

# Modelo C4 Completo – Sistema de Questionários Online

## Nível 1 – Contexto (Mermaid)

```
```mermaid
C4Context
title Sistema de Questionários Online - Contexto

Person(respondente, "Respondente", "Usuário que responde aos questionários")
Person(analista, "Pesquisador / Analista", "Cria, analisa e interpreta resultados dos questionários")

System(plataforma, "Plataforma de Questionários Online", "Onde os questionários são criados, publicados")

System_Ext(redes, "Plataformas de Anúncio / Redes Sociais", "Origem do tráfego que leva o Respondente ao sistema")

Rel(respondente, plataforma, "Responde questionários")
Rel(analista, plataforma, "Gerencia pesquisas e analisa resultados")
Rel(redes, respondente, "Encaminha para o questionário")
```
```

## Nível 2 – Containers (Mermaid)

```
```mermaid
C4Container
title Sistema de Questionários Online - Containers

Person(respondente, "Respondente", "Acessa e responde formulários.")
Person(analista, "Pesquisador / Analista", "Cria pesquisas e interpreta resultados.")

System_Boundary(sistema, "Plataforma de Questionários Online") {

    Container(frontend, "Frontend Web", "Angular/React/Vue", "Interface onde respondentes e analistas interagem")
    Container(api, "API Backend", ".NET 9 / Node.js", "Fornece endpoints para criar pesquisas, registrar respondentes e analistas")
    ContainerDb(db, "Banco de Dados", "MySQL/SQL Server", "Armazena pesquisas, perguntas, respostas e resultados")

}

System_Ext(redes, "Plataformas de Anúncio / Redes Sociais", "Origem de tráfego e divulgação de links de anúncios")

Rel(respondente, frontend, "Responde questionários via")
Rel(analista, frontend, "Gerencia pesquisas via")
Rel(frontend, api, "Consume API REST", "HTTPS/JSON")
Rel(api, db, "CRUD", "SQL")
Rel(redes, frontend, "Direciona tráfego")
```
```

## Nível 3 – Componentes da API (Mermaid)

```
```mermaid
C4Component
title API Backend - Componentes

Container(api, "API Backend", ".NET 9 / Node.js", "Lida com lógica de negócios e dados")

Component(controllerSurvey, "SurveyController", "Exposição de endpoints de pesquisas")
```
```

```

Component(serviceSurvey, "SurveyService", "Regras de negócio de pesquisas")
Component(repoSurvey, "SurveyRepository", "Acesso ao banco de pesquisas")

Component(controllerResults, "ResultsController", "Endpoints de resultados e gráficos")
Component(serviceResults, "ResultsService", "Processamento e agregação de respostas")
Component(repoResults, "ResultsRepository", "Acesso a respostas consolidadas")

Rel(controllerSurvey, serviceSurvey, "Chama")
Rel(serviceSurvey, repoSurvey, "Consulta/Grava")

Rel(controllerResults, serviceResults, "Chama")
Rel(serviceResults, repoResults, "Consulta/Grava")
```

```

## Nível 4 – Código (Estrutura Simplificada)

```

```csharp
// SurveyController.cs
[ApiController]
[Route("api/survey")]
public class SurveyController : ControllerBase {
    private readonly ISurveyService _service;
    public SurveyController(ISurveyService service) => _service = service;

    [HttpGet("{id}")]
    public async Task<IActionResult> GetSurvey(int id)
        => Ok(await _service.GetSurveyViewModelAsync(id));
}

// SurveyService.cs
public class SurveyService : ISurveyService {
    private readonly ISurveyRepository _repo;
    public SurveyService(ISurveyRepository repo) => _repo = repo;

    public Task<SurveyViewModel> GetSurveyViewModelAsync(int id)
        => _repo.GetSurveyViewModelAsync(id);
}
```

```