UAST Unidade Acadêmica de Serra Talhada - PE
UFRPE
Bacharelado em Sistemas de Informação

MPOO

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)
Profº: Richarlyson D'Emery **Data:** 17 / 10 / 2022

Aluno: _____

Nota: _____

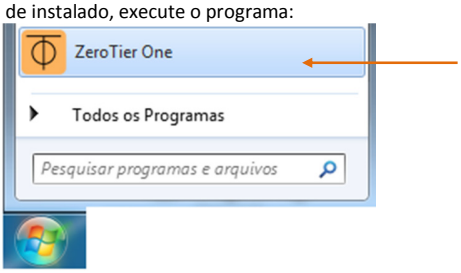
VERIFICAÇÃO DE APRENDIZAGEM FINAL

CONFIGURAÇÃO DO AMBIENTE DE ACOMPANHAMENTO DA V.A.

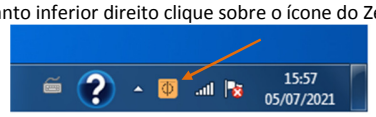
Para acompanhamento da realização da 3ª V.A., deverão ser seguidas as seguintes instruções:

1) Instalação e uso de **Zerotier:**

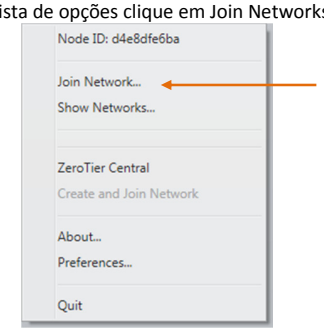
- Baixe o software Zerotier em: https://www.zerotier.com/download/
- Depois de instalado, execute o programa:



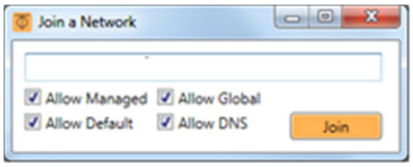
- No canto inferior direito clique sobre o ícone do Zerotier



- Na lista de opções clique em Join Networks:

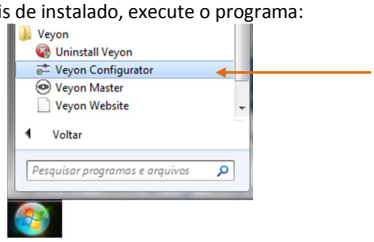


- Selecione todas as caixas de seleção e digite o código **a84ac5c10ab37ee0** e clique em Join:

