111.111-44"/>		
Nome Cliente		
<input type="text" value="Daniel ferreira antonio dos dias ferreira rodrigues"/>		
RG	Telefone	
<input type="text" value="11-111-111-1"/>	<input type="text" value="11-111-111-1"/>	
Endereço	Numero	Complemento
<input type="text" value="Av. capres sales de antonio"/>	<input type="text" value="34567"/>	<input type="text" value="11-111-111-1"/>
Bairro	Cidade	UF
<input type="text" value="Vila tavolaro"/>	<input type="text" value="Ribeirão Pires"/>	<input type="text" value="SP"/>
<input type="button" value="Salvar Dados"/>		

Fluxo Alternativo

AVISO- Cliente não cadastrado
Será encaminhado para o cadastro de cliente
<input type="button" value="OK"/>

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.

Fluxo Principal

Cadastrar Medicamento

Nome Medicamento

Parecetamol 10 mg

Fabricante

Medley

Prazo de Validade

08/11/2020

Preço

R\$ 50,99

Salvar Medicamento

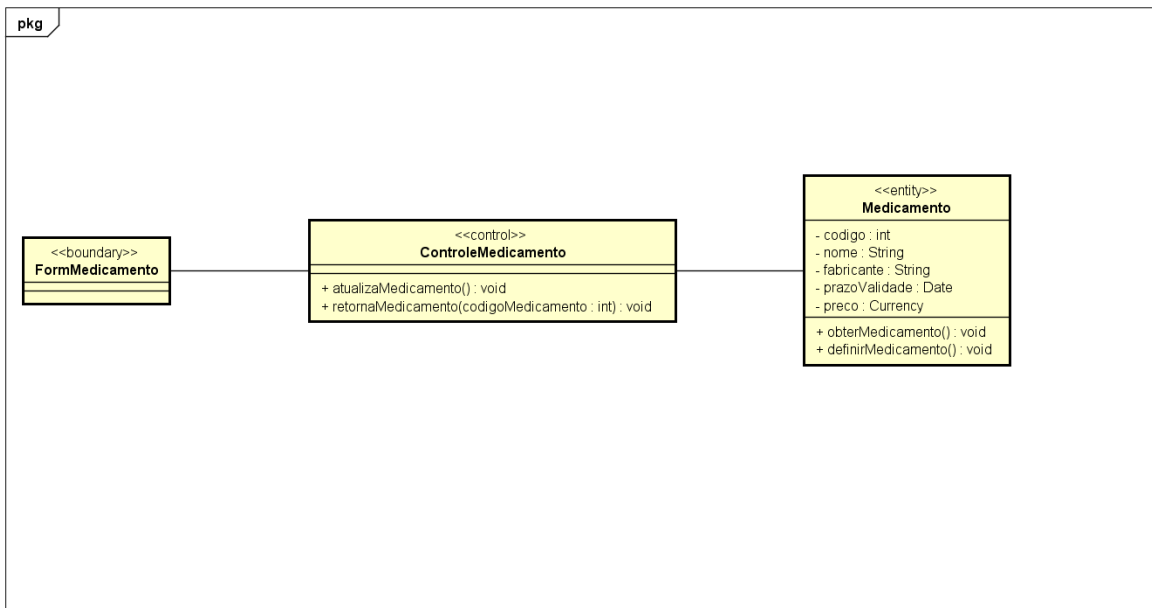
Fluxo Alternativo

AVISO- DATA INVALIDA

Digite a data novamente

OK

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.



