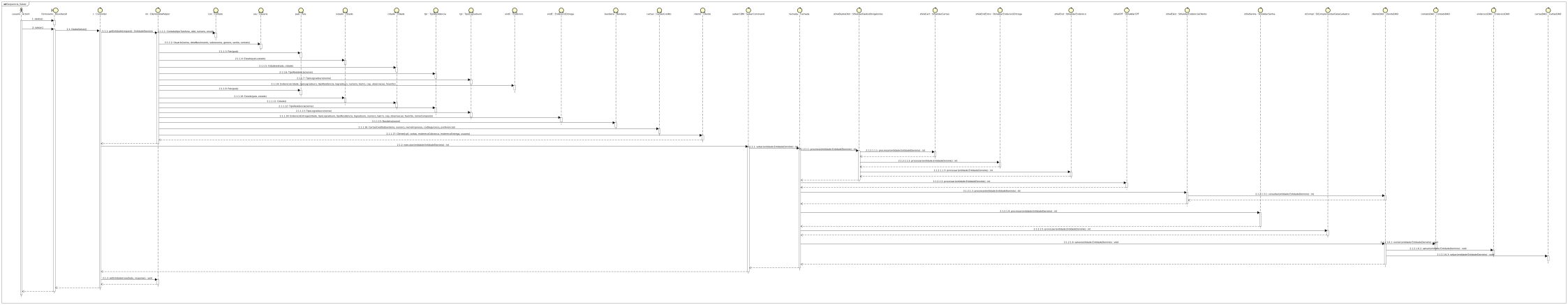
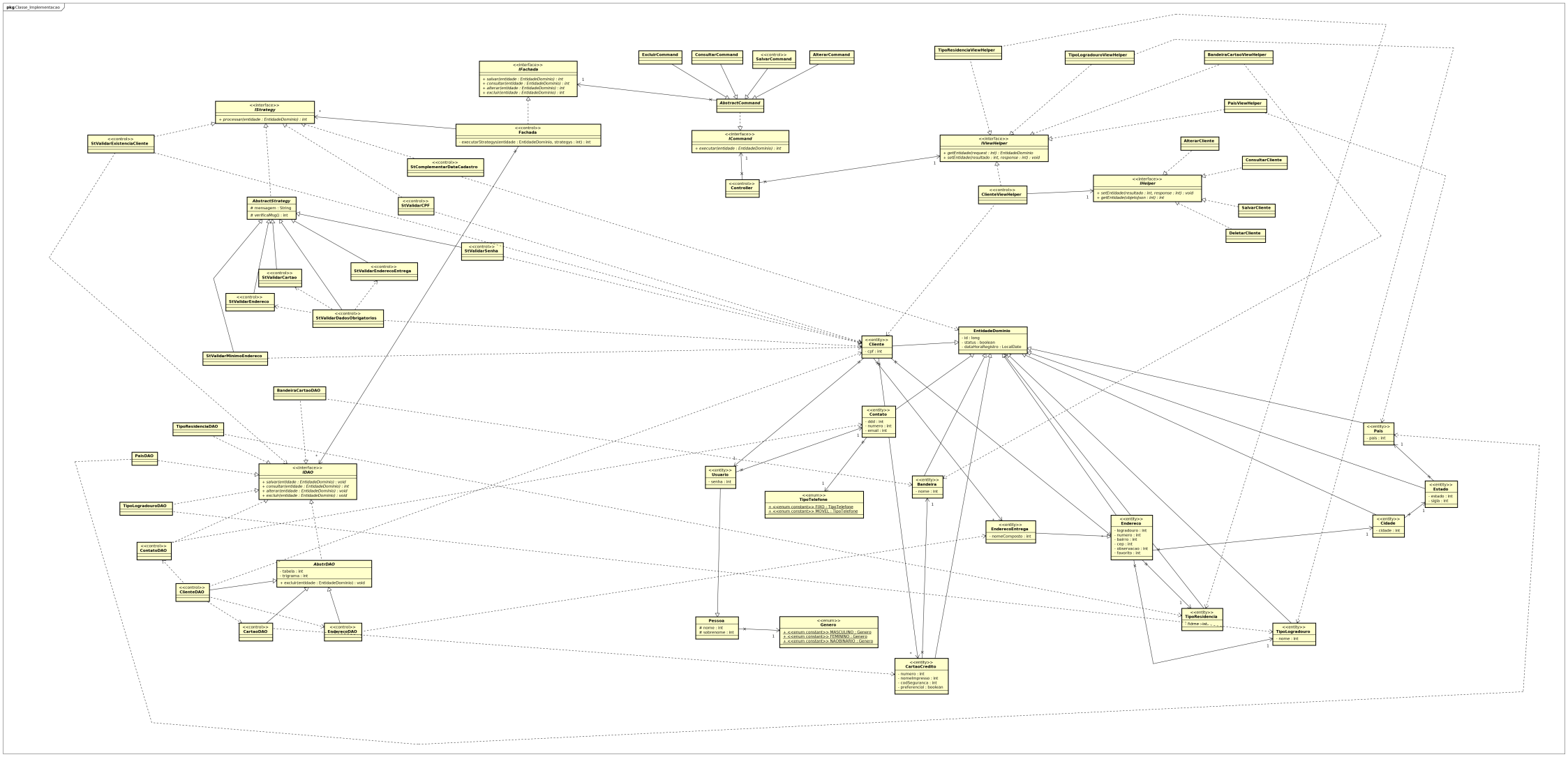
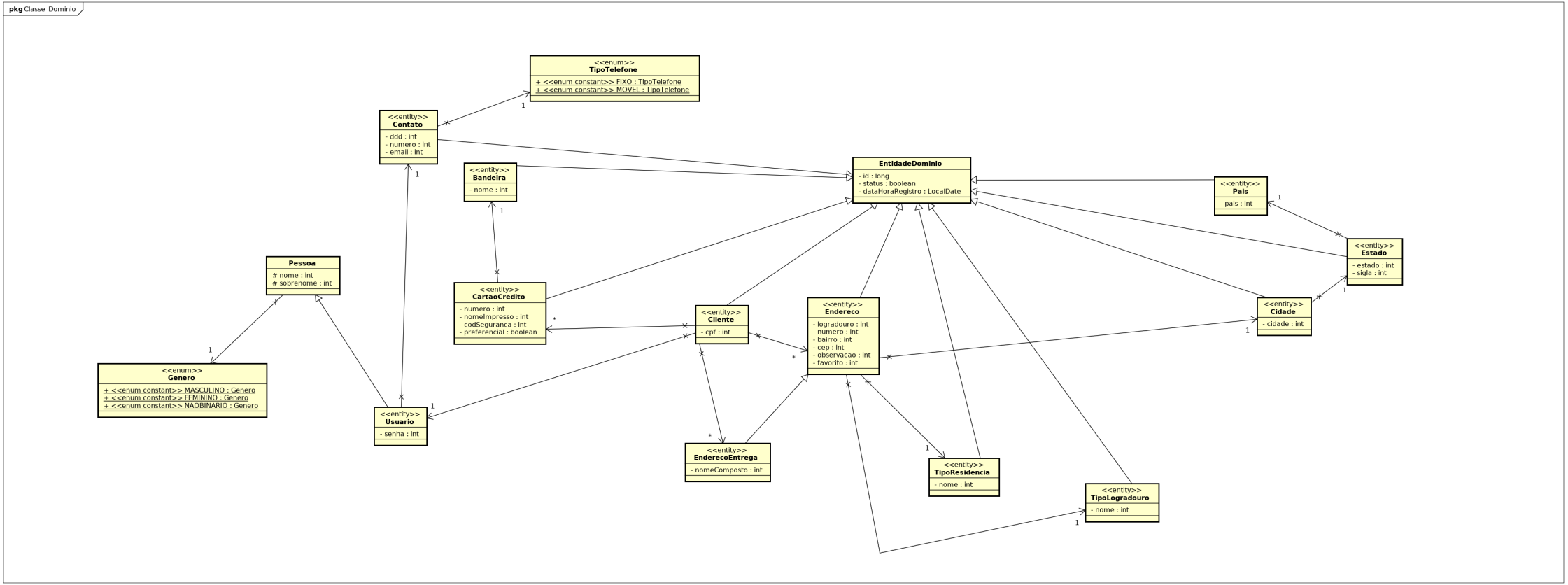

sd Sequence_Solar
usuario : Actor
1 : Iniciar()
3: Salir()
formulario : BorrarEvid
4 : Cerrar()
i : CtrlCliente
2.1 : CrearCliente()
ch : ChecksCuestiopr
2.1.1 : getFichaAdelantaqued : EntidadDominio
con : Cliente
2.1.1.1 : ContactoTipo(Telefono, dato, numero, pais)
dni : Cliente
2.1.1.2 : Clase(Nombre, dateBaseUnaria, valoresnna, genero, varios, varioss)
dir : Cliente
2.1.1.3 : Pais(pais)
edad : Cliente
2.1.1.4 : Edad(edad,cuando)
cliente : Cliente
2.1.1.5 : Cliente(idCalle, cliente)
tp : TipoDirecciones
2.1.1.6 : TipoDireccion(tipodirecion)
tpr : TipoProvincia
2.1.1.7 : Tipo(provinciaprovincia)
ovd : CodPostales
2.1.1.8 : Codpostal(idCalle, idCiudad, idProvincia, idTipoDireccion, tipo, numero, bloq, cpp, obsercoccpa, fasoldo)
2.1.1.9 : Pais(pais)
2.1.1.10 : Edad(edad, edad)
2.1.1.11 : Cliente()
2.1.1.12 : Tipo(tipodirecion,tipo)
2.1.1.13 : Tipo(provinciaprovincia)
2.1.1.14 : Cliente(idCiudad, idTipoDireccion, idTipoProvincia, idCodPostal, tipo, numero, bloq, cpp, obsercoccpa, fasoldo, solicitCompuesto)
2.1.1.15 : Bandera(numero)
2.1.1.16 : CarSolicitUbicaciones(id, numero, nombreVarios, ccdlegacrecs, prvdirecccpa)
2.1.1.17 : Cliente(pd, solidas, endereco,ciudadcoca, endereco,idCompa, cuando)
2.1.2 : CrearLocalizaUbicaciones(EntidadDominio : id)
solicitUBC : SolicitCreateUbicaciones
2.1.2.1 : solicitCreateUbicaciones(EntidadDominio)
fichaAdelantada : FichaAdelantada
