



INSTITUTO
POLITÉCNICO
DA MAIA **IPMAIA**

INSTITUTO POLITÉCNICO DA MAIA

RELATÓRIO FINAL

***ESTÁGIO DESENVOLVIDO NA ENTIDADE
CÂMARA MUNICIPAL DA MAIA***

Hugo Lima nº 31713

LICENCIATURA EM TECNOLOGIAS DE INFORMAÇÃO WEB E MULTIMÉDIA

Pedro Pimenta

Rui Carreira

Porto, Junho de 2021

FICHA DE CARACTERIZAÇÃO

Instituição de Ensino Superior:

INSTITUTO POLITÉCNICO DA MAIA - IPMAIA

Licenciatura em Tecnologias de Informação Web e Multimédia

Entidade Acolhedora:

CÂMARA MUNICIPAL DA MAIA

Departamento onde realizou o estágio:

BaZe

Nome Completo do Aluno:

HUGO PINHEIRO MAGALHÃES DE LIMA

N.º de Aluno:

31713

Nome do Orientador na Entidade Acolhedora:

Pedro Pimenta

Nome do Supervisor do IPMAIA:

Rui Carreira

Duração do Estágio:

250 h

Período de Realização do Estágio:

01 / 03 / 2021 a 18 / 06 / 2021

Validação:

Assinatura do Aluno: _____

Assinatura do Orientador: _____

Assinatura do Supervisor: _____

Autenticação
(carimbo da entidade)

Após a conclusão da Formação em Contexto de Trabalho/Estágio, recebeu um convite para ficar a desempenhar funções na entidade?

☐ Sim

☐ Não

SUMÁRIO

O presente relatório descreve as atividades desenvolvidas e o processo das mesmas ao longo do estágio curricular que decorreu durante o 2º semestre do 3º ano letivo integrado no Curso da Licenciatura de Tecnologias de Informação Web e Multimédia, do Instituto Politécnico da Maia – IPMAIA.

Este estágio foi realizado remotamente na Câmara Municipal da Maia, no projeto Balanço Zero de Carbono (BaZe) durante o período de aproximadamente 4 meses, incluindo uma carga horária de 16 horas semanais, perfazendo assim as 250 horas estipuladas para a unidade curricular do curso.

Durante este estágio curricular, as atividades realizadas foram supervisionadas e orientadas pelo Engenheiro Pedro Pimenta, tendo sido fornecido, ao longo de cada semana, *feedback* relativamente ao que foi elaborado para assegurar que o raciocínio e as metodologias usadas estão corretas. Quando as soluções obtidas não eram as ideais, foram fornecidas sugestões de melhoria ou até estratégias diferentes para serem abordadas.

Este relatório descreve brevemente a caracterização da empresa, as tecnologias usadas e atividades realizadas pela mesma e de uma forma mais pormenorizada, as atividades realizadas pelo estudante. Nesta última descrição, é relatado o processo de desenvolvimento de código e o uso de uma base de dados externa para guardar os respetivos dados.

O estágio foi uma experiência enriquecedora no conhecimento, assinalando um percurso a nível académico. Sem dúvida foi um contributo para crescer como pessoa e como profissional.

AGRADECIMENTOS

O presente estágio curricular foi possível devido à influência de várias pessoas que me acompanharam.

Gostaria de agradecer ao Eng. Pedro Pimenta, responsável pela minha orientação durante o estágio, mostrando disponibilidade e espírito de ajuda nos momentos que foram encontrados de maior dificuldade.

Agradeço também ao professor Rui Carreira, o meu supervisor, sendo também o coordenador do curso.

Não só, mas também aos meus colegas Fábio Azevedo e Pedro Maia, que estagiaram na mesma entidade acolhedora, podendo contar com a ajuda de cada um, nas fases que foram encontrados mais problemas.

Gostaria também de dar um especial agradecimento à minha família, que demonstraram sempre o seu apoio no decorrer do curso e estágio.

ÍNDICE

FICHA DE CARACTERIZAÇÃO	2
SUMÁRIO	3
AGRADECIMENTOS	4
ÍNDICE	5
ÍNDICE DE FIGURAS	6
INTRODUÇÃO/OBJETIVOS	7
DESCRIÇÃO E CARACTERIZAÇÃO DA EMPRESA	8
DESCRIÇÃO DAS TECNOLOGIAS UTILIZADAS PELA EMPRESA	9
DESCRIÇÃO DAS ATIVIDADES REALIZADAS PELA EMPRESA	10
DESCRIÇÃO DAS ATIVIDADES REALIZADAS PELO ESTUDANTE	11
CONCLUSÕES	30
BIBLIOGRAFIA	31
ANEXOS	32

ÍNDICE DE FIGURAS

Figura 1 Ficheiro JSON tranferido do INE	13
Figura 2 Código para transferir vários ficheiros a partir do array dos anos.....	17
Figura 3 Função que permite fazer a ligação por Http.....	18
Figura 4 Código para saber o último ano e preencher array dos anos	19
Figura 5 1ª versão do modelo relacional	20
Figura 6 2ª versão do modelo relacional	20
Figura 7 3ª versão do modelo relacional	21
Figura 8 Modelo relacional realizado no MySQL Workbench	22
Figura 9 Query para inserir e criar uma chave única na tabela "valores"	22
Figura 10 Última versão do modelo relacional	23
Figura 11 Configuração para acesso à base de dados local	23
Figura 12 Query para inserir dados na tabela localidades	24
Figura 13 Tabela relativamente às freguesias da Maia	25
Figura 14 Resultados dos dados inseridos relativamente ao ano de 2014.....	25
Figura 15 Tabelas a usar na base de dados externa	27
Figura 16 Exemplo da definição de um input em MSSQL	27
Figura 17 Query para verificar o id da freguesia a partir do nome.....	28
Figura 18 Query para a verificação de um determinado registo	28
Figura 19 Dados inseridos na tabela obitos na base de dados externa	28
Figura 20 Memória usada e duração da execução dos anos de 2014 a 2019.....	29
Figura 21 Memória usada e duração da execução dos anos de 2014 a 2020.....	29

INTRODUÇÃO/OBJETIVOS

O presente relatório de estágio é importante porque é elaborado no âmbito da Unidade Curricular Estágio no 2º semestre do 3º ano letivo do Curso da Licenciatura de Tecnologias de Informação Web e Multimédia, do Instituto Politécnico da Maia – IPMAIA.

O avanço tecnológico possibilita a evolução para uma era mais digital, permitindo assim economizar os recursos naturais disponibilizados ao ser humano e tornar as informações mais fáceis de serem acedidas.

