

3.7. Modelagem de Classes de Projeto

Sumário:

[1- Módulo de Agendamento](#)

[1.1 Código UML](#)

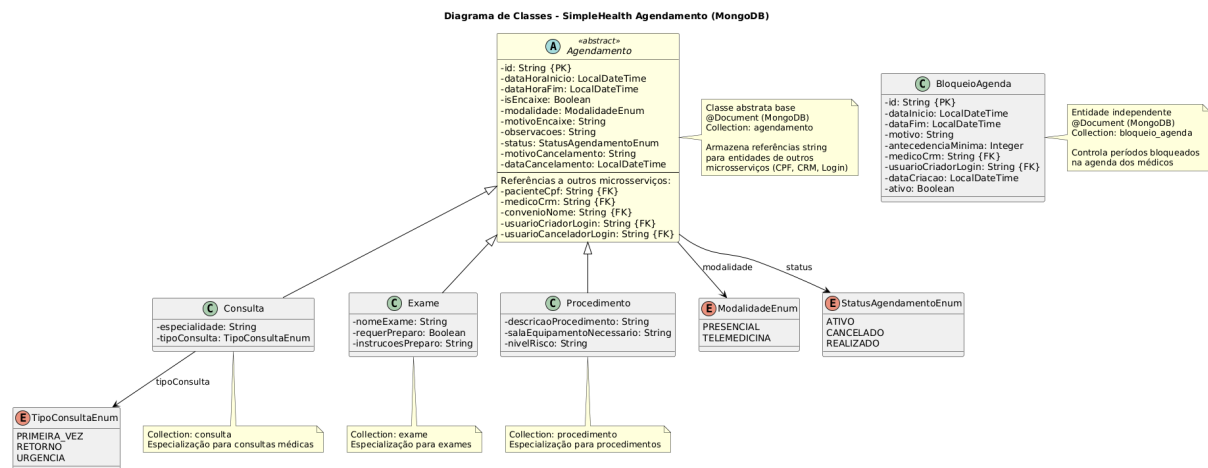
[2 - Módulo de Cadastro](#)

[2.1 Código UML](#)

[3 - Módulo de Estoque](#)

[3.1 Código UML](#)

1- Módulo de Agendamento



1.1 Código UML

@startuml Diagrama de Classes - Microserviço Agendamento

title Diagrama de Classes - SimpleHealth Agendamento (MongoDB)

' Configurações de estilo

skinparam classAttributeIconSize 0

skinparam class {

 BackgroundColor<<abstract>> LightYellow

 BorderColor Black

 ArrowColor Black

}

' Classe abstrata Agendamento

abstract class Agendamento <<abstract>> {

 - id: String {PK}

- dataHoraInicio: LocalDateTime
- dataHoraFim: LocalDateTime
- isEncaixe: Boolean
- modalidade: ModalidadeEnum
- motivoEncaixe: String
- observacoes: String
- status: StatusAgendamentoEnum
- motivoCancelamento: String
- dataCancelamento: LocalDateTime
-
- {field} Referências a outros microserviços:
- pacienteCpf: String {FK}
- medicoCrm: String {FK}
- convenioNome: String {FK}
- usuarioCriadorLogin: String {FK}
- usuarioCanceladorLogin: String {FK}

}

' Classe Consulta

```
class Consulta {
    - especialidade: String
    - tipoConsulta: TipoConsultaEnum
}
```

' Classe Exame

```
class Exame {
    - nomeExame: String
    - requerPreparo: Boolean
    - instrucoesPreparo: String
}
```

' Classe Procedimento

```
class Procedimento {
    - descricaoProcedimento: String
    - salaEquipamentoNecessario: String
    - nivelRisco: String
}
```

' Classe BloqueioAgenda (independente)

```
class BloqueioAgenda {
    - id: String {PK}
    - dataInicio: LocalDateTime
    - dataFim: LocalDateTime
    - motivo: String
    - antecedenciaMinima: Integer
    - medicoCrm: String {FK}
    - usuarioCriadorLogin: String {FK}
    - dataCriacao: LocalDateTime
}
```

```

- ativo: Boolean
}

' Enums
enum ModalidadeEnum {
    PRESENCIAL
    TELEMEDICINA
}

enum StatusAgendamentoEnum {
    ATIVO
    CANCELADO
    REALIZADO
}

enum TipoConsultaEnum {
    PRIMEIRA_VEZ
    RETORNO
    URGENCIA
}

' Relacionamentos de herança
Agendamento <|-- Consulta
Agendamento <|-- Exame
Agendamento <|-- Procedimento

' Relacionamentos com Enums
Agendamento --> ModalidadeEnum : modalidade
Agendamento --> StatusAgendamentoEnum : status
Consulta --> TipoConsultaEnum : tipoConsulta

note right of Agendamento
    Classe abstrata base
    @Document (MongoDB)
    Collection: agendamento

    Armazena referências string
    para entidades de outros
    microserviços (CPF, CRM, Login)
end note

note right of BloqueioAgenda
    Entidade independente
    @Document (MongoDB)
    Collection: bloqueio_agenda

    Controla períodos bloqueados
    na agenda dos médicos

