



Instituto Infnet

Projeto da Disciplina

Arquitetura .NET

25E4_3

Ricardo José Nunes

Sumário

Sumário.....	2
Histórico de atividades.....	3
Visão do Projeto.....	4
PROBLEMA.....	5
PROPOSTA DE SOLUÇÃO.....	5
OBJETIVOS.....	7
DIAGRAMAS.....	8
DIAGRAMA DE CONTEXTO (C4 - NÍVEL 1).....	8
DIAGRAMA DE CONTAINERS (C4 - NÍVEL 2).....	9
DIAGRAMA DE COMPONENTES (C4 - NÍVEL 3).....	10
DIAGRAMA DE CÓDIGO (C4 - NÍVEL 4).....	11
PROJETO.....	11
Testabilidade do Acesso a Dados.....	12
PROCEDIMENTOS DE ACESSO E AVALIAÇÃO.....	12

Histórico de atividades

Descrição	Data	Responsável
Criação do documento	22/11/2025	Ricardo José Nunes
Definição da visão geral do projeto	22/11/2025	Ricardo José Nunes
Elaboração da seção de introdução	22/11/2025	Ricardo José Nunes
Identificação dos problemas	06/12/2025	Ricardo José Nunes
Divulgação da proposta de solução	13/12/2025	Ricardo José Nunes
Indicação dos objetivos	14/12/2025	Ricardo José Nunes
Entrega do projeto	15/12/2025	Ricardo José Nunes

Visão do Projeto

Nesta seção, será apresentada uma visão geral do projeto, abordando os principais aspectos que norteiam sua concepção e desenvolvimento. A **Introdução** contextualiza o sistema proposto, destacando sua finalidade e público-alvo. Em seguida, o **Problema** descreve os desafios e as necessidades enfrentados pela startup, justificando a necessidade do projeto. Os **Objetivos** detalham as metas a serem alcançadas, tanto gerais quanto específicas, quanto ao escopo que define as funcionalidades incluídas e excluídas do sistema. Os **Stakeholders** identificam os principais atores envolvidos e impactados pelo projeto.

INTRODUÇÃO

Neste projeto você terá a responsabilidade de um arquiteto de software. Sua missão será criar um sistema de Questionários Online para uma startup. Por questões contratuais, um dos pré-requisitos para a construção da solução é que ela seja feita utilizando os componentes do .NET framework.

O sistema de questionários online terá como finalidade principal a elaboração de pesquisas públicas. Um dos alvos da startup é pesquisa pública sobre as eleições, onde seriam feitos anúncios em diversas redes sociais convidando as pessoas a responderem a pesquisa. O questionário teria uma estrutura simples de perguntas com resposta no modelo múltipla escolha. Como o alvo das pesquisas são milhões de pessoas, é preciso se preocupar com questões de escala também. Depois do período de coleta de respostas, o sistema deve disponibilizar de forma sumarizada, para alguns usuários, o resultado da pesquisa.

Para nos guiar na construção deste sistema temos um diagrama utilizando C4 Model para descrever os componentes arquiteturais do sistema. Assuma que você deve apresentar a solução proposta para os desenvolvedores do projeto e para um usuário do sistema.

PROBLEMA

Uma startup necessita realizar coleta de informações sobre eleições, e para isto é necessário construir sistema de questionários online, que terá como finalidade principal a elaboração de pesquisas públicas. Um dos alvos da startup é a pesquisa pública sobre as eleições, onde serão feitos anúncios em diversas redes sociais convidando as pessoas a responderem a pesquisa.

Como o alvo das pesquisas são milhões de pessoas, é preciso se preocupar com questões de escala também. Depois do período de coleta de respostas, o sistema deve disponibilizar de forma sumarizada, para alguns usuários, o resultado da pesquisa.

PROPOSTA DE SOLUÇÃO

O sistema de questionário tem como objetivo construir uma aplicação robusta e para contemplar tais necessidades, usaremos:

- Domain Drive-Design (DDD)
 - O domínio é o centro do sistema, não a tecnologia. O DDD ajuda a isolar e proteger as regras do negócio e que permita rápida evolução, pois sua implementação tende a ter um baixo acoplamento.
 - Mudanças não quebram o sistema.
 - O DDD permite ter uma linguagem comum ou linguagem ubíqua, evitando mal entendidos e retrabalho .
 - Todos enxergam a camada de domínio central e assim todos trabalham alinhados.
 - Protege o domínio das outras camadas, como:
 - Criar regras na parte de infraestrutura: Direto no banco de dados ou na camada de aplicação, no qual deveria apenas se preocupar na estrutura dos dados.