powered by Astah

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

Fluxo Principal

Atualizar Medicamento

Código medicamento

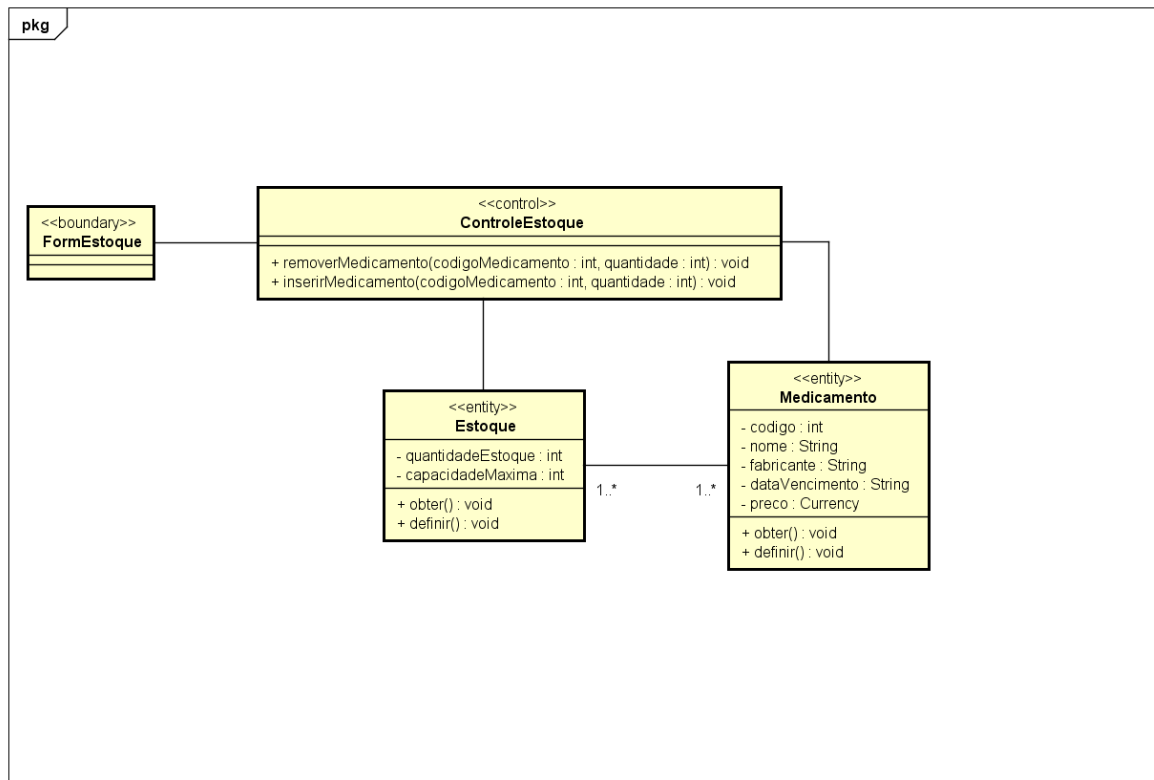
Nome Medicamento

Fabricante

Fluxo Alternativo

AVISO- PRODUTO NÃO CADASTRADO
Sera encaminhado para o cadastro de medicamento
OK

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.

Fluxo Principal

Controle de Estoque	
Medicamento	
<input type="text" value="Paracetamol 12g"/>	
Quantidade	
<input type="text" value="10"/>	
<input type="button" value="Remover"/>	<input type="button" value="Adicionar"/>

Fluxo Alternativo (A1)

