

**\*Título do Trabalho de Projecto  
Usando a Primeira Letra na Forma Maiúscula\***

|  |  |
| --- | --- |
|  | \*Fernando Pessoa\* |
|  | \*Ricardo Reis\* |

|  |  |
| --- | --- |
| Orientadores | \*Àlvaro de Campos\* |
|  | \*Alberto Caeiro, SoftCompany\* |

Relatório \*de progresso\* \*beta\* final\* realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2014/2015

\*Maio\* de 2015

**Instituto Superior de Engenharia de Lisboa**  
Licenciatura em Engenharia Informática e de Computadores

**\*Título do Trabalho de Projecto  
Usando a Primeira Letra na Forma Maiúscula\***

|  |  |
| --- | --- |
| \*75463\* | \*Fernando António Nogueira Pessoa\* |
| \*72453\* | \*Ricardo Manuel Augusto dos Santos Reis\* |
|  |  |

|  |  |
| --- | --- |
| Orientadores: | \*Álvaro José Silva Veiga de Campos\* |
|  | \*Alberto Joaquim Alves Caeiro, SoftCompany\* |

Relatório \*de progresso\* \*beta\* final\* realizado no âmbito de Projecto e Seminário,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Verão 2014/2015

\*Maio\* de 2015

# Resumo

O presente documento foi desenvolvido no âmbito da disciplina Sistemas de Informação II de Licenciatura do Curso de Engenharia Informática de Computadores. Trata-se de um relatório da Segunda Fase do Trabalho Prático que visa aplicar a matéria adquirida nas aulas sobre a linguagem SQL, mecanismos transacionais e camadas de acesso a dados com controlo transacional. Para tal, o projeto foi contextualizado num desenvolvimento de uma aplicação com acesso a um sistema de informação para a gestão de mercados financeiros.

Durante o seu desenvolvimento,

Breve descrição do projecto, dos resultados importantes e das conclusões: o objectivo é dar ao leitor uma visão global do projecto (não deve exceder uma página).

**Palavras-chave:**

**ACID: Atomicidade, Consistência, Isolamento e Durabilidade**

**ADO.NET: ActivezX Data Object**

**GUI: Graphic User Interface**

**ORM: Object Relational Mapping**

# Abstract

Abstract text (1 page).

**Keywords:** sorted keyword list, delimited by ;.

**Índice**

[Resumo v](#_Toc27951545)

[Abstract vii](#_Toc27951546)

[Lista de Figuras x](#_Toc27951547)

[Lista de Tabelas xii](#_Toc27951548)

[1. Introdução 1](#_Toc27951549)

[1.1 Organização do documento (Incompleto) 2](#_Toc27951550)

[2. Formulação do Problema 3](#_Toc27951551)

[2.1 Desenvolvimento das Aplicações – Questões surgidas 3](#_Toc27951552)

[2.2 Restrições de Interface do Utilizador 3](#_Toc27951553)

[2.3 Tecnologias usadas 4](#_Toc27951554)

[2.3.1 ADO.NET em modo “conectado” 4](#_Toc27951555)

[2.3.2 ADO.NET Entity Framework 5](#_Toc27951556)

[2.4 Contextualização 6](#_Toc27951557)

[3. Solução 1 - Aplicação com Menu de Configuração 8](#_Toc27951558)

[3.1 Arquitetura da Aplicação 8](#_Toc27951559)

[Referências 10](#_Toc27951560)

[A.1 Diagrama Sequencial da Solução 1 11](#_Toc27951561)

# 

# Lista de Figuras

[Figura 1 - Modelo Conectado e Desconectado 4](#_Toc27951417)

[Figura 2 - Arquitetura do ADO.NET Entity Framework [2] 5](#_Toc27951418)

[Figura 3 - Diagrama sequencial da aplicação com um menu de configuração 9](#_Toc27951419)

# Lista de Tabelas

# Introdução

No âmbito da disciplina Sistemas de Informação II, foi proposto fazer um trabalho composto por duas fases cujos os objetivos são:

* Utilizar a linguagem SQL a um nível avançado, incluindo SQL/PSM:
* Compreender e usar os mecanismos transacionais (ACID) percebendo como podem ser usados para lidar com a concorrência;
* Construir camadas de acesso a dados com controlo transacional;

Nesta fase, segunda, pretende-se:

* Desenvolver uma aplicação em C#;
* Utilizar processamento transacional, concretizado através de mecanismos disponíveis na plataforma .NET;
* Utilizar ADO.NET em modo “conectado”;
* Utilizar ADO.NET Entity Framework para acessos a dados;
* Garantir a correta libertação de ligações e recursos, quando estes não estejam a ser utilizados;
* Garantir a correta implementação das restrições de integridade e/ou lógica de negócio;
* Organizar o código de acesso a dados usando padrões de desenho como Data Mapper e Lazy load;

Para tal, o trabalho foi contextualizado na necessidade de desenvolver um sistema, para a empresa *Pilim*, de informação para a gestão de mercados financeiros.

