**Requisitos Funcionais**

Os **requisitos funcionais** descrevem **o que o sistema deve fazer**, ou seja, as funcionalidades relacionadas às tabelas.

**Tabela Cargo**

**Função geral:** Armazenar os diferentes cargos existentes no hospital, clínica ou empresa.

**Requisitos funcionais:**

1. O sistema deve permitir **cadastrar novos cargos** com um nome único.
2. O sistema deve permitir **consultar** todos os cargos existentes.
3. O sistema deve permitir **atualizar o nome** de um cargo existente.
4. O sistema deve permitir **excluir um cargo**, desde que não esteja sendo usado por nenhum funcionário (restrição referencial).
5. O sistema deve **garantir unicidade** do nome do cargo (não podem existir dois cargos com o mesmo nome).

**Tabela Funcionário**

**Função geral:** Armazenar os dados cadastrais de todos os funcionários da instituição.

**Requisitos funcionais:**

1. O sistema deve permitir **registrar novos funcionários**, vinculando-os a um cargo existente.
2. O sistema deve permitir **consultar informações** de um funcionário específico.
3. O sistema deve permitir **atualizar dados** pessoais e profissionais do funcionário (telefone, endereço, cargo etc.).
4. O sistema deve permitir **excluir um funcionário** do banco de dados.
5. O sistema deve **validar CPF e RG** para que não existam duplicatas (campo CPF é único).
6. O sistema deve **garantir integridade referencial** entre Funcionário e Cargo (todo funcionário deve ter um id\_cargo existente).
7. O sistema deve permitir **armazenar informações específicas de registro profissional**, como:

**COREN** → para enfermeiros.

**CRM** → para médicos.

**Requisitos Não Funcionais**

Os **requisitos não funcionais** definem **como o sistema deve funcionar**, isto é, suas características técnicas e de qualidade.

**Para ambas as tabelas:**

1. **Banco de dados relacional:**  
    As tabelas devem ser armazenadas em um banco relacional (MySQL) com **InnoDB**, permitindo integridade referencial e transações seguras.
2. **Charset UTF-8 (utf8mb4):**  
    Garante compatibilidade com caracteres acentuados (ex: João, São Paulo) e até emojis, se necessário.
3. **Performance:**

A consulta por **chave primária (id\_cargo, id\_funcionario)** deve ser rápida.

Índices automáticos garantem desempenho em buscas e relacionamentos.

1. **Segurança dos dados:**

O CPF é único para evitar duplicações.

As restrições NOT NULL impedem que informações obrigatórias fiquem em branco.

1. **Padronização dos dados:**

Campos como UF\_funcionario, CEP, CPF, RG, CRM e COREN têm tamanho fixo para garantir consistência.

Tipos de dados (como DATE e CHAR) ajudam na validação automática.

1. **Manutenibilidade:**  
    O uso de chaves estrangeiras e normalização facilita futuras alterações sem corromper dados.