Model::Main

model

Usuario
-id: Integer
-login: String
-password: String
-cpf: String
-nome: String
-telefone: String
-email: String
-cadastrado: LocalDateTime
-editado: LocalDateTime

«enumeration»
TipoUsuario
ADMIN
MEDICO
ATENDENTE

-tipoUsuario

Consulta
-id: Integer
-descricao: String
-dataAgendamento: LocalDateTime
-dataFim: LocalDateTime

+isConsultaRealizada(): boolean
+isValidForRegistroBD(): boolean

Paciente
-id: String
-cpf: String
-nome: String
-telefone: String
-cidade: String
-estado: String
-numero: String
-cadastrado: LocalDateTime
-editado: LocalDateTime

Medico
-crm: String

-medico -consultas
1   1..*

-consultas -paciente
1..*

persistence

PacienteDao
-getConnection(): Connection
+create(): boolean
+existsByCpf(cpf: String): boolean
+extractPacienteByResultSet(resultSet: ResultSet): Paciente
+findAll(): List<Paciente>
+delete(id: Integer)
+findByNomeOrCpf(pacienteSearchText: String)

«interface»
GenericDao
+create(data: T): boolean
+delete(id: Integer): boolean

«abstract»
AbstractGenericDao
#getConnection(): Connection

controller

PacienteController
-pacienteDao: PacienteDao
+cadastrarPaciente(paciente: Paciente): boolean
+listarPacientes(): List<Paciente>
+excluirPaciente(id: Integer): boolean
+buscarPacientesByStringSearch(pacienteSearchText: String): List<Paciente>

exception

PacienteJaExisteException

view

dialog

JDialogGeneric
+abrirDialog(): void
+mostrarMensagemCpfInvalido(): void
+mostrarMensagemLoginInvalido(): void
+mostrarMensagemEmailInvalido(): void
+mostrarMensagemErroInesperado(): void

JDialogCadastroPaciente
-JCidade: JLabel
-JEstado: JLabel
-JNumero: JLabel
-JCPF: JLabel
-CadastroPacienteButton: JButton
-CpfTextField: JFormattedTextField
-jLabel7: JLabel
-jNome: JLabel
-jPanel1: JPanel
-jTelefone: JLabel
-TextFieldCidade: JTextField
-TextFieldEstado: JTextField
-TextFieldNome: JTextField
-TextFieldNumero: JTextField
-TextFieldTelefone: JFormattedTextField
+main(args: String): void
-initComponents(): void
-CadastroPacienteButtonActionPerformed(evt: ActionEvent): void

home

JHome
-medicoController: MedicoController
-usuarioController: UsuarioController
-consultaController: ConsultaController
-jBuscaCadastroButton: JButton
-jCadastroAtendente: JMenuItem
-jCadastroConsulta: JMenuItem
-jCadastroMedico: JMenuItem
-jCadastroPaciente: JMenuItem
-jListarRowButton: JButton
-jFiltroConsultaBox: JComboBox<String>
-jFinalizaSessaoButton: JButton
-jFinalizarConsultaButton: JButton
-jLabel1: JLabel
-jLabel2: JLabel
-jLabel3: JLabel
-jLabel4: JLabel
-jLabel5: JLabel
-jListaVendentesButton: JButton
-jListaMedicosButton: JButton
-jListaPacientesButton: JButton
-jListaConsultasButton: JButton
-jMenuBar1: JMenuBar
-jMenuCadastro: JMenu
-jScrollPane1: JScrollPane
-jSearchTextField: JTextField
-jTabelaListagens: JTable
-pacienteController: PacienteController
-initComponents(): void
-CadastroPacienteActionPerformed(evt: ActionEvent): void
-jListarConsultasButtonActionPerformed(evt: ActionEvent): void
-atualizarTabelaComListaDeConsultas(): void
-atualizarTabelaComListaDePacientes(): void
-atualizarTabelaComListaDeAtendentes(): void
-atualizarTabelaComListaDeMedicos(): void
-CadastroMedicoActionPerformed(evt: ActionEvent): void
-CadastroAtendenteActionPerformed(evt: ActionEvent): void
-CadastroConsultaButtonActionPerformed(evt: ActionEvent): void
-jListaPacientesButtonActionPerformed(evt: ActionEvent): void
-jListaVendentesButtonActionPerformed(evt: ActionEvent): void
-jListaMedicosButtonActionPerformed(evt: ActionEvent): void
-jBuscaCadastroButtonActionPerformed(evt: ActionEvent): void
-jFinalizarConsultaButtonActionPerformed(evt: ActionEvent): void
-jFinalizaSessaoButtonActionPerformed(evt: ActionEvent): void
-jExcluirButtonActionPerformed(evt: ActionEvent): void
+abrirDialog(): void
+main(args: String): void

table

PacienteTableModel
-colunas: String[] {readOnly}
-pacientes: List<Paciente>
+getRowCount(): int
+getColumnCount(): int
+getEntityId(rowIndex: int): Integer
+getColumnName(column: int): String
+getPaciente(rowIndex: int): Paciente
+getValueAt(rowIndex: int, columnIndex: int): Object

util

CpfUtil
+isValidCpf(cpf: String): boolean

DateUtil
+formatarPadraoData(store: DateTimeFormatter)
+formatarParaSqlDateTime(localDateTime): String
+parseDateAndHourToLocalDateTime(dateTime: String): LocalDateTime