

Documento Técnico – Sistema de Pesquisa

1. Visão Geral

Este documento descreve **toda a arquitetura de banco de dados** e a **implementação em C# (.NET)** utilizando **Entity Framework Core**, **Repository Pattern** e **LINQ**, com base no diagrama fornecido.

O sistema tem como objetivo permitir: - Cadastro de empresas - Criação de pesquisas por empresa - Criação de perguntas e opções - Registro de respostas de entrevistados - Associação das respostas às opções escolhidas

2. Modelo de Dados (Banco de Dados)

2.1 Tabela: Empresa

```
CREATE TABLE Empresa (  
    IdEmpresa INT IDENTITY PRIMARY KEY,  
    NomeEmpresa VARCHAR(150) NOT NULL,  
    Documento VARCHAR(14) NOT NULL, -- CNPJ ou CPF  
    Ativo BIT NOT NULL DEFAULT 1  
);
```

2.2 Tabela: Pesquisa

```
CREATE TABLE Pesquisa (  
    IdPesquisa INT IDENTITY PRIMARY KEY,  
    Descricao VARCHAR(200) NOT NULL,  
    Ativo BIT NOT NULL DEFAULT 1,  
    DataInicio DATE NULL,  
    DataFinal DATE NULL,  
    IdEmpresa INT NOT NULL,  
    CONSTRAINT FK_Pesquisa_Empresa FOREIGN KEY (IdEmpresa)  
        REFERENCES Empresa(IdEmpresa)  
);
```

2.3 Tabela: Pergunta

```
CREATE TABLE Pergunta (  
    IdPergunta INT IDENTITY PRIMARY KEY,  
    Descricao VARCHAR(300) NOT NULL,  
    Ativo BIT NOT NULL DEFAULT 1,
```

```
IdPesquisa INT NOT NULL,  
CONSTRAINT FK_Pergunta_Pesquisa FOREIGN KEY (IdPesquisa)  
REFERENCES Pesquisa(IdPesquisa)  
);
```

2.4 Tabela: Opcao

```
CREATE TABLE Opcao (  
    IdOpcao INT IDENTITY PRIMARY KEY,  
    Descricao VARCHAR(200) NOT NULL,  
    Peso INT NOT NULL,  
    Ativo BIT NOT NULL DEFAULT 1,  
    IdPergunta INT NOT NULL,  
    CONSTRAINT FK_Opcao_Pergunta FOREIGN KEY (IdPergunta)  
    REFERENCES Pergunta(IdPergunta)  
);
```

2.5 Tabela: Resposta

```
CREATE TABLE Resposta (  
    IdResposta INT IDENTITY PRIMARY KEY,  
    IdEmpresa INT NOT NULL,  
    CpfEntrevistado VARCHAR(11) NOT NULL,  
    DataResposta DATETIME NOT NULL DEFAULT GETDATE(),  
    CONSTRAINT FK_Resposta_Empresa FOREIGN KEY (IdEmpresa)  
    REFERENCES Empresa(IdEmpresa)  
);
```

2.6 Tabela: OpcaoResposta

```
CREATE TABLE OpcaoResposta (  
    IdOpcaoResposta INT IDENTITY PRIMARY KEY,  
    IdResposta INT NOT NULL,  
    IdOpcao INT NOT NULL,  
    IdPergunta INT NOT NULL,  
    CONSTRAINT FK_OpcaoResposta_Resposta FOREIGN KEY (IdResposta)  
    REFERENCES Resposta(IdResposta),  
    CONSTRAINT FK_OpcaoResposta_Opcao FOREIGN KEY (IdOpcao)  
    REFERENCES Opcao(IdOpcao)  
);
```

3. Entidades C# (Domain Models)

3.1 Empresa

```
public class Empresa
{
    public int IdEmpresa { get; set; }
    public string NomeEmpresa { get; set; }
    public string Documento { get; set; }
    public bool Ativo { get; set; }

    public ICollection<Pesquisa> Pesquisas { get; set; }
}
```

3.2 Pesquisa

```
public class Pesquisa
{
    public int IdPesquisa { get; set; }
    public string Descricao { get; set; }
    public bool Ativo { get; set; }
    public DateTime? DataInicio { get; set; }
    public DateTime? DataFinal { get; set; }

    public int IdEmpresa { get; set; }
    public Empresa Empresa { get; set; }

    public ICollection<Pergunta> Perguntas { get; set; }
}
```