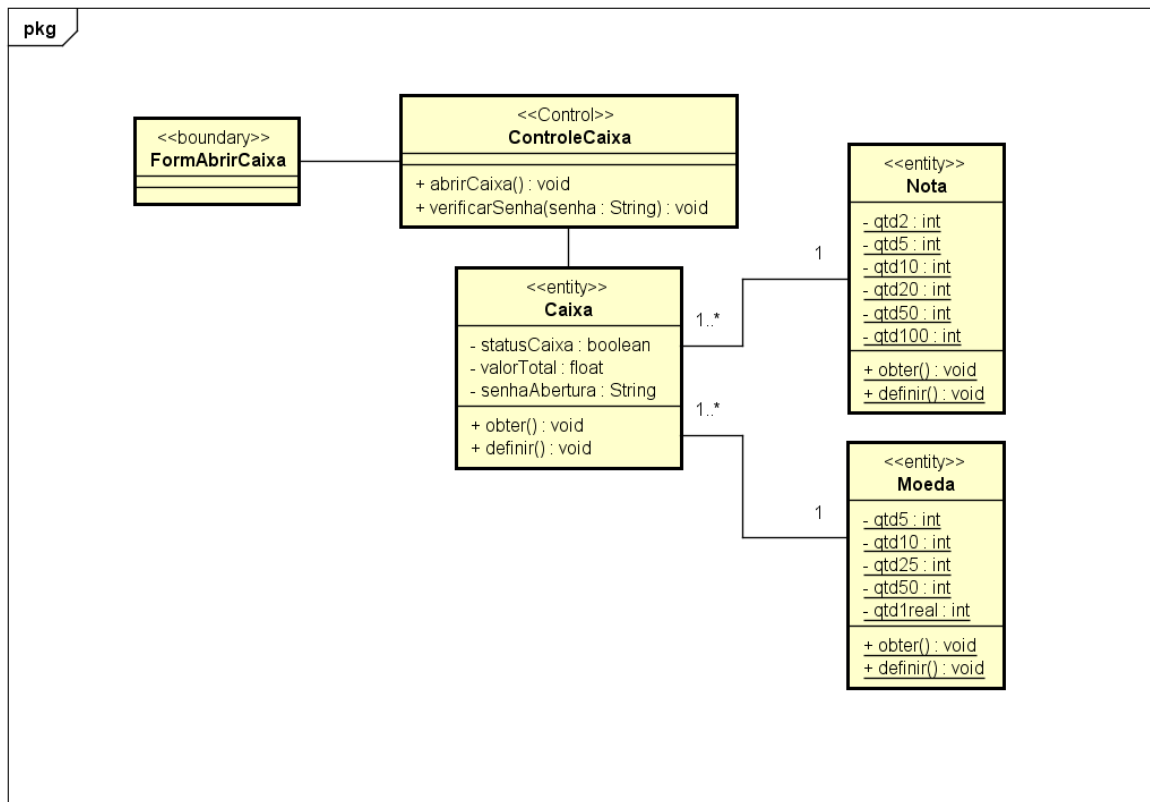
AVISO- MEDICAMENTO SEM ESTOQUE
O medicamento requerido não tem no Estoque
<input type="button" value="OK"/>

Fluxo Alternativo (A2)

AVISO- PRODUTO NÃO CADASTRADO
Sera encaminhado para o cadastro de medicamento
<input type="button" value="OK"/>

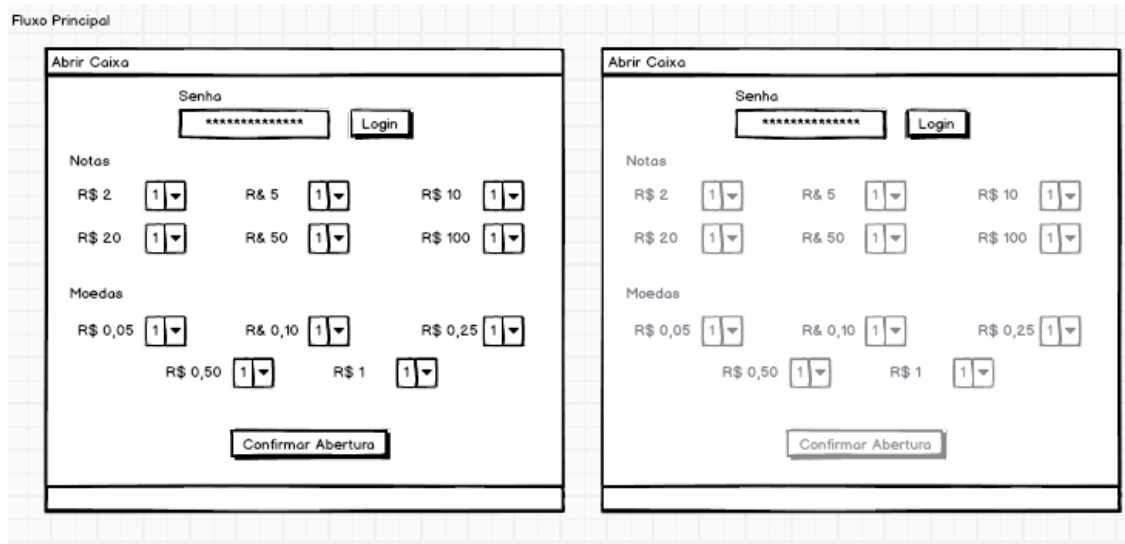
]