2.1.2.1.1 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.1 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.2 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.3 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.4 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.5 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.6 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.7 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.8 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.9 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.10 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.11 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.12 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.13 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.14 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.15 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.16 : procesarsolicitUbicaciones(EntidadDominio) : id
2.1.2.1.1.17 : procesarsolicitUbicaciones(EntidadDominio) : id


pkg Classe_Implementacao
<<interface>>
IFachada
+ salvar(entidade : EntidadeDominio) : int
+ consultar(entidade : EntidadeDominio) : int
+ alterar(entidade : EntidadeDominio) : int
+ excluir(entidade : EntidadeDominio) : int
ExcluirCommand
ConsultarCommand
<<control>>
SalvarCommand
AlterarCommand
AbstractCommand
<<interface>>
ICommand
+ executar(entidade : EntidadeDominio) : int
<<control>>
Controller
<<interface>>
IStrategy
+ processar(entidade : EntidadeDominio) : int
<<control>>
Fachada
- executarStrategys(entidade : EntidadeDominio, strategys : int) : int
<<control>>
StValidarExistenciaCliente
<<control>>
StComplementarDataCadastro
<<control>>
StValidarCPF
AbstractStrategy
mensagem : String
verificaMsg() : int
<<control>>
StValidarSenha
<<control>>
StValidarEnderecoEntrega
<<control>>
StValidarCartao
<<control>>
StValidarEndereco
<<control>>
StValidarDadosObrigatorios
StValidarMinimoEndereco
TipoResidenciaViewHelper
TipoLogradouroViewHelper
BandeiraCartaoViewHelper
PaisViewHelper
<<interface>>
IViewHelper
+ getEntidade(request : int) : EntidadeDominio
+ setEntidade(resultado : int, response : int) : void
AlterarCliente
ConsultarCliente
<<interface>>
IHelper
+ setEntidade(resultado : int, response : int) : void
+ getEntidade(objetojson : int) : int
<<control>>
ClienteViewHelper
SalvarCliente
DeletarCliente
EntidadeDominio
- id : long
- status : boolean
- dataHoraAlteraGlobo : LocalDate
<<entity>>
Cliente
- cpf : int
<<entity>>
Contato
- ddd : int
- numero : int
- email : int
<<entity>>
Usuario
- senha : int
<<enum>>
TipoTelefone
+ <<enum constant>> FIXO : TipoTelefone
+ <<enum constant>> MÓVEL : TipoTelefone
<<entity>>
Bandeira
- nome : int
<<entity>>
EnderecoEntrega
- nomeComposto : int
<<entity>>
Endereco
- logradouro : int
- numero : int
- bairro : int
- cep : int
- observacao : int
- favorito : int
<<entity>>
Pais
- pais : int
<<entity>>
Estado
- estado : int
- sigla : int
<<entity>>
Cidade
- cidade : int
<<entity>>
TipoResidencia
- nome : int
<<entity>>
TipoLogradouro
- nome : int
<<entity>>
CartaoCredito
- numero : int
- nomeImpresso : int
- codSeguranca : int
- preferencial : boolean
Pessoa
nome : int
sobrenome : int
