

Dicionário de Dados

Sigla e Nome do Projeto: **Psicool – Sistema de Gestão de Clínicas de Psicologia**

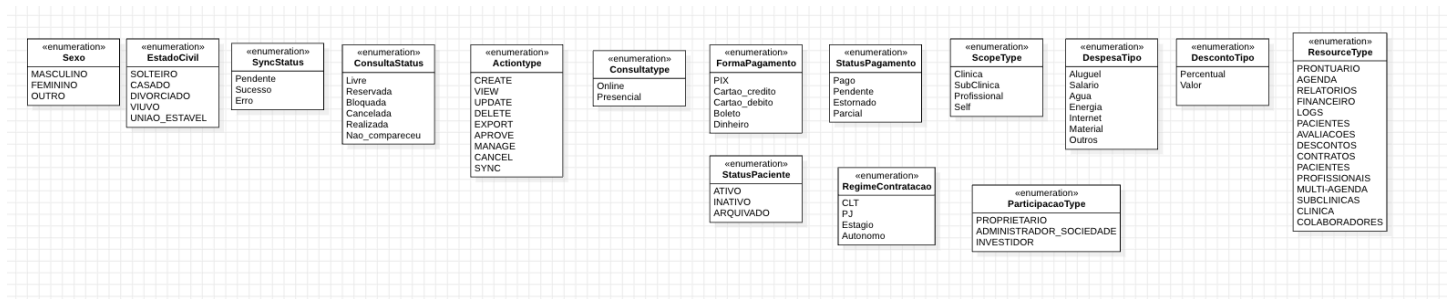
Introdução

Este documento apresenta o dicionário de dados do sistema Psicool, descrevendo de forma padronizada os tipos de entidade, seus atributos e restrições de integridade. A modelagem conceitual foi realizada a partir do Diagrama de Classes UML, utilizado como diagrama conceitual de dados. Para cada entidade, são especificados o mnemônico e o nome do atributo, o tipo de dado e tamanho, a cardinalidade mínima e máxima, bem como observações e restrições de negócio associadas.

Diagrama Conceitual de Dados

O diagrama conceitual de dados do sistema Psicool é representado pelo Diagrama de Classes abaixo.

Segue os Enums criados até então.



Entidade: Anotacoes

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
data_ant: data da anotação	Data	1,1	Data em que a anotação foi registrada.
anotacao: texto da anotação	Char(*)	0,*	Conteúdo textual; multivalorado, pode haver várias anotações.

Entidade: AgendaIndividual

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
ID: identificador da agenda individual	Char(36)	1,1	UUID do registro de agenda.
startAt: início da consulta	DataHora	1,1	Data e hora de início do atendimento.
duracaominutos: duração em minutos	Int(3)	1,1	Duração da consulta em minutos.
status: status da consulta	Enum ConsultaType	1,1	Domínio ConsultaType definido no modelo.
observacoes: observações	Char(*)	0,*	Notas adicionais; multivalorado.

Entidade: Agenda-multi

Classe de apoio para agrupamento de agendas, **sem atributos** definidos no diagrama.

Entidade: Clinica

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
CNPJ: CNPJ da clínica	Char(14)	1,1	{readOnly}. Chave de negócio.
nome_fantasia: nome fantasia	Char(*)	1,1	Nome fantasia da clínica.
razao_social: razão social	Char(*)	1,1	Razão social registrada.
inscricao_estadual: inscrição estadual	Char(*)	0,1	Opcional.
email: e-mail	Char(*)	1,*	Pelo menos um e-mail cadastrado.
telefone: telefone	Char(*)	1,*	Pelo menos um telefone cadastrado.
proprietario: sócios/proprietários	Composto ClinicoSocio	0,*	Lista de vínculos ClinicoSocio associados à clínica.

