

**Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores**

**Trabalho Prático Parte 1**

|  |  |  |
| --- | --- | --- |
| 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

04 – Dezembro – 2021

<< 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 objectivo de cada classe.

Índice

[1. Introdução 5](#_Toc93760461)

[2. Desenvolvimento 5](#_Toc93760462)

[Presentation Layer 7](#_Toc93760463)

[Business Layer 8](#_Toc93760464)

[Model Layer 9](#_Toc93760465)

[ADO.NET 10](#_Toc93760466)

[Entity Framework 6 11](#_Toc93760467)

[Testes de desempenho 12](#_Toc93760468)

[3. References 13](#_Toc93760469)

Lista de Figuras

Figura 1 – Diagrama de classes 5

Figura 2 – Diagrama Presentation Layer 7

Figura 3 – Diagrama Business Layer 8

Figura 4 – Diagrama Model Layer 9

Figura 5 - Diagrama ADO.NET 10

Figura 5 - Diagrama Entity Framework 6. 11

Lista de Tabelas

Tabela 1 -Teste de Desempenho 12

Listagens

**No table of figures entries found.**

# Introdução

O enunciado pretende que se realize uma aplicação em C# que de use diferentes implementações de acesso a 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.

# Desenvolvimento

Diagram

Description automatically generatedO 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 negocio existentes como por exemplo a data de inicio de uma intervenção ter de ser maior do data de aquisição do activo. Qualquer comunicação com a base de dados é realizada a partir de qualquer class que implemete a interface IDatabase que é passada no constructor.

A camada de modelos representa o modelo de dados do problemas. Serão estes objectos que serão passados entre camadas.

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 negocio na mesma class.

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

* Serviço de activos -> interface de activos para acesso a base de dados -> implementada pela a camada de ADO.net e Entity Framework

