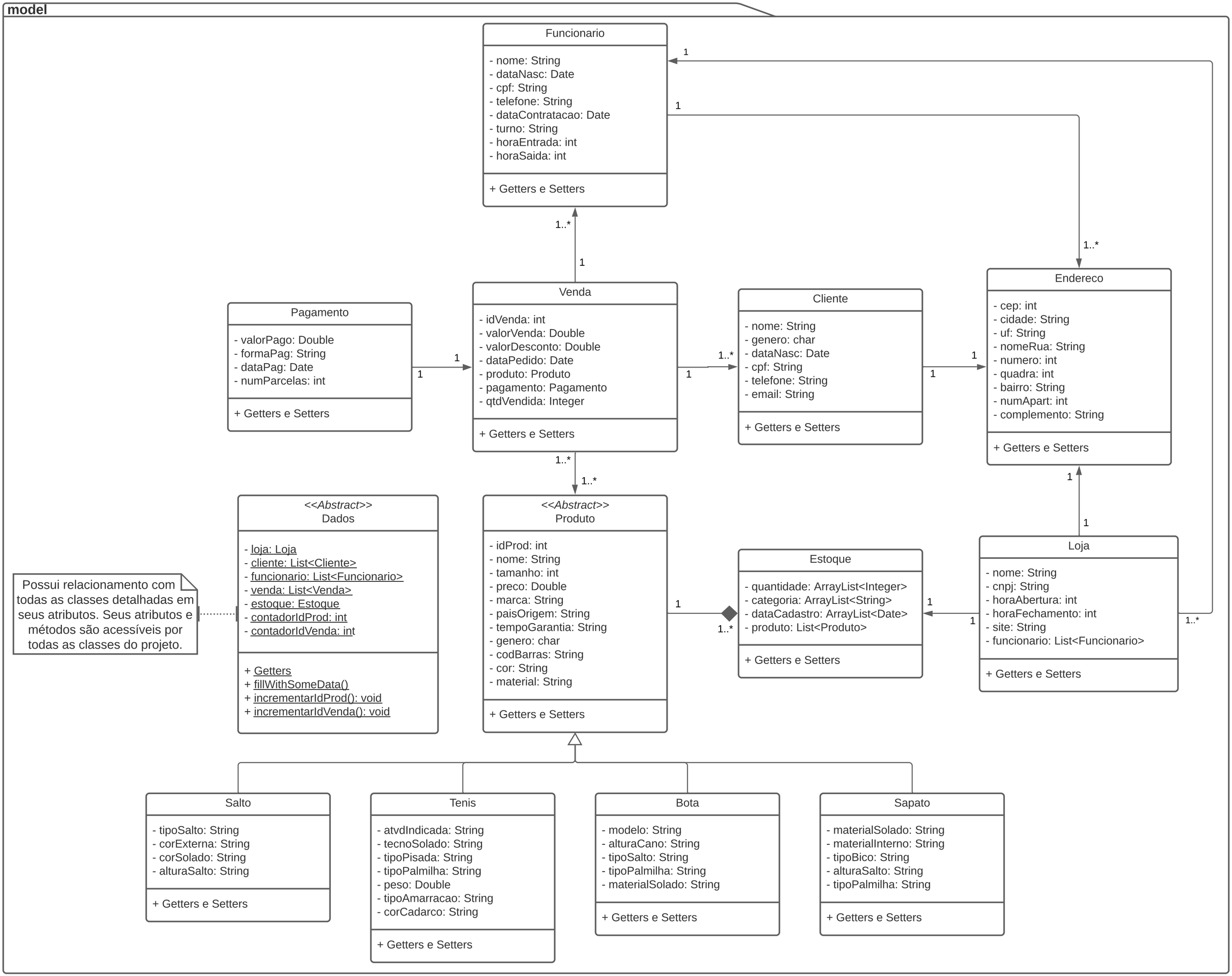
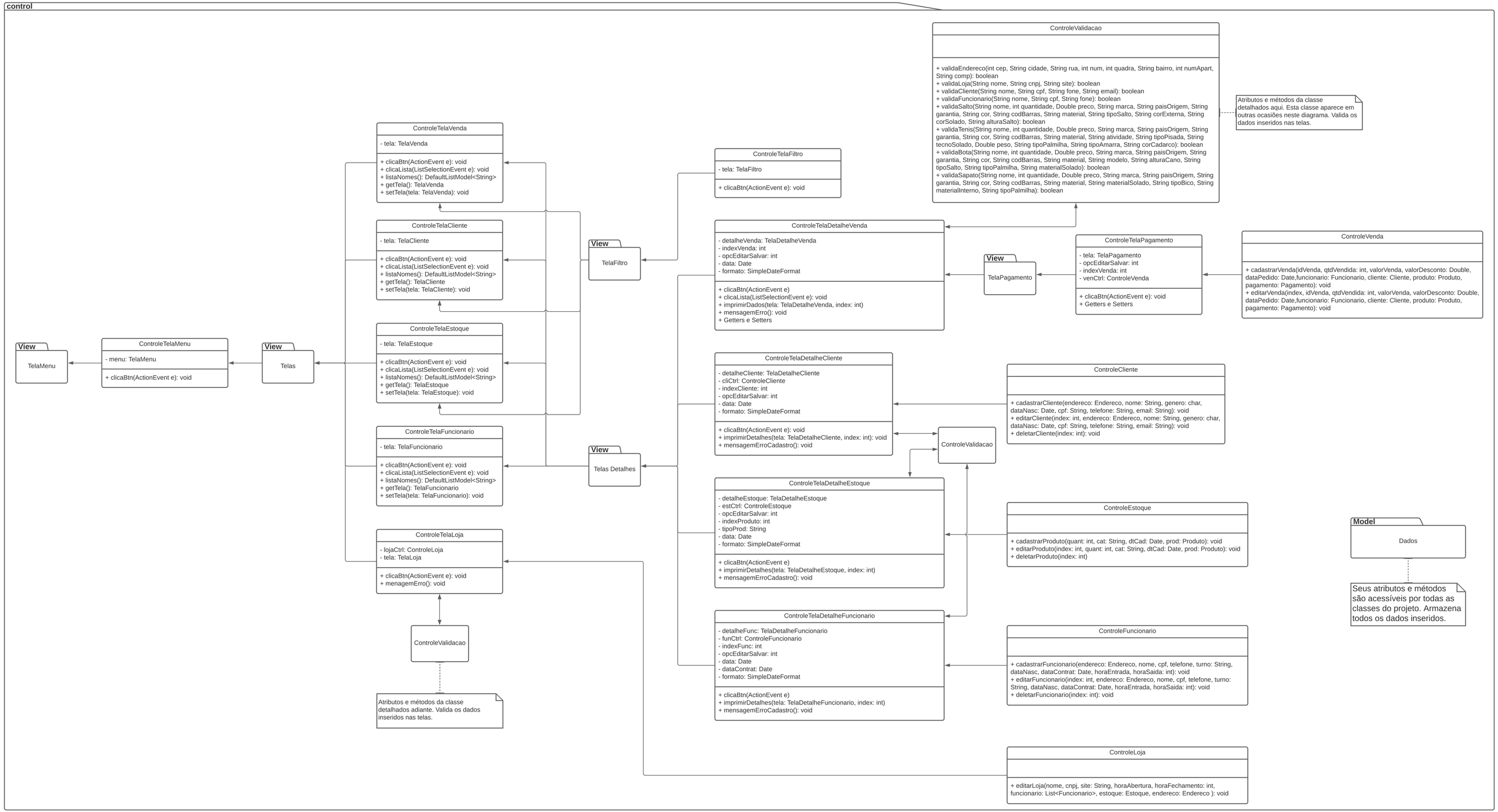
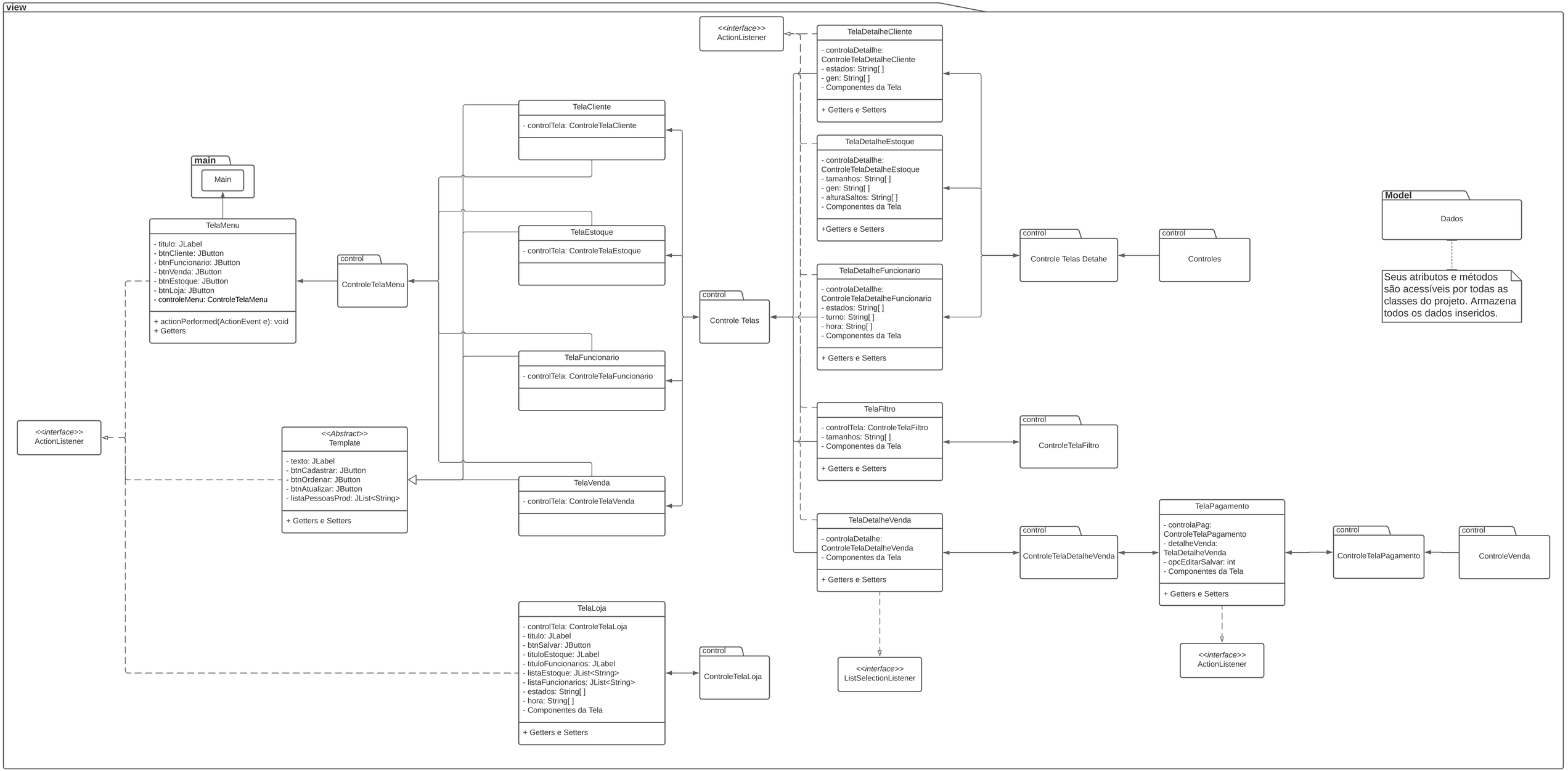
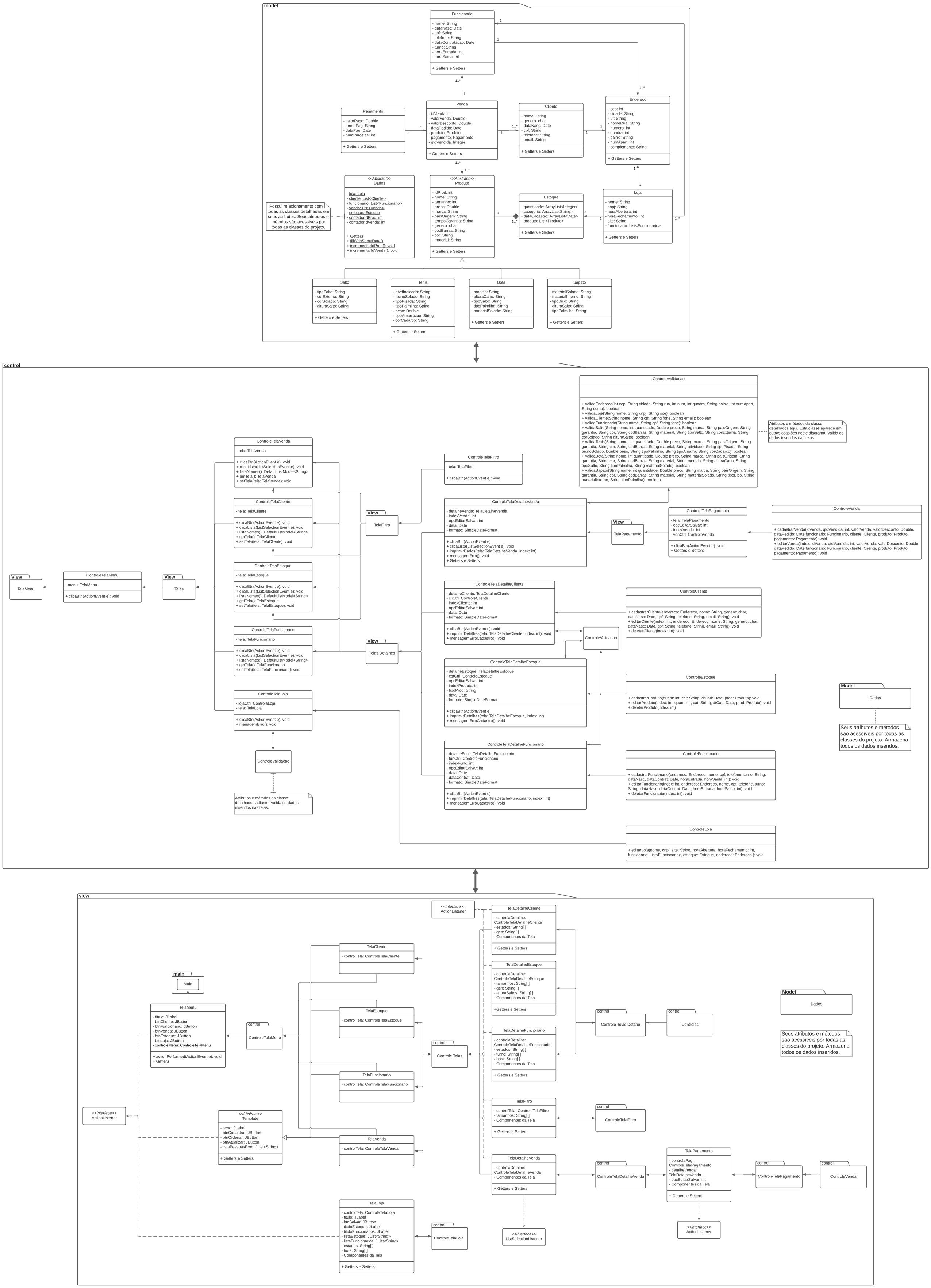

model
Funcionario
- nome: String
- dataNasc: Date
- cpf: String
- telefone: String
- dataContratacao: Date
- turno: String
- horaEntrada: int
- horaSaida: int
+ Getters e Setters
Pagamento
- valorPago: Double
- formaPag: String
- dataPag: Date
- numParcelas: int
+ Getters e Setters
Venda
- idVenda: int
- valorVenda: Double
- valorDesconto: Double
- dataPedido: Date
- produto: Produto
- pagamento: Pagamento
- qtdVendida: Integer
+ Getters e Setters
Cliente
- nome: String
- genero: char
- dataNasc: Date
- cpf: String
- telefone: String
- email: String
+ Getters e Setters
Endereco
- cep: int
- cidade: String
- uf: String
- nomeRua: String
- numero: int
- quadra: int
- bairro: String
- numApart: int
- complemento: String