Neste projeto, o BaZe, permite essa transição, a evolução para uma *SmartCity*. O estudo e fornecimento das informações como a qualidade do ar, consumo energético, entre outros, permite a verificação e análise relativamente a indicadores importantes para uma boa sustentabilidade.

Aqui entra em ação um *Data Engineer*, que tem como objetivos a compreensão e análise de várias fontes de dados possíveis para obter os dados pretendidos, quais as ferramentas e tecnologias a usar para a recolha destes dados e saber como e onde guardar a informação recolhida.

A importância deste processo permite que um possível cliente possa aceder aos dados que pretende, estando estes apresentados de uma forma mais simplificada, ou os dados serem futuramente apresentados ao público, podendo um cidadão curioso, visualizar qualquer informação pertinente relativamente a uma zona da Maia.

DESCRIÇÃO E CARACTERIZAÇÃO DA EMPRESA

A Câmara da Maia, situada no município da Maia, pertencente ao distrito do Porto, na região Norte e integrada na Área Metropolitana do Porto. Consiste numa área de 82,99 km², estando dividido em 10 freguesias

O município da Maia é considerado, por um lado rico nas suas raízes culturais, históricas e religiosas e por outro, é dos mais avançados no país, representado um fator importante na economia, com a sua indústria, inovação e novas tecnologias, revelando-se um exemplo a nível nacional no desenvolvimento económico e ambiental.

DESCRIÇÃO DAS TECNOLOGIAS UTILIZADAS PELA EMPRESA

No decorrer do estágio, foram usadas várias tecnologias e ferramentas de forma a acompanhar o estagiário.

Como forma de um histórico, foi disponibilizada uma folha, usando o *Google Drive* com várias informações pertinentes ao estágio, como o acompanhamento de cada reunião, documentos disponibilizados pelo Orientador e documentos relativamente a cada estagiário para facilitar o acesso, entre outros.

Foi usado o *Google Hangouts* como chat, para fazer comunicações mais rápidas, como agendar futuras reuniões.

Por fim, para a realização das reuniões semanais, foi usado o *Google Meet*, que é um serviço que permite comunicação por videochamada, com o objetivo de, por parte dos estagiários, ser feito o resumo dos resultados semanais obtidos.

DESCRIÇÃO DAS ATIVIDADES REALIZADAS PELA EMPRESA

A Câmara Municipal da Maia é uma entidade executiva da Maia, constituída por diversos departamentos de administração onde prestam vários serviços públicos, tais como segurança, ambiente, educação entre outros.

É considerado um dos melhores espaços onde a importância de um novo rumo ao conhecimento, ao planear e viver na cidade ao desenvolvimento da sustentabilidade e respetiva descarbonização, ganha uma outra dimensão. Tudo isto é conseguido através de um conjunto de ações que são baseadas na inovação e participação de todos os que fazem parte de uma sociedade, nomeadamente as indústrias e os cidadãos.

Um dos serviços destacados é o projeto BaZe que é um exemplo desta inovação, permitindo a aplicação e demonstração de soluções que permitem ao município fazer a análise dos dados obtidos e ir ao encontro de um desenvolvimento de uma boa sustentabilidade.

Representa ideias dinâmicas com o espírito de criar um quotidiano com um melhor ambiente e melhor qualidade de vida.

DESCRIÇÃO DAS ATIVIDADES REALIZADAS PELO ESTUDANTE

No início do estágio, foi proposto, pelo Engenheiro Pedro Pimenta, que cada estagiário escolhesse o tema que gostaria de tratar durante a atividade curricular, nomeadamente *Data Engineering*, *Data Science* e um sistema para gerir a ordem dos dispositivos que vão recolher as informações.

Foi definido para os estagiários do IPMAIA, que o período de estágio fosse dividido em dois ciclos, apresentando em cada um deles, o que foi realizado, detalhando os resultados obtidos.

O tema que o estagiário decidiu abordar foi *Data Engineering*, onde é suposto estudar várias fontes de dados disponíveis, para poder comparar qual a melhor fonte a usar, relativamente à credibilidade e qualidade de informação e de seguida, tratar esses dados e guardar numa base de dados.

Foi dada a liberdade de escolha da ferramenta/tecnologia a usar, tendo sido escolhido JavaScript, por ser uma linguagem potente e com muita procura no mercado de trabalho e como foi dada a hipótese de seguir os interesses dos estagiários, foi considerada uma boa oportunidade para aprender e aplicar os conhecimentos que foram ganhos durante este período de estágio.

Foi também usado Nodejs, que é um *software* de código aberto que permite executar JavaScript fora de um *browser*.

Em primeiro lugar, foi disponibilizada uma série de documentação, como o REOT (Relatório do Estado de Ordenamento do Território) da Maia e ISO 37120 (International Organization for Standardization) para que os estagiários a estudassem, servindo como exemplo e para fornecer alguma orientação, do que é pretendido tratar. Depois do estudo dos documentos mencionados, teve que ser escolhido qual o tema que se iria tratar, para que o cargo de *Data Engineer* pudesse ser exercido.

O tema que se decidiu abordar foi “Óbitos por local de residência” após ter sido visualizado alguns exemplos no ISO 37120 relacionados com óbitos. Serve de exemplo, o tema “Número de mortes causadas por acidentes industriais”. Foi considerado que o registo de óbitos é um dado populacional importante, principalmente devido à fase atual de pandemia. Após a escolha do tema a tratar, a identificação de uma boa fonte de dados seria a próxima etapa.

A fase de procura da fonte de dados a trabalhar é muito importante por ser a matéria prima de um *Data Engineer*. Garantir uma boa fonte de dados que seja credível e com a informação necessária para trabalhar é fundamental.

Com base na informação do documento REOT, muitos dados têm como origem a fonte do INE (Instituto Nacional de Estatística). Após efetuada uma pesquisa na plataforma do INE por “óbitos”, foi encontrado o registo “Óbitos (N.º) por Local de residência (NUTS - 2013) e Sexo; Anual”.

Esta base de dados contém informação detalhada, podendo selecionar várias regiões, incluindo as freguesias do concelho da Maia, Área Metropolitana do Porto (AMP), ou qualquer outro local de Portugal.

As regiões estão divididas por NUTS (*Nomenclature of Territorial Units for Statistics* - Nomenclatura das Unidades Territoriais para Fins Estatísticos), no qual se apresenta no nível 3, que corresponde às entidades intermunicipais, como a Área Metropolitana do Porto e também estão presentes as freguesias. Relativamente ao NUTS, estão incluídos níveis inferiores (NUTS 1 e 2). Estes níveis, identificam respetivamente Portugal continental e as ilhas dos Açores e da Madeira (NUTS 1), as regiões (Norte, Centro, área metropolitana de Lisboa, Alentejo e Algarve, NUTS 2) e estão também presentes os concelhos. Os restantes dados apresentados são o sexo, masculino e feminino, e o ano de 2014 até 2019.

