



# INSTITUTO POLITÉCNICO DA MAIA **IPMAIA**

**INSTITUTO POLITÉCNICO DA MAIA**

**RELATÓRIO FINAL**

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

Fábio Daniel Alves Azevedo  
LICENCIATURA EM TECNOLOGIAS DE INFORMAÇÃO WEB E MULTIMÉDIA  
Pedro Correia Cravo Pimenta  
Rui Alexandre Salgado Ruela Branco Carreira

Castelo da Maia 28/06/2021

# FICHA DE CARACTERIZAÇÃO

**Instituição de Ensino Superior:**

INSTITUTO POLITÉCNICO DA MAIA - IPMAIA

**Licenciatura em Tecnologias de Informação Web e Multimédia**

**Entidade Acolhedora:**

**Departamento onde realizou o estágio:**

**Nome Completo do Aluno:**

**N.º de Aluno:**

**Nome do Orientador na Entidade Acolhedora:**

**Nome do Supervisor do IPMAIA:**

**Duração do Estágio:**

\_\_\_\_ h

**Período de Realização do Estágio:**

\_\_ / \_\_ / \_\_\_\_ a \_\_ / \_\_ / \_\_\_\_

**Validação:**

Assinatura do Aluno: \_\_\_\_\_

Assinatura do Orientador: \_\_\_\_\_

Assinatura do Supervisor: \_\_\_\_\_

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 documento tem como finalidade apresentar o relatório de estágio necessário para a obtenção do certificado de Técnica Superior Profissional de Nível VI do Quadro Nacional de Qualificações. Este é um relatório de Estágio Curricular, integrado na Licenciatura de Tecnologia de Informação, Web e Multimédia, executado na entidade Câmara da Maia sediada no Concelho da Maia. O estágio realizou-se no período compreendido entre 1 de Março de 2021 até 16 de junho de 2021, com uma duração de 250 horas.

Durante o período de estágio curricular foram atribuídas várias tarefas dentro do projeto BaZe, o qual integrei como Data Engineer e das quais se destaca:

- Pesquisa e obtenção de dados sobre o a quantidade de ligações de internet no concelho da Maia.
- Desenvolvimento de um software em JavaScript que obtém os dados, filtra e envia para a base de dados da entidade.

Palavras-chave:

BaZe, Data Engineer, JavaScript

## Agradecimentos

Um sincero agradecimento ao por todo o apoio prestado na elaboração deste Relatório de Estágio Profissional e por toda a disponibilidade e empenho demonstrados.

A todos os meus colegas de estágio que me ajudaram e se mostraram disponíveis sempre que necessário. A todos os professores do IPMaia por sempre se terem mostrado disponíveis para ajudar em tudo o que lhes fosse possível. A todos os meus colegas e amigos de faculdade que me acompanharam durante todo o meu percurso académico e o tornaram inesquecível. À minha família por ter acreditado nas minhas capacidades e, apesar de complicado, abdicaram de muitos momentos e meios para que fosse possível atingir este grau.

## Índice

Introdução .....	7
Expectativas Iniciais .....	8
Descrição e caracterização da entidade acolhedora .....	9
Descrição das tecnologias utilizadas pela empresa.....	10
Descrição das atividades realizadas pela empresa .....	11
Projeto Baze .....	11
Conceito .....	11
Objetivos gerais.....	12
Objetivos específicos.....	12
Atividades Desenvolvidas.....	13
Sumário semanal das atividades de estágio .....	13
Pesquisa e obtenção de dados.....	16
Desenvolvimento .....	18
Refactoring.....	19
Conclusão .....	23
Referências .....	24
Anexos .....	25
Anexo 1 - Mercado de trabalho para JavaScript .....	26
Anexo 2 - Excerto de ficheiro INE .....	27
Anexo 3 – Obtenção de dados.....	28
Anexo 4 – Ficheiro de leitura .....	29
Anexo 5 – Conexão ao servidor MySQL .....	30
Anexo 6 – Conexão ao servidor MsSQL .....	31

## Índice de Figuras

Figura 1 Excerto da documentação da API .....	17
Figura 2 Excerto de página o INE.....	17
Figura 3 pedido HTTP .....	18
Figura 4 classe Dados.....	19
Figura 5 Base de dados local .....	19
Figura 6 Nova classe.....	20
Figura 7 função de escrita pra ficheiros .....	20
Figura 8 função de criação de objetos .....	20
Figura 9 Ligação MsSQL.....	21
Figura 10 exemplo de pedido MsSQL .....	21
Figura 11 medida de performance.....	22
Figura 12 teste de performance.....	22

## Introdução

O estágio curricular tem como propósito estabelecer a ligação entre os conteúdos adquiridos ao longo do percurso académico e as atividades em contexto de trabalho prático.

Este relatório de estágio provém da licenciatura em Tecnologias de informação, web e multimédia que contém cerca de 250 horas práticas em contexto de trabalho.