Entidade: ClinicoSocio

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
ID: identificador do vínculo de sócio	Char(36)	1,1	UUID.
tipoParticipacao: tipo de participação	Enum ParticipationType	1,1	Domínio ParticipationType.
percentualParticipacao: percentual	Decimal(5,2)	0,1	Percentual de participação societária.
desde: data de início da participação	Data	1,1	Início do vínculo.
ate: data de término da participação	Data	0,1	Opcional; preenchido na saída do sócio.
representanteLegal: é representante legal?	Booleano	1,1	Indica se o sócio é representante legal da clínica.

Entidade: Colaborador

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
salario: salário do colaborador	Decimal(10,2)	1,1	Valor do salário base.
matricula: matrícula interna	Char(*)	1,1	Identificador interno; regra de negócio de unicidade.
dataAdmissao: data de admissão	Data	1,1	Início do vínculo.
dataDesligamento: data de desligamento	Data	0,1	Preenchido quando houver desligamento.
regimeContrato: regime de contratação	Enum RegimeContratacao	1,1	Domínio RegimeContratacao (CLT, PJ etc.).
cargaHoraria: carga horária contratada	Int(3)	1,1	Ex.: horas semanais.
PercentualComissao: percentual de comissão	Decimal(5,2)	0,1	Opcional; percentual de comissão sobre serviços.
banco: banco	Char(*)	0,*	Multivalorado. Pode haver vários bancos cadastrados.
agencia: agência bancária	Char(*)	0,*	Multivalorado; normalmente associado a cada banco.
conta: número da conta	Char(*)	0,*	Multivalorado; associado a banco/agência.
chavePix: chave PIX	Char(*)	0,*	Uma ou mais chaves PIX cadastradas.
ativo: colaborador ativo?	Booleano	1,1	true = ativo, false = inativo/desligado.

Entidade: Endereco

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
CEP: código de endereçamento	Char(8)	1,1	CEP obrigatório.
rua: logradouro	Char(*)	1,1	Rua, avenida etc.
complemento: complemento	Char(*)	0,1	Opcional.
bairro: bairro	Char(*)	0,1	Opcional.
numero: número do imóvel	Char(10)	1,1	Número do endereço.

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
cidade: cidade	Char(*)	1,1	Cidade.
estado: estado	Char(2)	1,1	UF / estado.
pais: país	Char(*)	1,1	Valor padrão “Brasil”, conforme diagrama.

Entidade: Paciente

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
convenio: convênio	Char(*)	0,1	Nome/código do convênio.
numeroconvenio: número do convênio	Char(*)	1,1	Número da carteirinha.
validadeconvenio: validade do convênio	Data	1,1	Data de validade.
datacadastro: data de cadastro	Data	1,1	Data em que o paciente foi cadastrado.
lembreteWhatsapp: recebe via WhatsApp	Booleano	1,1	Indica se aceita lembretes por WhatsApp.
lembreEmail: recebe via e-mail	Booleano	1,1	Indica se aceita lembretes por e-mail.
statusPaciente: status do paciente	Enum StatusPaciente	1,1	Domínio StatusPaciente (ATIVO, INATIVO, ARQUIVADO).

Entidade: Permissao

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
ID: identificador da permissão	Char(36)	1,1	UUID da permissão.
resource: recurso protegido	Enum ResourceType	1,1	Domínio ResourceType (PRONTUARIO, AGENDA, PACIENTES, etc.).
action: tipo de ação	Enum Actiontype	1,1	Domínio Actiontype (CREATE, VIEW, UPDATE, DELETE, etc.).

Entidade: Pessoa

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
nome: nome completo	Char(*)	1,1	{readOnly} no modelo conceitual. Obrigatório.
nomesocial: nome social	Char(*)	0,1	Opcional.
CPF: número de CPF	Char(11)	1,1	{readOnly}. Deve ser único no sistema (chave de negócio).