Seguindo o exemplo do documento REOT, este compara os valores de cada freguesia do concelho da Maia, sendo que esta fonte de dados permite fazer a mesma representação. [\(Referência1\)](#)

Como se pode verificar na figura abaixo, o ficheiro JSON contém informação sobre o indicador, com o respetivo código, o nome, um *url* com informações adicionais sobre o mesmo, a data de extração e a última atualização.

Dentro do campo “Dados” tem o valor do ano, neste caso 2014 que é composto por um *array* de objetos. Em cada objeto, os dados que serão usados são: “geocod” que é um código associado a um respetivo local; “geodsg” que contém o nome do local; “dim_3_t” que define o género (H de homem, M de mulher e HM para a soma dos dois); “valor” onde tem os dados numéricos relativamente aos óbitos;

```
[ {
  "IndicadorCod" : "0008235",
  "IndicadorDsg" : "Óbitos (N.º) por Local de residência (NUTS - 2013) e Sexo; Anual - INE, Óbitos",
  "MetaInfUrl" : "https://www.ine.pt/bddXplorer/htdocs/minfo.jsp?var_cd=0008235&lingua=PT",
  "DataExtracao" : "2021-04-26T19:45:11.531+01:00",
  "DataUltimoAtualizacao" : "2020-04-27",
  "UltimoPref" : "2019",
  "Dados" : {
    "2014" : [ {
      "geocod" : "186121210",
      "geodsg" : "Tolosa",
      "dim_3" : "1",
      "dim_3_t" : "H",
      "valor" : "15"
    }, {
      "geocod" : "186121211",
      "geodsg" : "União das freguesias de Arez e Amieira do Tejo",
      "dim_3" : "1",
      "dim_3_t" : "H",
      "valor" : "7"
    }
  ]
} ]
```

Figura 1 Ficheiro JSON transferido do INE

Para a verificação de uma boa fonte de dados teve que ser feita mais pesquisas para encontrar mais fontes de dados, usando o google e ao pesquisar por “óbitos em portugal”, onde foram encontradas outras fontes, como o Eurostat (Gabinete de Estatísticas da União Europeia), estando estes dados representados apenas pelo NUTS 2. Como esta fonte não contém a informação pretendida, até às freguesias, foi descartada pois não tem a informação necessária para o desenvolvimento deste trabalho. [\(Referencia2\)](#)

Outra fonte de dados encontrada para a recolha de dados foi o pordata. [\(Referência3\)](#). Estes dados têm como origem a plataforma do INE e é apenas apresentado o registo de Portugal e o respetivo ano, não contendo o

registo de freguesias, nem nenhum nível de NUTS. Como os dados apresentados são os valores totais de todas as regiões como valor final, esta não será uma boa fonte de dados a trabalhar.

Uma vez que não foi possível encontrar outras fontes de dados que contenham valores das freguesias, que são dados importantes, foi decidido avançar com a primeira fonte encontrada, a do INE.

O “Portal” do INE contém uma API onde, depois de fazer o estudo do manual que está disponibilizado publicamente, é possível obter os dados que foram selecionados, onde gera um ficheiro JSON e é descarregado. O manual encontra-se nesta página onde ajuda o utilizador a guiar-se e a perceber como pode usufruir deste serviço. [\(Referência4\)](#) Esta API é constituída por códigos que terão de ser inseridos pelo utilizador para obter os dados que se pretende. Teve-se que verificar quais os códigos a usar para se obter o respetivo ficheiro JSON da fonte que se vai utilizar. Estes valores são o indicador: ou seja, o código associado a “Óbitos (N.º) por Local de residência (NUTS - 2013) e Sexo; Anual” que é identificado por 0008235; o código da dimensão temporal: como exemplo o ano de 2019 tem o código S7A2019; e o código da dimensão geográfica: um dos códigos será 11A130603, sendo este associado ao de Folgosa. Um exemplo do link URL para obter o ficheiro JSON relativamente ao município da Maia, no ano 2019 é [\(Referência5\)](#).

A impossibilidade de, na dimensão geográfica, selecionar múltiplas opções (ou a API do INE não permite ou não foi descoberto como), obrigou a inserir as localizações que se pretende individualmente, sendo pouco eficaz a transferências de vários ficheiros. O concelho da Maia é constituído por 10 freguesias, sendo que, teria que ser usado dez vezes um link diferente para cada um deles, relativamente apenas a um ano.

Servindo de exemplo, o código do Castelo da Maia é 11A130618 e o de Folgosa 11A130603, logo, terá que ser gerado um *link* diferente para cada um deles, sendo respetivamente [\(Referência6\)](#) e [\(Referência7\)](#). Apenas é alterado o valor que define a dimensão de “Dim2” para poder filtrar a localização geográfica pretendida. Foi identificado como desvantagem, a extração das localizações separadas e não num só ficheiro, perdendo sempre alguma eficiência na leitura de múltiplos ficheiros.

A escolha da fonte de dados a tratar permite avançar para o passo seguinte, sendo este analisar como obter os dados que são disponibilizados. Foi pensado a criação de um algoritmo que permite a extração das localizações pretendidas automaticamente, definindo um *array* com as dimensões geográficas que se pretende trabalhar.

Depois de alguma pesquisa relativamente a como transferir ficheiros usando JavaScript e Node.js, foram encontradas algumas dificuldades. Como foi o primeiro contacto com estas tecnologias, foi necessário perceber alguns fatores básicos, como definição de variáveis e como são criadas funções, para poder transferir todos os ficheiros para seguidamente juntar a informação.

O processo de aprender uma nova linguagem é sempre complicado por haver algumas diferenças na escrita do código. Foi pensado como se poderia percorrer a informação de cada indicador (localidade) e obter os dados que serão guardados na base de dados. Deparou-se com um problema ao usar a API do INE, pois outras APIs usam o URL para apresentar os dados JSON em HTML, enquanto que o INE é usado um URL (como já mencionado anteriormente), para gerar e transferir um ficheiro JSON. Presumia-se que a forma de aceder aos dados JSON seria a mesma, mas depois de alguns testes, não foi possível aceder à informação, pois eram apresentados erros.

Como os dados que se pretende usar estão num ficheiro que é descarregado a partir do URL, teve que ser pensado numa forma de percorrê-los para de seguida avançar com o processo de seleccionar os dados essenciais e inserir numa base de dados.