- Melhora testabilidade, pois com o domínio isolado, os testes unitários podem ser implementados independente de banco de dados, framework ou uso de comunicação por HTTP por exemplo.
- Diagramação com uso do C4 Model.
 - Permite que stakeholders possam ter compreensão do sistema, através do uso de uma linguagem ubíqua.
 - Permite uma comunicação em todos os níveis, desde do nível de negócio até ao nível mais técnico.
- Plataforma de desenvolvimento em .NET 10.0
 - O uso do .NET como plataforma de desenvolvimento se dá pelo seguintes fatores:
 - Robustez e confiabilidade: .NET é uma plataforma amplamente utilizada e testada, que oferece um alto nível de confiabilidade.
 - Flexibilidade e escalabilidade: .NET pode ser usado para criar aplicações de qualquer tamanho, desde aplicações simples até aplicações complexas.
 - Acessibilidade: .NET é uma plataforma gratuita e de código aberto, que está disponível para todos os desenvolvedores.
 - É multiplataforma, como uso em ambientes Linux ou Windows ou aplicativos móveis.
 - Aplicações web: .NET fornece uma ampla gama de ferramentas e recursos para criar aplicações web modernas e escaláveis, como ASP.NET Core e bibliotecas como Blazor WebAssembly.
 - Aplicações móveis: .NET fornece um conjunto de ferramentas e bibliotecas para criar aplicações móveis para Android, iOS e Windows.
 - Aplicações de desktop: .NET fornece um conjunto de ferramentas e bibliotecas para criar aplicações de desktop para Windows, macOS e Linux.
 - Aplicações de jogos: .NET fornece um conjunto de ferramentas e bibliotecas para criar jogos para Windows, macOS e Linux.

- Aplicações de serviços de nuvem: .NET fornece um conjunto de ferramentas e bibliotecas para criar aplicações de serviços de nuvem, como Azure e AWS.

O sistema de questionários proposto contará com o seguinte conjunto de entidades DDD (Domain Driven-Design):

1. Bounded Contexts

- Survey Context
 - Onde questionários são criados e gerenciados.
- Survey Report Context
 - Onde os resultados são sumarizados e processados.

2. Agregados e Entidades do domínio

- Survey (Agregado raiz)
 - Representa um questionário.
 - Regras:
 - Só pode ser respondido se estiver Active.
 - Um survey Active não pode ter perguntas editadas.
- Question (Entidade dentro de Survey)
 - Pergunta do questionário.
- Option (Value Object)
 - Alternativas de múltipla escolha.

OBJETIVOS

O sistema de questionários online terá como finalidade principal a elaboração de pesquisas públicas

Objetivos do sistema

Criar uma estrutura de entidades com relacionamentos bem definidos.

- Implementar operações CRUD para o questionário online.
- Relatório dos resultados das pesquisas.
- Uso de componentes .NET adequados.

Funcionalidades incluídas

- Cadastro, consulta, resultados das pesquisas e validações para as operações no sistema.
 - Validação de campos.
 - Tratamento de erros.
- Associação e relacionamentos das entidades e objetos de valor.
- Exposição de APIs REST documentadas via Swagger.
- Armazenamento dos dados em banco de dados relacional.

DIAGRAMAS

DIAGRAMA DE CONTEXTO (C4 - NÍVEL 1)

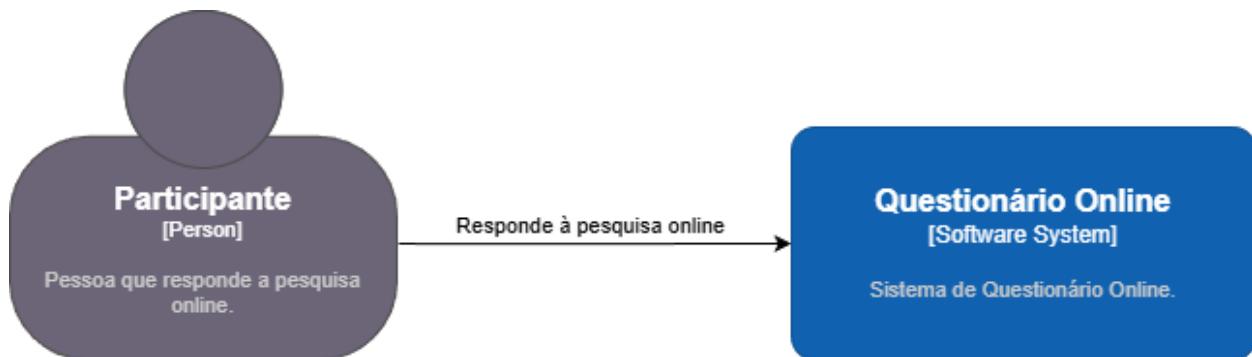


DIAGRAMA DE CONTAINERS (C4 - NÍVEL 2)