Estágio que teve a duração de aproximadamente 4 meses, tendo decorrido de 1 de Março de 2021 a 13 de junho de 2021, totalizando assim as 250 horas na entidade, com o intuito de preparar os formandos para a vida ativa e para um futuro profissional no mercado de trabalho.

Este relatório está então organizado de forma a descrever as experiências, os diferentes tipos de atividades realizadas, e a aprendizagem adquirida durante o Estágio.

## Expectativas Iniciais

O estágio é sempre algo que fomenta alguma expectativa, algumas ideias realistas outras nem tanto. Quando chegou a altura de começar o estágio estava pessoalmente algo empolgado com o facto de ir trabalhar na área que estudei sendo esta desenvolvimento e programação e, portanto, estava á espera que todo o meu trabalho fosse de alguma maneira associado a essa área e que fosse algo desafiante e empolgante.



## Descrição e caracterização da entidade acolhedora

A Câmara Municipal da Maia é o órgão autárquico do concelho da Maia e cabe-lhe promover o desenvolvimento do município em todas as áreas da vida, como a saúde, a educação, a ação social e habitação, o ambiente e saneamento básico, o ordenamento do território e urbanismo, os transportes e comunicações, o abastecimento público, o desporto e cultura, a defesa do consumidor e a proteção civil.

## Descrição das tecnologias utilizadas pela empresa

### Comunicação

Em termos de comunicação entre equipa a empresa utiliza as plataformas da google como plataforma de eleição sendo que o Google Drive é a plataforma de eleição para partilha de ficheiros, utiliza também o Hangouts que é plataforma da google de mensagens instantânea, o Google Meet para as reuniões entre equipa assim como o Google Calendar para a marcação das mesmas. É de mencionar que todas estas ferramentas são gratuitas sendo que o único requisito de uso é ter uma conta Google.

### Desenvolvimento

No que toca a desenvolvimento dentro do projeto o qual estou integrado temos o uso de bases de dados da Microsoft (Windows Server Standard). O Windows Server Standard é um sistema operacional para servidor que permite a um computador operar os serviços de rede, tais como servidor de impressão, controlador de domínio, servidor de web e servidor de arquivo. Ele serve ainda como plataforma para produtos de servidor adquiridos separadamente, tais como o Exchange Server e o SQL Server. A edição Standard foi elaborada para organizações pequenas e médias com servidores físicos ou levemente virtualizados.

## Descrição das atividades realizadas pela empresa

### Projeto Baze

Balanco Zero de Carbono é um programa de ações e projetos integrados no conceito de laboratório vivo implementadas em contexto real. O Living Lab Maia.

É um projeto que permite a aplicação, demonstração, apreciação e apropriação de soluções integradas e transversais que promovem a descarbonização enquanto fator determinante na construção da sustentabilidade.

O plano de implementação do Living Lab Maia: Net Zero Carbon City inclui iniciativas tangíveis e intangíveis, de carácter inovador, disruptivo e com impacto na descarbonização do espaço urbano.

A proposta é marcadamente multisectorial e multidisciplinar, abrangendo todas as áreas com influência na transição do paradigma de urbanidade em curso, nomeadamente a Energia, Economia Circular e Ambiente, Mobilidade e de uma forma transversal a cidadania - sensibilização, partilha e envolvimento da comunidade. (Baze, 2021)

### Conceito

O BaZe - Living Lab Maia: Net Zero Carbon City combinará ações de mobilidade sustentável, eficiência energética, energias renováveis, economia circular e de envolvimento do cidadão, assentes em modelos de negócio de promoção da descarbonização, que se esperam viáveis e replicáveis, por forma a construir um ecossistema urbano, escalável e replicável.

Todo este ecossistema será continuamente monitorizado garantindo que o consumo e produção de recursos seja gerido de forma inteligente e sustentável, apoiando a otimização de processos que visem a redução dos impactos económicos, sociais e ambientais inerentes à procura global por energia. Este será suportado por uma arquitetura de *smart meters* (dispositivos eletrónicos que registam informação diversificada, comunicando-a), considerando o potencial do fenómeno da Internet das Coisas (IoT) e dos sistemas de computação na Cloud para capacitar os gestores do território com ferramentas avançadas de monitorização, controlo e otimização das infraestruturas e equipamentos.

As soluções irão oferecer uma experiência interativa para experimentar, em espaço urbano, momentos de compreensão e aprendizagem relativamente à gestão de recursos da urbe. Por outro lado, e num conceito mais desmaterializado, serão realizadas inúmeras ações de capacitação, assim como irão ser promovidos espaços de co-criação de e para o cidadão, traduzindo conhecimento, ações e linguagem de alto nível em ações concretas de fácil compreensão, mas de elevado valor acrescentado.