Inicialmente foi avaliado modos de poder abrir o URL, deparando com a função “location.href”, que permite definir o endereço que se pretende aceder, forçando a abertura de um browser com o respetivo URL, transferindo assim o ficheiro JSON.

Esta função foi rapidamente descartada após a avaliação do desempenho e tempo que demoraria a usar todos os indicadores necessários.

Para combater este problema, foi descoberto o módulo XMLHttpRequest, que permite fazer um pedido Http e poder observar os dados sem ser necessário a utilização de um browser. Nesta fase, primeiro é feita a criação do JSON com o uso do módulo “fs” (*File System*), permitindo trabalhar com ficheiros e “writeFileSync”, de modo a tratar cada um sincronamente e ao usar o módulo XMLHttpRequest, que permite aceder aos dados da internet, é inserindo a informação nesse ficheiro. Durante esta fase, foi também verificado, como teste, que já é possível visualizar os valores, garantindo que o ficheiro JSON foi descarregado e lido corretamente.

Com a criação deste JSON, contendo os dados, apenas resta a definição de um *loop* para poder repetir a ação anterior, consoante as freguesias disponibilizadas na API do INE.

Seguindo o exemplo anterior, foi criado um *array* com todos os indicadores que se irão usar, sendo todos eles relativamente às freguesias do concelho da Maia, num total de 10 indicadores, identificados como: 11A130601, 11A130618, 11A130619, 11A130603, 11A130608, 11A130609, 11A130620, 11A130617, 11A130613, 11A130616.

Depois de definir o *array*, a função criada anteriormente foi inserida dentro de um ciclo para poder descarregar todos os ficheiros pretendidos, alterando apenas o valor associado ao “Dim2” para percorrer os valores do *array*.

Durante o estágio, foram decorrendo reuniões semanais com o orientador e os restantes estagiários, para esclarecimento de dúvidas e partilhar ideias entre os elementos. Nestas reuniões são apresentados os avanços e resultados que foram obtidos, podendo assim receber *feedback* do que foi realizado e orientações, como outros métodos que se podem usar ou até alterar a estratégia

que está a ser usada, no caso de problemas que possam ser encontrados no futuro. Neste caso, foi solicitada uma alteração naquilo que já foi realizado, tendo sido pedido que se transferisse apenas um ficheiro para cada ano.

A transferência de apenas um ficheiro implica um documento mais pesado, pois contém todos os dados relativamente a um respetivo ano, mas, por outro lado, significa menos pedidos à API do INE podendo haver um melhor desempenho. Ao não incluir o valor de “Dim2”, que é referente ao código da dimensão geográfica, o INE gera um ficheiro JSON com todos os dados relativamente a esse ano.

Esta adaptação, de transferir apenas um ficheiro, implica a alteração do *array* com as freguesias para um *array* com os anos. Este *array* foi então modificado para conter as datas de 2014 até 2019, onde é usado um *while loop* de modo a percorrer todos os anos no *array* e os valores no URL serem modificados para descarregar os ficheiros com base no ano.

```
while(i < dim1_cods.length)
{
    var URL = host_url+varcd_cod+"&Dim1="+dim1_cods[i]+"&lang=PT";

    var fs = require('fs');
    fs.writeFileSync("obitos"+firstdate+".json", GetURL(URL), function(err)
    {
        if (err)
        {
            console.log(err);
        }
    });
    i++;
    firstdate++;
}
```

Figura 2 Código para transferir vários ficheiros a partir do array dos anos

A função “GetURL” permite efetuar o pedido XMLHttpRequest que já foi mencionado anteriormente.

```
function GetURL(newURL)
{
    var req = new XMLHttpRequest();
    req.responseType = 'json';
    req.open('GET', newURL, false);
    req.send(null);
    return req.responseText;
}
```

Figura 3 Função que permite fazer a ligação por Http

Após efetuar a alteração pedida, teve que ser pensado num algoritmo para poder percorrer todos os ficheiros que foram descarregados anteriormente. Seguindo a mesma estrutura usada para descarregar os ficheiros, criou-se um ciclo para ler os ficheiros, que após o *download* são identificados como, por exemplo para o ano de 2014, “obitos2014.json”, onde o nome “óbitos” é estático e o ano é concatenado consoante o ficheiro que é lido, para poder diferenciar os ficheiros dos restantes que são descarregados, para evitar que sejam substituídos por já conter um ficheiro com o mesmo nome.

Por fim, para o *array* dos anos serem definidos dinamicamente, é feito outro pedido ao INE, sem a definição do “Dim1”, que é o responsável pelo indicador do ano, onde por padrão, é retornado o ano mais recente, podendo assim, verificar a partir do ano 2014 (referente à primeira data disponibilizada), quantos anos estão disponíveis, preenchendo o *array* com os valores do primeiro ano, até ao ano mais recente.

Deste modo, foi possível fazer com que todos os ficheiros disponíveis fossem descarregados e lidos, podendo assim obter os dados que se pretende para depois guardar numa base de dados.

```
function GetLastDate()
{
    var URL = host_url+varcd_cod+"&lang=PT";

    var fs = require('fs');
    const path = require('path');
    fs.writeFileSync("obitos"+firstdate+".json", GetURL(URL), function(err)
    {
        if (err)
        {
            console.log(err);
        }
    });

    let ficheiro = fs.readFileSync(path.resolve(__dirname, 'obitos'+firstdate+'.json'));
    let data = JSON.parse(ficheiro);
    keys = Object.keys(data[0].Dados);
}

//Colocar no array os Dim1 disponiveis
function FillDates()
{
    do
    {
        dim1_cods.push(S7A+firstdate);
        firstdate++;
    }while(firstdate <= keys);
}
```

Figura 4 Código para saber o último ano e preencher array dos anos

Com base nos dados que se pretende trabalhar, teve-se que definir a estrutura para a base de dados, tendo sido elaborado um modelo relacional. Esta fase permite perceber a relação dos dados que estão disponíveis e selecionar a informação mais relevante. Este processo foi demorado pelo facto de ter sido elaborado várias versões até ser encontrada a mais eficiente e correta. Os dados pertinentes, (já mencionados anteriormente) são compostos pelos seguintes valores: localidade, onde contém o nome da localização geográfica e o seu respetivo código; o ano, onde contém o ano em que foi feito o registo; o sexo, onde é identificado o género; e valor, contendo o dado principal com o número de óbitos. Depois de realizar esta análise dos dados a trabalhar, foi necessário estruturar como cada valor é interligado. No decorrer deste período, foram desenvolvidos, testados e apresentados vários modelos relacionais.