<<enum>>
Genero
+ <<enum constant>> MASCULINO : Genero
+ <<enum constant>> FEMININO : Genero
+ <<enum constant>> NAOINFORMADO : Genero
BandeiraCartaoDAO
TipoResidenciaDAO
PaisDAO
<<interface>>
IDAO
+ salvar(entidade : EntidadeDominio) : void
+ consultar(entidade : EntidadeDominio) : int
+ alterar(entidade : EntidadeDominio) : void
+ excluir(entidade : EntidadeDominio) : void
TipoLogradouroDAO
<<control>>
ContatoDAO
AbstrDAO
- tabela : int
- trigrama : int
+ excluir(entidade : EntidadeDominio) : void
<<control>>
ClienteDAO
<<control>>
CartaoDAO
<<control>>
EnderecoDAO


pkg Classe_Dominio
<<enum>>
TipoTelefone
+ <<enum constant>> FIXO : TipoTelefone
+ <<enum constant>> MOVEL : TipoTelefone
<<entity>>
Contato
- ddd : int
- numero : int
- email : int
<<entity>>
Bandeira
- nome : int
EntidadeDominio
- id : long
- status : boolean
- dataHoraRegistro : LocalDate
<<entity>>
Pais
- pais : int
<<entity>>
Estado
- estado : int
- sigla : int
Pessoa
nome : int
sobrenome : int
<<entity>>
CartaoCredito
- numero : int
- nomeImpresso : int
- codSeguranca : int
- preferencial : boolean
<<entity>>
Cliente
- cpf : int
<<entity>>
Endereco
- logradouro : int
- numero : int
- bairro : int
- cep : int
- observacao : int
- favorito : int
<<entity>>
Cidade
- cidade : int
<<enum>>
Genero
+ <<enum constant>> MASCULINO : Genero
+ <<enum constant>> FEMININO : Genero
+ <<enum constant>> NAOBINARIO : Genero
<<entity>>
Usuario
- senha : int
<<entity>>
EnderecoEntrega
- nomeComposto : int
<<entity>>
TipoResidencia
- nome : int
<<entity>>
TipoLogradouro
- nome : int
1
*