Área Departamental de Engenharia de Eletrónica e Telecomunicações e de Computadores

Trabalho Prático Parte 2

Autores:	46973	Jorge Alexandre Luzio Simões
	46976	Paulo Jorge da Cruz da Eufémia
	47199	Tiago Luís Lima da Silva

Relatório para a Unidade Curricular de Sistemas de Informação 2 da Licenciatura em Engenharia Informática e de Computadores

Professor: Afonso Remédios

22 – Janeiro – 2022

<< Esta página foi intencionalmente deixada em branco >>

Resumo

O objetivo deste documento é demonstrar a resolução da segunda fase do projeto assim como justificar as decisões tomadas na realização do mesmo.

Será apresentado o diagrama de classes e será explicado o objetivo de cada classe.

Índice

Lista de Figuras

Lista de Tabelas

Listagens

No table of figures entries found.

1. Introdução

O enunciado pretende que se realize uma aplicação em C# que use diferentes implementações de acesso à base de dados, nomeadamente ADO.NET e Entity Framework 6. As funcionalidades a serem implementadas são as do enunciado da parte 1 da alinha 2e a 2i.

2. Desenvolvimento

O diagrama seguinte representa a estrutura deste projeto.

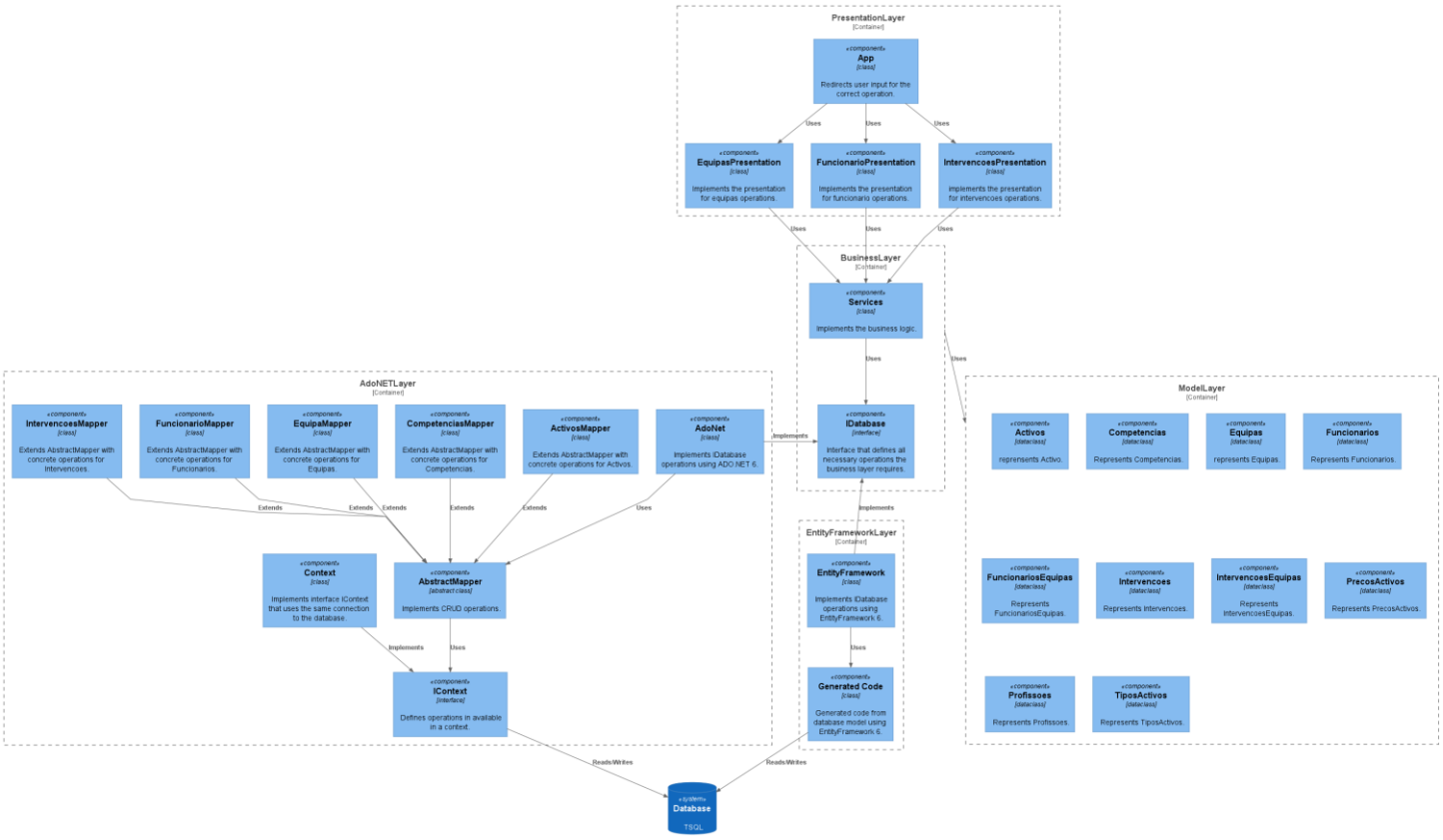

Figura 1 – Diagrama de classes

A camada de apresentação é responsável pela criação da interface e parse dos dados introduzidos pelo utilizador.