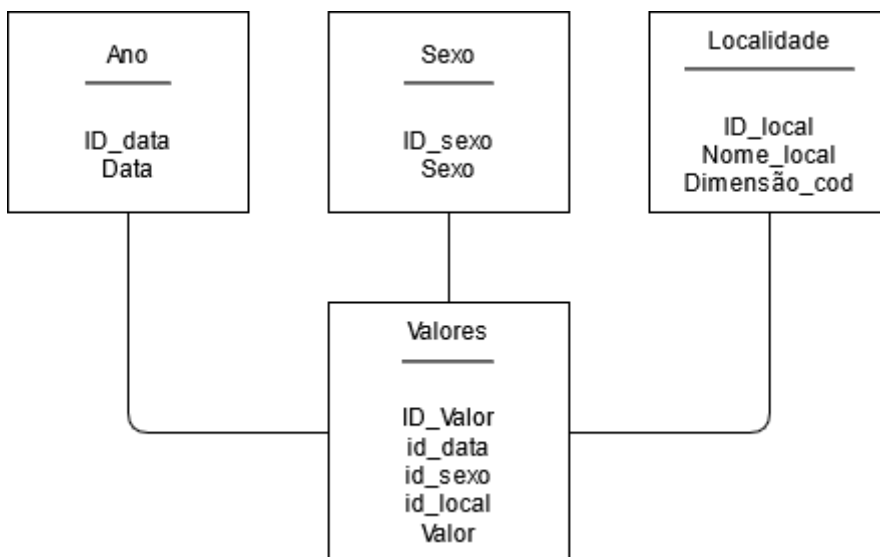


Figura 5 1ª versão do modelo relacional

O primeiro modelo relacional criado serviu como representação de como os dados são apresentados pelo INE, utilizando apenas os dados que são mais importantes para armazenar na base de dados.

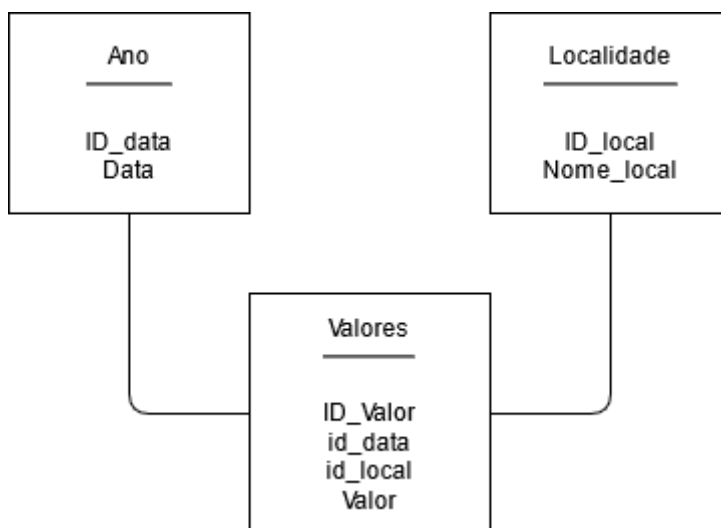


Figura 6 2ª versão do modelo relacional

Na 2ª versão, para simplificação da tabela, foi pensado que poderia ser armazenado apenas o valor total dos óbitos, podendo assim retirar o componente “sexo”, e retirado o campo “Dimensão_cod”, por apenas ser necessário o nome do local, simplificando a tabela.

Seguindo vários exemplos da documentação no início do estágio, a estrutura sofreu várias alterações, foram encontradas falhas e melhorias que podiam ser aplicadas.

A importância das reuniões que foram realizadas permitiu o acompanhamento dos estagiários, onde, sem dúvida, teve um grande impacto no estágio curricular. Foram apresentadas algumas sugestões e dicas na qual o estagiário tomou em consideração.

Inicialmente, como fase de teste, todos os dados serão persistidos para uma base de dados, obrigando a adaptar a tabela das localidades, para evitar conflitos de locais com o mesmo nome, como por exemplo, Ribeira Grande (situada nos Açores) que contém uma freguesia chamada Maia, foi então decidido voltar a inserir o código da localidade que é definido pelo INE de forma a poder diferenciar e especificar a localidade em questão. Foi também adicionado uma tabela para a identificação da fonte de dados que está a ser usada.

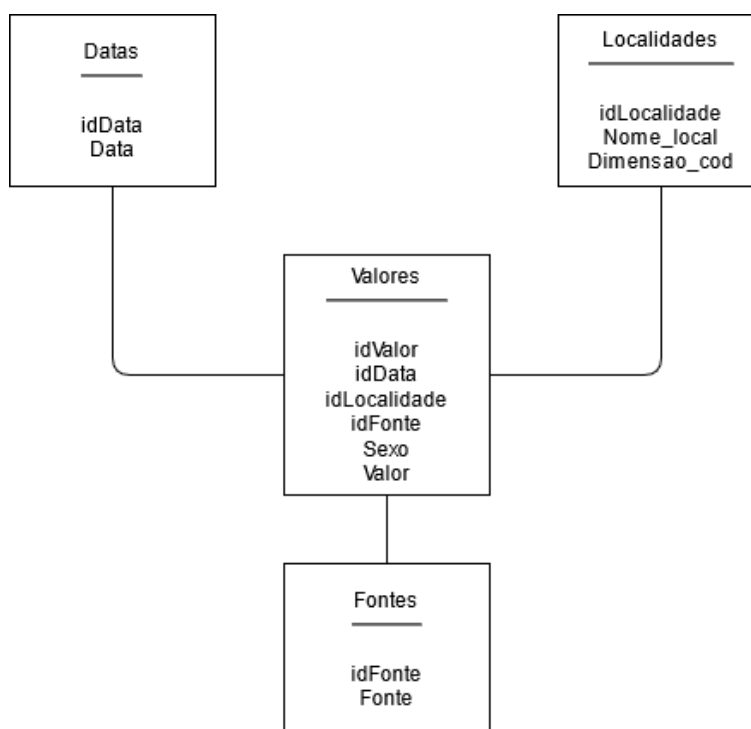


Figura 7 3ª versão do modelo relacional

Para uma melhor representação do modelo relacional, foi usado o programa *MySQL Workbench*, onde foi possível, de uma forma mais detalhada, definir as tabelas e os respetivos campos com a identificação do *data type*, assim como as respetivas relações.