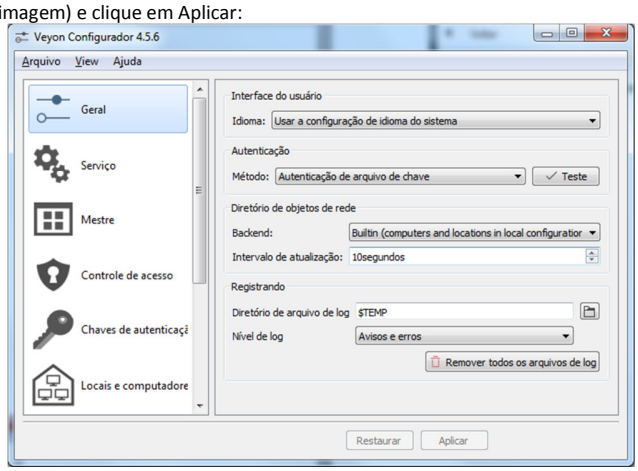


2) Instalação e uso de **Veyon**

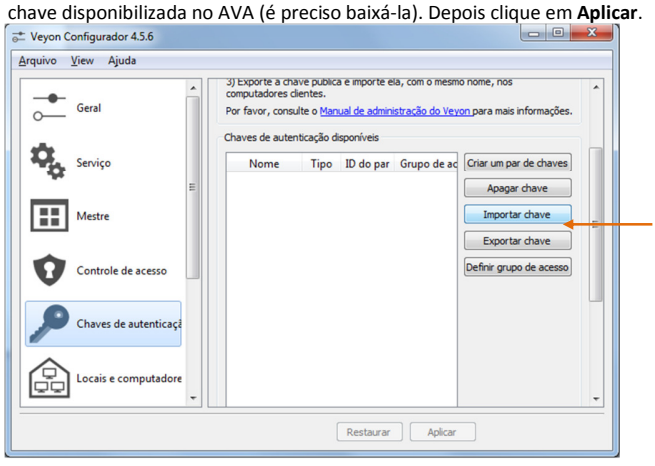
- Baixe o software Veyon em: https://veyon.io/en/download/
- Depois de instalado, execute o programa:



- No Menu Geral, habilite as configurações no menu **Geral** (conforme a imagem) e clique em Aplicar:



- No Menu Chaves de autenticação clique em **Importar chave** e adicione a chave disponibilizada no AVA (é preciso baixá-la). Depois clique em **Aplicar**.



UAST Unidade Acadêmica de Serra Talhada - PE UFRPE Desde 2006

BSi Bacharelado em Sistemas de Informação

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)

Profº: Richarlyson D'Emery **Data:** 17 / 10 / 2022

Aluno: _____

Nota:_____

VERIFICAÇÃO DE APRENDIZAGEM FINAL

Instruções gerais:

Salve as implementações a cada modificação, caso aconteça alguma falha de energia o trabalho será preservado. Lembre-se que uma vez removido o arquivo do eclipse, seu contendo será perdido.

A prova é prática e deverão ser entregues no SIGAA: [V.A. FINAL - Prova Prática] em Semana 16. Deverá ser entregue todo o projeto Eclipse contendo os códigos-fonte implementados em Java. *A Nota máxima desta prova é de 10,0 pontos*. A Pontuação da Prova está distribuída entre os conceitos vistos na disciplina MPOO CCMP5012.

A composição da nota da 3ª V.A. obedecerá a seguinte pontuação: V.A. FINAL = 100% * Prova Prática

---

1) No Eclipse limpe todos os projetos existentes.
   - Crie um novo **projeto chamado**
     br. vaFinalmpoo.edu.NomeSobrenome
     Este deverá ter uma **pasta de pacotes** chamada **sistemaMPOOShop** contendo todos os arquivos necessários para as respectivas questões.
   - Ao finalizar a prova **compacte o projeto** contendo toda a codificação do projeto (arquivos texto, *bytecodes* e imagens) e envio-o no SIGAA:
     [V.A. FINAL - Prova Prática]
     O SIGAA aceitará submissões até às 22h

O Problema:

A Empresa "MPOO Development Edu" recebeu do proprietário da rede "MPOO Shop" a demanda de criação de um protótipo de sistema que permite aos seus clientes receber descontos e da possibilidade de sua utilização. Sendo assim, você assumirá o papel de programador e apresentará um protótipo que atenda a essa demanda, mostre que você está preparado no desenvolvimento de sistemas Orientado a Objetos em Java. Para isso, tem-se o diagrama de classes com a modelagem da solução (Apêndice A) e a descrição de GUI's e funcionalidades do sistema a serem apresentadas.

Descrição:

2) É descrição do sistema:

Design Pattern: MVC (0,5 ponto)

a) Utilize os padrões de projeto: Model-View-Controller (MVC)

Classes, atributos e métodos construtores (0,75 ponto)
Encapsulamento e métodos de acesso (0,5 ponto)
static e no-static (0,25 ponto)
Abstract (0,5 ponto)
Herança (0,5 ponto)
Polimorfismo de objetos (0,75 ponto)
Agregação e Cardinalidade por ArrayList (0,75 pontos)
Composição (0,5 ponto)
Definição de métodos e suas implementações (0,75 ponto)