## Presentation Layer

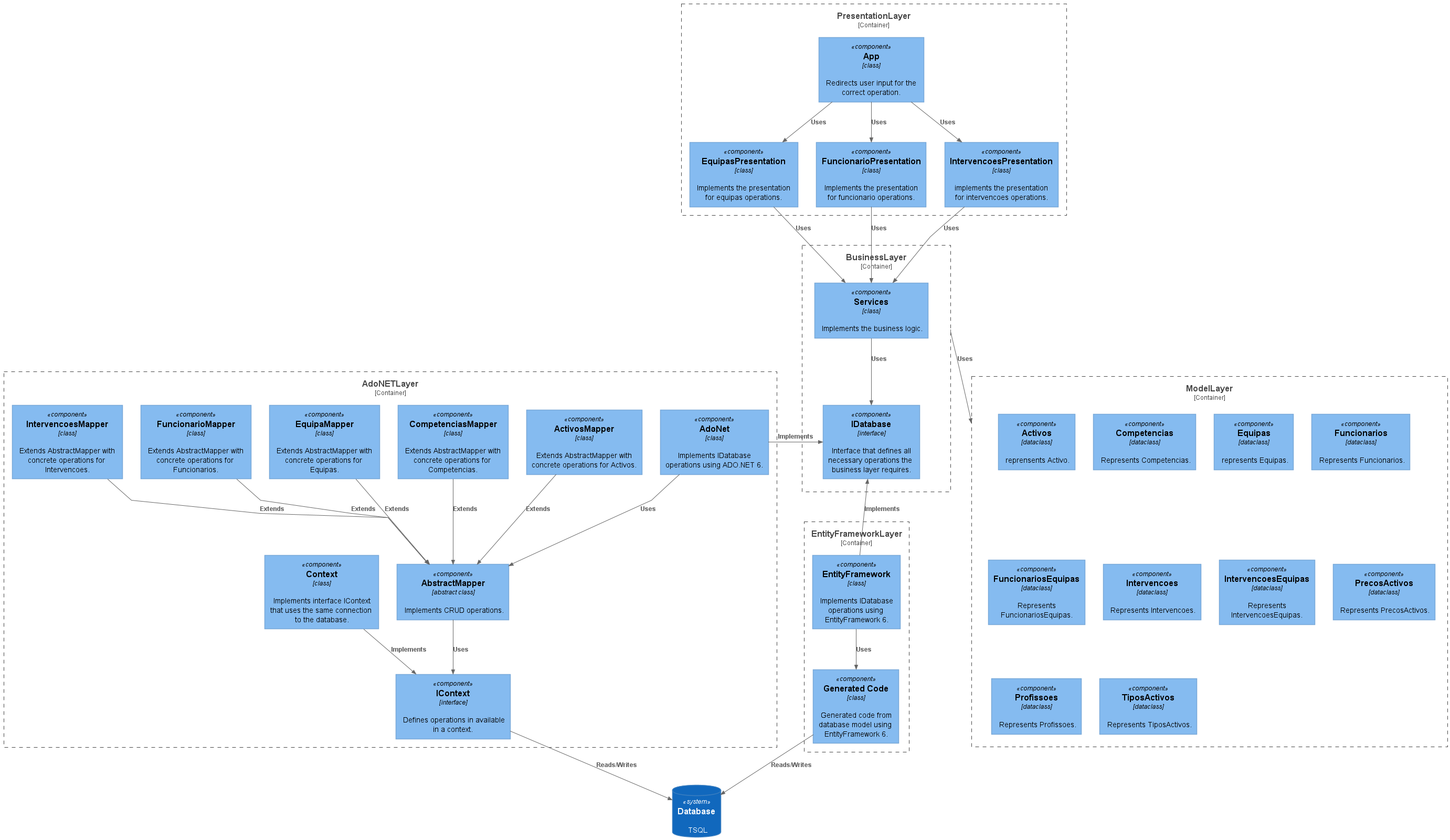


Figura 2 – Diagrama Presentation Layer

As classes neste modulo são responsáveis por criar a interface (em consola) que é mostrada ao utilizador como também fazer parse da informação e por fim passar esta informação para a camada de negocio correspondente.

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

Cada class de apresentação é responsavel por chamar o serviço correspondente da camada de negocio.

## Business Layer

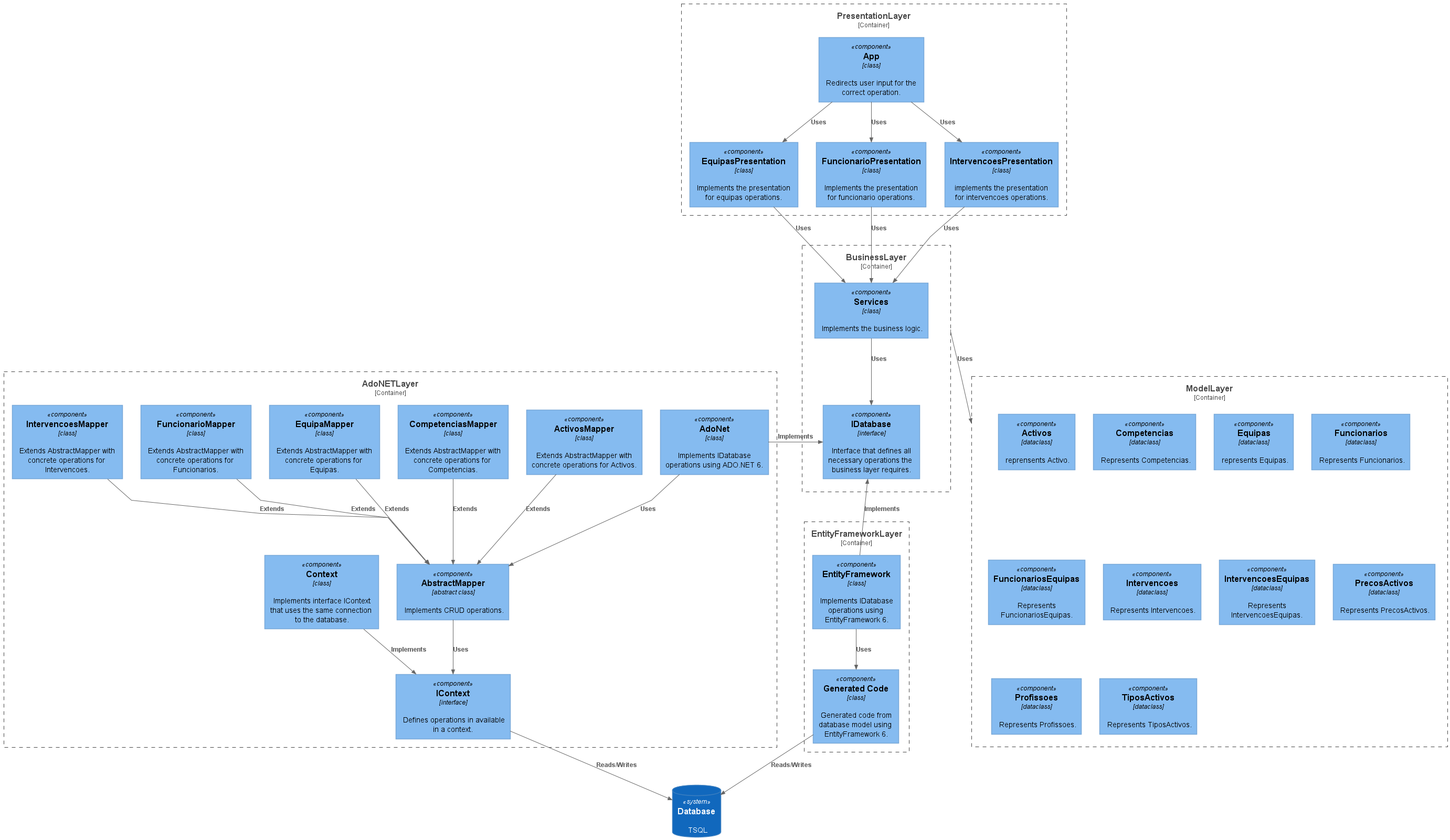


Figura 3 – Diagrama Business Layer

As classes deste modulo são responsáveis por implementação das regras de negocio.

