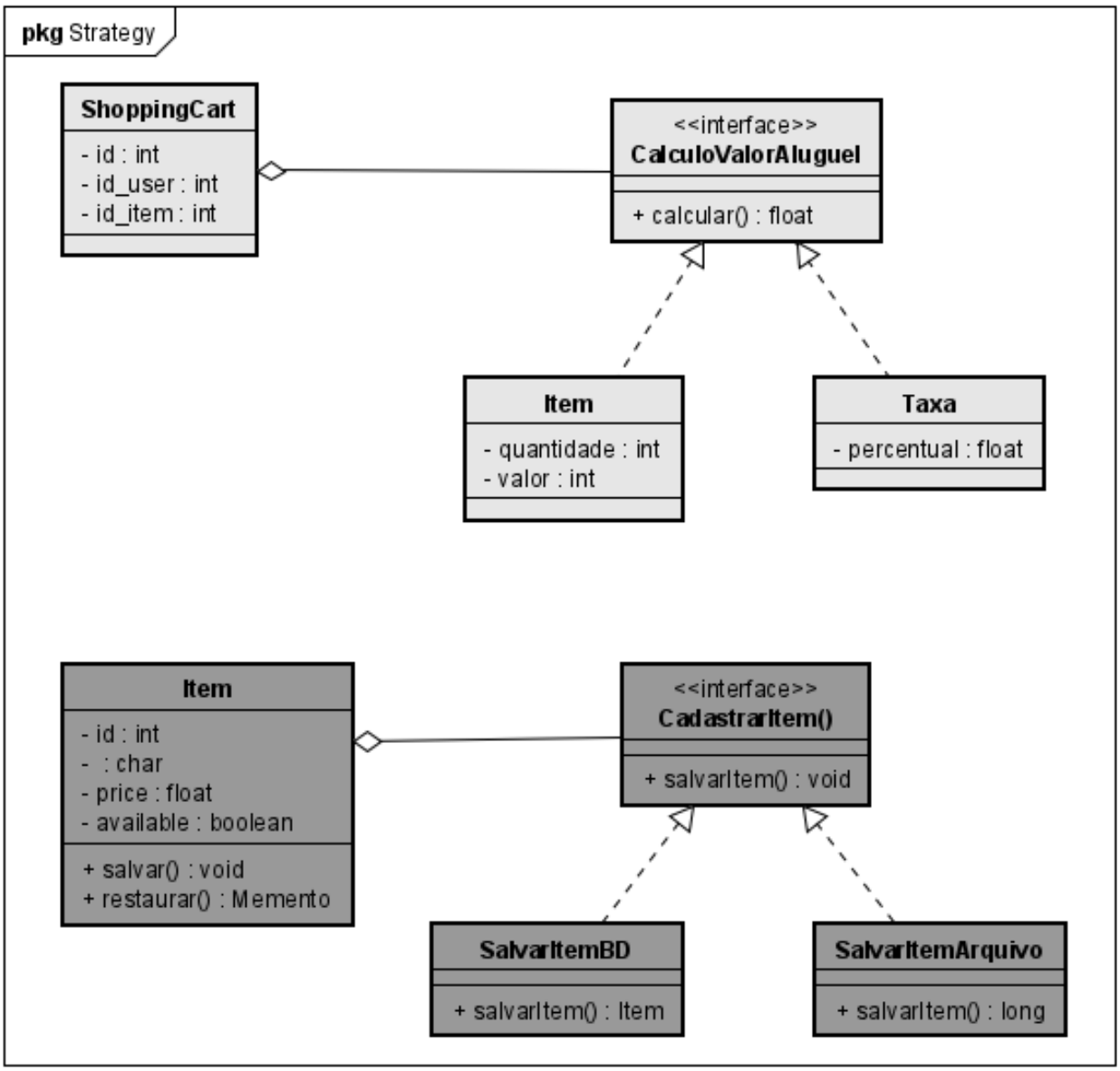