Na primeira fase do trabalho, foi projetado e construído um sistema de armazenamento e organização de dados em SQL, de modo a que todos os dados do sistema sejam consistentes com as necessidades da empresa.

Na segunda fase, foi desenvolvido uma aplicação que acede ao sistema anteriormente criado. Para o âmbito da disciplina, foram desenvolvidas duas aplicações com tecnologias diferentes de acesso. Na primeira implementação foi usado o ADO.NET em modo “conectado” e na segunda ADO.NET Entity Framework.

## 1.1 Organização do documento (Incompleto)

O restante relatório é constituído por cinco capítulos principais.

O primeiro capitulo trata-se da formulação do problema, onde é relatado todos os problemas encontrados no inicio e durante a execução da segunda fase do trabalho.

O segundo capitulo descreve a primeira solução implementada e justifica a razão para qual a solução foi abandonada.

O terceiro capitulo apresenta a solução encontrada e implementada para esta fase.

O quarto capitulo demonstra todas as análises efetuadas durante a implementação da solução implementada, manifestando que testes foram efetuados para garantir a boa execução da aplicação e responde também algumas perguntas mencionadas no enunciado desta segunda fase do trabalho.

No quinto e último capitulo são colocadas todas as conclusões que os alunos obtiveram durante a execução desta fase.

# Formulação do Problema

Pretende-se que nesta segunda fase do trabalho se crie aplicações que usem diferentes *frameworks* de acesso a dados. As aplicações devem ter como metas a reutilização de código, fácil de manutenção e eficiência, e serem independentes do modo de acesso a dados.

Na secção 2.1, relata-se as questões que houve no inicio e na implementação das aplicações desenvolvidas. Na secção 2.2, é exposto as restrições que a interface do utilizador pode ter. Na secção 2.3, é explicado as tecnologias que são usadas para o desenvolvimento da aplicação.

## 2.1 Desenvolvimento da Segunda Fase – Questões surgidas

Ao analisar o enunciado do trabalho, vários problemas surgiram para desenvolver as aplicações como:

* Vai ser necessário modificar alguma implementação realizada na fase anterior?
* Fazer uma aplicação em consola ou numa GUI?
* Para aceder aos dados, o utilizador introduz as suas credenciais? As credenciais estão guardadas num ficheiro proprietário? Ou as credenciais estão já guardadas no código (*hardcoded)*?
* Faz-se uma aplicação que tenha um menu de configuração onde o utilizador consegue escolher que tipo de tecnologia a aplicação usa ou duas aplicações com tecnologias diferentes?
* No inicio da aplicação, testa-se o acesso aos dados?

Estas foram umas de muitas questões que surgiram no inicio da implementação, no entanto outras foram surgindo durante a implementação

.

## 2.2 Restrições de Interface do Utilizador

Enquanto se planeava a interface que o utilizador iria ter com a aplicação para aceder aos dados do sistema, certos problemas surgiram, como:

* O utilizador interage diretamente com o sistema de armazenamento de dados, podendo construir as suas próprias *queries*?
* Deixamos o utilizador restringido a certos comandos sendo depois a responsabilidade de um programador futuro acrescentar outros comandos que o utilizador precise?

## 2.3 Tecnologias usadas

Para ir de encontro às metas e objetivos anteriormente delineados, foi necessário fazer um estudo das tecnologias que são necessárias usar, ADO.NET em modo “conectado” e ADO.NET Entity Framework.

Tradicionalmente, o processamento de dados baseia-se principalmente em um modelo de duas camadas baseado em conexão. À medida que o processamento de dados utiliza cada vez mais arquiteturas de várias camadas, os programadores estão a mudar para uma abordagem desconectada para fornecer melhor escabilidade para as suas aplicações [1].