```

end note

note bottom of Consulta

Collection: consulta

Especialização para consultas médicas

end note

note bottom of Exame

Collection: exame

Especialização para exames

end note

note bottom of Procedimento

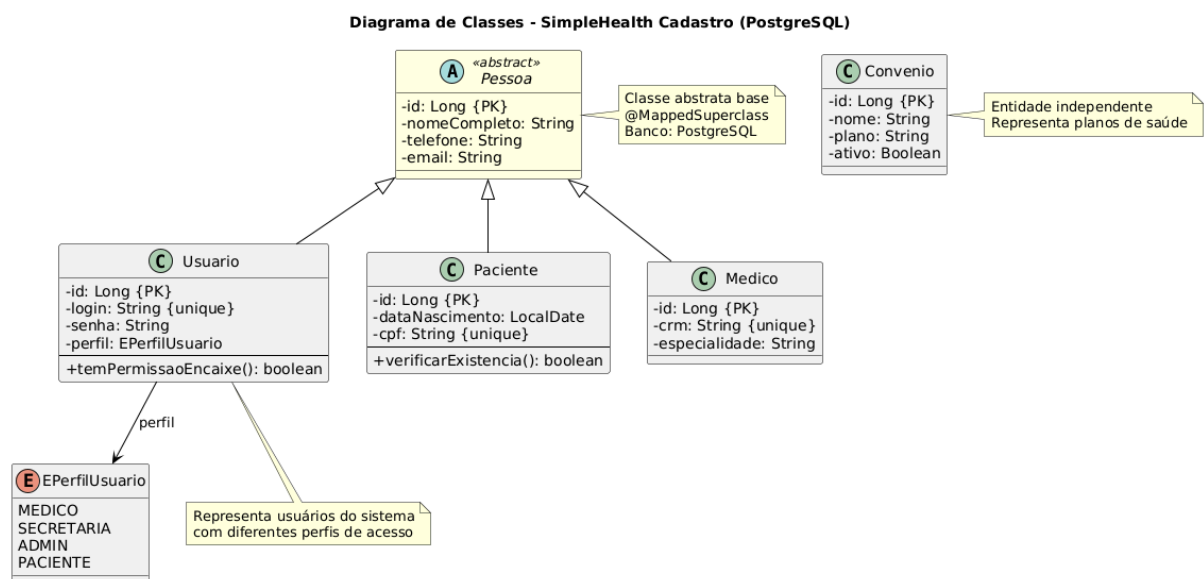
Collection: procedimento

Especialização para procedimentos

end note

@enduml

2 - Módulo de Cadastro



2.1 Código UML

@startuml Diagrama de Classes - Microserviço Cadastro

title Diagrama de Classes - SimpleHealth Cadastro (PostgreSQL)

' Configurações de estilo

skinparam classAttributeIconSize 0

```
skinparam class {
    BackgroundColor<<abstract>> LightYellow
    BorderColor Black
    ArrowColor Black
}
```

' Classe abstrata Pessoa

```
abstract class Pessoa <<abstract>> {
    - id: Long {PK}
    - nomeCompleto: String
    - telefone: String
    - email: String
}
```

' Classe Usuario

```
class Usuario {
    - id: Long {PK}
    - login: String {unique}
    - senha: String
    - perfil: EPerfilUsuario
    --
    + temPermissaoEncaixe(): boolean
}
```

' Classe Paciente

```
class Paciente {
    - id: Long {PK}
    - dataNascimento: LocalDate
    - cpf: String {unique}
    --
    + verificarExistencia(): boolean
}
```

' Classe Medico

```
class Medico {
    - id: Long {PK}
    - crm: String {unique}
    - especialidade: String
}
```

' Classe Convenio (independente)

```
class Convenio {
    - id: Long {PK}
    - nome: String
    - plano: String
    - ativo: Boolean
}
```

```

' Enum EPerfilUsuario
enum EPerfilUsuario {
    MEDICO
    SECRETARIA
    ADMIN
    PACIENTE
}