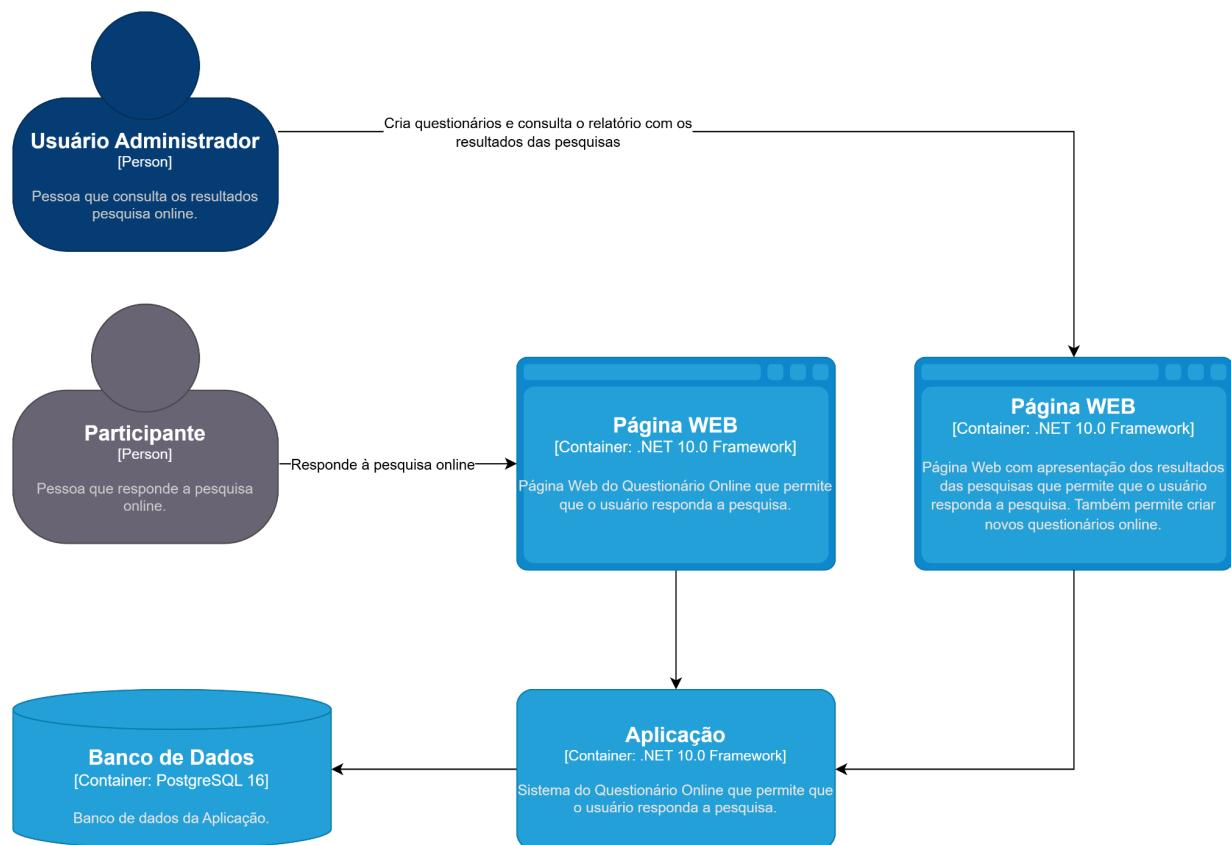


DIAGRAMA DE COMPONENTES (C4 - NÍVEL 3)