### 2.3.1 ADO.NET em modo “conectado”

O modo conectado do ADO.NET é um modo que foi explicitamente desenhado para manipulação de dados, apenas leitura e um acesso rápido a dados. Como é ilustrado na Figura 1, este modo é composto por um objeto *Connection*, que providencia conectividade à fonte de dados, um objeto *Command*, que permite o acesso aos comandos da base de dados para retornar e modificar dados, executar procedimentos e enviar ou recuperar informações de parâmetros, um objeto *Data Reader*, apresenta uma *Stream* de alta performance de dados da fonte de dados, e um objeto *Data Adapter* que promove uma ligação ponte entre a fonte e dados e um objeto *Data Set* [1].

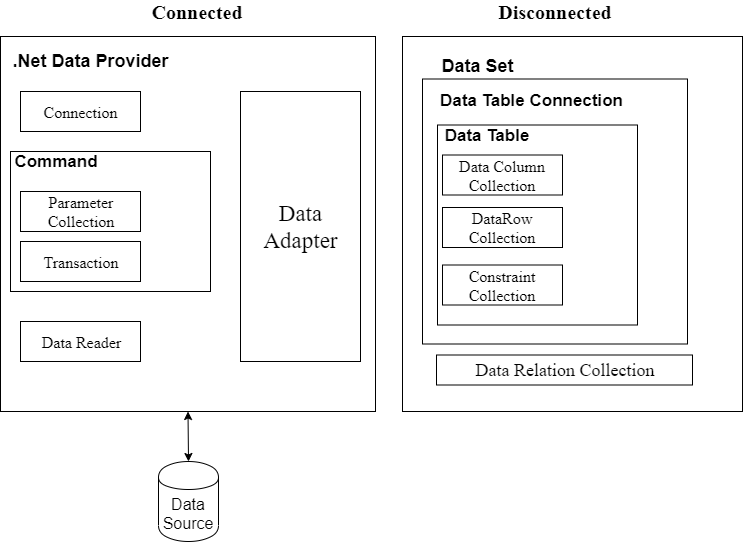


Figura 1 - Modelo Conectado e Desconectado

No entanto, como pedido no enunciado, só vai ser utilizado o ADO.NET em modo “conectado”, o que implica que não se vai usar objetos do modo “desconectado”. Assim, não vai ser utilizado objetos *Data Adapter*.

### 2.3.2 ADO.NET Entity Framework

O Entity Framework é uma ferramenta da Microsoft, pertencente ao pacote ADO.NET, que permite ao programador trabalhar com banco de dados relacional de maneira abstrata [2].

Esta ferramenta utiliza uma técnica para diminuir a complexidade e o constrangimento da programação orientada a objetos com bases de dados relacionais denominada por ORM (*Object Relational Mapping*) ou mapeamento de objeto relacional. A partir desta técnica é possível apresentar tabelas como classes e os registos de cada tabela como registos nas classes correspondentes.

Com esta ferramenta, é possível desenvolver aplicações com uma estrutura em camadas e, assim, o programador pode focar mais nas regras do negócio, abstraindo da camada da base de dados, como é representado na Figura 2.

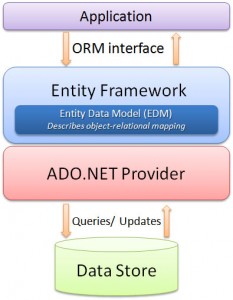


Figura - Arquitetura do ADO.NET Entity Framework [2]

Para trabalhar com o Entity Framework, existem 3 possíveis abordagens, *Database-First*, *Model-First* e *Code-First*.

Na primeira abordagem, a base de dados foi criada previamente e as entidades são surgidas baseadas na base de dados. Na segunda abordagem, é criado um modelo conceitual primeiro e através deste é gerado um script para criar a base de dados. Na terceira abordagem, são criados classes primeiro e através destas toda a lógica da entidade é criado, permitindo que com este modelo seja gerado a base de dados.

## 2.4 Contextualização

O presente trabalho contextualiza-se na necessidade de desenvolver um sistema de informação para a gestão de mercados financeiros para a empresa *Pilim*.

Este sistema é constituído por um mercado financeiro. Um mercado financeiro pode ser constituído por um conjunto de instrumentos. Tanto os instrumentos como os mercados financeiros têm um registo diário com registos próprios e fundamentais para a sua avaliação quantitativa e qualitativa. Existe também um conjunto de informação denominada por triplos onde identifica um instrumento existente ou inexistente na base dados com informação do seu valor numa certa altura temporal.

