

**Trabalho Prático Sistemas de Informação II**

(Fase I)

|  |  |
| --- | --- |
|  | Joana Campos |
|  | Nuno Cardeal  Carolina Couto |

|  |  |
| --- | --- |
| Docentes | Afonso Remédios |
|  | Nuno Datia |

Relatório de progresso realizado no âmbito de Sistemas de Informação II,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Inverno 2019/2020

Novembro de 2019

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

**Trabalho Prático Fase I**

|  |  |
| --- | --- |
| Grupo | 14 |
| 44792 | Joana Filipa Alves de Campos |
| 44863  44871 | Nuno Rafael Mourão Cardeal  Carolina Alexandra Roque Couto |

|  |  |
| --- | --- |
| Docentes | Afonso dos Santos Remédios |
|  | Nuno Miguel Soares Datia |

Relatório de progresso realizado no âmbito de Sistemas de Informação II,  
do curso de licenciatura em Engenharia Informática e de Computadores  
Semestre de Inverno 2019/2020

Novembro de 2019

# Resumo

Este trabalho foca-se na criação de um sistema de gestão de base de dados para uma empresa, *Pilim*, de gestão de mercados financeiros. Esta fase do trabalho foca-se na criação do modelo, quer físico quer concetual do sistema de informação e do desenvolvimento do *software* para a criação em base de dados do mesmo em linguagem *Transact Structured Query Language*, o mesmo *software* deve contar com algumas funcionalidades que permitam um correto e estável funcionamento do sistema por forma que numa segunda fase deste trabalho seja possível o desenvolvimento de uma aplicação para o preenchimento e a manipulação da base de dados. Sendo este um trabalho educacional não é pretendido que o mesmo fique a funcionar por completo, mas sim que os seus desenvolvedores aprendam e expliquem como este trabalho pode realmente ser bem executado, dando azo aos mesmos de poderem tirar e discutir as suas próprias interpretações sobre os problemas que vão surgindo durante o trabalho.

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:** lista de palavras-chave, ordenadas alfabeticamente, separadas por ;.

# Abstract

Abstract text (1 page).

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

# Agradecimentos

Texto dos agradecimentos. É opcional.

**Índice**

[Resumo iv](#_Toc24303627)

[Abstract vi](#_Toc24303628)

[Agradecimentos viii](#_Toc24303629)

[Lista de Figuras xii](#_Toc24303630)

[Lista de Tabelas xiv](#_Toc24303631)

[1. Introdução 1](#_Toc24303632)

[1.1 Enunciado 1](#_Toc24303633)

[1.2 Espectativas 2](#_Toc24303634)

[1.3 Organização do documento 2](#_Toc24303635)

[2. Formulação do Problema 3](#_Toc24303636)

[2.1 Modelos de Dados 3](#_Toc24303637)

[2.2 Bases de Dados Dinâmicas 3](#_Toc24303638)

[2.3 Transações 4](#_Toc24303639)

[2.4 Níveis de Isolamento 4](#_Toc24303640)

[2.5 Vistas 4](#_Toc24303641)

[2.6 Gatilhos 4](#_Toc24303642)

[2.7 Funções 4](#_Toc24303643)

[2.8 Importância dos Testes 4](#_Toc24303644)

[3. Solução Proposta 5](#_Toc24303645)

[3.1 Modelo EA 5](#_Toc24303646)

[3.2 Modelo Relacional 5](#_Toc24303647)

[3.3 Criação do modelo físico 5](#_Toc24303648)

[3.4 Remoção da base de dados 5](#_Toc24303649)

[3.5 Preenchimento Inicial da Base de Dados 6](#_Toc24303650)

[3.6 Inserção, Atualização e Remoção de um Cliente 6](#_Toc24303651)

[3.6.1 Inserção e Atualização de um Cliente 6](#_Toc24303652)

[3.6.2 Remoção de um Cliente 6](#_Toc24303653)

[4. Avaliação Experimental 6](#_Toc24303654)

[4.1 Nome da primeira secção deste capítulo 6](#_Toc24303655)

[4.2 A segunda secção deste capítulo 7](#_Toc24303656)

[4.2.1 A primeira sub-secção desta secção 7](#_Toc24303657)

[4.2.2 A segunda sub-secção desta secção 7](#_Toc24303658)

[4.3 Análise de resultados 7](#_Toc24303659)

[5. Conclusões 9](#_Toc24303660)

[Referências 10](#_Toc24303661)

[A.1 Modelo EA 11](#_Toc24303662)

[A.2 Modelos de dados 14](#_Toc24303663)

# 

# Lista de Figuras

[Figura 1 - Legenda da figura com o logotipo do ISEL. 1](#_Toc416101905)

[Figura 2 - Legenda da figura com o logotipo do ISEL – versão 2. 5](#_Toc416101906)

[Figura 3 – Diagrama de casos de utilização. 15](#_Toc416101907)

# Lista de Tabelas

[Tabela 1 -Um exemplo de legenda de tabela. Prazos de entrega de Projecto e Seminário, 4](#_Toc416101908)

# Introdução

Este é o início do capítulo.