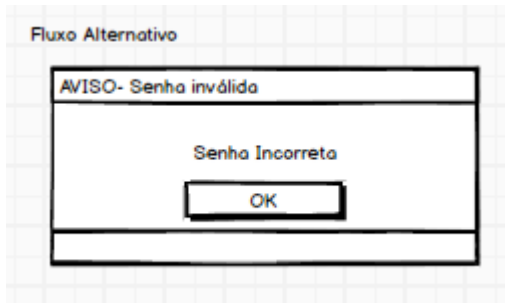
11- Modele uma VCP para o caso de uso utilizando a categorização BCE para o CSU6. A classe de controle deve apresentar um método no mínimo e as classes de entidade devem apresentar seus devidos atributos e um método no mínimo.



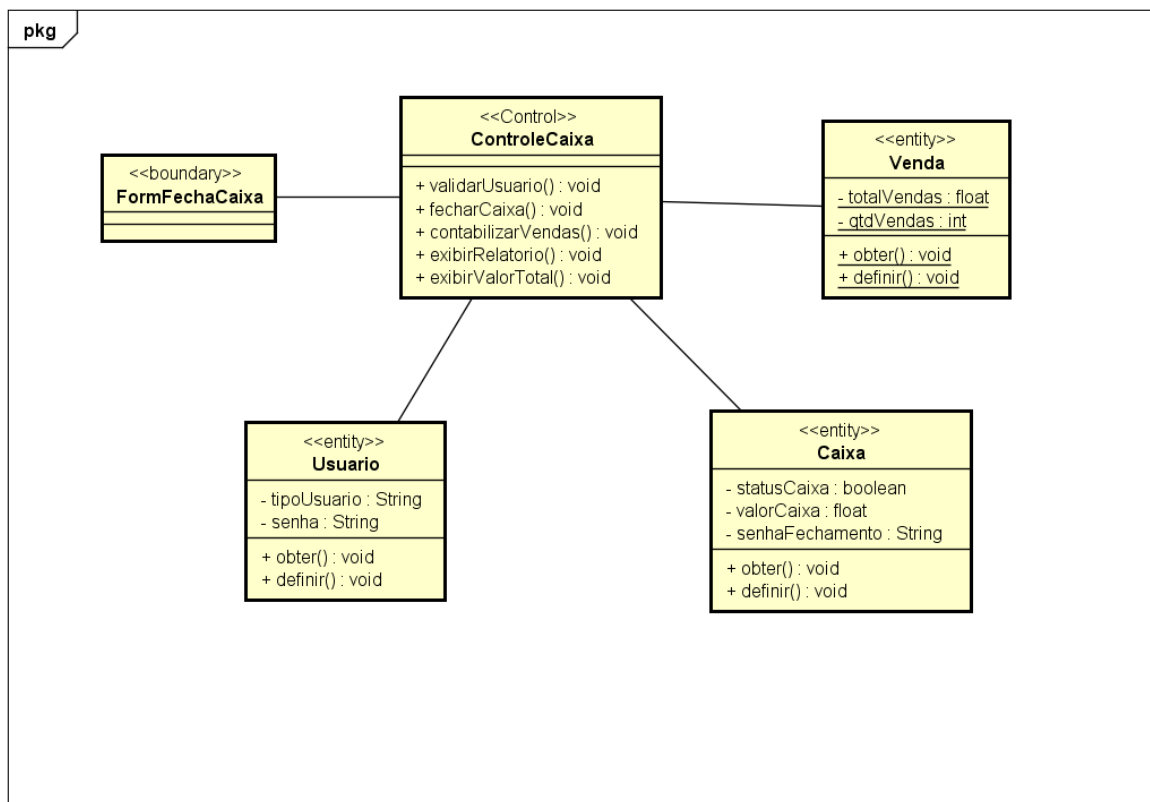
powered by Astah

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.