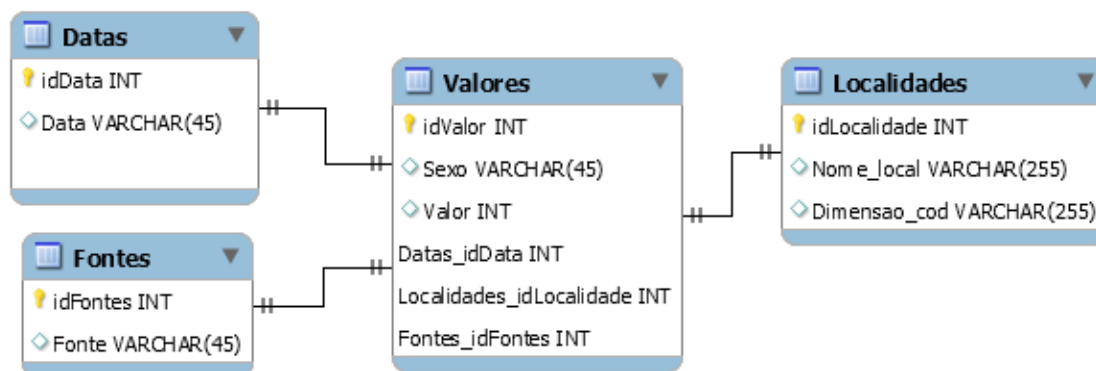


Figura 8 Modelo relacional realizado no MySQL Workbench

Foi comunicado que o modelo relacional ainda não estava otimizado e pedido que também fosse acrescentado uma tabela relativamente às freguesias da Maia, sendo obrigado a continuar a otimizar e melhorar este componente até atingir a versão ideal.

O uso da chave primária para o campo “idValor” permite que este campo seja único, não permitindo valores repetidos. Para evitar que sejam inseridos os mesmos valores novamente, foi encontrada uma solução que foi tornar este campo um “VARCHAR”, permitindo assim não só números como também letras, e usando os valores como a data, o código do local e o “dim_3_t”, que define o sexo, pode-se criar uma chave única, que define se já existe um registo com um determinado ano, local e sexo.

```
const newval = { idValor: keys[0]+data[0].Dados[firstdate][i].geocod+data[0].Dados[firstdate][i].dim_3_t
connection.query('INSERT IGNORE INTO valores SET ?', newval, (err, res) => {
```

Figura 9 Query para inserir e criar uma chave única na tabela “valores”

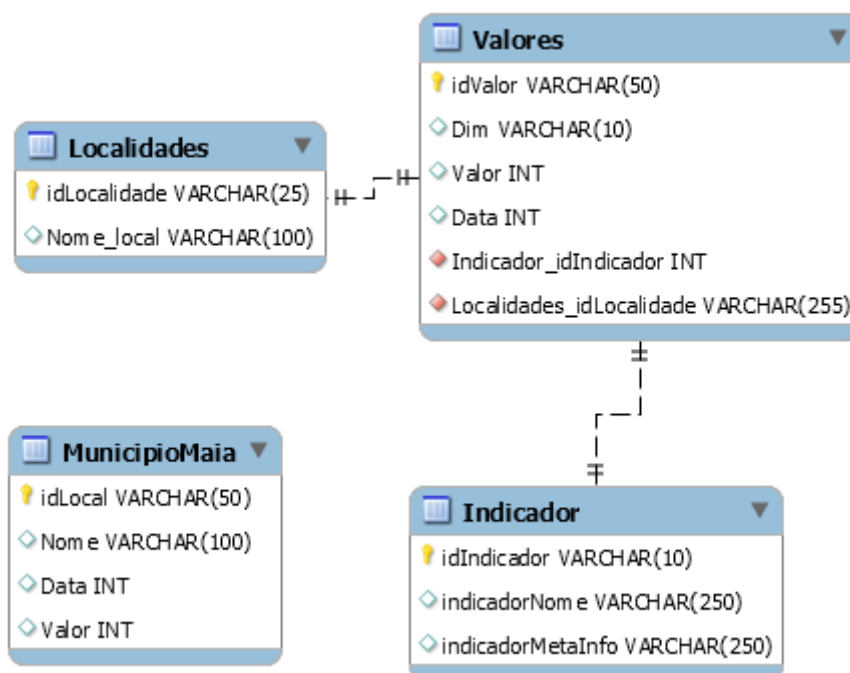


Figura 10 Última versão do modelo relacional

Assim que foi dada a aprovação do modelo relacional, foi criada uma base de dados local para teste, de forma a poder testar e inserir os valores nos respetivos campos e tabelas. Para a criação desta base de dados local, foi usado o XAMPP. Foi necessário uma configuração para associar o programa à base de dados que foi criada, seguindo a seguinte imagem como referência, onde foi também usado MySQL (da Oracle Corporation), que permite fazer a gestão de uma base de dados. A base de dados foi denominada por “obitosteste”.

```

const mysql = require('mysql');
const connection = mysql.createConnection({
  multipleStatements: true,
  host: 'localhost',
  user: 'root',
  database: 'obitosteste'
});

connection.connect((err) => {
  if(err){
    console.log('Error connecting to Db\n');
    return;
  }
  console.log('Connection established\n');
});

```

Figura 11 Configuração para acesso à base de dados local

Com a finalização da leitura dos dados de cada ficheiro, foi necessário preparar o sistema para poder inserir os valores que se pretende guardar na base de dados. Para isto, com a ligação ao servidor XAMPP que foi criada anteriormente, foi criado uma base de dados no “phpMyAdmin” com as respetivas tabelas, seguindo o modelo relacional que já foi apresentado. Usou-se o MySQL para fazer os *inserts* na base de dados, selecionando os respetivos valores que são fornecidos no ficheiro JSON do INE.

Estes dados serão distribuídos e guardados pelas respetivas tabelas que foram criadas anteriormente, onde foi definido um ciclo para percorrer todos os objetos do *array* e ao realizar uma *query* SQL, estes dados são inseridos nas tabelas. Servindo de exemplo, a imagem abaixo mostra uma *query* sql para inserir os dados do *geocod* (código do local) e o *geodsg* (nome do local) na tabela localidades, onde terá todos os dados de NUTS I a III, inclusive freguesias.

```
const local = { idLocalidade: info[i].geocod , nomeLocal: info[i].geodsg}  
connection.query('INSERT IGNORE INTO localidades SET ?', local, (err, res)
```

Figura 12 Query para inserir dados na tabela localidades

Todo o processo realizado anteriormente foi para efeitos de teste, para verificar como serão selecionados os dados e perceber a estrutura de informação que é fornecida pelo INE. Como a documentação que foi disponibilizada no início do estágio é focada para os dados da Maia, o produto final terá de conter apenas os dados relativamente ao concelho da Maia. Assim que foi possível inserir todos os dados nas tabelas corretamente, foi pedido uma tabela final, que contém apenas os dados da Maia, e que esta se apresentasse da forma mais simples possível, mantendo apenas informação relevante e excluir dados redundantes. Foi considerado, com base nos diagramas anteriores, que os dados mais importantes seriam o nome do local, o ano e o número de óbitos.