```

```

' Relacionamentos de herança
Pessoa <|-- Usuario
Pessoa <|-- Paciente
Pessoa <|-- Medico

```

```

' Relacionamentos de composição/associação
Usuario --> EPerfilUsuario : perfil

```

```

note right of Pessoa
    Classe abstrata base
    @MappedSuperclass
    Banco: PostgreSQL
end note

```

```

note right of Convenio
    Entidade independente
    Representa planos de saúde
end note

```

```

note bottom of Usuario
    Representa usuários do sistema
    com diferentes perfis de acesso
end note

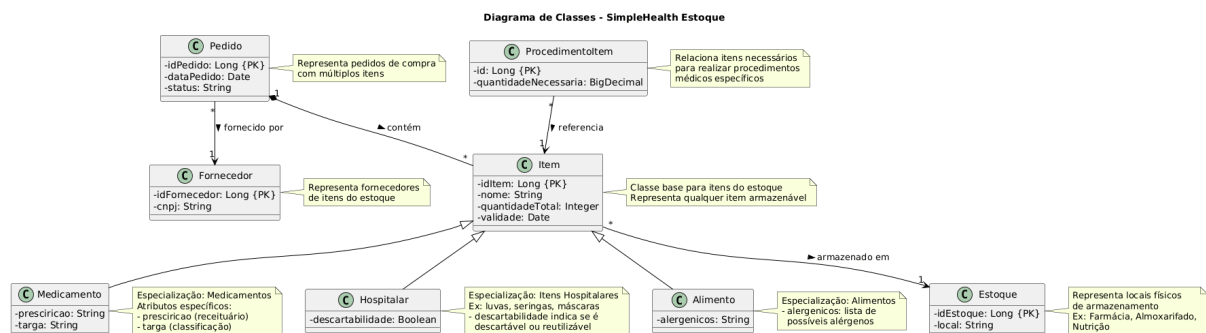
```

```

@enduml

```

3 - Módulo de Estoque



3.1 Código UML

@startuml Diagrama de Classes - Microserviço Estoque

title Diagrama de Classes - SimpleHealth Estoque

```
' Configurações de estilo
skinparam classAttributeIconSize 0
skinparam class {
    BackgroundColor<<abstract>> LightYellow
    BorderColor Black
    ArrowColor Black
}
```

```
' Classe base Item
class Item {
    - idItem: Long {PK}
    - nome: String
    - quantidadeTotal: Integer
    - validade: Date
}
```

```
' Classes especializadas de Item
class Medicamento {
    - prescricao: String
    - targa: String
}
```

```
class Hospitalar {
    - descartabilidade: Boolean
}
```

```
class Alimento {
    - alergenicos: String
}
```

```
' Classe Estoque
class Estoque {
    - idEstoque: Long {PK}
    - local: String
}
```

```
' Classe Fornecedor
class Fornecedor {
    - idFornecedor: Long {PK}
    - cnpj: String
}
```

' Classe Pedido

```
class Pedido {  
  - idPedido: Long {PK}  
  - dataPedido: Date  
  - status: String  
}
```

' Classe ProcedimentoItem

```
class ProcedimentoItem {  
  - id: Long {PK}  
  - quantidadeNecessaria: BigDecimal  
}
```

' Relacionamentos de herança

Item <|-- Medicamento

Item <|-- Hospitalar

Item <|-- Alimento

' Relacionamentos de composição/associação

Item "*" --> "1" Estoque : armazenado em >

Pedido "1" *-- "*" Item : contém >

Pedido "*" --> "1" Fornecedor : fornecido por >

ProcedimentoItem "*" --> "1" Item : referencia >

note right of Item

Classe base para itens do estoque

Representa qualquer item armazenável

end note

note right of Medicamento

Especialização: Medicamentos

Atributos específicos:

- prescricao (receituário)

- targa (classificação)

end note

note right of Hospitalar

Especialização: Itens Hospitalares

Ex: luvas, seringas, máscaras

- descartabilidade indica se é
descartável ou reutilizável

end note

note right of Alimento

Especialização: Alimentos

- alergenicos: lista de
possíveis alérgenos

end note

note right of Estoque
Representa locais físicos
de armazenamento
Ex: Farmácia, Almoxarifado,
Nutrição
end note

note right of Fornecedor
Representa fornecedores
de itens do estoque
end note

note right of Pedido
Representa pedidos de compra
com múltiplos itens
end note

note right of ProcedimentoItem
Relaciona itens necessários
para realizar procedimentos
médicos específicos
end note

@enduml