powered by Astah

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

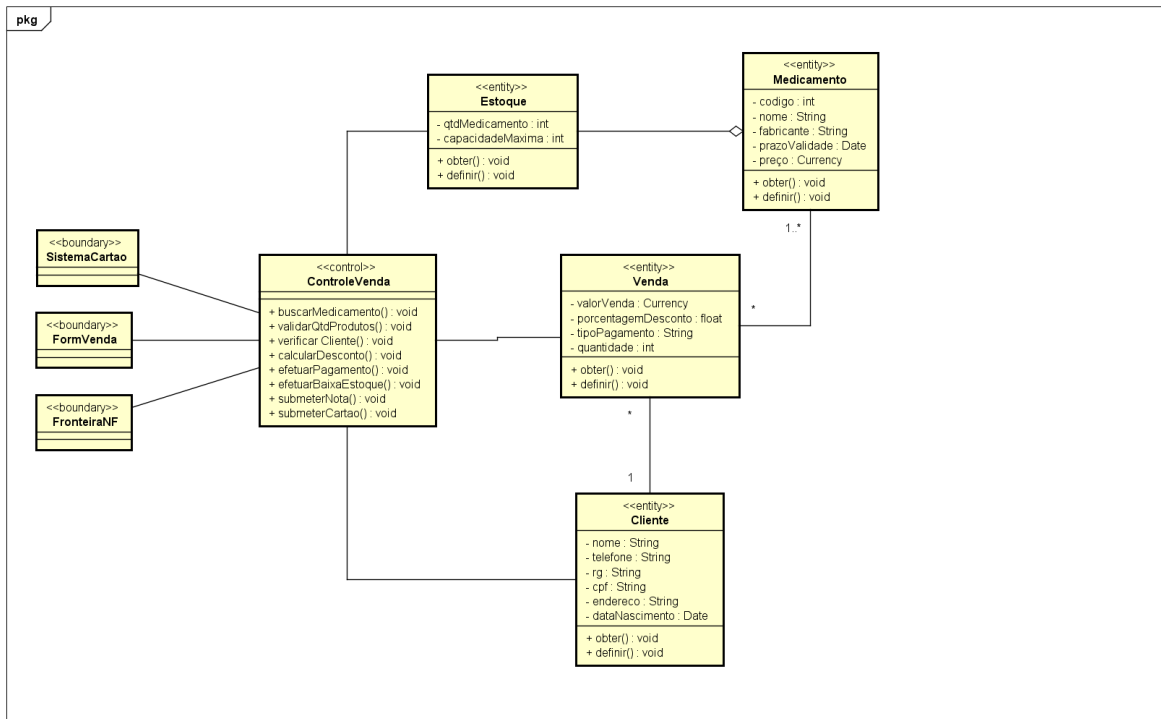
Fluxo Principal

Fechar Caixa	
Medicamento	Quantidade
Rivotril	10
Rivotril	10
Rivotril	10
Rivotril	10
Rivotril	10
Rivotril	10
Valor de venda	Valor no caixa
R\$ 10000,00	R\$ 10000,00
Confirmar Fechamento	

Fluxo Alternativo

AVISO- Senha inválida
Senha Incorreta
OK

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.

Fluxo Principal

Efetuar Venda

Medicamento

Cód. Medicamento

Rivotril

09876877

Buscar

Medicamento

Cód. Medicamento

Rivotril 4mg

12731637137

Rivotril 10mg

1283912839

Rivotril 20 mg

19238129381

Quantidade

Nome do cliente

12

Antonio Gomes Fernando Silva

Verificar

Verifica cadastro

Valor total

R\$60,00

Efetuar Pagamento

Efetuar Pagamento

☐ Cartão

☒ Dinheiro

Confirmar

Fluxo Alternativo (A1)

AVISO- Medicamento em falta

Medicamento não está em estoque

OK

Fluxo Alternativo (A2)