Exemplo de indentação do segundo parágrafo.

## 1.1 Enunciado

A empresa *Pilim* pretende desenvolver um sistema de informação para a gestão de mercados financeiros. Um mercado financeiro é caracterizado por um código (único), uma descrição e um nome curto. Para cada dia e para cada mercado, são registados o valor do índice/mercado (soma de todos os valores de abertura dos seus instrumentos, calculado no fim do dia), o valor de abertura (valor de índice do dia útil anterior) e a variação diária (em euros). Um mercado é constituído por um conjunto de instrumentos financeiros (e.g. ações de empresas), doravante designado simplesmente por instrumento. Um instrumento é caracterizado por um código único, designado de *International Securities Identification Numbers* (ISIN)1 e uma descrição. A empresa recebe triplos de um sistema externo, constituídos por

*< identificador; datatempo; valor >* onde

*identificador* identifica um instrumento que pode ou não estar registado no sistema,

*datatempo* representa um instante temporal com granularidade ao segundo e

*valor* indica o valor do instrumento em euros.

Todos os triplos têm de ficar registados no sistema de informação da empresa de forma persistente. Para cada instrumento é mantido o registo diário do valor mínimo, valor máximo, valor de abertura e valor de fecho. Existem associados a cada instrumento dados fundamenteis, constituídos pelo valor de variação diária (diferença entre os valores máximo e mínimo), valor atual, média a 6 meses do valor de fecho de cada dia, valor variação a 6 meses, percentagem de variação diária e percentagem de variação a 6 meses. O sistema de informação mantém um registo de clientes, sendo cada um caracterizado por um número de identificação fiscal (único), um número de cartão de cidadão (único) e um nome.

Cada cliente tem um conjunto de contactos (pelo menos 1). Um contacto é caracterizado por um código (único) e por uma descrição. Para os contactos telefónicos é necessário manter o número de telefone e o indicativo. Para os e-mails apenas é necessário manter o endereço. Os clientes têm um portfólio de instrumentos, designados de posições, onde é registada a quantidade de cada instrumento (e.g. 50 ações). Um portfólio tem um nome (único para cada cliente) e um valor total, que resulta do somatório do produto do valor atual dos instrumentos e da quantidade de cada um. Deve ser possível saber o valor total de cada posição. O sistema deve garantir que sempre que o valor diário de um instrumento é alterado, é mantida coerência com o valor diário registado para o mercado.

1 Para mais detalhes consultar https://www.isin.org/

## 1.2 Espectativas

É esperado que no final do trabalho consigamos realizar os seguintes pontos:

* Desenvolver um modelo de dados adequado aos requisitos, normalizado até à 3NF;
* Conceber e implementar uma solução baseada em bases de dados dinâmicas, adequadas aos requisitos;
* Utilizar corretamente controlo transacional;
* Utilizar corretamente níveis de isolamento;
* Utilizar corretamente vistas, justificando o seu uso na solução;
* Utilizar corretamente procedimentos armazenados, justificando o seu uso na solução;
* Utilizar corretamente gatilhos, justificando o seu uso na solução;
* Utilizar corretamente funções, justificando o seu uso na solução;
* Desenvolver código de teste, em T-SQL, para cada uma das funcionalidades pretendidas;
* Escrever um relatório técnico sobre o trabalho desenvolvido.

## 1.3 Organização do documento

O restante relatório encontra-se organizado da seguinte forma: Na secção 2 existe a formulação do problema onde é apresentado tudo o que necessitamos de usar para cumprir as expectativas deste trabalho e ainda uma justificação do porquê de ser útil usar tais ferramentas na implementação desta fase do trabalho. A secção 3 é a explicação de como usamos cada ferramenta e do porquê de a termos usado. Na secção 4 indicamos uma avaliação experimental onde corremos vários testes e avaliamos se estão a ter ou não o resultado pretendido, e em caso negativo perceber e explicar como poderia estar a suceder bem. Por fim na 5ª secção retiraremos as conclusões sobre este trabalho.