+ Getters e Setters
<<Abstract>>
Dados
- loja: Loja
- cliente: List<Cliente>
- funcionario: List<Funcionario>
- venda: List<Venda>
- estoque: Estoque
- contadorIdProd: int
- contadorIdVenda: int
+ Getters
+ fillWithSomeData()
+ incrementarIdProd(): void
+ incrementarIdVenda(): void
Possui relacionamento com todas as classes detalhadas em seus atributos. Seus atributos e métodos são acessíveis por todas as classes do projeto.
<<Abstract>>
Produto
- idProd: int
- nome: String
- tamanho: int
- preco: Double
- marca: String
- paisOrigem: String
- tempoGarantia: String
- genero: char
- codBarras: String
- cor: String
- material: String
+ Getters e Setters
Estoque
- quantidade: ArrayList<Integer>
- categoria: ArrayList<String>
- dataCadastro: ArrayList<Date>
- produto: List<Produto>
+ Getters e Setters
Loja
- nome: String
- cnpj: String
- horaAbertura: int
- horaFechamento: int
- site: String
- funcionario: List<Funcionario>
+ Getters e Setters
Salto
- tipoSalto: String
- corExterna: String
- corSolado: String
- alturaSalto: String
+ Getters e Setters
Tenis
- atvdIndicada: String
- tecnoSolado: String
- tipoPisada: String
- tipoPalmilha: String
- peso: Double
- tipoAmarracao: String
- corCadarco: String
+ Getters e Setters
Bota
- modelo: String
- alturaCano: String
- tipoSalto: String
- tipoPalmilha: String
- materialSolado: String
+ Getters e Setters
Sapato
- materialSolado: String
- materialInterno: String
- tipoBico: String
- alturaSalto: String
- tipoPalmilha: String
+ Getters e Setters
1
1..*


control
ControleValidacao
+ validaEndereco(int cep, String cidade, String rua, int num, int quadra, String bairro, int numApart, String comp): boolean
+ validaLoja(String nome, String cnpj, String site): boolean
+ validaCliente(String nome, String cpf, String fone, String email): boolean
+ validaFuncionario(String nome, String cpf, String fone): boolean
+ validaSalto(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String tipoSalto, String corExterna, String corSolado, String alturaSalto): boolean
+ validaTenis(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String atividade, String tipoPisada, String tecnoSolado, Double peso, String tipoPalmilha, String tipoAmarra, String corCadarco): boolean
+ validaBota(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String modelo, String alturaCano, String tipoSalto, String tipoPalmilha, String materialSolado): boolean
+ validaSapato(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String materialSolado, String tipoBico, String materialInterno, String tipoPalmilha): boolean
Atributos e métodos da classe detalhados aqui. Esta classe aparece em outras ocasiões neste diagrama. Valida os dados inseridos nas telas.
