

Avaliação de Conhecimentos - Programador C#

Candidato: Jorge Antonio Carneiro diniz

Cargo: Programador C# JR

A avaliação escolhida foi “AvaliacaoConhecimentosBackEnd.pdf”. Que tinha a seguinte atividade a ser resolvida: *Criar uma listagem de fornecedores relacionados a uma empresa;*

Para a resolução da atividade, optei pelo desenvolvimento de uma API Rest utilizando .NET 6, e banco de dados SQL server. Na camada de infraestrutura utilizei o ORM da Microsoft, Entity Framework Core para realizar as operações implementadas nos repositórios. As validações de domínio foram desenvolvidas com a biblioteca Fluent Validation, e criei views para validações ainda na controller para otimizar a resposta a erros, utilizando os atributos Data Annotation, do próprio .NET.

No fim, o resultado foi esse:

The top screenshot displays the Swagger UI for the 'TesteVagaJr' API. It lists several endpoints under the 'Empresa' group:

- POST /api/v1/empresas/adicionar-empresas
- DELETE /api/v1/empresas/remover-empresas/{id}
- GET /api/v1/empresas/pegar-todas-empresas
- POST /api/v1/empresas/adicionar-fornecedor
- GET /api/v1/empresas/pegar-todos-fornecedores-empresa/{id}
- GET /api/v1/users/filtrar-fornecedor-por-nome
- GET /api/v1/users/filtrar-fornecedor-por-numero-documento
- GET /api/v1/users/filtrar-fornecedor-por-data-cadastro

The bottom screenshot shows the Visual Studio IDE with the 'TesteVagaJr' project open. The 'EmpresasController.cs' file is displayed, showing the implementation of the 'AdicionarEmpresa' endpoint:

```
using AutoMapper;
using Microsoft.AspNetCore.Mvc;
using TesteVagaJr.Core.Exceptions;
using TesteVagaJr.Services.Dtos;
using TesteVagaJr.Services.Interfaces;
using TesteVagaJr.Utilities;
using TesteVagaJr.ViewModels;

namespace TesteVagaJr.Controllers;

public class EmpresaController : MainController
{
    private readonly IEmpresaService _empresaService;
    private readonly IMapper _mapper;

    public EmpresaController(IEmpresaService empresaService, IMapper mapper)
    {
        _empresaService = empresaService;
        _mapper = mapper;
    }

    [HttpPost]
    [Route("/api/v1/empresas/adicionar-empresas")]
    public async Task<IActionResult> AdicionarEmpresa([FromBody] CadastrarEmpresaViewModel empresaViewModel)
    {
        try
        {
            var empresaDto = _mapper.Map<EmpresaDto>(empresaViewModel);
            var empresaAdicionar = await _empresaService.AdicionarEmpresa(empresaDto);
        }
    }
}
```

Muito obrigado pela oportunidade de participar do processo :)