A camada de negócio é responsável por validar todas a regras de negócio existentes como por exemplo a data de início de uma intervenção ter de ser maior da data de aquisição do ativo. Qualquer comunicação com a base de dados é realizada a partir de qualquer classe que implemente a interface IDatabase que é passada no construtor dos serviços.

A camada de modelos representa o modelo de dados do problema. Serão estes objetos que serão passados entre cada camada.

A camada ADO.NET e Entity Framework implementam a interface IDatabase usando a tecnologia correspondente.

Visto que as operações que são disponibilizadas para o utilizador são poucas optamos por colocar todas as operações fornecidas pela camada de negócio na mesma classe.

Mas caso houvesse mais operações estas seriam separadas na mesma logica que as da camada de apresentação, onde teríamos diferentes tipos de serviço, por exemplo:

- Serviço de ativos -> interface de ativos para acesso a base de dados -> implementada pela camada de ADO.NET e Entity Framework

Presentation Layer

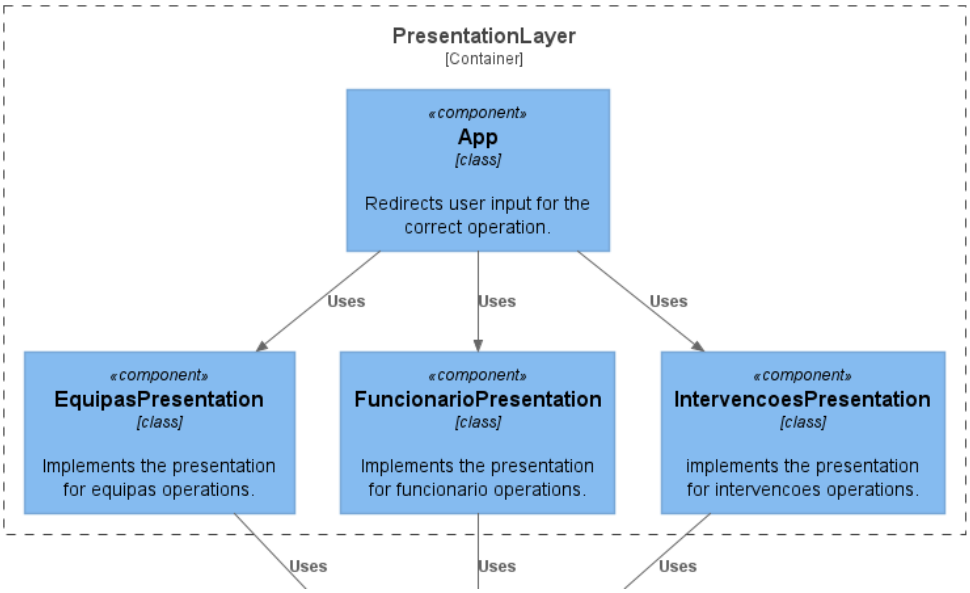

Figura 2 – Diagrama Presentation Layer

As classes neste módulo são responsáveis por criar a interface (em consola) que é mostrada ao utilizador, como também fazer parse da informação por fim, passar esta informação para a camada de negócio correspondente, usando as classes da camada de modelo.