# Formulação do Problema

Com este trabalho é pretendido que os alunos aprendam a desenvolver modelos de dados adequados a diferentes problemas e normalizados corretamente na 3NF, aprender como funcionam bases de dados dinâmicas e conceber uma de acordo com os modelos de dados realizados, utilizar o controlo transacional, os níveis de isolamento, vistas, gatilhos e funções em *T-SQL* e criar na mesma linguagem testes para o código desenvolvido. Na secção 2.1 fala acerca dos modelos de dados e da sua importância para o desenrolar deste trabalho. A secção 2.2 apresenta como são elaboradas bases de dados dinâmicas. Na secção 2.3 aborda-se a necessidade de transações. A secção 2.4 trata dos aspetos referentes aos níveis de isolamento. A secção 2.5 refere a boa utilização de vistas. Nas secções 2.6 e 2.7 abordam-se os conteúdos referentes a gatilhos e funções respetivamente. Por fim a secção 2.8 sublinha a importância da utilização de testes para o desenvolvimento do trabalho.

Estamos no início do novo capítulo. Aqui podemos colocar algum texto introdutório e de resumo do conteúdo do capítulo. Por exemplo, a secção 2.1 trata aspectos referentes às citações de bibliografia. Na secção 2.2 apresenta-se um exemplo de enumeração de conteúdos. O uso de tabelas é exemplificado na secção 2.3. Nas secções 2.4 e 2.5 abordam-se expressões matemáticas e o uso de figuras de grandes dimensões.

## 2.1 Modelos de Dados

Existem 2 tipos de modelos de dados, o concetual e o físico. Tal como o próprio nome indica o modelo concetual serve para uma ajuda na conceção do sistema de gestão de base de dados, dando uma ideia visual de como a base de dados está conectada, sendo usado para este trabalho o Modelo Entidade-Associação que dá uma boa representação de quais são os diversos pontos no sistema de gestão de base de dados que têm de ter uma maior atenção na altura do seu desenvolvimento. O modelo físico é uma representação mais real de como é o sistema de gestão de base de dados, sendo que para o seu entendimento tem de haver uma maior atenção do leitor visto que é um modelo mais descritivo e que tem um maior aprofundamento de relações. O modelo físico escolhido foi o Modelo Relacional que tem uma boa ambiguidade com o Modelo Entidade-Associação e as características do modelo físico.

## 2.2 Bases de Dados Dinâmicas

Para este trabalho é necessário usar bases de dados dinâmicas, visto serem as bases de dados mais usadas em sistemas de gestão, uma base de dados dinâmica é uma base de dados onde se pode inserir e remover valores de uma tabela por forma que a própria base de dados mantenha a coerência em outras tabelas, para isso é necessária a utilização de gatilhos e transações visto que estes servem para dar estabilidade à base de dados e vigiar quando existe inserção o remoção de dados.

## 2.3 Transações

As transações servem para manter a estabilidade de uma base de dados de tal maneira que se acontecer algum erro durante a execução do código a base de dados retorna ao seu último estado mais estável.

## 2.4 Níveis de Isolamento

Existem diversos tipos de isolamento, estes servem para conter as transações de modo que elas trabalhem só e apenas com o que é pretendido, se uma transação tentar fazer uma ação fora do seu nível de isolamento a mesma não será executada.

## 2.5 Vistas

Vistas são tabelas “fictícias” nas quais não se pode inserir, remover nem atualizar dados, visto que estas tabelas na verdade não existem, mas retiram informação de tabelas existentes, são bastante uteis na comparação de dados de diferentes tabelas ou na criação de listas

## 2.6 Gatilhos

Os gatilhos são a maneira que um programador tem de saber se a base de dados foi alterada, eles, quando criados, ficam sempre a vigiar alguma alteração à entidade que ficou de vigiar, como tabelas, vistas ou até mesmo bases de dados inteiras, se a alteração escolhida pelo programador tiver acontecido então executa o código que está dentro do gatilho.

Geralmente estes servem para manter a coerência na base de dados quando existe uma atualização, remoção ou inserção de dados.

## 2.7 Funções

Funções são troços de código que servem maioritariamente para manipular e processar dados, uma função pode receber parâmetros e tem sempre de retornar um resultado, no final da função todas as modificações feitas são realizadas na base de dados.

## 2.8 Importância dos Testes

Os testes são importantes para ter a certeza que quando acabado sistema de gestão de base de dados tem todas as funcionalidade esperadas, de modo que ela possa cumprir a sua função corretamente quando for necessária, assim, se houver alguma inconsistência na base de dados é nesta parte que ela é descoberta e corrigida, por forma a ficar uma base de dados estável, coerente e dinâmica.

# Solução Proposta

Para a realização deste trabalho começamos por desenvolver os modelos Entidade-Associação e Relacional e colocar os mesmos na 3NF por forma que fosse possível ter uma base de dados bem planeada e de fácil manipulação. De seguida começamos o desenvolvimento da base de dados, em *T-SQL*, realizando *scripts* que criassem tabelas e as apagassem, preenchessem-nas e também removessem todo o seu conteúdo. Criou-se transações para quando se inserisse removesse ou se atualizasse dados de um cliente ou de um mercado, de forma a não perder dados caso estas ações não fossem corretamente realizadas, criou-se diversas funções e procedimentos que nos auxiliassem na manipulação da base de dados e na sua procura. Por fim implementou-se também testes para o auxílio de uma melhor compreensão se tudo funcionava como suposto.