A class Service implementa todas as operações que estão disponiveis 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 metodos que são precisos fornecer para que a class Service funcionar. A logica anteror aplica-se aqui tambem onde caso existam mais operações esta classe iria ser desdobrada em varias.

## Model Layer

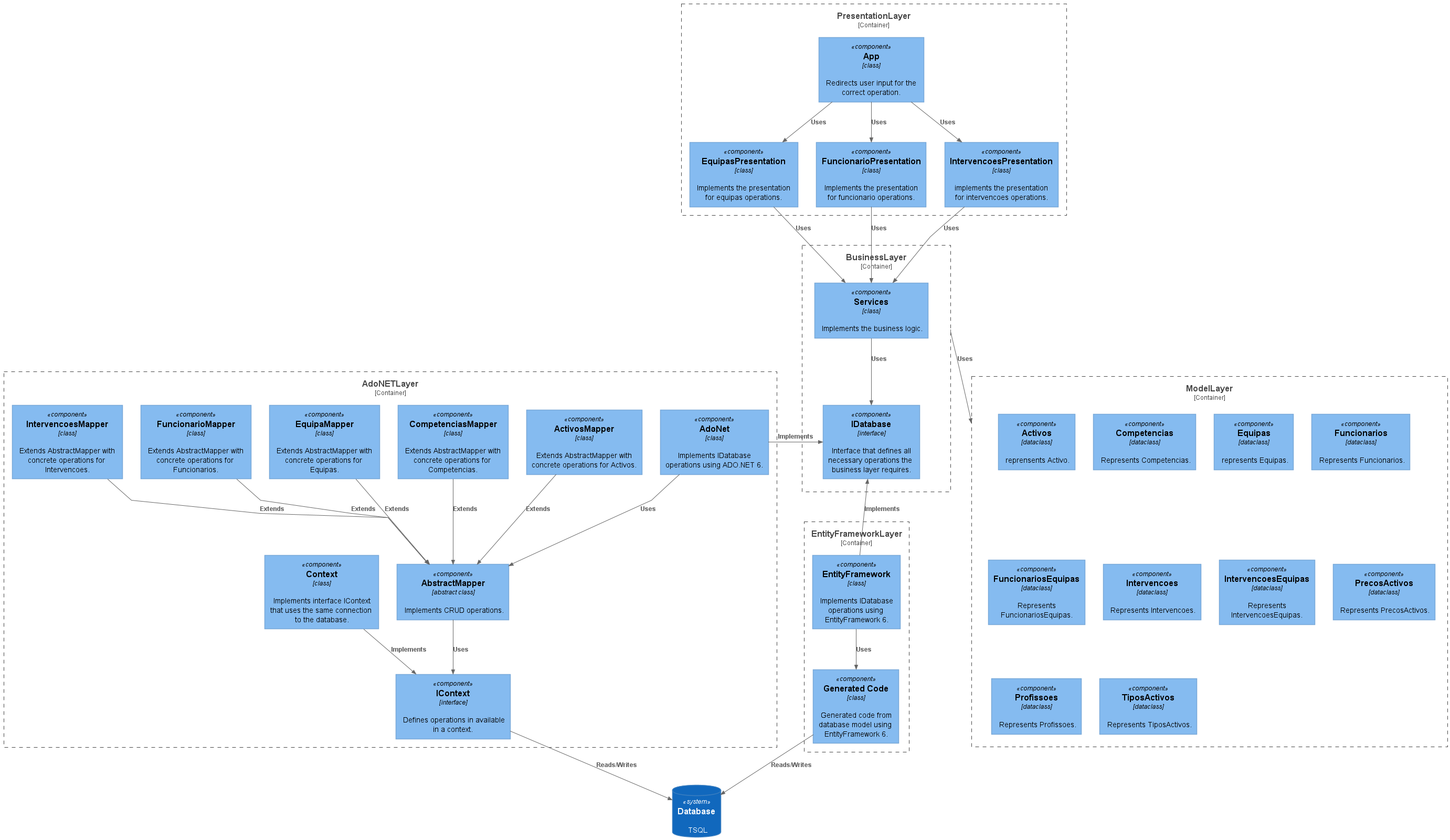


Figura 4 – Diagrama Model Layer

Este modulo contém todas as classes de dados existentes para este projecto. Estas classes são utilizadas para transferência de dados entre as outras camadas.