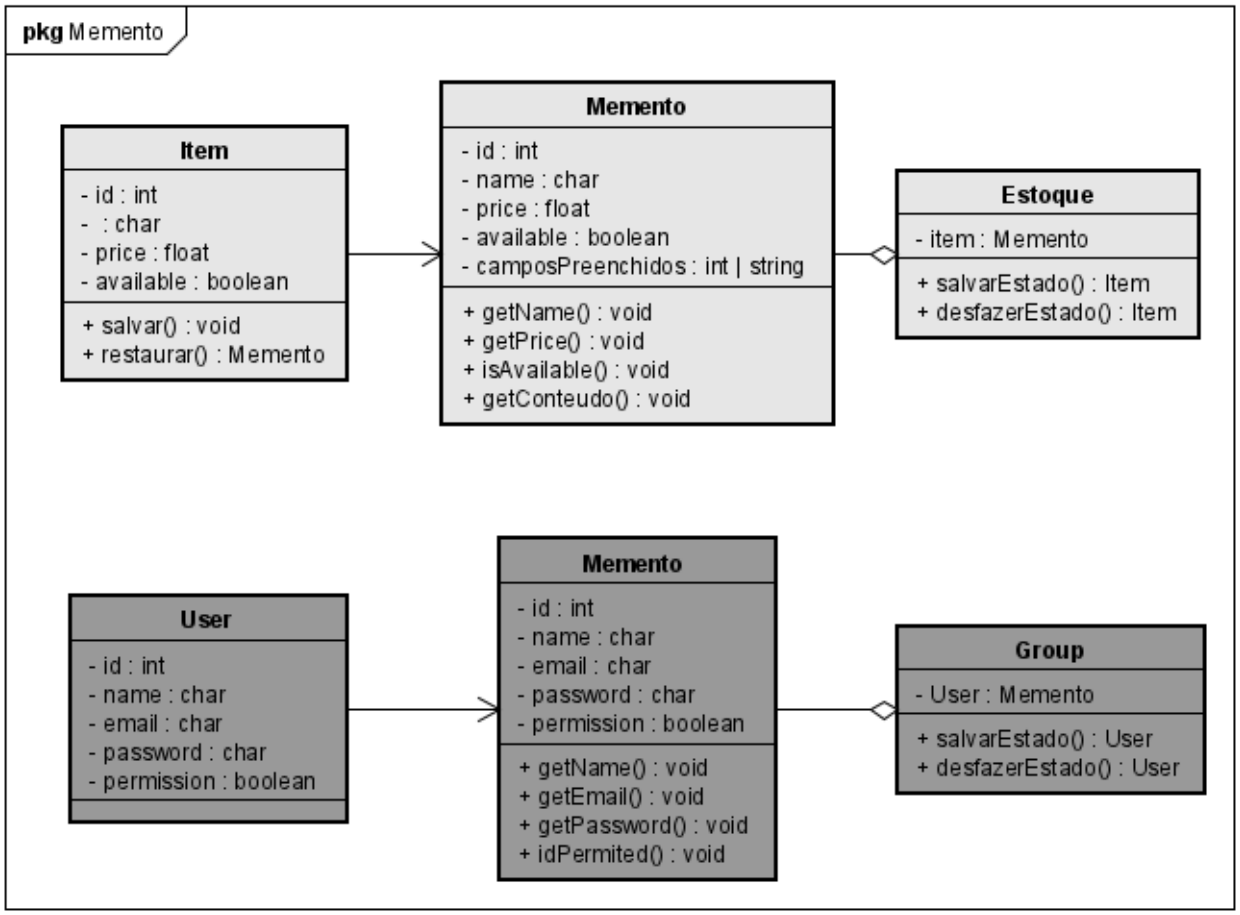
Design Patterns - Individual