AVISO- Cliente inexistente
Cliente não Cadastrado
<input type="button" value="OK"/>

ATENÇÃO - Gerar Desconto

ATENÇÃO - Gerar Desconto
Senha Gerente
<input type="password" value="*****"/>
<input type="button" value="Login"/>

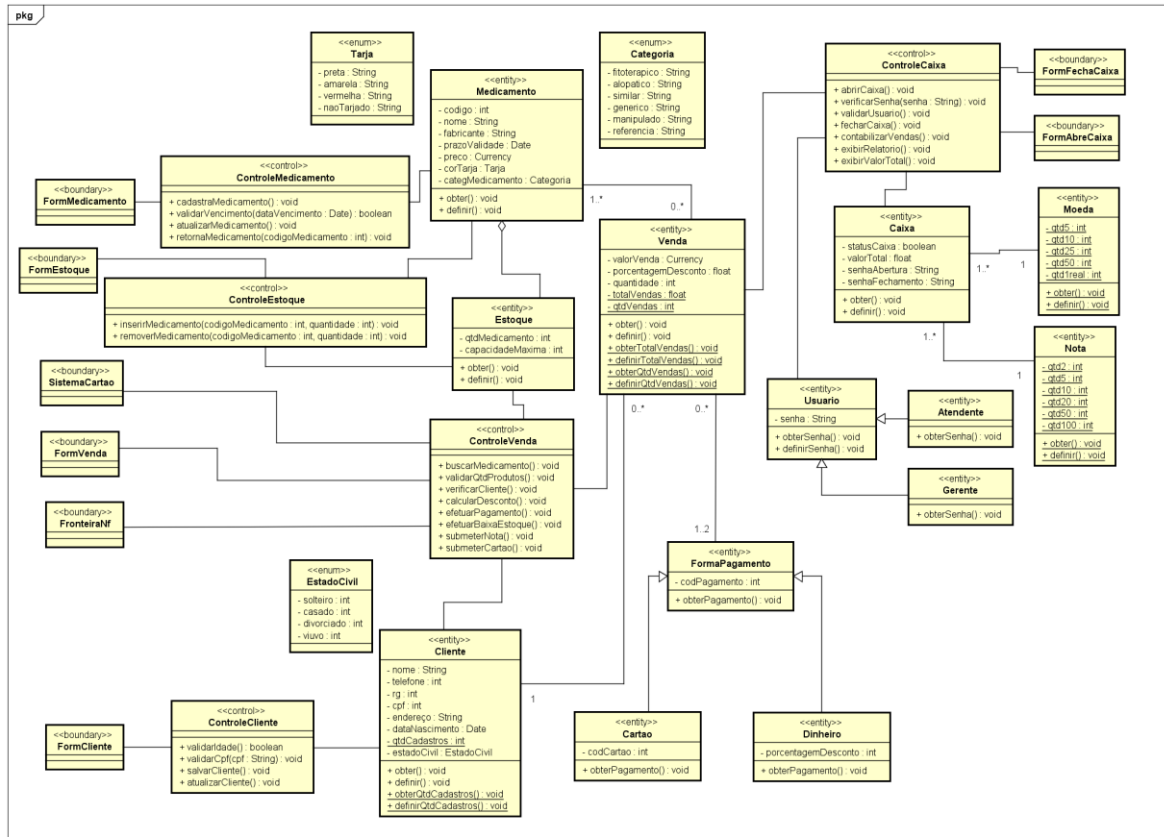
Fluxo Alternativo (A3)

Dinheiro
Valor total
<input type="text" value="R\$60,00"/>
Valor desconto
<input type="text" value="R\$3,00"/>
<input type="button" value="Confirmar"/>