Foi criada então, a tabela denominada de “Obitos_Maia”, que contém os dados mais relevantes para a identificação dos registos que poderão ser consultados futuramente. Os dados são compostos pelo ano, o número de óbitos e o nome da freguesia. Assim que foi lançada uma última proposta, foi testado e adaptado no código para que os dados inseridos sejam apenas relativamente às freguesias

da Maia. Depois de realizar uma *insert query* os dados que foram já mencionados são inseridos na respetiva tabela.

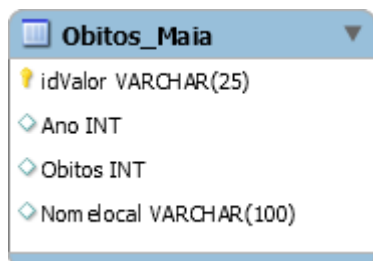


Figura 13 Tabela relativamente às freguesias da Maia

idValor	Ano	Obitos	NomeLocal
201411A130601HM	2014	176	Águas Santas
201411A130603HM	2014	30	Folgosa
201411A130608HM	2014	37	Milheirós
201411A130609HM	2014	79	Moreira
201411A130613HM	2014	15	São Pedro Fins
201411A130616HM	2014	37	Vila Nova da Telha
201411A130617HM	2014	105	Pedrouços
201411A130618HM	2014	106	Castêlo da Maia
201411A130619HM	2014	230	Cidade da Maia
201411A130620HM	2014	42	Nogueira e Silva Escura

Figura 14 Resultados dos dados inseridos relativamente ao ano de 2014

Como forma de melhorar as funcionalidades, foi implementado um sistema onde, ao correr o programa, é possível escolher uma data (desde que esta data esteja presente nas datas disponíveis), e apenas o ficheiro com essa data é lido. Não só, mas também, é possível definir uma data de início e outra de fim, para indicar um intervalo, onde serão selecionados todos os ficheiros dentro do mesmo. Esta funcionalidade permite dar alguma liberdade relativamente aos anos que se pretende guardar informação.

Foi feito um pedido para que se alterasse a forma de criação das chaves primárias, em vez de se concatenar vários valores para formar uma chave única, para ser pensado numa outra solução. A forma atual de inserção dos dados numa base de dados, implica a verificação dos campos e a criação de uma coluna extra com a concatenação dos dados, verificando assim se uma determinada linha de dados já existe. A solução encontrada foi fazer um *select*, onde são verificados os dados principais (como o ano e o local), e se for encontrado algum registo na base de dados com esses valores, significa que já existe na base de dados esse registo. Foi também adaptado para, no caso de ter encontrado um determinado registo, este registo é atualizado, usando o *update*.

Depois da fase de testes, foi enviado ao estagiário as credenciais de acesso a uma base de dados externa que será o destino dos dados para que a Câmara da Maia tenha acesso. Teve que ser adaptado o código pois estava a ser usado o MySQL para a ligação da base de dados local e a que foi fornecida apresentava-se em MSSQL. Para aceder à base de dados fornecida teve que ser instalado o módulo “mssql” e verificar a documentação, disponibilizada em [\(Referência8\)](#).

Após o estudo da documentação e verificar algumas diferenças que são apresentadas entre o MySQL e MSSQL foi possível realizar a nova configuração e aceder à base de dados externa com sucesso. Foi necessário o uso das credenciais fornecidas, para a autenticação, o uso da porta “1433” e garantir que o certificado do servidor (trustServerCertificate) é verdadeiro, para validar uma conexão segura, senão é apresentado um erro.

Esta base de dados é composta por várias tabelas, nas quais algumas são usadas pelo estagiário. A tabela “Freguesia”, que contém a identificação das 10 freguesias da Maia e a tabela “obitos” onde será o destino dos dados relativamente aos óbitos.

id	nome	area
1	Águas Santas	820.33
2	Castêlo da Maia	1991.04
3	Cidade da Maia	1041.68
4	Folgosa	1011.45
5	São Pedro Fins	470.68
6	Nogueira e Silva Escura	882.66
7	Milheirós	360.42
8	Pedrouços	254.09
9	Moreira	867.27
10	Vila Nova da Telha	594.31

ano	Freg_id	obtM	obtH	obtTotal
1	2	3	4	5

Figura 15 Tabelas a usar na base de dados externa

Apesar de ser um dado redundante, foi considerado a inclusão dos óbitos totais nesta tabela, de forma a ser mais rápida o uso desta informação, sem ser necessário a verificação dos campos para ser realizado uma soma.

A forma como são inseridos os valores nas *queries* teve também que sofrer alterações devido à modificação que foi feita para MSSQL. Foi verificado na documentação que é necessário criar *inputs* de forma a poder obter os valores que se pretende e identificar o valor que se deseja para o parâmetro.

```
.input('input_parameter', sql.NVarChar, info[i].geodsg)
```

Figura 16 Exemplo da definição de um input em MSSQL

Assim que foi possível fazer esta análise e as *queries* realizadas com sucesso, foram feitas algumas alterações ao código para o uso correto das tabelas fornecidas pela base de dados. Como na tabela “obitos”, a freguesia está identificada pelo id (Freg_id) teve que se implementar uma *query* para verificar qual o valor que está a ser lido no ficheiro, e poder obter o respetivo id. Para isto,

a definição de uma outra *query* para, a partir do nome da freguesia, selecionar o id na tabela, permite enviar o respetivo resultado para a *query* seguinte.

```
let result1 = await pool.request()
.input('input_parameter', sql.NVarChar, info[i].geodsg)
.query('select Freguesia.id FROM Freguesia WHERE freguesia.nome = @input_parameter')
```

Figura 17 Query para verificar o id da freguesia a partir do nome

Com isto, o “result1” contém o valor da *query*, sendo este valor apresentado como um objeto. A partir deste ponto, é necessário obter o resultado anterior e fazer a verificação mencionada anteriormente, se já existe um determinado ano e freguesia como registo.

```
let result2 = await pool.request()
.input('input_parameter', sql.Int, keys)
.input('input_parameter2', sql.Int, result1.recordset[0]['id'])
.query('select ano, Freg_id from obitos WHERE ano = @input_parameter AND Freg_id = @input_parameter2')
```

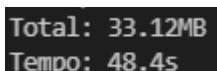
Figura 18 Query para a verificação de um determinado registo