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
data_nasc: data de nascimento	Data	1,1	{readOnly}.
telefone: telefone(s) de contato	Char(*)	1,*	Pelo menos um telefone. Atributo multivalorado.
email: e-mail principal	Char(*)	1,1	Formato de e-mail válido; usualmente único na prática.
sexo: sexo/gênero	Enum Sexo	1,1	Domínio enumerado Sexo (valores definidos na enum).
estadocivil: estado civil	Enum EstadoCivil	1,1	Domínio enumerado EstadoCivil.
profissao: profissão	Char(*)	0,1	Descrição da profissão declarada.

Entidade: Prontuario

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
ID: identificador do prontuário	Char(36)	1,1	UUID. {readOnly}.
data_criacao: data de criação	Data	1,1	{readOnly}. Data de abertura do prontuário.
data_att: data de atualização	Data	1,1	Data da última atualização.
responsavel: usuário responsável	Composto Usuario	1,1	Referência ao usuário responsável.
paciente: paciente dono do prontuário	Composto Paciente	1,1	{readOnly}. Paciente ao qual o prontuário pertence.
versao: versão do prontuário	Int(3)	1,1	Número de versão.
consentimento: texto de consentimento	Char(*)	0,1	Texto opcional de consentimento.
bloqueado: indicador de bloqueio	Booleano	1,1	Indica se o prontuário está bloqueado para edição.
tags: tags	Char(*)	0,1	Lista textual de tags/palavras-chave.

Entidade: Psicologo

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
CRP: registro profissional	Char(*)	1,1	Número do CRP.
especialidade: especialidade do psicólogo	Char(*)	1,1	Área de especialização.
CRP_UF: UF do CRP	Char(2)	1,1	Unidade federativa do registro.
atendeOnline: atende on-line?	Booleano	1,1	Indica se oferece atendimentos on-line.
atendePresencial: atende presencialmente?	Booleano	1,1	Indica se oferece atendimentos presenciais.
valorconsulta: valor da consulta	Decimal(10,2)	1,1	Valor padrão de consulta.
duracaopadrao_min: duração padrão	Int(3)	1,1	Duração típica da sessão, em minutos.

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
(min)			
intervalo_entre_cons_min: intervalo (min)	Int(3)	1,1	Intervalo mínimo entre consultas.
limitesconsulta_dia: limite consultas/dia	Int(3)	1,1	Máximo de consultas por dia.
aceitaconvenio: aceita convênio?	Booleano	1,1	Indica se atende via convênio.
texto_apresentacao: texto de apresentação	Char(*)	0,1	Texto opcional de apresentação/biografia.

Entidade: Role

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
name: nome do papel	Char(*)	1,1	Nome do papel (ADMIN, PSICOLOGO etc.).
scopetype: tipo de escopo	Enum ScopeType	1,1	Domínio ScopeType.

Entidade: RolePermissao

Classe associativa entre Role e Permissao **sem atributos próprios** no diagrama.

Entidade: RoleUsuario

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
ID: identificador do vínculo	Char(36)	1,1	UUID do registro RoleUsuario.
startAt: início da vigência	DataHora	1,1	Início de validade do papel.
endAt: fim da vigência	DataHora	0,1	Opcional; nulo indica vigência aberta.
scopetype: tipo de escopo	Enum ScopeType	1,1	Domínio ScopeType (Clinica, SubClinica, Profissional, Self).
scopeRefID: identificador de escopo	Char(36)	1,1	UUID da entidade à qual o escopo se refere.

Entidade: SolicitacaoAcessoProntuario

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
ID: identificador da solicitação	Char(36)	1,1	UUID. {readOnly}.
solicitante: paciente solicitante	Composto Paciente	1,1	{readOnly}. Paciente que solicita o acesso.
prontuario: prontuário solicitado	Composto Prontuario	1,1	{readOnly}. Prontuário alvo da solicitação.
status: status da solicitação	Char(*)	1,1	Status textual (PENDENTE, APROVADO,

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
			NEGADO etc.).
motivo: motivo da solicitação	Char(*)	0,*	Campo multivalorado – pode registrar vários motivos.
aprovadopor: usuário que aprovou	Composto Usuario	1,1	Usuário responsável pela aprovação.
aprovadoem: data de aprovação	Data	1,1	Data da aprovação.
obs: observações	Char(*)	0,*	Campo multivalorado para observações adicionais.