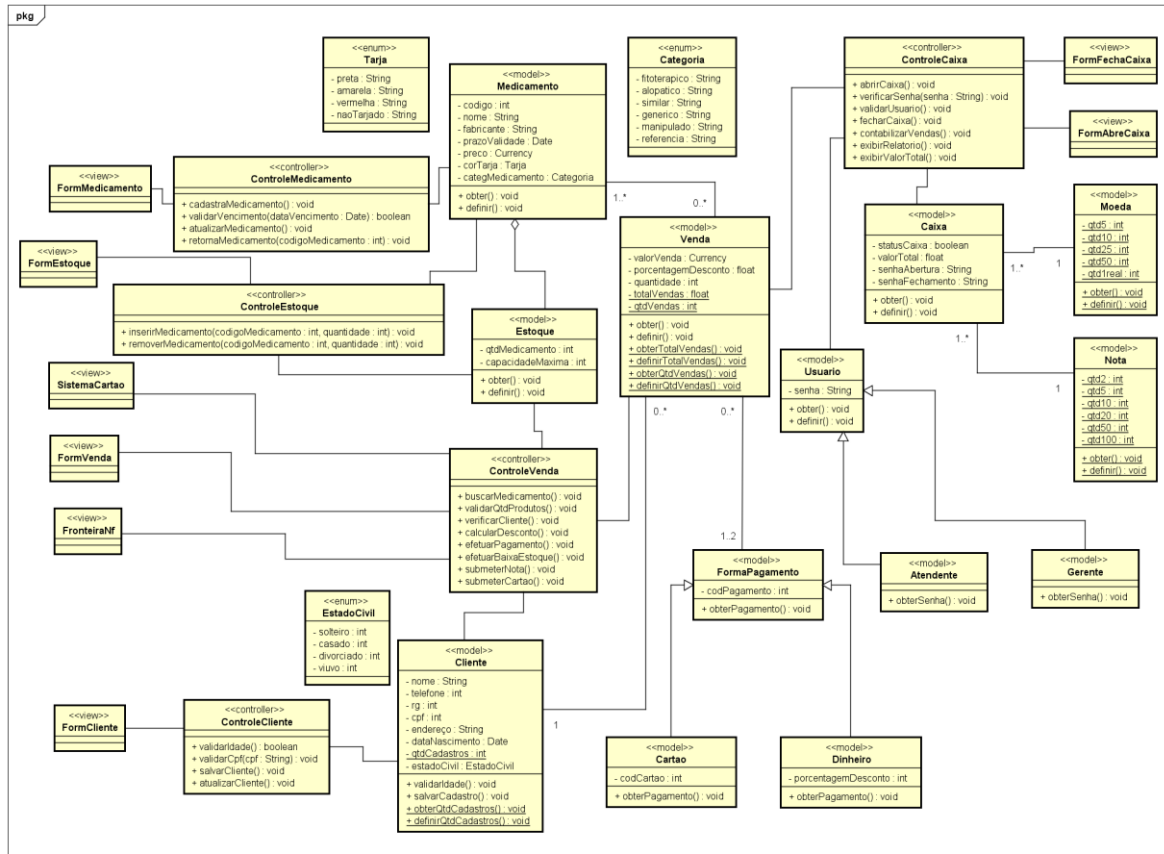
Fluxo Alternativo (A4)

Dados Cartao
Numero
<input type="text" value="0104093991"/>
Valor total
<input type="text" value="R\$60,00"/>
<input type="button" value="Confirmar"/>