## 3.1 Modelo EA

Começamos por realizar o Modelo Entidade-Associação pois permitiu-nos que depois de tirarmos uma primeira interpretação do enunciado fazer alterações, explorar mais o enunciado e tirar interpretações diferentes e mais aprofundadas, de forma que quando tivéssemos um Modelo EA bem estruturado pudéssemos passar para o modelo físico visto que existe uma maior facilidade de criar um modelo físico olhando para um modelo concetual.

O nosso Modelo EA está representado no Anexo I e é constituído da seguinte forma:

## 3.2 Modelo Relacional

A partir do Modelo EA foi possível realizar o Modelo Relacional para assim depois se proceder à realização do modelo físico. Depois de alguns ajustes e normalizações até atingir a 3NF o nosso modelo Relacional está apresentado no Anexo II deste relatório.

## 3.3 Criação do modelo físico

Depois do Modelo Relacional concluído criou-se o modelo físico criando todas as tabelas na base de dados disponibilizada pelos docentes, e com os respetivos atributos, chaves primárias e chaves estrangeiras retirados do Modelo Relacional.

## 3.4 Remoção da base de dados

Fez-se um *script* para a remoção do modelo físico caso o mesmo exista, removendo as tabelas por ordem contrária à que foram criadas, de forma a não existir conflito das chaves estrangeiras.

## 3.5 Preenchimento Inicial da Base de Dados

Foi criado um terceiro *script* para o preenchimento inicial da base de dados, este é apenas auxiliar nos testes que se irão realizar mais tarde, caso seja necessária a entrega da base de dados a um cliente este *script* não deveria ser facultado e a base de dados deveria ir vazia.

## 3.6 Inserção, Atualização e Remoção de um Cliente

Para a inserção, atualização e remoção de um cliente na base de dados decidiu-se separar em dois procedimentos, uma para a inserção e atualização e outro para a remoção, em ambos os procedimentos utiliza-se transações com isolação de nível de leitura e escrita por forma À mesma poder ler e comparar dados na base de dados e saber se pode escrever ou não e onde pode escrever, utilizamos transações pois a nossa implementação conta com uma possível exceção e caso a mesma aconteça a base de dados deve voltar ao estado anterior.

### 3.6.1 Inserção e Atualização de um Cliente

Para este procedimento recebemos os dados fundamentais de um cliente *nif*, *ncc* e *name* e é verificado se o cliente já está inserido na base de dados, caso o mesmo já se encontre apenas se faz uma atualização dos seus dados, caso contrário ele é inserido pela primeira vez como um novo cliente.

### 3.6.2 Remoção de um Cliente

Para a implementação deste procedimento apenas é necessário receber como parâmetro a chave primária da tabela *CLIENT*, o *nif*, e durante a transação é visto se o cliente está inserido na base de dados, e caso esteja é removido da mesma, removendo o cliente com o *nif* igual ao recebido e a tabela de ligação entre o cliente e o seu portefólio, *CLIENT\_PORTFOLIO*, também tem este tuplo removido, no entanto escolheu-se não remover o portefólio da tabela *PORTFOLIO* pois se mais tarde o cliente reentrar na base de dados já tem o seu portefólio criado.

## 3.7 Inserção, Atualização e Remoção de um Mercado

Esta implementação tem a mesma ideia da implementação acima, usando o mesmo tipo de transações e nível de isolamento em cada uma delas, com algumas pequenas alterações relativas aos parâmetros recebidos e às comparações feitas.

### 3.7.1 Inserção e Atualização de um Mercado

Este procedimento recebe os parâmetros *description*, *name* e *code* que são os atributos da tabela *MARKET* e de maneira similar ao procedimento de inserção e atualização do cliente é verificado se o mercado já existe para poder apurar se cria ou atualiza o mercado na base de dados.

### 3.7.2 Remoção de um Mercado

Quando é necessário apagar o mercado da base de dados similarmente à remoção de um cliente vê-se se o mesmo se encontra na base de dados, visto que se não se encontrar não tem sido apagá-lo. Mas na altura de apagar este tuplo é preciso fazer muito mais remoções e verificações, visto que também é necessário remover os instrumentos associados a esse mercado, e para isso remove-se todos os tuplos de *DAILYREG* e de *POSITION* associados ao *isin* do instrumento que tenha o mesmo *mrktcode*, na tabela *INSTRUMENT*, e *code*, na tabela *MARKET* remove-se também os instrumentos e os registos das tabelas *INSTRUMENT* e *DAILYMARKET*, respetivamente que tenham os mesmos códigos de mercado. Para concluir remove-se o mercado da tabela *MARKET* que tenha o código passado como parâmetro.