Ary Farah

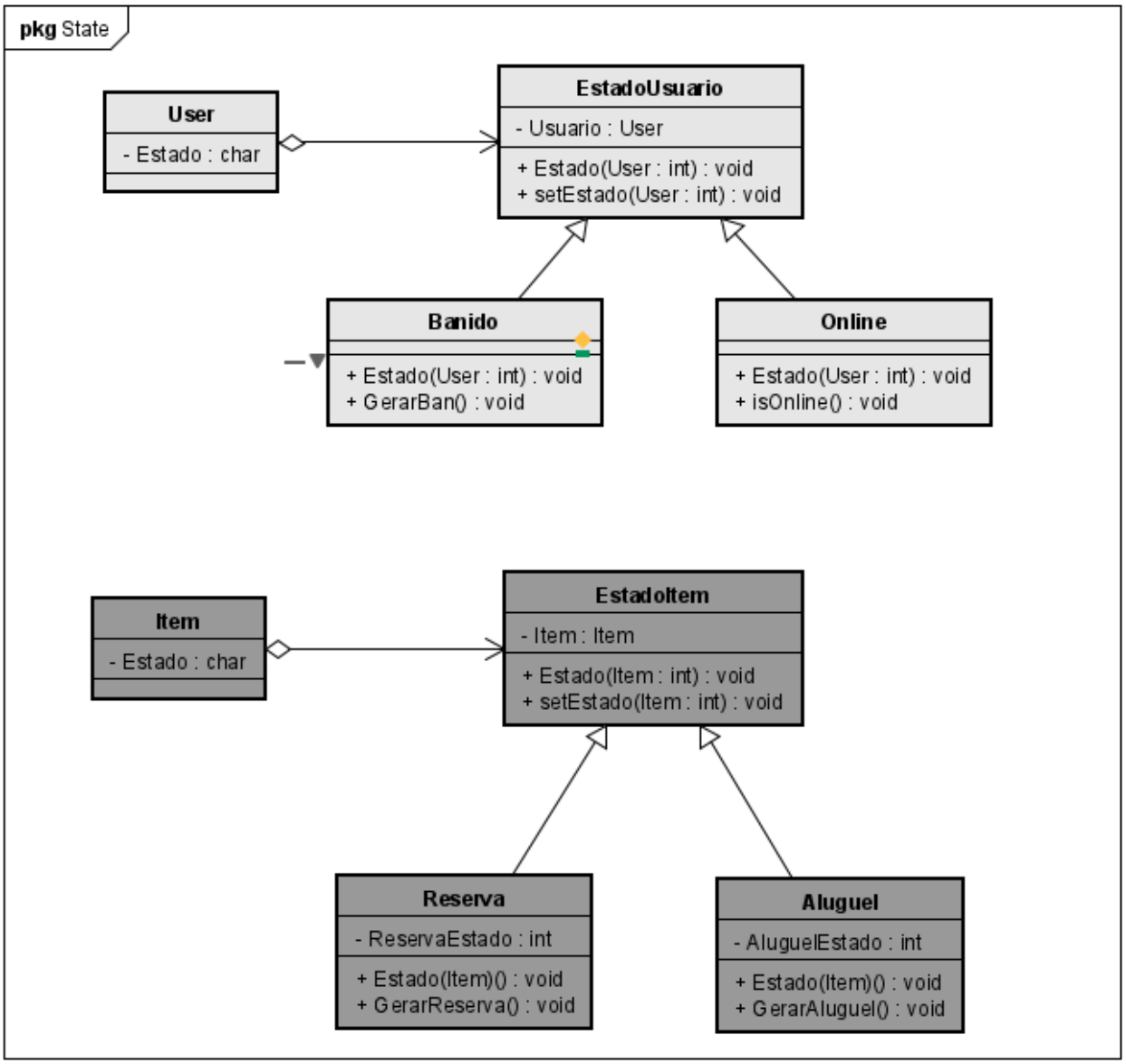
Strategy


pkg Strategy
ShoppingCart
- id : int
- id_user : int
- id_item : int
<<interface>>
CalculoValorAluguel
+ calcular() : float
Item
- quantidade : int
- valor : int
Taxa
- percentual : float
Item
- id : int
- : char
- price : float
- available : boolean
+ salvar() : void
+ restaurar() : Memento
<<interface>>
CadastrarItem()
+ salvarItem() : void
SalvarItemBD
+ salvarItem() : Item
SalvarItemArquivo
+ salvarItem() : long