Entidade: SubClinica

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
CNPJ: CNPJ da subclínica	Char(14)	1,1	{readOnly}. Chave de negócio.
nome_fantasia: nome fantasia	Char(*)	1,1	Nome fantasia da subclínica.
razao_social: razão social	Char(*)	1,1	Razão social registrada.
inscricao_estadual: inscrição estadual	Char(*)	0,1	Opcional.
email: e-mail	Char(*)	1,*	Pelo menos um e-mail cadastrado.
telefone: telefone	Char(*)	1,*	Pelo menos um telefone cadastrado.
gerente: colaborador gerente	Composto Colaborador	1,*	Um ou mais colaboradores associados como gerente(s).

Entidade: Usuario

Mnemônico e nome do atributo	Tipo (tam)	Card	Restrições e Observações
ID: identificador do usuário	Char(36)	1,1	UUID.
username: nome de usuário	Char(*)	1,1	Único no sistema.
passwordHash: hash da senha	Char(*)	1,1	Armazena apenas o hash da senha.
isActive: usuário ativo?	Booleano	1,1	true = ativo; false = inativo/bloqueado.
lastlogin: data/hora último login	DataHora	1,1	Data/hora do último login.

Relacionamentos entre as classes

Abaixo, uma visão textual de se não **todos**, **a maioria dos relacionamentos** do diagrama, com cardinalidades e quem se relaciona com quem.

Herança

1. Paciente **é** uma Pessoa
 - Especialização de Pessoa.
 - Todo Paciente herda os atributos de Pessoa.
2. Colaborador **é** uma Pessoa
 - Especialização de Pessoa.
 - Todo Colaborador herda os atributos de Pessoa.
3. Psicologo **é** um Colaborador
 - Especialização de Colaborador.
 - Todo Psicologo herda atributos de Colaborador e, indiretamente, de Pessoa.

Segurança e controle de acesso

- 1 . Pessoa — Usuario
 - Cardinalidade: Pessoa 1 — 1 Usuario
 - Interpretação: cada Pessoa pode ter, no máximo, **uma** conta de usuário; cada Usuario está associado a uma Pessoa.
- 2 . Usuario — RoleUsuario
 - Cardinalidade: Usuario 1 — 0..* RoleUsuario
 - Cada Usuario pode ter vários vínculos RoleUsuario (papéis em diferentes escopos).
 - Cada RoleUsuario referencia um único Usuario.
- 3 . Role — RoleUsuario
 - Cardinalidade: Role 1 — 0..* RoleUsuario
 - Cada Role pode estar associado a muitos RoleUsuario.
 - Cada RoleUsuario referencia exatamente **um** Role.
- 4 . Role — RolePermissao
 - Cardinalidade: Role 1 — 0..* RolePermissao
 - Cada Role pode agregar diversas permissões via RolePermissao.
 - Cada instância de RolePermissao está ligada a **um** Role.
- 5 . Permissao — RolePermissao
 - Cardinalidade: Permissao 1 — 0..* RolePermissao
 - Cada Permissao pode ser atribuída a vários Role por meio de RolePermissao.
 - Cada RolePermissao está ligada a uma Permissao.

3. Paciente, Colaborador, Psicólogo e prontuário

1 . Psicólogo — Paciente

- Cardinalidade: Colaborador 1 — 0..* Paciente
- Interpretação: um Colaborador (psicólogo) pode ser responsável por **zero ou vários** pacientes; cada Paciente está associado a um psicólogo.

2 . Paciente — Prontuario

- Cardinalidade: Paciente 1 — 1 Prontuario
- Interpretação: cada Paciente possui um único Prontuario e cada Prontuario pertence a um **único** Paciente.