ControleTelaVenda
- tela: TelaVenda
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaVenda
+ setTela(tela: TelaVenda): void
ControleTelaCliente
- tela: TelaCliente
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaCliente
+ setTela(tela: TelaCliente): void
ControleTelaEstoque
- tela: TelaEstoque
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaEstoque
+ setTela(tela: TelaEstoque): void
ControleTelaFuncionario
- tela: TelaFuncionario
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaFuncionario
+ setTela(tela: TelaFuncionario): void
ControleTelaLoja
- lojaCtrl: ControleLoja
- tela: TelaLoja
+ clicaBtn(ActionEvent e): void
+ menagemErro(): void
ControleValidacao
Atributos e métodos da classe detalhados adiante. Valida os dados inseridos nas telas.
View
TelaMenu
ControleTelaMenu
- menu: TelaMenu
+ clicaBtn(ActionEvent e): void
View
Telas
View
TelaFiltro
View
Telas Detalhes
ControleTelaFiltro
- tela: TelaFiltro
+ clicaBtn(ActionEvent e): void
ControleTelaDetalheVenda
- detalheVenda: TelaDetalheVenda
- indexVenda: int
- opcEditarSalvar: int
- data: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e)
+ clicaLista(ListSelectionEvent e): void
+ imprimirDados(tela: TelaDetalheVenda, index: int)
+ mensagemErro(): void
+ Getters e Setters
View
TelaPagamento
ControleTelaPagamento
- tela: TelaPagamento
- opcEditarSalvar: int
- indexVenda: int
- venCtrl: ControleVenda
+ clicaBtn(ActionEvent e): void
+ Getters e Setters
ControleVenda
+ cadastrarVenda(idVenda, qtdVendida: int, valorVenda, valorDesconto: Double, dataPedido: Date,funcionario: Funcionario, cliente: Cliente, produto: Produto, pagamento: Pagamento): void
+ editarVenda(index, idVenda, qtdVendida: int, valorVenda, valorDesconto: Double, dataPedido: Date,funcionario: Funcionario, cliente: Cliente, produto: Produto, pagamento: Pagamento): void
ControleTelaDetalheCliente
- detalheCliente: TelaDetalheCliente
- cliCtrl: ControleCliente
- indexCliente: int
- opcEditarSalvar: int
- data: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e): void
+ imprimirDetalhes(tela: TelaDetalheCliente, index: int): void
+ mensagemErroCadastro(): void
ControleCliente
+ cadastrarCliente(endereco: Endereco, nome: String, genero: char, dataNasc: Date, cpf: String, telefone: String, email: String): void
+ editarCliente(index: int, endereco: Endereco, nome: String, genero: char, dataNasc: Date, cpf: String, telefone: String, email: String): void
+ deletarCliente(index: int): void
ControleValidacao
ControleTelaDetalheEstoque
- detalheEstoque: TelaDetalheEstoque
- estCtrl: ControleEstoque
- opcEditarSalvar: int
- indexProduto: int
- tipoProd: String
- data: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e)
+ imprimirDetalhes(tela: TelaDetalheEstoque, index: int)
+ mensagemErroCadastro(): void
ControleEstoque
+ cadastrarProduto(quant: int, cat: String, dtCad: Date, prod: Produto): void
+ editarProduto(index: int, quant: int, cat: String, dtCad: Date, prod: Produto): void
+ deletarProduto(index: int)
Model
Dados
Seus atributos e métodos são acessíveis por todas as classes do projeto. Armazena todos os dados inseridos.