## 3.8 Atualização dos Valores Diários de cada Instrumento

Neste procedimento começa-se por calcular os valores diários a colocar na tabela *DAILYREG* para o instrumento que tenha o mesmo *id* e *datetime*, na tabela *EXTTRIPLE*,passados como parâmetro calculando os atributos *minval*, valor mínimo na tabela *EXTTRIPLE* para aquele instrumento naquele dia, *maxval*, valor máximo na tabela *EXTTRIPLE* para aquele instrumento naquele dia, *openingval*, primeiro valor inserido na tabela *EXTTRIPLE* para aquele instrumento naquele dia, e *closingval*, último valor inserido na tabela *EXTTRIPLE* para aquele instrumento naquele dia. De seguida é feita uma verificação se já existe em *DAILYREG* e *EXTTRIPLE* um instrumento com o mesmo *id* na mesma data passados como parâmetros e caso exista faz uma atualização na tabela *DAILYREG* com os valores obtidos, se não existir na tabela *INSTRUMENT* então não é inserido na tabela *DAILYREG* se existir significa que foi a primeira inserção em *EXTTRIPLE* para aquele instrumento naquele dia, por isso procedesse à inserção dos valores obtidos na tabela *DAILYREG*.

## 3.9 Cálculo da Média a 6 meses de um Instrumento

Para esta função decidiu-se passar a quantidade de tempo em dias que se tem de andar para trás como parâmetro, para dar liberdade ao utilizador de escolher a quantidade de tempo caso queira uma média maior ou menor de 6 meses. A data inicial é calculada através do auxilio da função *getdate()* do *t-sql* fazendo uma diferença do dia com a quantidade de dias passados como parâmetro. Por fim retorna-se uma seleção da média de todos os *closingval* na tabela *DAILYREG* que tenham uma data, *dailydate*, mais recente que a data inicial, esta média é calculada com auxilio da função *avg()* do *t-sql*, onde é passado como parâmetro o atributo *closingval*.

## 3.10 Atualização dos Dados Fundamentais de um Instrumento

Esta função recebe como parâmetros o *isin* do instrumento e a *date* do registo do instrumento para aferir a partir de que data deve ser calculado os dados fundamentais, estes dados são retornados numa tabela, visto que são dados derivados não tem sentido juntar a uma tabela têm sentido juntar a uma tabela do modelo físico. Os dados *dailyvar* e *var6m*, variação diária e variação a seis meses respetivamente são os primeiros a serem calculados pois serão necessários para o cálculo de outros dados. O atributo *dailyvar* é calculado a partir da diferença dos valores mínimos e máximo registados em *DAILYREG* para o *isin* passado como parâmetro, já o *var6m* é calculado através da diferença do valor máximo nos últimos seis meses que exista na tabela *DAILYREG* e do valor mínimo da mesma tabela para a mesma quantidade de tempo, para esta operação utiliza-se o auxílio da função *getdate()* do *t-sql*. De seguida insere-se os valores na tabela retornada que são respetivamente o *dailyvar* e o *var6m*, já calculados, o *currval*, que é calculado numa função auxiliar criada, que está descrita neste relatório mais abaixo, e é denominada por *get\_Currval()* recebendo como parâmetro o *isin*, o *avg6m* calculado através da função *Average()*, descrita acima que recebe a data e o *isin* passados como parâmetros, e as percentagens de variação *dailyvarperc* e *var6mperc* que são calculados a partir da divisão de *dailyvar* e *var6m* pelos seus correspondentes valores mínimos.

# Avaliação Experimental

A avaliação da nossa solução é apresentada neste capítulo. Aqui mostramos

como as nossas grandes ideias funcionaram

Exemplo de indentação do segundo parágrafo.

## 4.1 Nome da primeira secção deste capítulo

Texto da secção.

Continuação do texto noutro parágrafo.

## 4.2 A segunda secção deste capítulo

Na segunda secção deste capítulo, vamos abordar o enquadramento, o contexto e as funcionalidades.

### 4.2.1 A primeira sub-secção desta secção

As sub-secções são úteis para mostrar determinados conteúdos de forma

organizada. Contudo, o seu uso excessivo também não contribui para a facilidade

de leitura do documento[[1]](#footnote-1).

### 4.2.2 A segunda sub-secção desta secção

Esta é a segunda sub-secção desta secção, a qual termina aqui.

## 4.3 Análise de resultados

A análise de resultados segue aqui, nos próximos parágrafos de forma detalhada.