O valor “keys” contém o ano do ficheiro que está a ser lido no momento e o “result1.recordset[0][“id”]” contém o resultado da *query* anterior, ou seja, o id do nome da freguesia que foi lido. Assim que estas verificações são feitas, é possível diferenciar se um determinado registo existe na base de dados ou não, sendo feito então um “insert” ou “update” consoante o resultado obtido.

```
{ ano: 2014, Freg_id: 1, obtM: 84, obtH: 92, obtTotal: 176 },
{ ano: 2014, Freg_id: 4, obtM: 10, obtH: 20, obtTotal: 30 },
{ ano: 2014, Freg_id: 7, obtM: 18, obtH: 19, obtTotal: 37 },
{ ano: 2014, Freg_id: 9, obtM: 34, obtH: 45, obtTotal: 79 },
{ ano: 2014, Freg_id: 5, obtM: 6, obtH: 9, obtTotal: 15 },
{ ano: 2014, Freg_id: 10, obtM: 19, obtH: 18, obtTotal: 37 },
{ ano: 2014, Freg_id: 8, obtM: 49, obtH: 56, obtTotal: 105 },
{ ano: 2014, Freg_id: 2, obtM: 54, obtH: 52, obtTotal: 106 },
{ ano: 2014, Freg_id: 3, obtM: 110, obtH: 120, obtTotal: 230 },
{ ano: 2014, Freg_id: 6, obtM: 23, obtH: 19, obtTotal: 42 },
{ ano: 2015, Freg_id: 1, obtM: 85, obtH: 91, obtTotal: 176 },
{ ano: 2015, Freg_id: 4, obtM: 17, obtH: 7, obtTotal: 24 },
{ ano: 2015, Freg_id: 7, obtM: 11, obtH: 22, obtTotal: 33 },
{ ano: 2015, Freg_id: 9, obtM: 35, obtH: 38, obtTotal: 73 },
{ ano: 2015, Freg_id: 5, obtM: 7, obtH: 6, obtTotal: 13 },
{ ano: 2015, Freg_id: 10, obtM: 22, obtH: 22, obtTotal: 44 },
{ ano: 2015, Freg_id: 8, obtM: 69, obtH: 48, obtTotal: 117 },
{ ano: 2015, Freg_id: 2, obtM: 64, obtH: 72, obtTotal: 136 },
{ ano: 2015, Freg_id: 3, obtM: 134, obtH: 134, obtTotal: 268 },
{ ano: 2015, Freg_id: 6, obtM: 34, obtH: 25, obtTotal: 59 },
```

Figura 19 Dados inseridos na tabela obitos na base de dados externa

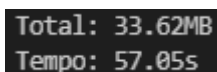
Na fase final, foi pedido o registo dos tempos e memória usada durante a execução do programa, incluindo as transferências dos ficheiros. Esta verificação permite avaliar o desempenho do código desenvolvido. Inicialmente foi usado o “console.time” para iniciar um temporizador e “console.timeEnd” para terminá-lo e ser apresentado no terminal o resultado. Foi encontrado um problema no registo dos tempos porque este era retornado antes das *queries* terminarem, impossibilitando esta verificação. Ao verificar novamente a documentação, devido às funções assíncronas, não estava a ser usado o operador “await”, que permite uma “pausa” até que seja retornado o valor esperado. Como as *queries* estão dentro de um *loop*, com o uso do “await”, assim que sejam terminadas, é mostrado o tempo apenas no fim da execução das queries. De forma a verificar a memória foi pedido que se usasse o módulo “performance” ([Referência9](#)) mas foi verificado que este módulo é usado para a *web browsers* e APIs, tendo sido encontrado uma alternativa que foi o “process” ([Referência10](#)). O “process” já vem incorporado no nodejs, não sendo necessário o uso de módulos, fornecendo informação relativamente ao processo que está a decorrer. A função “memoryUsage” permite fazer a verificação da memória, em *bytes*, que é alocada ao programa durante a sua execução. É retornado um objeto que contém vários parâmetros, onde foi usado “memoryUsage.heaptotal”, para retornar apenas o valor relativamente à memória total usada no momento.



```
Total: 33.12MB
Tempo: 48.4s
```

Figura 20 Memória usada e duração da execução dos anos de 2014 a 2019

Apenas na aproximação do fim do estágio curricular, o INE disponibilizou os dados relativamente ao ano de 2020, podendo assim fazer a verificação e uma comparação no uso dos dados de 2014 a 2019, com os anos de 2014 a 2020. É possível verificar um pequeno aumento no uso da memória e aproximadamente 9 segundos de diferença.



```
Total: 33.62MB
Tempo: 57.05s
```

Figura 21 Memória usada e duração da execução dos anos de 2014 a 2020

CONCLUSÕES

Com a finalização deste estágio, posso concluir que foi sem dúvida uma experiência enriquecedora e produtiva, quer a nível pessoal como profissional. Relativamente ao 1º ciclo que foi desenvolvido, os objetivos foram cumpridos, elaborando com sucesso a recolha de dados de uma fonte de dados, a sua filtragem e inserção numa base de dados

Infelizmente, devido à carga horária e quantidade de trabalhos que foram propostos nas unidades curriculares que foram realizadas em conjugação com o estágio, não foi possível realizar o 2º ciclo no desenvolvimento de trabalho como foi proposto. Apesar deste aspeto, no geral, o balanço deste estágio foi positivo, proporcionando uma qualidade autodidata pelo uso de tecnologias totalmente desconhecidas, em termos práticos.

A realização deste estágio foi considerada muito importante para, assim que sejam encontradas dificuldades, haver uma capacidade e vontade de as superar, permitindo uma preparação para a vida no mercado de trabalho.

BIBLIOGRAFIA

(Referência1)

https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_indicadores&indOcorrCo d=0008235&contexto=bd&selTab=tab2.

(Referência2)

https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=hlth_cd_asdr2&lang =en

(Referência3)

<https://www.pordata.pt/Portugal/%c3%93bitos+de+residentes+em+Portugal+tot al+e+por+sexo-3500>

(Referência4)

https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_api&INST=322751522&x lang=pt

(Referência5)

https://www.ine.pt/ine/json_indicador/pindica.jsp?op=2&varcd=0008235&Dim1= S7A2019&Dim2=11A130603&lang=PT

(Referência6)

https://www.ine.pt/ine/json_indicador/pindica.jsp?op=2&varcd=0008235&Dim1= S7A2019&Dim2=11A130618&lang=PT

(Referência7)

https://www.ine.pt/ine/json_indicador/pindica.jsp?op=2&varcd=0008235&Dim1= S7A2019&Dim2=11A130603&lang=PT

(Referência8) <https://www.npmjs.com/package/mssql>

(Referência9) <https://developer.mozilla.org/en-S/docs/Web/API/Performance>

(Referência10) <https://nodejs.org/api/process.html>

ANEXOS