A classe App é responsável por fazer o roteamento das operações pedidas para as classes de apresentação correta.

Cada classe de apresentação é responsável por chamar o serviço correspondente da camada de negócio.

Business Layer

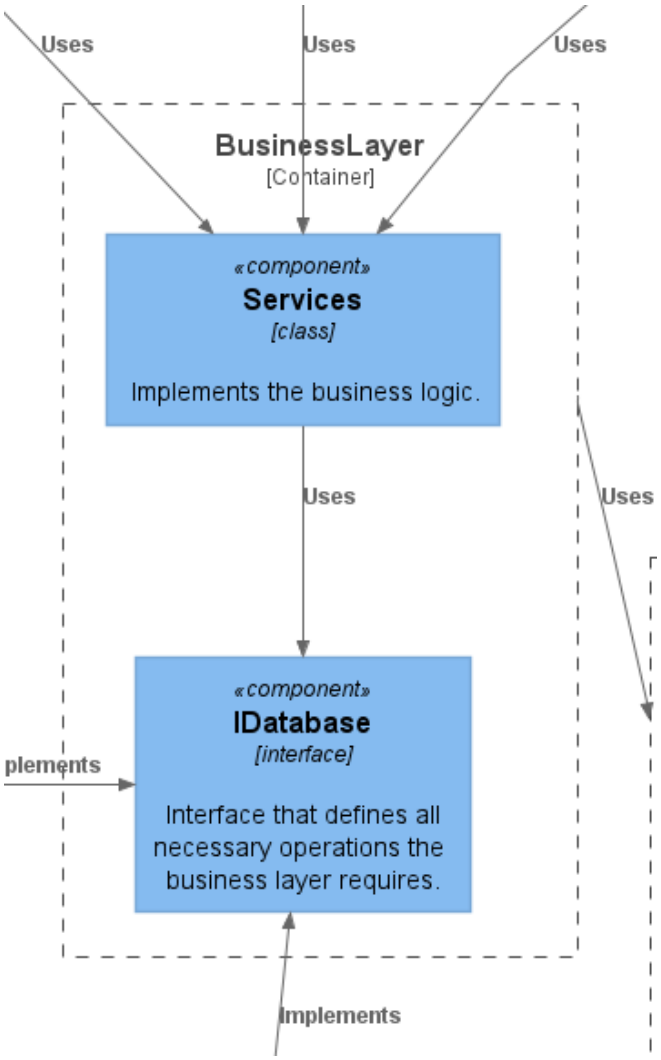

Figura 3 – Diagrama Business Layer

As classes deste módulo são responsáveis por implementação das regras de negócio.

A classe Service implementa todas as operações que estão disponíveis ao utilizador. Como referido antes caso houvesse mais operações estas seriam dividas em diferentes tipos de serviço um para cada modelo.

A interface IDatabase declara os métodos que são precisos fornecer para que a classe Service consiga disponibilizar as operações pretendidas. A logica anterior aplica-se aqui também, onde caso existam mais operações esta classe iria ser desdobrada em várias.