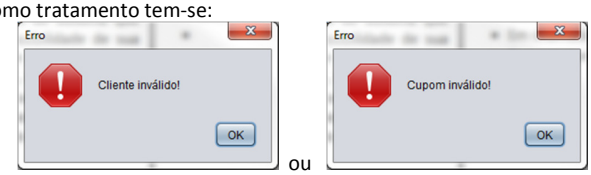
b) Implemente as definições do modelo do diagrama de classes do Apêndice A. É descrição:
   - BaseDados é uma classe que contém estruturas de dados para Pessoas e Compras. Também permite a definição de um Brinde.
   - cpf é a chave primária de uma pessoa física;
   - Todo cupom de desconto tem um código que o identifica;
   - Toda compra tem um id auto incrementável;
   - Toda pessoa jurídica tem uma franquia do tipo auto incrementável;
   - Todos os cupons do sistema definidos estão em EstoqueCupom;
   - Cliente contém CupomDesconto, sendo esta uma relação de composição.

Interface (0,75 ponto)

c) O sistema é composto pela interface CupomManaged. Mostre que você tem potencial de ser um programador de MPOO Development Edu apresentando suas utilizações e distinção quando estas possuem definições abstratas e estáticas.
   - validarCupom valida se um cupom é válido
   - valorCupom relaciona o valor para o código de um cupom
   - pegarCupom permite que um cliente reivindique um cupom desde que: o cliente exista e o cupom seja válido

Métodos e Manipulação de Exceção (0,75 ponto)

d) Faça a devida manipulação de Exception quando:
   - Um cliente informado não é válido
   - Um código de um cupom não é válido
   - Em chamadas de métodos exigir `try/catch`.
   - Como tratamento tem-se:



Componentes Gráficos (1,5 pontos)

É descrição das GUI's do sistema da Budega:

e) Possui a tela de opções:

MenuView (520x100)

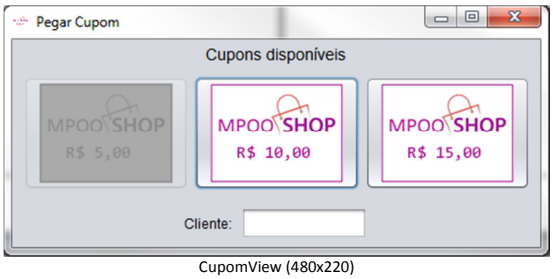
Utilize:

```
UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel")
```

f) A opção "Pegar Cupom" permite um cliente reivindicar um cupom de desconto:

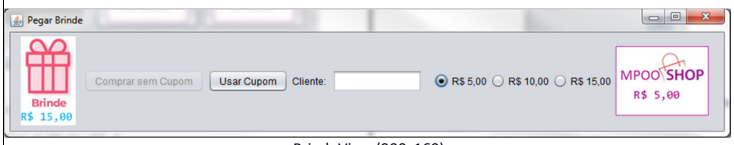
CupomView (480x220)

Quando um cupom não tem quantidade disponível, sua opção deve aparecer desabilitada:



CupomView (480x220)

g) A opção "Pegar Brinde" exibe:



BrindeView (900x160)

Observe que a opção para "Comprar sem Cupom" é desabilitada.

Para o usuário pegar um brinde é preciso ter selecionado a opção de R$15,00 e este cupom precisa ter sido reivindicado pelo cliente.

Caso o usuário acione "Usar Cupom" este deverá verificar se o cliente é válido (pertence à Base), caso contrário deve-se exibir a mensagem para cliente inválido da letra "d)".

Também se deve verificar se o cliente tem o cupom disponível, caso contrário deve-se exibir a mensagem correspondente da letra "d)"

Tratamento de Eventos (1,5 pontos)
Design Pattern: Bônus: Adapter (0,25 ponto)

h) Os tratamentos dos eventos das telas (sistema.view) devem estar nos respectivos controladores (sistema.controller) conforme diagrama de classes do Apêndice A:
- Para MenuView tem-se MenuController, sendo:
  - As opções "Pegar Cupom" e "Pegar Brinde" como classe interna privada.
  - A opção "Sair" como classe interna anônima
  - Encerrar pela tecla Escape por classe interna privada.
- Para CupomView tem-se CupomController, sendo:
  - botões de cupom tratado na própria classe (classe realiza interface)
- BrindeView tem-se BrindeController, sendo:
  - botão "Usar Cupom" tratado na própria classe (classe realiza interface).