3 . Paciente — Anotacoes

- Cardinalidade: Paciente 1 — 0..* Anotacoes
- Um paciente pode ter diversas anotações; cada Anotacoes está ligada a um único paciente.

4 . Paciente — SolicitacaoAcessoProntuario

- Cardinalidade: Paciente 1 — 0..* SolicitacaoAcessoProntuario
- Um Paciente pode fazer várias solicitações de acesso ao próprio prontuário; cada solicitação tem um paciente solicitante.

5 . Prontuario — SolicitacaoAcessoProntuario

- Cardinalidade: Prontuario 1 — 0..* SolicitacaoAcessoProntuario
- Um Prontuario pode aparecer em várias solicitações de acesso; cada solicitação refere-se a **um** prontuário.

4. Agenda e solicitações de agendamento

1 . Psicologo — AgendaIndividual

- Cardinalidade: Psicologo 1 — 0..* AgendaIndividual
- Cada psicólogo possui zero ou mais itens de agenda individual; cada item de AgendaIndividual pertence a um psicólogo.

2 . AgendaIndividual — Agenda-multi

- Cardinalidade: AgendaIndividual * — 1 Agenda-multi (não definido do diagrama).
- Representa o vínculo entre um horário individual e uma “agenda múltipla” agregadora.

3 . Paciente — SolicitacaodeAgendamento

- Cardinalidade: Paciente 1 — 0..* SolicitacaoAgendamento
- Um paciente pode gerar diversas solicitações de agendamento; cada solicitação pertence a um paciente.

4 . SolicitacaoAgendamento — AgendaIndividual

- Cardinalidade: AgendaIndividual 1 — 0..*
- Uma mesma entrada de agenda pode estar associada a várias solicitações (por exemplo, fila de espera ou múltiplos interessados); cada solicitação aponta para uma AgendaIndividual.

5. Clínica, subclínica, endereço e sócios

1 . Clinica — Endereco

- Cardinalidade: Clinica 1 — 1 Endereco
- A clínica principal possui um endereço.

2 . SubClinica — Endereco

- Cardinalidade: SubClinica 1 — 1 Endereco
- Cada SubClinica também possui exatamente um endereço próprio.

3 . Clinica — SubClinica

- Cardinalidade: Clinica 1 — 0..* SubClinica
- Uma clínica pode ter nenhuma ou várias subclínicas; cada SubClinica pertence a exatamente uma Clinica.

4 . Clinica — ClinicoSocio

- Cardinalidade: Clinica 1 — 0..* ClinicoSocio (no lado de ClinicoSocio aparece *).
- Cada clínica tem um ou mais sócios (proprietario: ClinicoSocio[*]); cada ClinicoSocio está vinculado a uma clínica.

// Algumas melhorias como a relacao entre um colaborador com papel de gerente relacionado a subclinica, implementar multi-agenda e a agenda completa do psicologo que é um conjunto composto de agendaindividual. Implementar os UUID em algumas entidades, implementar a entidade de avaliação...

3. Métodos das Classes (Get/Set + Regras de Negócio)

3.1 Entidade: Pessoa

Getters/Setters básicos

- getNome(), setNome(nome)
- getNomeSocial(), setNomeSocial(nomeSocial)
- getCPF()
- getDataNasc()
- getTelefone(), setTelefone(listaTelefones)
- getEmail(), setEmail(email)
- getSexo(), setSexo(sexo)
- getEstadoCivil(), setEstadoCivil(estadoCivil)
- getProfissao(), setProfissao(profissao)

Métodos de negócio

- atualizarContato(telefones[], email)
Atualiza telefones e e-mail principal, garantindo pelo menos um telefone.
- atualizarDadosPessoais(nomeSocial, sexo, estadoCivil, profissao)
Atualiza em conjunto os principais dados pessoais configuráveis.
- calcularIdade(dataReferencia)
Calcula e retorna a idade em anos completos com base em data_nasc.
- isMaiorDeIdade(dataReferencia)
Retorna verdadeiro se a idade for ≥ 18 anos na data de referência.