Model Layer

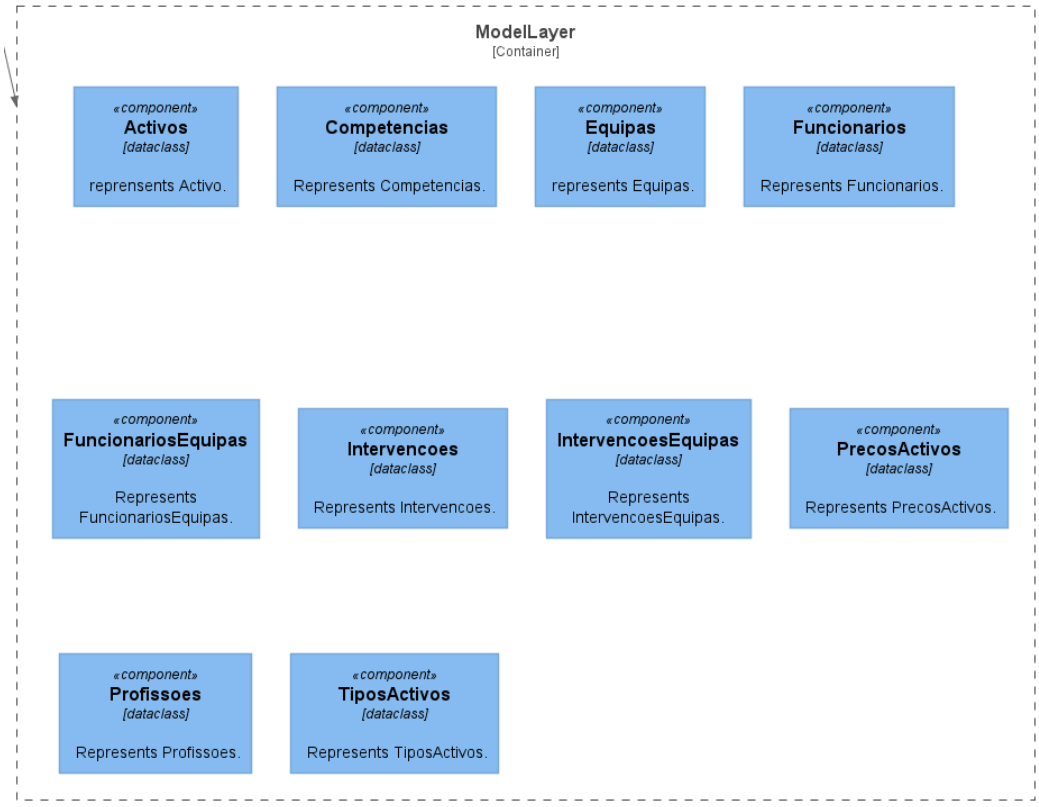

Figura 4 – Diagrama Model Layer

Este módulo contém todas as classes de dados que representam o modelo do problema. Estas classes são utilizadas para transferência de dados entre camadas.

ADO.NET

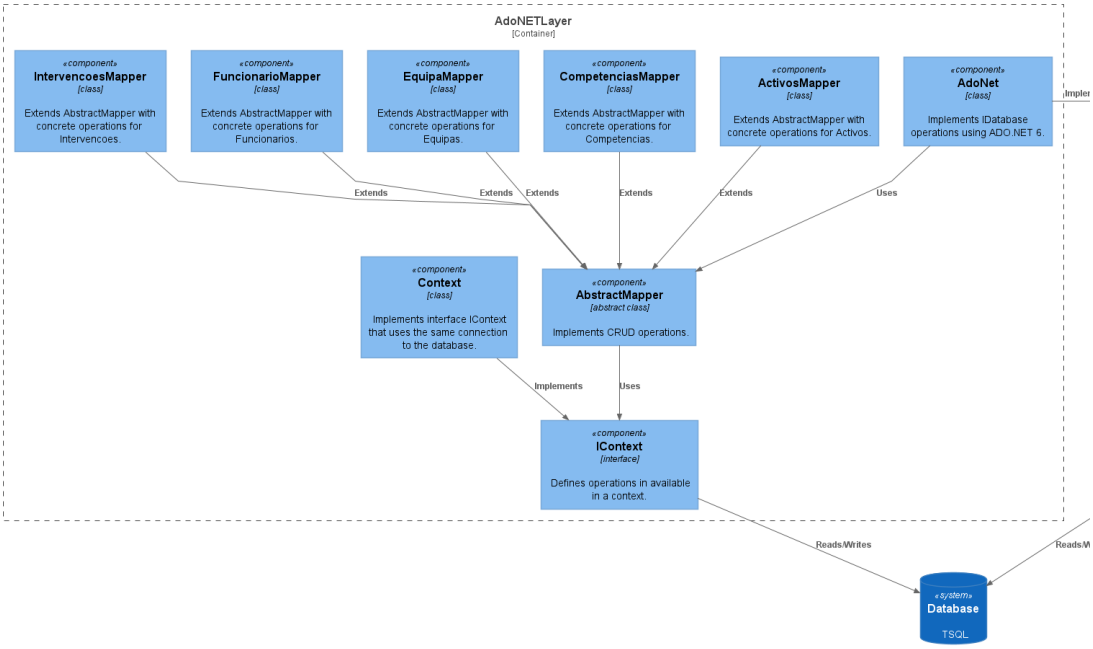

Figura 5 - Diagrama ADO.NET

Este módulo contém a implementação das operações da base dados utilizados pela camada de negócio recorrendo a ADO.NET para sua implementação.