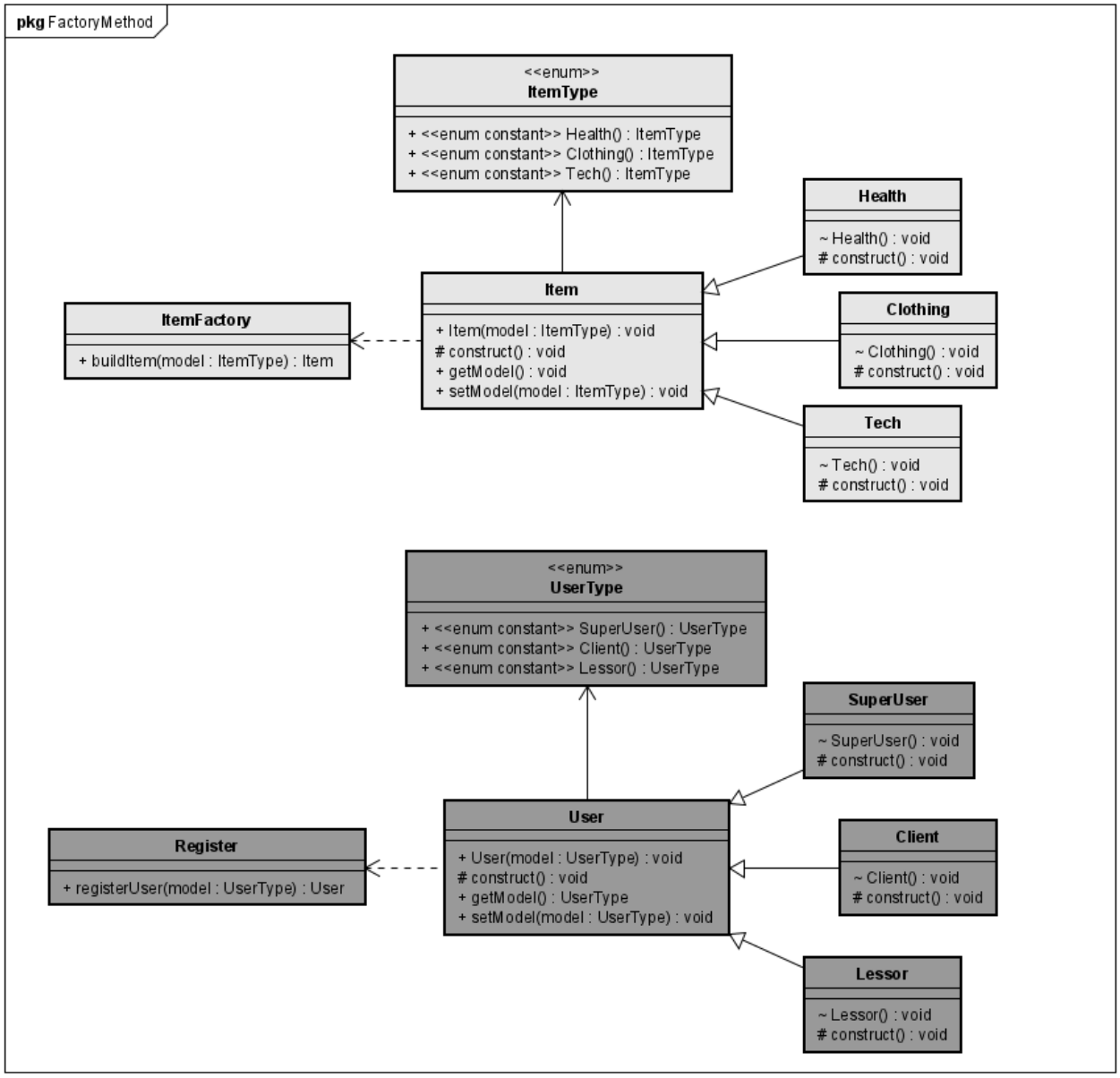
Memento


pkg Memento
Item
- id : int
- : char
- price : float
- available : boolean
+ salvar() : void
+ restaurar() : Memento
Memento
- id : int
- name : char
- price : float
- available : boolean
- camposPreenchidos : int | string
+ getName() : void
+ getPrice() : void
+ isAvailable() : void
+ getConteudo() : void
Estoque
- item : Memento
+ salvarEstado() : Item
+ desfazerEstado() : Item
User
- id : int
- name : char
- email : char
- password : char
- permission : boolean
Memento
- id : int
- name : char
- email : char
- password : char
- permission : boolean
+ getName() : void
+ getEmail() : void
+ getPassword() : void
+ idPermited() : void
Group
- User : Memento
+ salvarEstado() : User
+ desfazerEstado() : User