3.2 Entidade: Paciente

Getters/Setters básicos

- getConvenio(), setConvenio(convenio)
- getNumeroConvenio(), setNumeroConvenio(numero)
- getValidadeConvenio(), setValidadeConvenio(dataValidade)
- getDataCadastro(), setDataCadastro(dataCadastro)
- getLembreteWhatsapp(), setLembreteWhatsapp(flag)
- getLembreEmail(), setLembreEmail(flag)
- getStatusPaciente(), setStatusPaciente(status)

Métodos de negócio

- registrarCadastro(dataCadastro)
Define a data em que o paciente foi cadastrado no sistema.
- configurarConvenio(convenio, numeroConvenio, validade)
Cadastra ou atualiza os dados do convênio.
- cancelarConvenio()
Remove as informações de convênio, marcando o paciente como sem convênio.

- configurarLembretes(lembrereWhatsapp, lembrereEmail)
Atualiza as preferências de recebimento de lembretes.
- ativarPaciente() / inativarPaciente() / arquivarPaciente()
Ajustam o statusPaciente para ATIVO, INATIVO ou ARQUIVADO.
- aceitaAlgumLembrete()
Retorna verdadeiro se o paciente aceitar ao menos um tipo de lembrete.

3.3 Entidade: Colaborador

Getters/Setters básicos

- getSalario(), setSalario(salario)
- getMatricula(), setMatricula(matricula)
- getDataAdmissao(), setDataAdmissao(data)
- getDataDesligamento(), setDataDesligamento(data)
- getRegimeContrato(), setRegimeContrato(regime)
- getCargaHoraria(), setCargaHoraria(horas)
- getPercentualComissao(), setPercentualComissao(percentual)
- getBanco(), setBanco(listaBancos)
- getAgencia(), setAgencia(listaAgencias)
- getConta(), setConta(listaContas)
- getChavePix(), setChavePix(listaChaves)
- isAtivo(), setAtivo(flag)

Métodos de negócio

- registrarAdmissao(dataAdmissao, salario, regimeContrato, cargaHoraria)
Configura os dados iniciais do vínculo.
- registrarDesligamento(dataDesligamento)
Informa o desligamento do colaborador.
- atualizarRemuneracao(novoSalario, novoPercentualComissao)
Atualiza salário base e percentual de comissão.
- adicionarContaBancaria(banco, agencia, conta, chavePix)
Adiciona um conjunto de dados bancários.
- limparDadosBancarios()
Remove todos os dados bancários associados.
- ativarColaborador() / inativarColaborador()
Controla o flag ativo conforme o status do colaborador.

3.4 Entidade: Psicologo

Getters/Setters básicos

- getCRP(), setCRP(crp)
- getCRP_UF(), setCRP_UF(uf)

- getEspecialidade(), setEspecialidade(especialidade)
- getAtendeOnline(), setAtendeOnline(flag)
- getAtendePresencial(), setAtendePresencial(flag)
- getValorConsulta(), setValorConsulta(valor)
- getDuracaoPadraoMin(), setDuracaoPadraoMin(minutos)
- getIntervaloEntreConsMin(), setIntervaloEntreConsMin(minutos)
- getLimitesConsultaDia(), setLimitesConsultaDia(limite)
- getAceitaConvenio(), setAceitaConvenio(flag)
- getTextoApresentacao(), setTextoApresentacao(texto)

Métodos de negócio

- configurarRegistroProfissional(crp, crpUf)
Define o CRP e a UF de registro.
- configurarModalidadesAtendimento(atendeOnline, atendePresencial)
Define em quais modalidades o psicólogo atende.
- configurarPoliticaConsultas(valorConsulta, duracaoPadrao, intervaloEntreConsultas, limiteConsultasDia)
Centraliza a configuração dos parâmetros padrão de consulta.
- definirAceiteConvenio(aceitaConvenio)
Indica se o profissional aceita convênios.
- atualizarApresentacao(textoApresentacao)
Atualiza o texto público de apresentação/biografia.