A classe AdoNet é a implementação concreta de IDatabase utilizado no modelo de negócios. Esta classe utiliza os mappers para realização destas operações.

Entity Framework 6

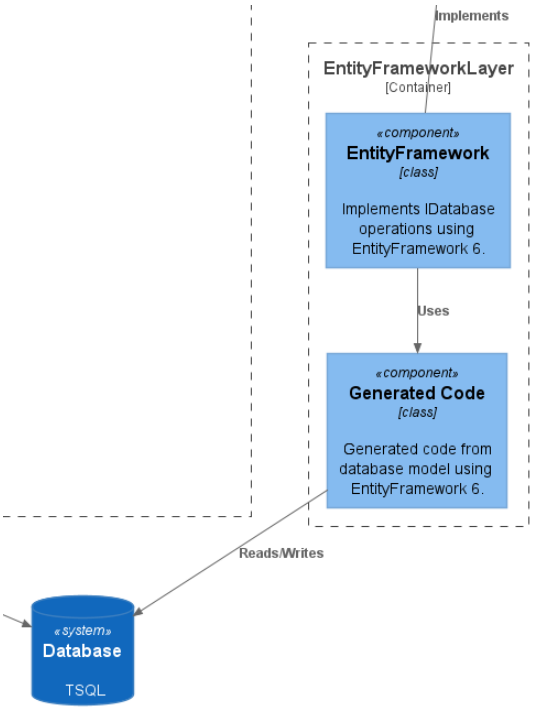

Figura 6 - Diagrama Entity Framework 6.

Este módulo contém a implementação das operações da base dados utilizados pela camada de negócio utilizando a Entity Framework 6 para sua implementação.

O equivalente aos mappers e modelo de dados utilizados no ADO.NET foram gerados automaticamente através da base de dados com a Entity Framework. Por isso a única classe que tivemos de criar foi a EntityFramework que implementa a interface IDatabase recorrendo às classes geradas automaticamente.

Teste de desempenho

Foi realizado teste de desempenho para a alinha 1c entre as tecnologias ADO.NET e Entity FrameWork a tabela seguinte mostra os seguintes resultados.

Tabela 1 -Teste de Desempenho

#	ADO.NET	Entity FrameWork
1	134 ms	218 ms
2	91 ms	126 ms
3	113 ms	110 ms
4	84 ms	116 ms
5	98 ms	103 ms
Média	104 ms	134.6 ms

Como esperado o teste de desempenho demonstra que o ADO.NET é relativamente mais rápido que a Entity FrameWork visto que o ADO.NET está diretamente conectado à base dados enquanto a Entity Framework tem de fazer uma tradução das queries LINQ para SQL o que aumenta o tempo de processamento necessário.

Mas como ADO.NET está diretamente conectado à base de dados tambem implica um maior desenvolvimento de código visto que não temos a geração de código automática que a Entity Framework disponibiliza.

Notas

Configuração da ligação a base de dados

Para alteração da connection string é necessário mudar o ficheiro de configuração App.config no módulo PresentationLayer e IntegrationTests na tag connectionStrings.

Função getFreeEquipa

Devido a uma limitação da Entity Framework funções de SQL que retorna escalares não são implementadas automaticamente na geração de código através da base dados.

Para contornar este problema nós adicionamos o seguinte método manualmente ao ficheiro "TrabSI2.Context.cs" no módulo EntityFrameworkLayer.

```
public int obtainCodigoDeEquipaLivre(int descricao)
{
    var value = Database.SqlQuery<int>("select
ISNULL(dbo.obtainCodigoDeEquipaLivre(@descricao), -1) equipaId", new
System.Data.SqlClient.SqlParameter("@descricao", descricao)).Single();
    return value;
}
```

3. References

[1] Remédios, Afonso; Datia, Nuno;, “Trabalho prático v1.00,” [Online]. Available: https://2122moodle.isel.pt/pluginfile.php/1108507/mod_resource/content/10/TrabalhoSI2-2122I.pdf.