ControleTelaDetalheFuncionario
- detalheFunc: TelaDetalheFuncionario
- funCtrl: ControleFuncionario
- indexFunc: int
- opcEditarSalvar: int
- data: Date
- dataContrat: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e)
+ imprimirDetalhes(tela: TelaDetalheFuncionario, index: int)
+ mensagemErroCadastro(): void
ControleFuncionario
+ cadastrarFuncionario(endereco: Endereco, nome, cpf, telefone, turno: String, dataNasc, dataContrat: Date, horaEntrada, horaSaida: int): void
+ editarFuncionario(index: int, endereco: Endereco, nome, cpf, telefone, turno: String, dataNasc, dataContrat: Date, horaEntrada, horaSaida: int): void
+ deletarFuncionario(index: int): void
ControleLoja
+ editarLoja(nome, cnpj, site: String, horaAbertura, horaFechamento: int, funcionario: List<Funcionario>, estoque: Estoque, endereco: Endereco): void


view
main
Main
TelaMenu
- titulo: JLabel
- btnCliente: JButton
- btnFuncionario: JButton
- btnVenda: JButton
- btnEstoque: JButton
- btnLoja: JButton
- controleMenu: ControleTelaMenu
+ actionPerformed(ActionEvent e): void
+ Getters
control
ControleTelaMenu
<<interface>>
ActionListener
<<Abstract>>
Template
- texto: JLabel
- btnCadastrar: JButton
- btnOrdenar: JButton
- btnAtualizar: JButton
- listaPessoasProd: JList<String>
+ Getters e Setters
TelaCliente
- controlTela: ControleTelaCliente
TelaEstoque
- controlTela: ControleTelaEstoque
TelaFuncionario
- controlTela: ControleTelaFuncionario
TelaVenda
- controlTela: ControleTelaVenda
TelaLoja
- controlTela: ControleTelaLoja
- titulo: JLabel
- btnSalvar: JButton
- tituloEstoque: JLabel
- tituloFuncionarios: JLabel
- listaEstoque: JList<String>
- listaFuncionarios: JList<String>
- estados: String[ ]
- hora: String[ ]
- Componentes da Tela
+ Getters e Setters
control
ControleTelaLoja
control
Controle Telas
<<interface>>
ActionListener
TelaDetalheCliente
- controlaDetalhe: ControleTelaDetalheCliente
- estados: String[ ]
- gen: String[ ]
- Componentes da Tela
+ Getters e Setters
TelaDetalheEstoque
- controlaDetalhe: ControleTelaDetalheEstoque
- tamanhos: String[ ]
- gen: String[ ]
- alturaSaltos: String[ ]
- Componentes da Tela
+Getters e Setters
TelaDetalheFuncionario
- controlaDetalhe: ControleTelaDetalheFuncionario
- estados: String[ ]
- turno: String[ ]
- hora: String[ ]
- Componentes da Tela
+ Getters e Setters
TelaFiltro
- controlTela: ControleTelaFiltro
- tamanhos: String[ ]
- Componentes da Tela
+ Getters e Setters
TelaDetalheVenda
- controlaDetalhe: ControleTelaDetalheVenda
- Componentes da Tela
+ Getters e Setters
<<interface>>
ListSelectionListener
control
Controle Telas Detahe
control
Controles
control
ControleTelaFiltro
control
ControleTelaDetalheVenda
TelaPagamento
- controlaPag: ControleTelaPagamento
- detalheVenda: TelaDetalheVenda
- opcEditarSalvar: int
- Componentes da Tela
+ Getters e Setters
<<interface>>
ActionListener
control
ControleTelaPagamento
control
ControleVenda
Model
Dados
Seus atributos e métodos são acessíveis por todas as classes do projeto. Armazena todos os dados inseridos.


model
Funcionario
- nome: String
- dataNasc: Date
- cpf: String
- telefone: String
- dataContratacao: Date
- turno: String
- horaEntrada: int
- horaSaida: int
+ Getters e Setters
Pagamento
- valorPago: Double
- formaPag: String
- dataPag: Date
- numParcelas: int
+ Getters e Setters
Venda
- idVenda: int
- valorVenda: Double
- valorDesconto: Double
- dataPedido: Date
- produto: Produto
- pagamento: Pagamento
- qtdVendida: Integer
+ Getters e Setters
Cliente
- nome: String
- genero: char
- dataNasc: Date
- cpf: String
- telefone: String
- email: String
+ Getters e Setters
Endereco
- cep: int
- cidade: String
- uf: String
- nomeRua: String
- numero: int
- quadra: int
- bairro: String
- numApart: int
- complemento: String
+ Getters e Setters
<<Abstract>>
Dados
- loja: Loja
- cliente: List<Cliente>
- funcionario: List<Funcionario>
- venda: List<Venda>
- estoque: Estoque
- contadorIdProd: int
- contadorIdVenda: int
+ Getters
+ fillWithSomeData()
+ incrementarIdProd(): void
+ incrementarIdVenda(): void
Possui relacionamento com todas as classes detalhadas em seus atributos. Seus atributos e métodos são acessíveis por todas as classes do projeto.