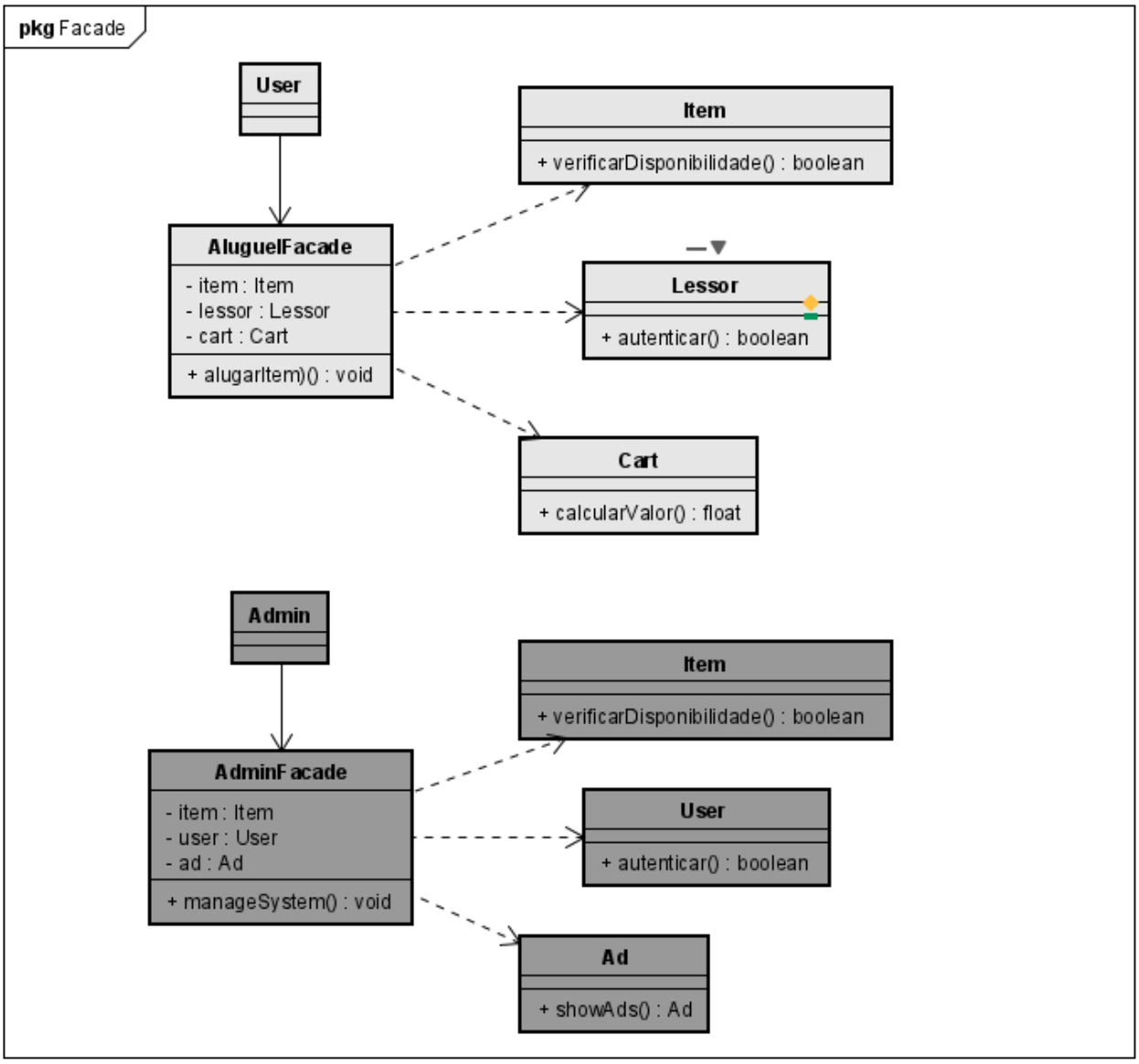
State



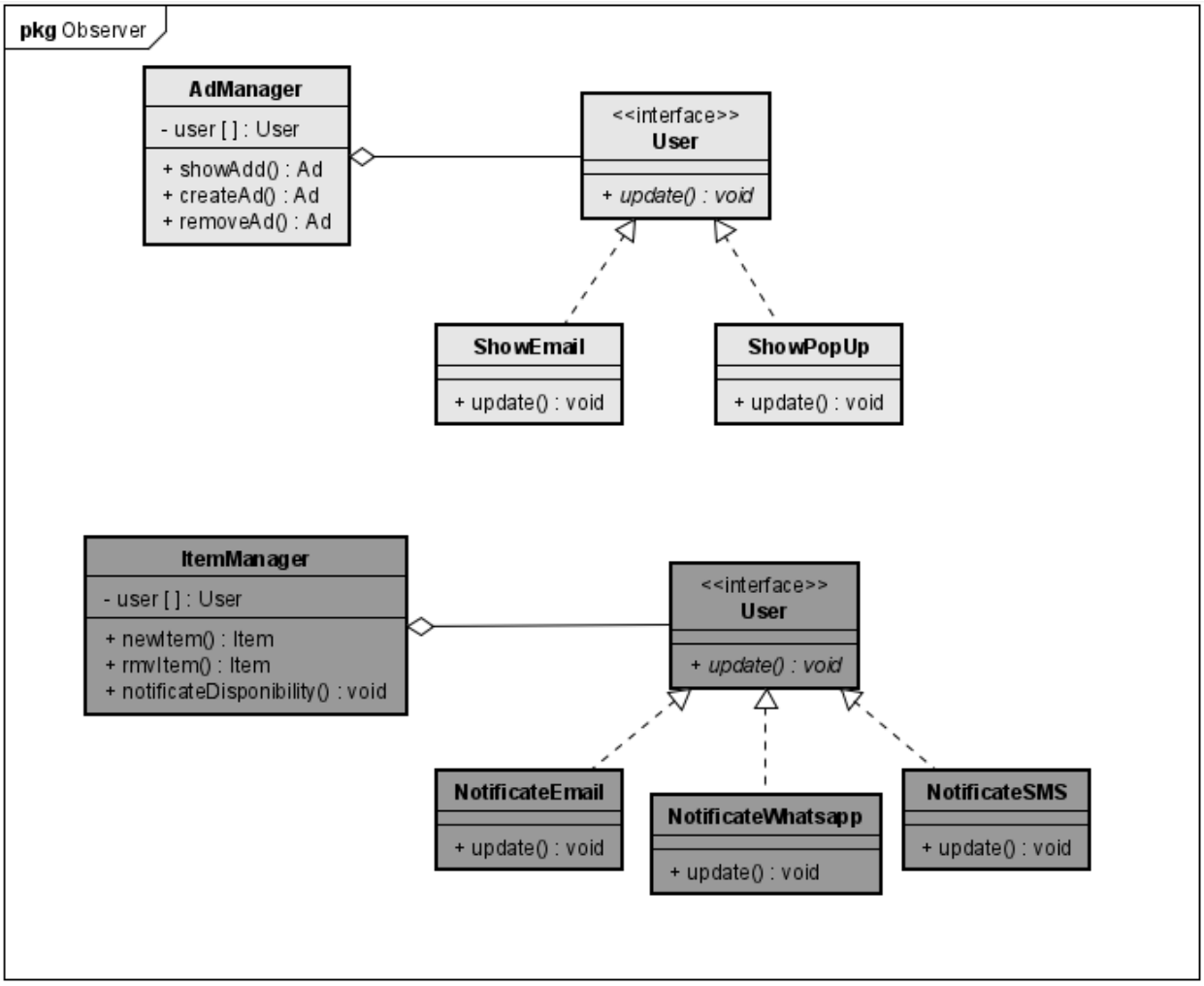
Factory Method


pkg FactoryMethod
<<enum>>
ItemType
+ <<enum constant>> Health() : ItemType
+ <<enum constant>> Clothing() : ItemType
+ <<enum constant>> Tech() : ItemType
Health
~ Health() : void
# construct() : void
Item
+ Item(model : ItemType) : void
# construct() : void
+ getModel() : void
+ setModel(model : ItemType) : void
Clothing
~ Clothing() : void
# construct() : void
ItemFactory
+ buildItem(model : ItemType) : Item
Tech
~ Tech() : void
# construct() : void
<<enum>>
UserType
+ <<enum constant>> SuperUser() : UserType
+ <<enum constant>> Client() : UserType
+ <<enum constant>> Lessor() : UserType
SuperUser
~ SuperUser() : void
# construct() : void
User
+ User(model : UserType) : void
# construct() : void
+ getModel() : UserType
+ setModel(model : UserType) : void
Register
+ registerUser(model : UserType) : User
Client
~ Client() : void
# construct() : void
Lessor
~ Lessor() : void
# construct() : void

