FUNDAÇÃO OSWALDO ARANHA

CENTRO UNIVERSITÁRIO DE VOLTA REDONDA

CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMACÃO

TRABALHO DE PROGRAMAÇÃO ORIENTADA A OBJETOS

DANILO DUARTE RIBEIRO

RAFAEL COUTO SILVA

THALYS FERREIRA CORREIA

AVALIAÇÃO BIMESTRAL DE POO: CadastroDeDev

VOLTA REDONDA

2020

# 

# 1.0 INTRODUÇÃO

O presente trabalho tem como objetivo apresentar a solução “cadastro-de-dev”, uma plataforma de cadastro de desenvolvedores, os quais são vinculados a linguagens de programação que conhecem e a empresas que desejam trabalhar. Esse trabalho também é a atividade final do 3° ano da disciplina de Programação Orientada a Objetos do curso de Sistemas de Informação do UniFOA (Centro Universitário de Volta Redonda).

Em seu desenvolvimento, foram utilizados diversos artifícios da linguagem C#, do framework .NET Core MVC (Model View Controller) e do banco de dados Oracle Database.

# 2.0 DESENVOLVIMENTO DA APLICAÇÃO

## 2.1 Construção e Recursos Utilizados

A aplicação foi desenvolvida utilizando o framework .NET Core MVC, ou seja, por meio da utilização de *Models*, *Views* e *Controllers* para cada classe. As classes de *model* criadas foram: Desenvolvedor, Empresa, Linguagem e DesenvolvedorLinguagem; e seus respectivos *controllers* são: DesenvolvedorController, EmpresaController, LinguagemController e DesenvolvedorLinguagemController.

Tais classes possuem *views* para que seja efetuado o CRUD de cada uma, isto é, Create, Read, Update e Delete; e seus respectivos métodos nos *controllers* responsáveis por executas essas ações.

Além disso, para realizar a integração com o banco de dados Oracle Database, foi criada a classe CadastroDeDevContext , onde estão contidas propriedades de cada uma das classes de *model*, além do relacionamento muitos para muitos entre as classes Desenvolvedor e Linguagem, estabelecendo chaves estrangeiras na classe intermediária DesenvolvedorLinguagem e; para garantir o funcionamento correto da integração, foi utilizada a ferramenta de *migration*.

Foi feito um relacionamento um para muitos entre as classes Empresa e Desenvolvedor, de maneira que cada desenvolvedor possa escolher uma empresa e uma empresa possa ser escolhida por vários desenvolvedores. Outro relacionamento criado, conforme já mencionado anteriormente, é um muitos para muitos entre as classes Desenvolvedor e Linguagem por meio da classe intermediária DesenvolvedorLinguagem, de maneira que cada desenvolvedor possa escolher várias linguagens de programação e cada linguagem de programação possa ser escolhida por vários desenvolvedores.

Também foram implementadas *procedures* e suas *views*, ShowDevs e ShowLangs uma para exibir os desenvolvedores vinculados a uma determinada empresa e outra para exibir as linguagens de programação que determinado desenvolvedor conhece e/ou domina, respectivamente.

## 2.2 Problemas Encontrados e Suas Respectivas Soluções

Foram encontrados problemas para realizar o relacionamento muitos para muitos entre as classes Desenvolvedor e Linguagem, por meio classe intermediária DesenvolvedorLinguagem, onde a classe Desenvolvedor não exibir as linguagens corretamente na *view* Create e não conseguia gravar o *id* da linguagem.

Para solucionarmos tal problema, foi necessário implementar *controller* e *views* para a classe intermediária, de forma que ela pudesse reconhecer a lista de desenvolvedores e a lista de linguagens cadastrados, exibi-los simultaneamente e gravar os campos selecionados.

Outro problema foi no acionamento da segunda *procedure* criada, Show, que contém parâmetros IN e OUT no PL/SQL. O erro ocorria devido à forma errônea de passar o parâmetro OUT e obter o seu retorno; para solucioná-lo, foi necessário declarar uma variável “cursor” que instanciasse um OracleParameter do tipo RefCursor e direção Output; remover a linha “DBMS\_SQL.RETURN\_RESULT(c\_lingdev);” da *procedure* no Oracle; substituir o método FromSqlInterpolated para FromSqlRaw; além de, no parâmetro desse método, chamar um novo objeto “cursor” passando a variável “cursor”.

Infelizmente, a equipe não consegui realizar a implementação das *functions* PL/SQL no código.

# 3.0 CONCLUSÕES

## 3.1 Danilo Duarte Ribeiro

No decorrer do desenvolvimento desse trabalho, concluí que a utilização dos recursos possíveis na linguagem C# e no framework .NET Core MVC, pode facilitar o desenvolvimento de páginas *web* além de integrar seus dados com algum banco de dados, tornando a implementação do CRUD mais simples e fragmentada em etapas, por meio do padrão MVC.

## 3.2 Rafael Couto Silva

Ao realizar esse trabalho pude compreender os benefícios de se usar o ASP .NET Core MVC, que trabalha utilizando o *front-end* e o *back-end* e um único sistema; facilitando e acelerando o processo de desenvolvimento de *sites* simples, feitos somente por uma pessoa.

## 3.3 Thalys Ferreira Correia

Com a realização deste trabalho pude concluir que, ao utilizar a arquitetura de software MVC dividindo o código em três camadas (*view*, *model* e *controller)*, o desenvolvimento dele é facilitado e, assim, o código fica mais organizado e mais limpo, seguindo um padrão.