i) As opções de encerrar do sistema (opção Sair de MenuView e tecla Esc) devem ter confirmação do usuário:

Thread: (0,5 ponto)

j) O sistema possui um robô (GerarCupom) que a cada 30 dias (GERAR = 2592000000L) executa-se a criação de 10 novos cupons. *Mas atenção:* não se deve confundir uma geração automática realizada por um robô *started* em uma aplicação com a possibilidade de definir valores diretamente em EstoqueCupom!

Instâncias (0,5 pontos)

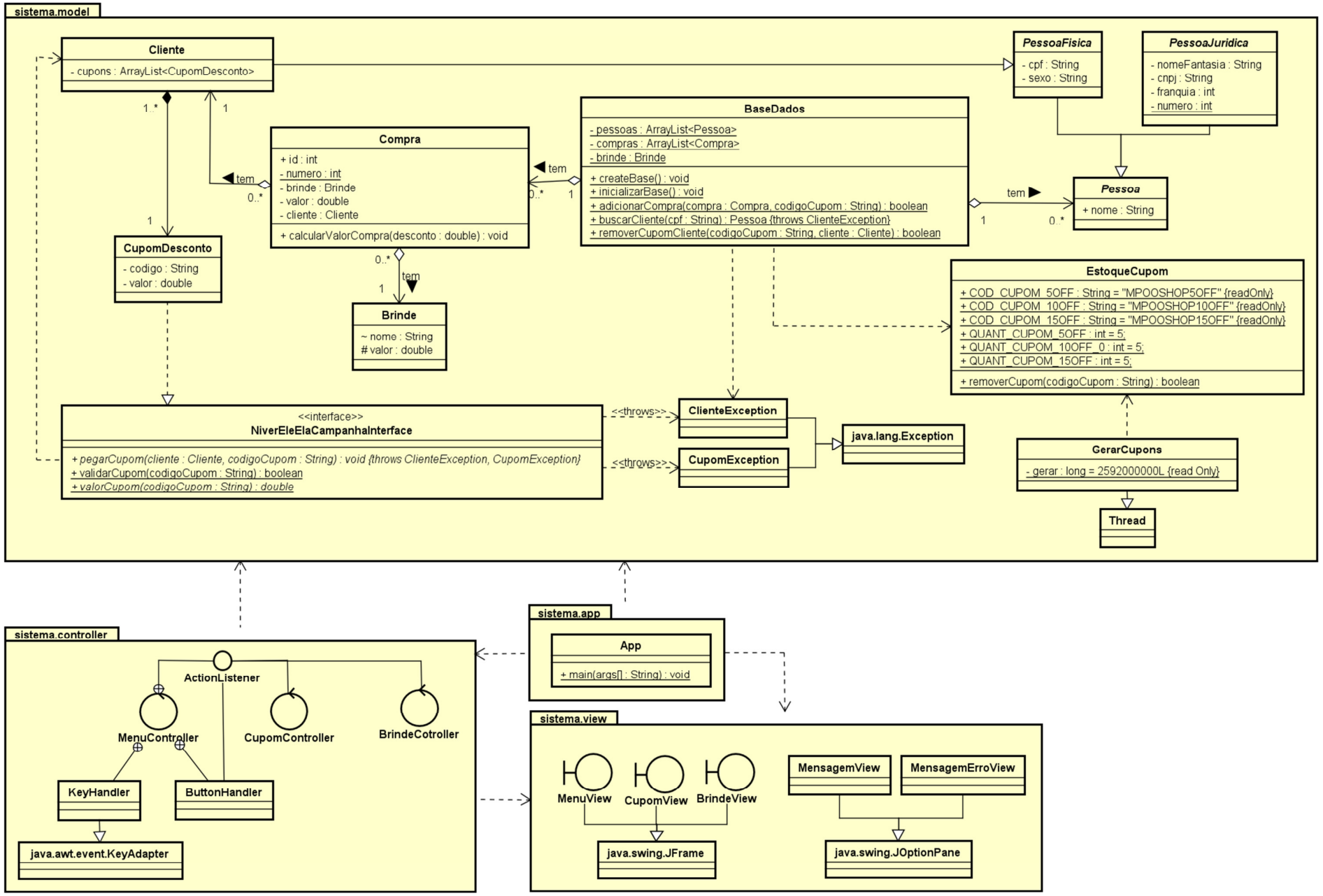
k) Na Base de Dados:
- Criação de um brinde: de nome Brinde e valor R$ 15.0
- Criação de
  - Um cliente com seus dados pessoais
  - Outro cliente de nome Rico DEmery, cpf 111.111.111-11 do sexo masculino
  - Uma pessoa jurídica de nome MPOO Shop, nome Fantasia MShop, cnpj 01.000.000/0001-11 sendo esta a primeira franquia.