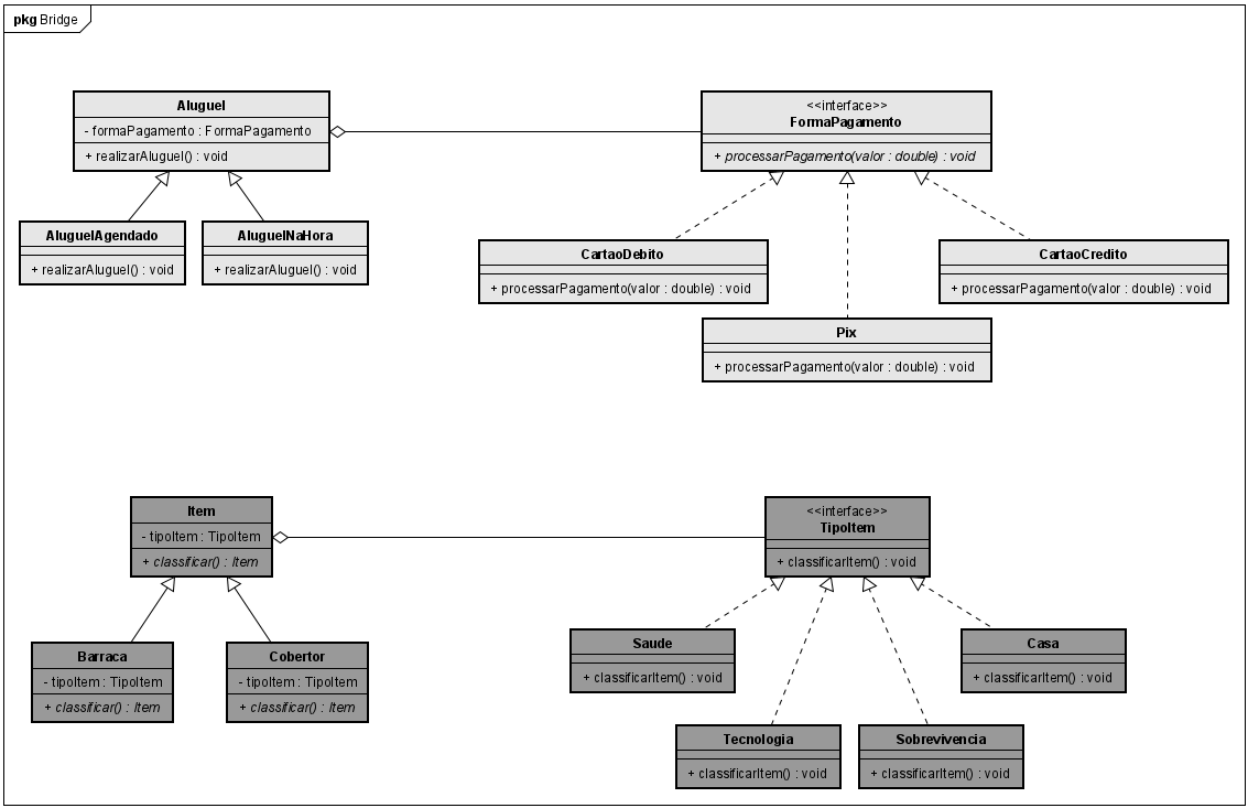
Facade


pkg Facade
User
Item
+ verificarDisponibilidade() : boolean
AluguelFacade
- item : Item
- lessor : Lessor
- cart : Cart
+ alugarItem)() : void
Lessor
+ autenticar() : boolean
Cart
+ calcularValor() : float
Admin
Item
+ verificarDisponibilidade() : boolean
AdminFacade
- item : Item
- user : User
- ad : Ad
+ manageSystem() : void
User
+ autenticar() : boolean
Ad
+ showAds() : Ad