A data limite de entrega da versão final em 19 de Setembro de 2015 tem subjacente a inscrição em época especial. Não se verificando esta situação, a data limite de entrega é em 25 de Julho de 2015. O júri de cada projecto é constituído por indicação do respectivo orientador, até 25 de Maio de 2015. A avaliação da versão beta será realizada até 29 de Junho de 2015.

A discussão da versão final do projecto é pública e inclui até 30 minutos de apresentação e demonstração seguidos de discussão até ao máximo de 120 minutos de duração total da prova (40 e 150 minutos, respectivamente, quando o grupo tem três estudantes, e, 20 e 90 minutos quando o trabalho é individual). O júri da prova é proposto pelo orientador do projecto, tem pelo menos três elementos e inclui o arguente, o orientador e um docente de Projecto e Seminário (podendo este, em caso de impedimento, delegar num docente da área departamental). As deliberações do júri são tomadas por maioria simples.

A melhoria de classificação, se realizada no semestre de Inverno seguinte, terá por base trabalho adicional e discussão oral avaliados pelo mesmo júri. Quando realizada noutro semestre, envolve a realização de novo projecto.

A coordenação global dos projectos e dos seminários é feita pelos docentes de Projecto e Seminário, de acordo com as orientações definidas pela comissão coordenadora do curso. No sítio desta unidade curricular, é mantida a informação relevante, incluindo prazos, calendário dos seminários, estado dos projectos e estudantes e orientadores envolvidos. No final de cada ano lectivo, o regente elabora e apresenta à comissão coordenadora do curso um relatório sucinto sobre o funcionamento da unidade curricular. Em cada ano lectivo são identificados os melhores projectos para promover a sua divulgação pública.

# Conclusões

Neste trabalho tratou-se o problema. Foi formulada a solução que assenta nos princípios de boas práticas aprendidos ao longo do curso.

A solução obtida atingiu resultados satisfatórios.

# Referências

|  |  |
| --- | --- |
| [1] | Wikipedia, “Big data --- Wikipedia, The Free Encyclopedia,” http://en.wikipedia.org/w/index.php?title=Big\_data&oldid=648786139, 2015. |
| [2] | X. Ding, X. Zhu e G. Wu, “Data mining with big data,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, n.º 1, pp. 97-107, 2014. |
| [3] | J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong e J. Zhang, “What Will 5G Be?,” *IEEE Journal on Selected Areas in Communications,* vol. 32, n.º 6, pp. 1065-1082, 2014. |
| [4] | L. Boytsov, “Indexing Methods for Approximate Dictionary Searching: Comparative Analysis,” *J. Exp. Algorithmics,* vol. 16, n.º may, p. 1.81, 2011. |
| [5] | T. Jurkiewicz e K. Mehlhorn, “On a Model of Virtual Address Translation,” *J. Exp. Algorithmics,* vol. 19, n.º jan, pp. 1-18, 2015. |
| [6] | J. Neumann, The Computer and the Brain, New Haven, CT, USA: Yale University Press, 1958. |
| [7] | B. Kernighan e P. Plauger, The Elements of Programming Style, New York, NY, USA: McGraw-Hill, Inc., 1982. |