Instâncias e chamadas de métodos (0,5 ponto)

l) Deve-se ter em App:
- A criação de instâncias MVC e chamadas de métodos (base, telas, controladores e thread).

```
public class Window{
    JButton button = new JButton();
    public static void main(String[] args) {
        Window window = new Window();
        window.button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                JOptionPane.showMessageDialog(null, "Boa Prova!");
            }
        });
        window.button.doClick();
    }
}
```

APÊNDICE A


sistema.model
Cliente
- cupons : ArrayList<CupomDesconto>
1..*
1
tem
0..*
1
CupomDesconto
- codigo : String
- valor : double
Compra
+ id : int
- numero : int
- brinde : Brinde
- valor : double
- cliente : Cliente
+ calcularValorCompra(desconto : double) : void
0..*
tem
1
Brinde
~ nome : String
# valor : double
tem
0..*
1
BaseDados
- pessoas : ArrayList<Pessoa>
- compras : ArrayList<Compra>
- brinde : Brinde
+ createBase() : void
+ inicializarBase() : void
+ adicionarCompra(compra : Compra, codigoCupom : String) : boolean
+ buscarCliente(cpf : String) : Pessoa {throws ClienteException}
+ removerCupomCliente(codigoCupom : String, cliente : Cliente) : boolean
tem
1
0..*
PessoaFisica
- cpf : String
- sexo : String
PessoaJuridica
- nomeFantasia : String
- cnpj : String
- franquia : int
- numero : int
Pessoa
+ nome : String
EstoqueCupom
+ COD_CUPOM_5OFF : String = "MPOOSHOP5OFF" {readOnly}
+ COD_CUPOM_10OFF : String = "MPOOSHOP10OFF" {readOnly}
+ COD_CUPOM_15OFF : String = "MPOOSHOP15OFF" {readOnly}
+ QUANT_CUPOM_5OFF : int = 5;
+ QUANT_CUPOM_10OFF_0 : int = 5;
+ QUANT_CUPOM_15OFF : int = 5;
+ removerCupom(codigoCupom : String) : boolean
<<interface>>
NiverEleElaCampanhaInterface
+ pegarCupom(cliente : Cliente, codigoCupom : String) : void {throws ClienteException, CupomException}
+ validarCupom(codigoCupom : String) : boolean
+ valorCupom(codigoCupom : String) : double
<<throws>>
<<throws>>
ClienteException
CupomException
java.lang.Exception
GerarCupons
- gerar : long = 2592000000L {read Only}
Thread
sistema.controller
ActionListener
MenuController
CupomController
BrindeCotroller
KeyHandler
ButtonHandler
java.awt.event.KeyAdapter
sistema.app
App
+ main(args[] : String) : void
sistema.view
MenuView
CupomView
BrindeView
java.swing.JFrame
MensagemView
MensagemErroView
java.swing.JOptionPane