<<Abstract>>
Produto
- idProd: int
- nome: String
- tamanho: int
- preco: Double
- marca: String
- paisOrigem: String
- tempoGarantia: String
- genero: char
- codBarras: String
- cor: String
- material: String
+ Getters e Setters
Estoque
- quantidade: ArrayList<Integer>
- categoria: ArrayList<String>
- dataCadastro: ArrayList<Date>
- produto: List<Produto>
+ Getters e Setters
Loja
- nome: String
- cnpj: String
- horaAbertura: int
- horaFechamento: int
- site: String
- funcionario: List<Funcionario>
+ Getters e Setters
Salto
- tipoSalto: String
- corExterna: String
- corSolado: String
- alturaSalto: String
+ Getters e Setters
Tenis
- atvdIndicada: String
- tecnoSolado: String
- tipoPisada: String
- tipoPalmilha: String
- peso: Double
- tipoAmarracao: String
- corCadarco: String
+ Getters e Setters
Bota
- modelo: String
- alturaCano: String
- tipoSalto: String
- tipoPalmilha: String
- materialSolado: String
+ Getters e Setters
Sapato
- materialSolado: String
- materialInterno: String
- tipoBico: String
- alturaSalto: String
- tipoPalmilha: String
+ Getters e Setters
control
ControleValidacao
+ validaEndereco(int cep, String cidade, String rua, int num, int quadra, String bairro, int numApart, String comp): boolean
+ validaLoja(String nome, String cnpj, String site): boolean
+ validaCliente(String nome, String cpf, String fone, String email): boolean
+ validaFuncionario(String nome, String cpf, String fone): boolean
+ validaSalto(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String tipoSalto, String corExterna, String corSolado, String alturaSalto): boolean
+ validaTenis(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String atividade, String tipoPisada, String tecnoSolado, Double peso, String tipoPalmilha, String tipoAmarra, String corCadarco): boolean
+ validaBota(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String modelo, String alturaCano, String tipoSalto, String tipoPalmilha, String materialSolado): boolean
+ validaSapato(String nome, int quantidade, Double preco, String marca, String paisOrigem, String garantia, String cor, String codBarras, String material, String materialSolado, String tipoBico, String materialInterno, String tipoPalmilha): boolean
Atributos e métodos da classe detalhados aqui. Esta classe aparece em outras ocasiões neste diagrama. Valida os dados inseridos nas telas.
ControleTelaVenda
- tela: TelaVenda
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaVenda
+ setTela(tela: TelaVenda): void
ControleTelaFiltro
- tela: TelaFiltro
+ clicaBtn(ActionEvent e): void
ControleTelaCliente
- tela: TelaCliente
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaCliente
+ setTela(tela: TelaCliente): void
View
TelaFiltro
ControleTelaDetalheVenda
- detalheVenda: TelaDetalheVenda
- indexVenda: int
- opcEditarSalvar: int
- data: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e)
+ clicaLista(ListSelectionEvent e): void
+ imprimirDados(tela: TelaDetalheVenda, index: int)
+ mensagemErro(): void
+ Getters e Setters
View
TelaPagamento
ControleTelaPagamento
- tela: TelaPagamento
- opcEditarSalvar: int
- indexVenda: int
- venCtrl: ControleVenda
+ clicaBtn(ActionEvent e): void
+ Getters e Setters
ControleVenda
+ cadastrarVenda(idVenda, qtdVendida: int, valorVenda, valorDesconto: Double, dataPedido: Date,funcionario: Funcionario, cliente: Cliente, produto: Produto, pagamento: Pagamento): void
+ editarVenda(index, idVenda, qtdVendida: int, valorVenda, valorDesconto: Double, dataPedido: Date,funcionario: Funcionario, cliente: Cliente, produto: Produto, pagamento: Pagamento): void
View
TelaMenu
ControleTelaMenu
- menu: TelaMenu
+ clicaBtn(ActionEvent e): void
View
Telas
ControleTelaEstoque
- tela: TelaEstoque
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaEstoque
+ setTela(tela: TelaEstoque): void
ControleTelaDetalheCliente
- detalheCliente: TelaDetalheCliente
- cliCtrl: ControleCliente
- indexCliente: int
- opcEditarSalvar: int
- data: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e): void
+ imprimirDetalhes(tela: TelaDetalheCliente, index: int): void
+ mensagemErroCadastro(): void
ControleCliente
+ cadastrarCliente(endereco: Endereco, nome: String, genero: char, dataNasc: Date, cpf: String, telefone: String, email: String): void
+ editarCliente(index: int, endereco: Endereco, nome: String, genero: char, dataNasc: Date, cpf: String, telefone: String, email: String): void
+ deletarCliente(index: int): void
ControleTelaFuncionario
- tela: TelaFuncionario
+ clicaBtn(ActionEvent e): void
+ clicaLista(ListSelectionEvent e): void
+ listaNomes(): DefaultListModel<String>
+ getTela(): TelaFuncionario
+ setTela(tela: TelaFuncionario): void
View
Telas Detalhes
ControleValidacao
ControleTelaDetalheEstoque
- detalheEstoque: TelaDetalheEstoque
- estCtrl: ControleEstoque
- opcEditarSalvar: int
- indexProduto: int
- tipoProd: String
- data: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e)
+ imprimirDetalhes(tela: TelaDetalheEstoque, index: int)
+ mensagemErroCadastro(): void
ControleEstoque
+ cadastrarProduto(quant: int, cat: String, dtCad: Date, prod: Produto): void
+ editarProduto(index: int, quant: int, cat: String, dtCad: Date, prod: Produto): void
+ deletarProduto(index: int)
Model
Dados
Seus atributos e métodos são acessíveis por todas as classes do projeto. Armazena todos os dados inseridos.