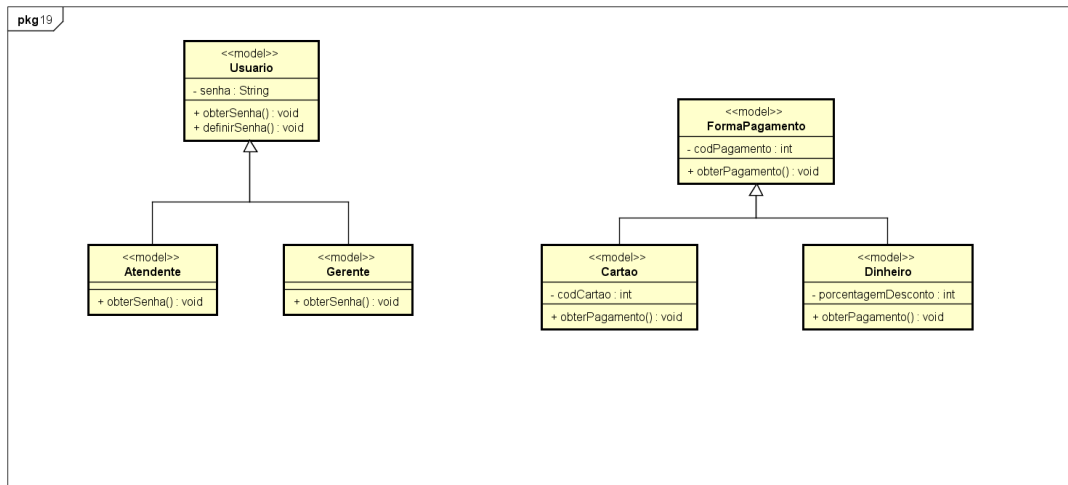
17- Modele um diagrama de classes de análise a partir das VCPs modeladas e mantenha a utilização da categorização BCE. Os devidos atributos e métodos devem continuar sendo exibidos.



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.



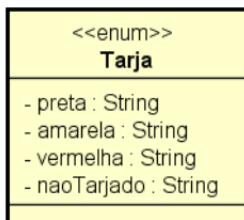
Para a gen/espec de usuário tanto o gerente quanto a atendente precisam de senhas para o sistema. Porém, as senhas podem ser obtidas de modo diferente. Então o método obterSenha() sofre polimorfismo.

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.

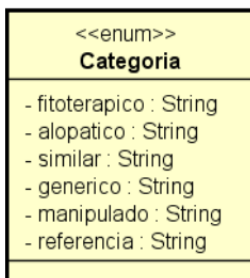
20- Modele três classes enumeradas e utilize as mesmas como tipos de atributos. Justifique a existência de cada uma das classes enumeradas modeladas.



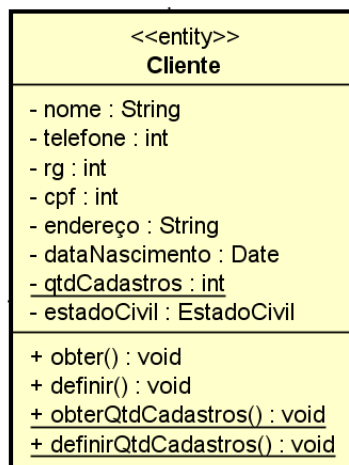
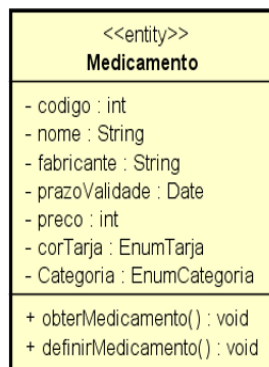
Foi criada a classe enumerada EstadoCivil, pois este atributo contém valores fixos, podendo assim listá-los para que seja selecionado uma das opções. Utilizada na classe Cliente



A classe Tarja, serve como enumerada para a identificação da cor da tarja de cada remédio. Sendo ela usada pela classe Medicamento.



A classe Categoria foi criada para distinguir os tipos de remédios. Sendo ela utilizada na classe Medicamento



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

<<entity>> Cliente
- nome : String - telefone : int - rg : int - cpf : int - endereço : String - dataNascimento : Date - <u>qtdCadastros : int</u> - estadoCivil : EstadoCivil
+ obter() : void + definir() : void + <u>obterQtdCadastros() : void</u> + <u>definirQtdCadastros() : void</u>

<<entity>> Venda
- valorVenda : Currency - porcentagemDesconto : float - quantidade : int - <u>totalVendas : float</u> - <u>qtdVendas : int</u>
+ obter() : void + definir() : void + <u>obterTotalVendas() : void</u> + <u>definirTotalVendas() : void</u> + <u>obterQtdVendas() : void</u> + <u>definirQtdVendas() : void</u>

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 métodos estáticos para exibir e modificar o valor, que são respectivamente obterTotalVendas() e definirTotalVendas(). E Também 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().