3.5 Entidade: Clinica

Getters/Setters básicos

- getCNPJ()
- getNomeFantasia(), setNomeFantasia(nome)
- getRazaoSocial(), setRazaoSocial(razao)
- getInscricaoEstadual(), setInscricaoEstadual(inscricao)
- getEmail(), setEmail(listaEmails)
- getTelefone(), setTelefone(listaTelefones)
- getProprietarios(), setProprietarios(listaClinicoSocio)

Métodos de negócio

- atualizarDadosCadastrais(nomeFantasia, razaoSocial, inscricaoEstadual)
Atualiza os dados principais da clínica (exceto CNPJ).
- atualizarContatos(emails[], telefones[])
Atualiza lista de e-mails e telefones, garantindo pelo menos um de cada.
- adicionarSocio(clinicoSocio) / removerSocio(clinicoSocio)
Gerencia os vínculos de sócios (entidade ClinicoSocio).
- definirEndereco(endereco)
Associa um Endereco à clínica.

3.6 Entidade: SubClinica

Getters/Setters básicos

- getCNPJ()
- getNomeFantasia(), setNomeFantasia(nome)
- getRazaoSocial(), setRazaoSocial(razao)
- getInscricaoEstadual(), setInscricaoEstadual(inscricao)
- getEmail(), setEmail(listaEmails)
- getTelefone(), setTelefone(listaTelefones)
- getGerentes(), setGerentes(listaColaboradores)

Métodos de negócio

- atualizarDadosCadastrais(nomeFantasia, razaoSocial, inscricaoEstadual)
- atualizarContatos(emails[], telefones[])
- definirEndereco(endereco)
- adicionarGerente(colaborador) / removerGerente(colaborador)
Controla quais colaboradores atuam como gerentes da subclínica.

3.7 Entidade: ClinicoSocio

Getters/Setters básicos

- getId()
- getTipoParticipacao(), setTipoParticipacao(tipo)
- getPercentualParticipacao(), setPercentualParticipacao(percentual)
- getDesde(), setDesde(data)
- getAte(), setAte(data)
- isRepresentanteLegal(), setRepresentanteLegal(flag)

Métodos de negócio

- registrarParticipacao(tipoParticipacao, percentual, desde)
Cria o vínculo societário com tipo, percentual e data de início.
- atualizarPercentualParticipacao(percentual)
- encerrarParticipacao(ate)
Preenche a data de término na saída do sócio.

3.8 Entidade: Endereco

Getters/Setters básicos

- getCEP(), setCEP(cep)
- getRua(), setRua(rua)
- getComplemento(), setComplemento(complemento)
- getBairro(), setBairro(bairro)
- getNumero(), setNumero(numero)
- getCidade(), setCidade(cidade)
- getEstado(), setEstado(estado)

- getPais(), setPais(pais)

Métodos de negócio

- atualizarEnderecoCompleto(cep, rua, numero, bairro, complemento, cidade, estado, pais)
Atualiza todos os dados do endereço, aplicando o default “Brasil” quando pais não for informado.

3.9 Entidade: Anotacoes

Getters/Setters básicos

- getDataAnt(), setDataAnt(data)
- getAnotacoes(), setAnotacoes(listaTextos)

Métodos de negócio

- registrarAnotacao(dataAnotacao, texto)
Cria uma nova anotação com data e conteúdo.
- atualizarTexto(texto)
Atualiza o conteúdo textual da anotação.