# A.1 Modelo EA

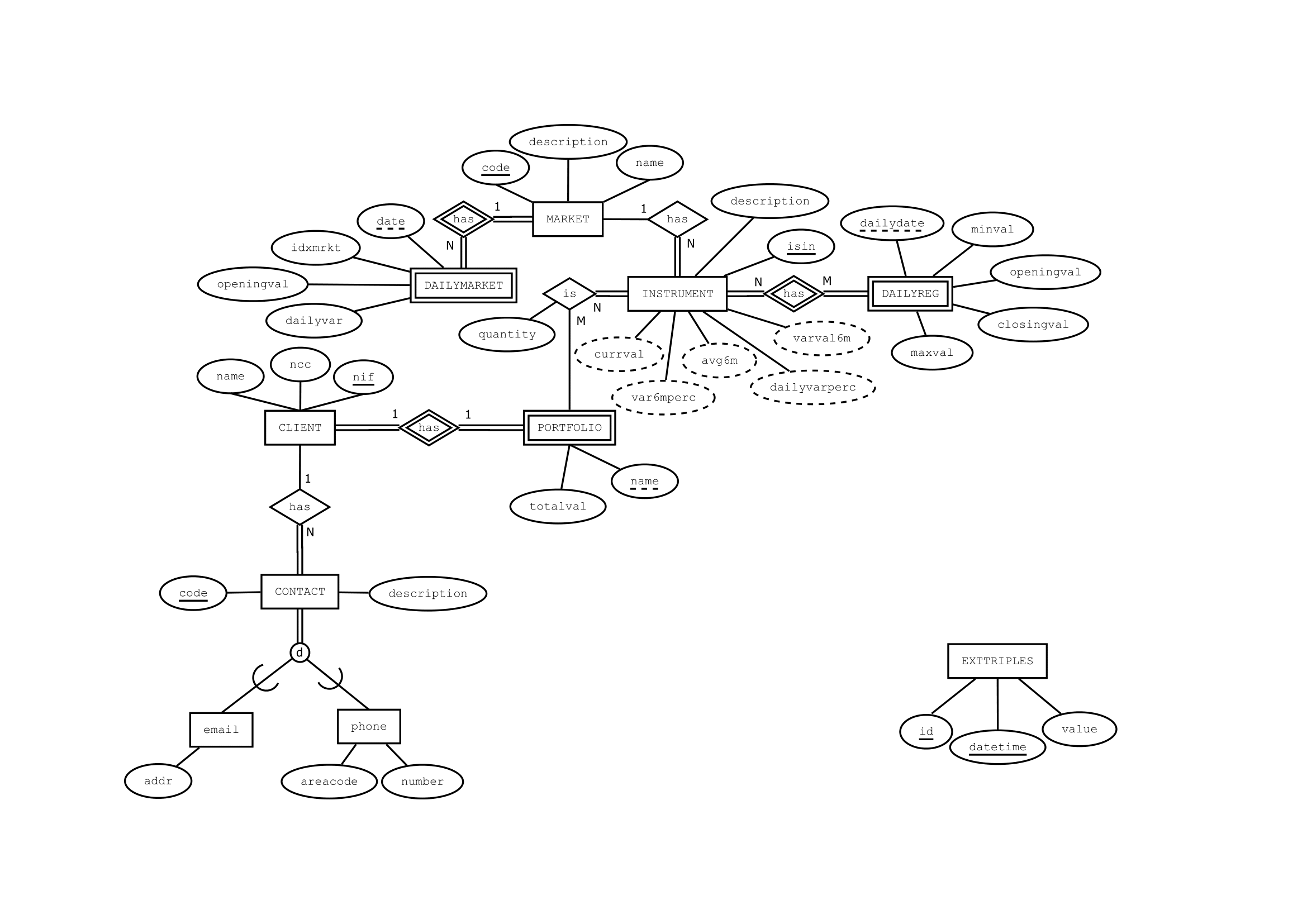


Figura 1 – Diagrama do Modelo EA.

# A.2 Modelos de dados

Estamos no início do anexo 2.

O *relatório* é um resumo do projecto global. Apenas como referência, é expectável cerca de 30 a 40 páginas A4 não devendo exceder 50 páginas.. A estrutura deve ser discutida e aceite pelo orientador. Os capítulos apresentados devem ter, em geral, a seguinte organização:

**Capítulo 1** — Introdução

Enquadramento do trabalho, metas, objectivos e especificações do projecto e resumo da solução. Concluir com a descrição breve dos restantes capítulos.

**Capítulo 2** — Formulação do problema

Introdução dos conhecimentos necessários para entendimento do trabalho, estabelecimento de terminologia e descrição detalhada do problema e do seu contexto. Síntese de abordagens anteriores do problema, caso existam, indicando as razões porque são insatisfatórias.

**Capítulo 3** — Grande ideia 1

Este capítulo pode ser subdividido em secções, designadamente:

1. Introdução: descrição da abordagem do problema e da metodologia utilizada; identificação das tarefas;
2. Elenco das características / Análise do problema: especificações, constrições, ferramenta de análise, etc*.*
3. Projecto: modelo para resolução do problema;
4. Implementação: a implementação do modelo como sistema computacional; descrição concisa do *hardware* e do *software*; dificuldades e contradições encontradas e sua resolução;
5. Avaliação: testes realizados e resultados experimentais (quando possível, o objectivo, a montagem e o método usado devem ser brevemente descritos); análise crítica dos resultados.

…

**Capítulo k+2** — Grande ideia k

**Capítulo k+3** — Conclusões

Recapitulação do trabalho desenvolvido. Referir claramente as observações e conclusões importantes. Discussão de ideias e recomendações para trabalho futuro.

**Referências**

Elenco dos livros e artigos citados no relatório. As referências são numeradas consecutivamente ao longo do relatório. O número da referência deve estar entre parêntesis rectos: [1].

**Anexos**

Os anexos devem incluir as partes importantes do *dossier do projecto*. O seu conteúdo depende da natureza do projecto, mas, em geral, pode incluir: listagem de programas, resultados de testes de *software*, exemplos de ecrãs de interface com o utilizador, esquemas dos circuitos, listagem de componentes, *data sheets* críticos, resultados de testes de *hardware*, desenhos mecânicos, análise económica, etc. (quando realizável, o *relatório* deve ser acompanhado da cópia do código, bibliotecas, etc. em suporte electrónico).