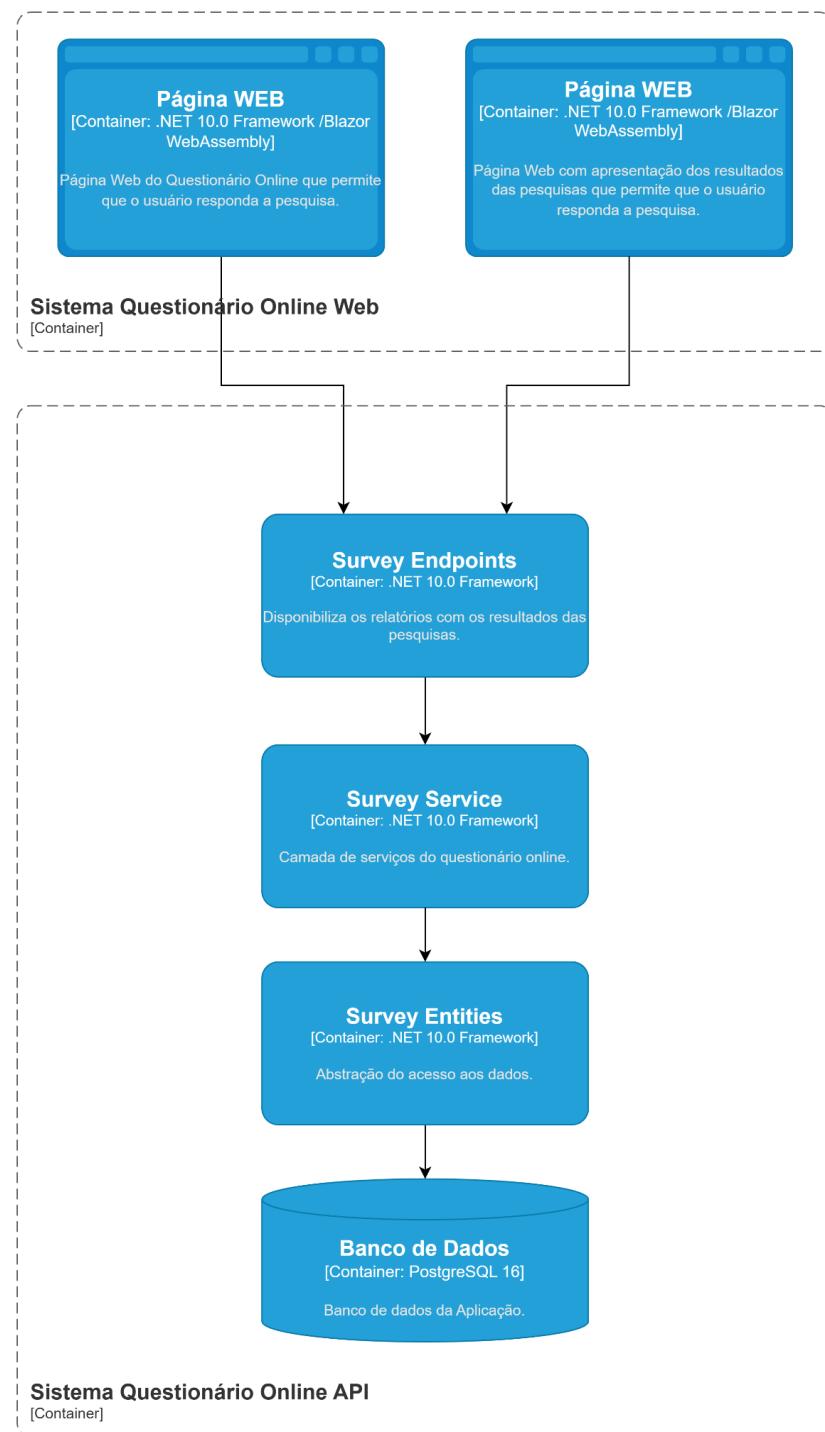
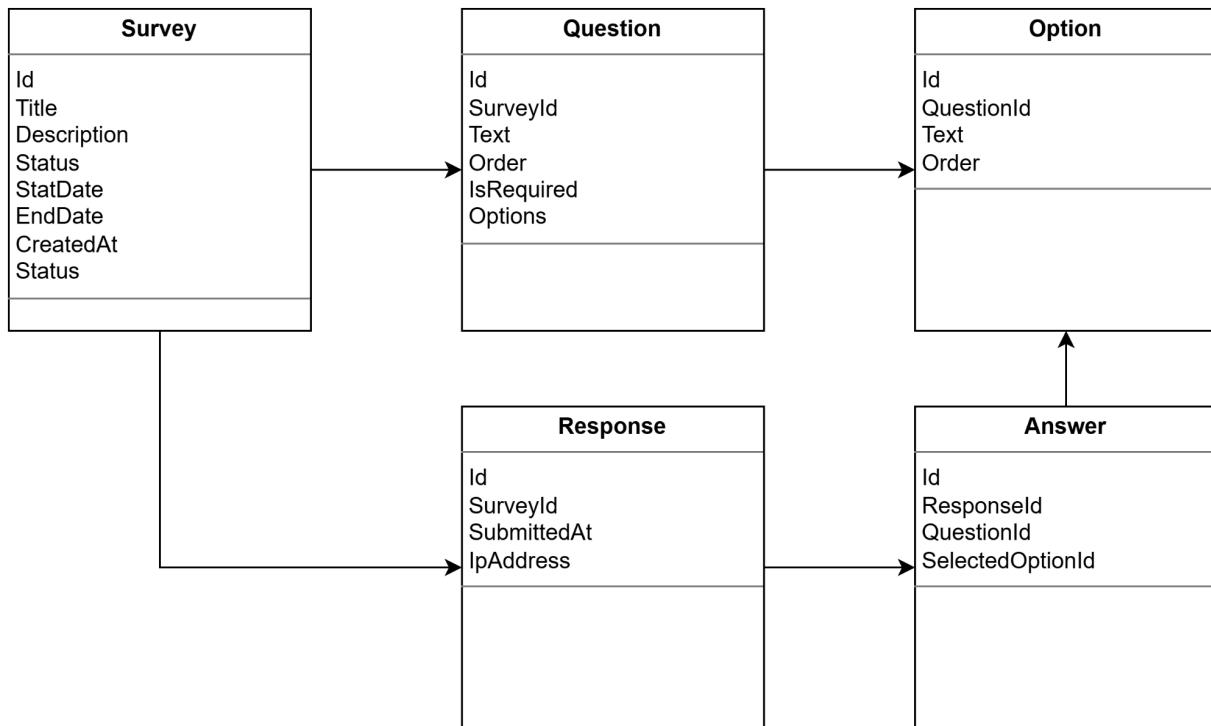