## ADO.NET

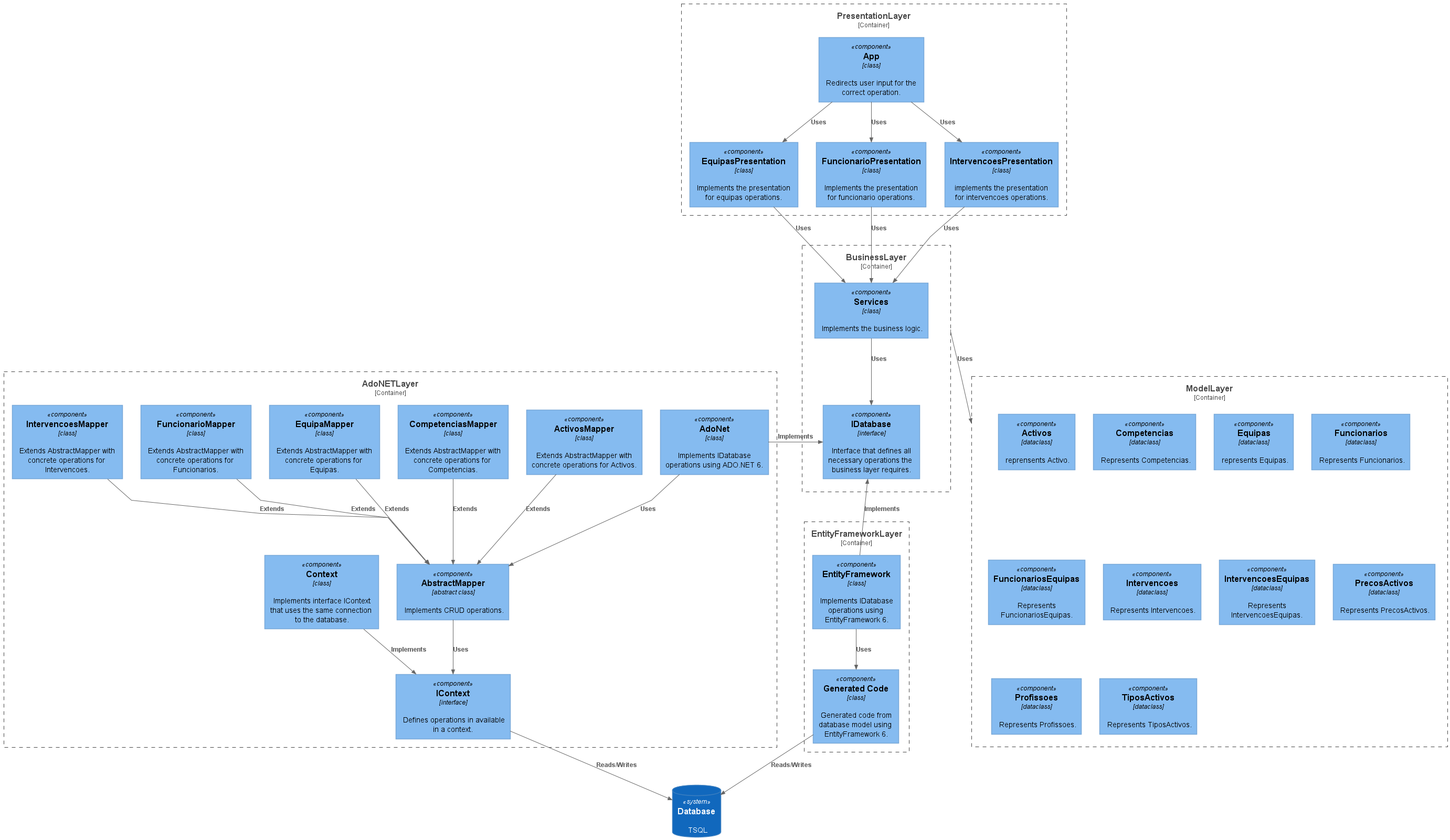


Figura 5 - Diagrama ADO.NET

Este modulo contém a implementação das operações da base dados utilizadas pela camada de negocio utilizando ADO.NET

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

## Entity Framework 6

Diagram

Description automatically generated

Figura 5 - Diagrama Entity Framework 6.

Este modulo contém a implementação das operações da base dados utilizadas pela camada de negocio utilizando a framework Entity Framework 6.

O equivalente aos mappers e modelo de dados utilizados no ADO.NET foram gerados automaticamente através da base de dados com a entity framework 6. Por isso a única classe que criamos foi a EntityFramework que implementa a interface IDatabase recorrendo as classes geradas automáticamente.

## 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 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| Média |  |  |

## Notas

### Configuração da ligação a base de dados

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

# 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. |