3.10 Entidade: AgendaIndividual

Getters/Setters básicos

- getId()
- getStartAt(), setStartAt(dataHora)
- getDuracaoMinutos(), setDuracaoMinutos(minutos)
- getStatus(), setStatus(status)
- getObservacoes(), setObservacoes(listaObs)

Métodos de negócio

- agendarConsulta(startAt, duracaoMinutos)
Configura início e duração de uma consulta com status inicial (por exemplo, AGENDADA).
- remarcarConsulta(novoStartAt)
- cancelarConsulta()
- concluirAtendimento()
- adicionarObservacao(texto)

3.11 Entidade: Agenda-multi

Getters/Setters básicos

- Não possui atributos próprios no diagrama; atua como classe agregadora.

Métodos de negócio

- adicionarItemAgenda(agendaIndividual)
- removerItemAgenda(agendaIndividual)

- listarItens()

3.12 Entidade: Prontuario

Getters/Setters básicos

- getId()
- getDataCriacao()
- getDataAtt(), setDataAtt(data)
- getResponsavel(), setResponsavel(usuario)
- getPaciente()
- getVersao(), setVersao(versao)
- getConsentimento(), setConsentimento(texto)
- isBloqueado(), setBloqueado(flag)
- getTags(), setTags(tags)

Métodos de negócio

- abrirProntuario(paciente, responsavel)
- registrarAtualizacao(responsavel)
Atualiza data_att e incrementa a versao.
- bloquearProntuario() / desbloquearProntuario()
- atualizarConsentimento(consentimento)
- atualizarTags(tags)

3.13 Entidade: Permissao

Getters/Setters básicos

- getId()
- getResource(), setResource(resource)
- getAction(), setAction(action)

Métodos de negócio

- definirPermissao(resource, action)
- confere(resource, action)
Retorna verdadeiro se a permissão corresponde ao par (recurso, ação) informado.

3.14 Entidade: Role

Getters/Setters básicos

- getName(), setName(name)
- getScopeType(), setScopeType(scopeType)

Métodos de negócio

- adicionarPermissao(permissao)

- removerPermissao(permissao)

3.15 Entidade: RolePermissao

Getters/Setters básicos

- getRole(), setRole(role)
- getPermissao(), setPermissao(permissao)

Métodos de negócio

- associar(role, permissao)
- desassociar()

3.16 Entidade: RoleUsuario

Getters/Setters básicos

- getId()
- getStartAt(), setStartAt(dataHora)
- getEndAt(), setEndAt(dataHora)
- getScopeType(), setScopeType(scopeType)
- getScopeRefId(), setScopeRefId(uuid)
- getUsuario(), setUsuario(usuario)
- getRole(), setRole(role)

Métodos de negócio

- atribuirRole(usuario, role, scopeType, scopeRefId, startAt)
- encerrarVigencia(endAt)
- estaVigente(dataHoraReferencia)

3.17 Entidade: SolicitacaoAcessoProntuario

Getters/Setters básicos

- getId()
- getSolicitante(), setSolicitante(paciente)
- getProntuario(), setProntuario(prontuario)
- getStatus(), setStatus(status)
- getMotivos(), setMotivos(listaMotivos)
- getAprovadoPor(), setAprovadoPor(usuario)
- getAprovadoEm(), setAprovadoEm(data)
- getObs(), setObs(listaObs)

Métodos de negócio

- registrarSolicitacao(paciente, prontuario, motivos[])
- aprovar(usuarioAprovador, dataAprovacao)

- negar(usuarioAprovador, dataAprovacao, obs)
- atualizarMotivos(motivos[])
- adicionarObservacao(obs)

3.18 Entidade: Usuario

Getters/Setters básicos

- getId()
- getUsername(), setUsername(username)
- getPasswordHash(), setPasswordHash(hash)
- isActive(), setActive(flag)
- getLastLogin(), setLastLogin(dataHora)

Métodos de negócio

- registrarNovoUsuario(username, passwordHash)
- alterarSenha(novoPasswordHash)
- ativarUsuario() / inativarUsuario()
- registrarLogin(dataHora)