**Mais algumas dicas…**

O júri para avaliação do projecto final de curso apreciará o projecto, a sua demonstração e o respectivo relatório final (valorizando a escrita enquanto forma de divulgação de conhecimento). O relatório, depois de aceite e discutido, ficará disponível na biblioteca do departamento, para consulta.

O relatório deve ser auto-suficiente, isto é, o professor ou qualquer aluno finalista deve ficar apto a perceber o trabalho que realizou sem ter de ir à biblioteca ler os artigos originais.

Não escreva para o orientador, conhecedor de todo o detalhe, ou para um principiante. Tente escrever para uma audiência constituída por estudantes finalistas. Mantenha em mente o nível de conhecimentos do leitor a que se dirige. O relatório será uma base de trabalho para estudantes em circunstâncias semelhantes. Não dificulte o trabalho do leitor nem o faça estúpido (obviamente, …). Também é impossível ser totalmente claro. Evite afirmações dogmáticas (exemplo: "O *software* é a parte mais importante do computador.").

O relatório técnico não é uma história: usualmente não segue a cronologia das actividades realizadas. Também não é um romance (atenção à adjectivação). O relatório é um documento formal, feito para descrever os aspectos importantes do trabalho realizado.

Não tente descrever a função de cada componente, por exemplo a frase "O circuito IC2 e os componentes a ele associados formam um amplificador inversor …" é adequada. Contudo, descreva detalhadamente a função de cada componente ou circuito invulgar ou crítico.

As ilustrações (figuras, tabelas, gráficos e exemplos) são auxiliares preciosos para a explicação, mas envolvem muito trabalho. As figuras e as tabelas devem ser legíveis, instrutivas, legendadas e ter título. Os exemplos devem ser suficientemente detalhados para ilustrar o conceito.

O texto deve, pelo menos, ser analisado por um corrector ortográfico: os erros de ortografia são inadmissíveis.

Recomenda-se a leitura de alguns artigos e ou livros bem escritos para adquirir sensibilidade para a arte de escrever. Os artigos premiados em conferência são, normalmente, bons exemplos de escrita.

A escrita do relatório demora sempre mais tempo do que o inicialmente previsto.

**Resumo**

No essencial, a ideia é que tem algo para vender e o “Resumo” é a montra: a mensagem deve ser suficientemente clara e encorajar o cliente a entrar — se ele não a perceber passará ao lado. O resumo inclui: a motivação para o trabalho, como o fez e os resultados principais. Devem ser evitados chavões e palavras longas, as referências são proibidas e não deve utilizar acrónimos. Tenha em conta que o leitor será influenciado quer pela informação contida no resumo quer pela maneira como este está escrito. Não há desculpas para frases curtas ou desligadas, erros de gramática ou erros de sintaxe.

Não é fácil escrever um bom resumo.

**Introdução**

Procure dar resposta às seguintes questões: qual é o problema? porque é importante? o que é que outros já fizeram? quais as ideias base da solução apresentada? como está organizado o resto do relatório?

**Formulação do problema**

Defina o problema. Introduza a terminologia. Discuta as propriedades básicas.

**Solução do problema**

Enumere as coisas que fez e que considere importantes. Não seja modesto mas também não exagere.

A correcta avaliação do projecto é um aspecto crítico.

**Conclusões**

Procure dar resposta às seguintes questões: quais, se for caso disso, as lições aprendidas? o que, se algo, foi explicado? em que medida os objectivos foram atingidos? se existe algo que agora faria de forma diferente? quais as vantagens e desvantagens do trabalho realizado face a exemplos da literatura? que ideias para trabalho futuro?

**Referências**

A ideia subjacente à referência é que esta poupa papel e que o leitor poderá obter o documento em qualquer biblioteca científica razoável. Assim, é critério essencial referir revistas disponíveis em bibliotecas de instituições de ensino superior ou outras instituições profissionais. Em geral, não é razoável a referência a actas de conferências, dado que estas raramente estão acessíveis em bibliotecas pelo que, para todos os efeitos, estão indisponíveis. As referências a “Comunicações Privadas” são inaceitáveis. A informação dada deve ser sempre suficientemente detalhada por forma a que o leitor possa adquirir a publicação ou consultá-la numa biblioteca. Referências a dissertações de doutoramento ou outras devem indicar a instituição e o seu endereço. Sendo a referência essencial para o trabalho, no caso desta ser difícil de obter, dever-se-á incluir no documento, ou em apêndice, os seus pontos essenciais.

Cite uma referência sempre que está a incluir algo que não é contribuição sua ou quer indicar um conjunto de referências que o leitor pode consultar, mas cujo conteúdo não pode ser descrito adequadamente no relatório.

1. Este é um exemplo de nota de rodapé. Devem ser usadas com moderação, pois retiram a atenção ao texto principal. [↑](#footnote-ref-1)