ControleTelaLoja
- lojaCtrl: ControleLoja
- tela: TelaLoja
+ clicaBtn(ActionEvent e): void
+ menagemErro(): void
ControleValidacao
Atributos e métodos da classe detalhados adiante. Valida os dados inseridos nas telas.
ControleTelaDetalheFuncionario
- detalheFunc: TelaDetalheFuncionario
- funCtrl: ControleFuncionario
- indexFunc: int
- opcEditarSalvar: int
- data: Date
- dataContrat: Date
- formato: SimpleDateFormat
+ clicaBtn(ActionEvent e)
+ imprimirDetalhes(tela: TelaDetalheFuncionario, index: int)
+ mensagemErroCadastro(): void
ControleFuncionario
+ cadastrarFuncionario(endereco: Endereco, nome, cpf, telefone, turno: String, dataNasc, dataContrat: Date, horaEntrada, horaSaida: int): void
+ editarFuncionario(index: int, endereco: Endereco, nome, cpf, telefone, turno: String, dataNasc, dataContrat: Date, horaEntrada, horaSaida: int): void
+ deletarFuncionario(index: int): void
ControleLoja
+ editarLoja(nome, cnpj, site: String, horaAbertura, horaFechamento: int, funcionario: List<Funcionario>, estoque: Estoque, endereco: Endereco): void
view
<<interface>>
ActionListener
TelaDetalheCliente
- controlaDetalhe: ControleTelaDetalheCliente
- estados: String[ ]
- gen: String[ ]
- Componentes da Tela
+ Getters e Setters
TelaCliente
- controlTela: ControleTelaCliente
main
Main
TelaDetalheEstoque
- controlaDetalhe: ControleTelaDetalheEstoque
- tamanhos: String[ ]
- gen: String[ ]
- alturaSaltos: String[ ]
- Componentes da Tela
+Getters e Setters
TelaMenu
- titulo: JLabel
- btnCliente: JButton
- btnFuncionario: JButton
- btnVenda: JButton
- btnEstoque: JButton
- btnLoja: JButton
- controleMenu: ControleTelaMenu
+ actionPerformed(ActionEvent e): void
+ Getters
control
ControleTelaMenu
TelaEstoque
- controlTela: ControleTelaEstoque
control
Controle Telas
control
Controle Telas Detahe
control
Controles
Model
Dados
Seus atributos e métodos são acessíveis por todas as classes do projeto. Armazena todos os dados inseridos.
TelaDetalheFuncionario
- controlaDetalhe: ControleTelaDetalheFuncionario
- estados: String[ ]
- turno: String[ ]
- hora: String[ ]
- Componentes da Tela
+ Getters e Setters
TelaFuncionario
- controlTela: ControleTelaFuncionario
<<interface>>
ActionListener
<<Abstract>>
Template
- texto: JLabel
- btnCadastrar: JButton
- btnOrdenar: JButton
- btnAtualizar: JButton
- listaPessoasProd: JList<String>
+ Getters e Setters
TelaFiltro
- controlTela: ControleTelaFiltro
- tamanhos: String[ ]
- Componentes da Tela
+ Getters e Setters
control
ControleTelaFiltro
TelaVenda
- controlTela: ControleTelaVenda
TelaDetalheVenda
- controlaDetalhe: ControleTelaDetalheVenda
- Componentes da Tela
+ Getters e Setters
control
ControleTelaDetalheVenda
TelaPagamento
- controlaPag: ControleTelaPagamento
- detalheVenda: TelaDetalheVenda
- opcEditarSalvar: int
- Componentes da Tela
+ Getters e Setters
control
ControleTelaPagamento
control
ControleVenda
TelaLoja
- controlTela: ControleTelaLoja
- titulo: JLabel
- btnSalvar: JButton
- tituloEstoque: JLabel
- tituloFuncionarios: JLabel
- listaEstoque: JList<String>
- listaFuncionarios: JList<String>
- estados: String[ ]
- hora: String[ ]
- Componentes da Tela
+ Getters e Setters
control
ControleTelaLoja
<<interface>>
ListSelectionListener
<<interface>>
ActionListener