O sistema também contem um armazenamento de clientes que têm contactos, email e/ou telefónico. Cada cliente pode ter também um ou mais portfólios. Cada portfólio pode ter um certo número de ações ou posições de um ou vários instrumentos.

Nesta segunda fase do trabalho, é preciso desenvolver uma aplicação para aceder a estes dados, obedecendo às necessidades e regras que a empresa impõe.

# Solução 1 - Aplicação com Menu de Configuração

Para ir ao encontro das metas e objetivos, foi pensado desenvolver uma única aplicação com um menu de configuração. Esta solução dá ao utilizador a oportunidade de escolher que tipo de tecnologia a aplicação usaria para aceder aos dados.

## 3.1 Arquitetura da Aplicação

Antes de se ter começado a fazer código, foi estudado que tecnologias se iam usar e como podemos implementá-las.

A primeira tecnologia que é necessário implementar

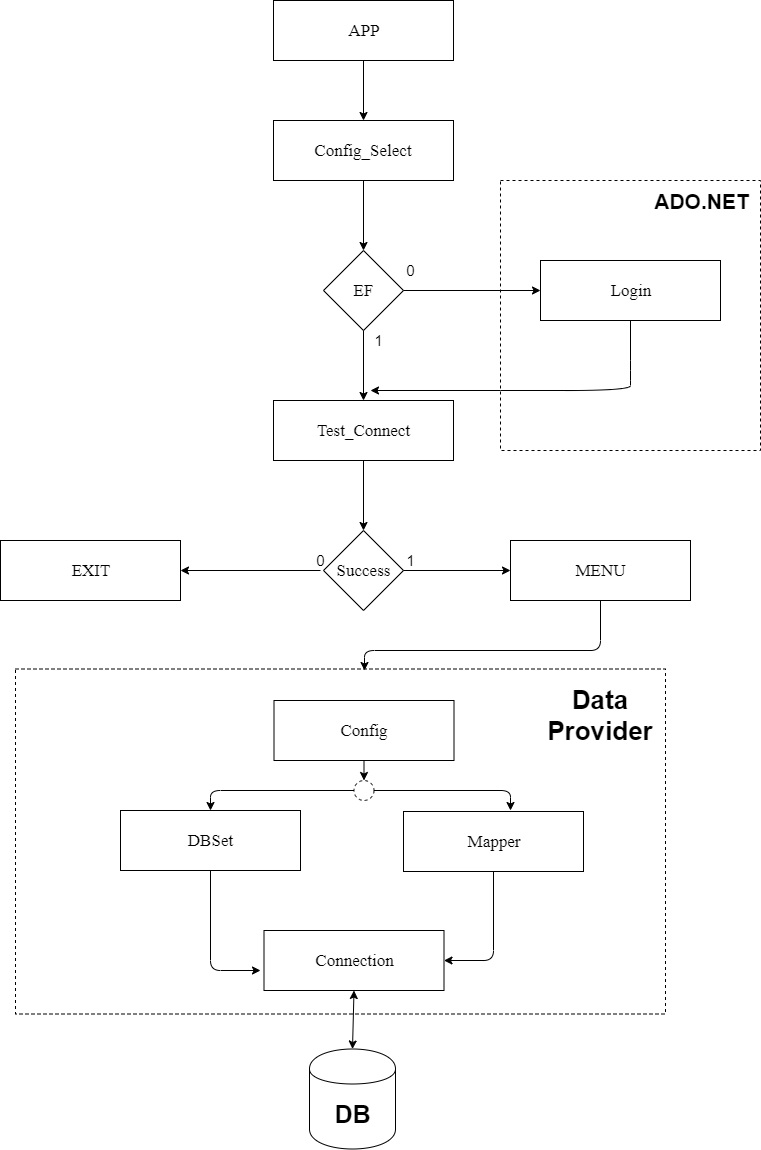


Figura - Diagrama sequencial da aplicação com um menu de configuração

# Referências

|  |  |
| --- | --- |
| [1] | Microsoft, 30 03 2017. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-architecture. [Acedido em 12 2019]. |
| [2] | T. Leite, Redspark, 25 11 2015. [Online]. Available: http://redspark.io/entendendo-o-entity-framework/. [Acedido em 12 2019]. |

# A.1 Diagrama Sequencial da Solução 1