DIAGRAMA DE CÓDIGO (C4 - NÍVEL 4)



PROJETO

O código-fonte está disponível no GitHub, para ter acesso ao projeto .NET Framework desenvolvido, acesse o repositório através do seguinte link: <https://github.com/ricardo-jinunes/infnet-questionario-online>.

Principais tecnologias utilizadas:

- Linguagem: .NET 10.0
- Frontend: Blazor WebAssembly
- Banco de Dados: PostgreSQL 16
- Persistência: Entity Framework Core
- Documentação: Swagger UI

- IDE: Visual Code Studio

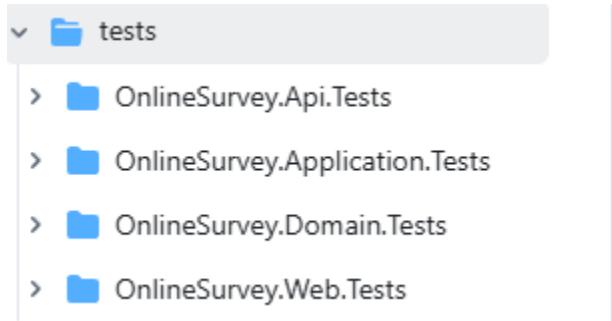
TESTABILIDADE DO ACESSO A DADOS

A testabilidade do acesso a dados foi considerada no projeto arquitetural por meio da separação clara entre as camadas de Domínio, Aplicação e Infraestrutura, conforme os princípios do Domain-Driven Design (DDD).

O uso do Entity Framework Core permite que o acesso ao banco de dados seja abstraido através de repositórios, os quais podem ser facilmente testados de forma isolada. Para testes automatizados, é possível utilizar:

Banco de dados em memória (InMemory Provider do EF Core), permitindo a execução de testes sem dependência de um banco de dados real.

Mocks de repositórios, possibilitando testes unitários das regras de negócio da camada de domínio sem acoplamento à infraestrutura, sendo estes que podem ser verificados através do diretório de *tests* do projeto:



Os testes podem ser aferidos através do comando:

```
> dotnet test
```

PROCEDIMENTOS DE ACESSO E AVALIAÇÃO

Faça o download do projeto:

- git clone <https://github.com/ricardo-jnunes/infnet-questionario-online.git>

- Ou baixe o arquivo ZIP diretamente pela interface do GitHub.

Importe o projeto em uma IDE e SDK compatível com .NET 10.0, como o Visual Studio.

Execução Local

1. Certifique-se de ter instalado o runtime do .NET 10.0.
2. No diretório raiz do projeto, execute o comando:
 - a. *docker compose up*
3. Após a inicialização, o servidor local estará disponível no endereço:
 - a. <http://localhost:5000>
4. Acesse a documentação interativa da API no Swagger UI:
 - a. <http://localhost:8080/swagger-ui/index.html>