3.3 Pergunta

```
public class Pergunta
{
    public int IdPergunta { get; set; }
    public string Descricao { get; set; }
    public bool Ativo { get; set; }

    public int IdPesquisa { get; set; }
    public Pesquisa Pesquisa { get; set; }

    public ICollection<Opcao> Opcoes { get; set; }
}
```

3.4 Opcao

```
public class Opcao
{
    public int IdOpcao { get; set; }
    public string Descricao { get; set; }
    public int Peso { get; set; }
    public bool Ativo { get; set; }

    public int IdPergunta { get; set; }
    public Pergunta Pergunta { get; set; }
}
```

3.5 Resposta

```
public class Resposta
{
    public int IdResposta { get; set; }
    public int IdEmpresa { get; set; }
    public string CpfEntrevistado { get; set; }
    public DateTime DataResposta { get; set; }

    public Empresa Empresa { get; set; }
    public ICollection<OpcaoResposta> OpcoesRespondidas { get; set; }
}
```

3.6 OpcaoResposta

```
public class OpcaoResposta
{
    public int IdOpcaoResposta { get; set; }

    public int IdResposta { get; set; }
    public Resposta Resposta { get; set; }

    public int IdOpcao { get; set; }
    public Opcao Opcao { get; set; }

    public int IdPergunta { get; set; }
}
```

4. DbContext

```
public class PesquisaContext : DbContext
{
    public DbSet<Empresa> Empresas { get; set; }
    public DbSet<Pesquisa> Pesquisas { get; set; }
    public DbSet<Pergunta> Perguntas { get; set; }
    public DbSet<Opcao> Opcoes { get; set; }
    public DbSet<Resposta> Respostas { get; set; }
    public DbSet<OpcaoResposta> OpcaoRespostas { get; set; }
}
```

5. Repository Pattern

5.1 Interface Genérica

```
public interface IRepository<T> where T : class
{
    Task<T> GetByIdAsync(int id);
    Task<IEnumerable<T>> GetAllAsync();
    Task AddAsync(T entity);
    Task UpdateAsync(T entity);
    Task DeleteAsync(T entity);
}
```

5.2 Implementação Base

```
public class Repository<T> : IRepository<T> where T : class
{
    protected readonly PesquisaContext _context;

    public Repository(PesquisaContext context)
    {
        _context = context;
    }

    public async Task AddAsync(T entity)
    {
        _context.Set<T>().Add(entity);
        await _context.SaveChangesAsync();
    }

    public async Task<IEnumerable<T>> GetAllAsync()
    {

```

```

        return await _context.Set<T>().ToListAsync();
    }

    public async Task<T> GetByIdAsync(int id)
    {
        return await _context.Set<T>().FindAsync(id);
    }

    public async Task UpdateAsync(T entity)
    {
        _context.Set<T>().Update(entity);
        await _context.SaveChangesAsync();
    }

    public async Task DeleteAsync(T entity)
    {
        _context.Set<T>().Remove(entity);
        await _context.SaveChangesAsync();
    }
}

```

6. Exemplos de LINQ

6.1 Buscar Pesquisa com Perguntas e Opções

```

var pesquisa = await _context.Pesquisas
    .Include(p => p.Perguntas)
    .ThenInclude(p => p.Opcoes)
    .FirstOrDefaultAsync(p => p.IdPesquisa == id);

```

6.2 Salvar Resposta da Pesquisa

```

var resposta = new Resposta
{
    IdEmpresa = empresaId,
    CpfEntrevistado = cpf,
    DataResposta = DateTime.Now,
    OpcoesRespondidas = opcoesSelecionadas.Select(o => new OpcaoResposta
    {
        IdOpcao = o.IdOpcao,
        IdPergunta = o.IdPergunta
    }).ToList()
};

```

```
_context.Respostas.Add(resposta);  
await _context.SaveChangesAsync();
```

7. Considerações Finais

- Estrutura preparada para **relatórios estatísticos**
 - Fácil extensão para **pesquisa NPS, escala, múltipla escolha**
 - Totalmente compatível com **Angular / React no frontend**
 - Segue boas práticas de **DDD + Repository + LINQ**
-

 Documento pronto para ser usado como **base oficial de desenvolvimento**