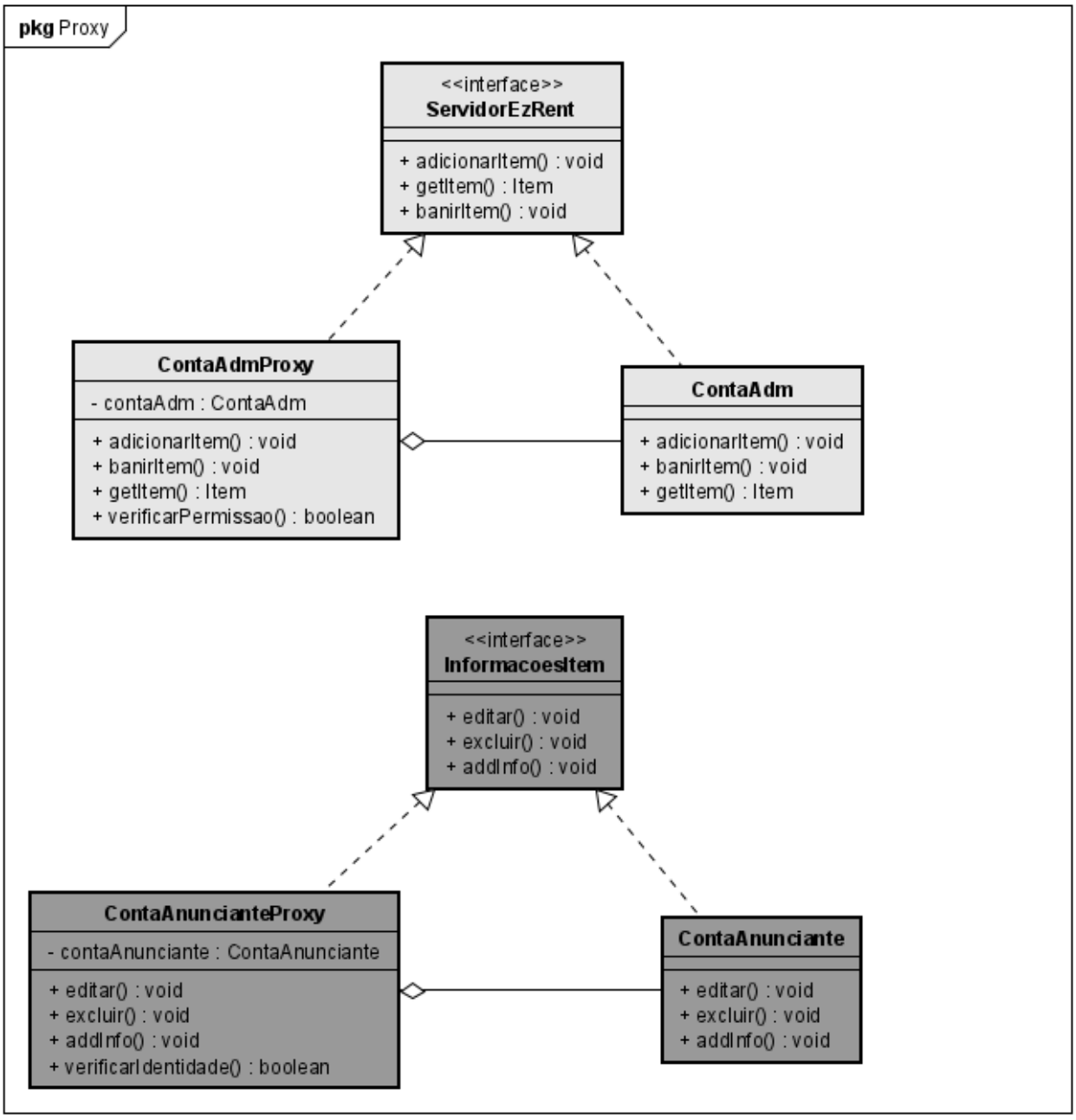
Observer


pkg Observer
AdManager
- user [ ] : User
+ showAdd() : Ad
+ createAd() : Ad
+ removeAd() : Ad
<<interface>>
User
+ update() : void
ShowEmail
+ update() : void
ShowPopUp
+ update() : void
ItemManager
- user [ ] : User
+ newItem() : Item
+ rmvItem() : Item
+ notificateDisponibility() : void
<<interface>>
User
+ update() : void
NotificateEmail
+ update() : void
NotificateWhatsapp
+ update() : void
NotificateSMS
+ update() : void

Bridge


pkg Bridge
Aluguel
- formaPagamento : FormaPagamento
+ realizarAluguel() : void
AluguelAgendado
+ realizarAluguel() : void
AluguelNaHora
+ realizarAluguel() : void
<<interface>>
FormaPagamento
+ processarPagamento(valor : double) : void
CartaoDebito
+ processarPagamento(valor : double) : void
CartaoCredito
+ processarPagamento(valor : double) : void
Pix
+ processarPagamento(valor : double) : void
Item
- tipoItem : TipoItem
+ classificar() : Item
Barraca
- tipoItem : TipoItem
+ classificar() : Item
Cobertor
- tipoItem : TipoItem
+ classificar() : Item
<<interface>>
TipoItem
+ classificarItem() : void
Saude
+ classificarItem() : void
Tecnologia
+ classificarItem() : void
Sobrevivencia
+ classificarItem() : void
Casa
+ classificarItem() : void

Proxy


pkg Proxy
<<interface>>
ServidorEzRent
+ adicionarItem() : void
+ getItem() : Item
+ banirItem() : void
ContaAdmProxy
- contaAdm : ContaAdm
+ adicionarItem() : void
+ banirItem() : void
+ getItem() : Item
+ verificarPermissao() : boolean
ContaAdm
+ adicionarItem() : void
+ banirItem() : void
+ getItem() : Item
<<interface>>
InformacoesItem
+ editar() : void
+ excluir() : void
+ addInfo() : void
ContaAnunciante
+ editar() : void
+ excluir() : void
+ addInfo() : void
ContaAnunciante
+ editar() : void
+ excluir() : void
+ addInfo() : void
+ verificarIdentidade() : boolean