O desenvolvimento do projeto e a simbiose das tecnologias interdisciplinares, têm o suporte de um conjunto de fatores que garantem o sucesso da implementação das medidas e ações propostas. Para além das entidades parceiras, que apresentam experiência em projetos de implementação e investigação nestes domínios, espera-se ainda a integração de diversos *stakeholders*, numa lógica de interação entre o município, os centros de conhecimento, as empresas, as indústrias e os cidadãos.

A definição das ações teve por base a ideia de, num espaço relativamente próximo, serem testadas múltiplas ações inovadoras e que no seu conjunto não haviam sido testadas em território nacional, em simultâneo, de forma tão integrada e tão próxima do cidadão. (Baze Conceito, 2021)

## Objetivos gerais

Os grandes objetivos da entidade passam por:

Reforçar a associação da imagem da cidade da Maia à inovação, sustentabilidade, inclusão e resiliência, com vista a melhorar a qualidade de vida dos cidadãos e das comunidades;

Fomentar a descarbonização da cidade através da implementação de soluções tecnológicas e ações imateriais que aumentem a eficiência e reduzam o consumo de energia, água e outras matérias, bem como reduzir a produção de resíduos e desperdício;

Fomentar a demonstração e teste de soluções tecnológicas integradas, em contexto real, que tenham potencial comprovado de ser escaladas para a cidade como um todo, com possibilidade de replicabilidade em qualquer cidade nacional e/ou internacional;

Sensibilizar a população para os benefícios da adoção de comportamentos sustentáveis e apropriação tecnológica, através do desenvolvimento de ações didáticas em registo pedagógico com as escolas, comunidades, instituições e empresas;

Promover a cidadania ativa, por via de um estímulo à participação dos cidadãos na vida das comunidades onde vivem e trabalham;

Promover um ambiente urbano sustentável e descarbonizado.

Envolver na ação, ativamente, o universo dos colaboradores do município fazendo dessa expressiva massa humana o primeiro fator de disseminação na comunidade local. (Objectivos Gerais, 2021)

## Objetivos específicos

Envolver e mobilizar o cidadão para a adoção de comportamentos sustentáveis.

Promover a gestão da procura e a eficiência energética, reduzindo as emissões de gases com efeito estufa das atividades e serviços realizados no espaço do Living Lab e na sua envolvente;

Promover a utilização de energias renováveis através da demonstração de sistemas de microprodução em regime de autoconsumo com apoio de armazenamento;

Testar ideias, tecnologias e processos inovadores em ambiente real e controlado;

Aumentar a conectividade ao nível das tecnologias de informação e comunicação recorrendo à Internet das Coisas (IoT) e sistemas de computação na *Cloud*;

Atrair e promover investimento em tecnologias e novos modelos de negócio associados à descarbonização;

Validar as soluções testadas e desenvolver a sua adaptabilidade e padronização com vista à construção do modelo de gestão tecnológica extensiva do território municipal;

Estimular o conjunto dos parceiros institucionais e tecnológicos para o exercício de extrapolação do "conhecimento líquido" obtido para ferramentas de apoio à reflexão prospetiva no ordenamento territorial;

Analisar o potencial de partilha e exportação dessa liquidez de know how para a escala metropolitana, nacional e rede de contactos internacionais. (Objectivos Especificos, 2021)

# Atividades Desenvolvidas

## Sumário semanal das atividades de estágio

### Semana 1

Inicialmente fez-se um estudo sobre os documentos fornecidos pela entidade acolhedora sendo eles:

ISO 37120 - Documento que define metodologias e indicadores para que seja possível medir o desempenho dos serviços municipais, assim como a qualidade de vida dos cidadãos do município.

REOT (Relatório do Estado do Ordenamento do Território) - Documento que contém vários indicadores já registados pelo município.

Apresentação protótipo BaZe - Slideshow com os pontos mais cruciais sobre o projeto Baze.

Carta educativa - Documento, elaborado pelos Departamento de Sociologia da FLUP e Instituto de Sociologia da Universidade do Porto, com informação detalhada sobre a educação no município da Maia.

### Semana 2

Continuação do estudo dos documentos fornecidos pela entidade acolhedora.

Como resultado deste estudo surgiu um tema ligado a telecomunicações que apesar de não ser um tema crucial segundo o documento ISO 37120 é ainda assim um tema o qual requer atenção e deve ser abordado. Posto a escolha do tema fez-se uma breve pesquisa de possíveis fontes de dados tendo com resultado preliminar o INE (Instituto Nacional de Estatística) e o site Pordata.

Posto escolhidos os temas fez-se um plano de trabalho e fez-se a escolha de quais as ferramentas de desenvolvimento que iriam ser utilizadas que neste caso foram Node + JavaScript, esta escolha foi fundamentada no facto de JavaScript ser uma linguagem bastante utilizada no mercado atualmente.

### Semana 3

Fez-se uma pesquisa sobre dados referentes ao tema que escolhido para o 1º ciclo (Dados de telecomunicação) e como resultado obteve-se o seguinte:

- INE
- Pordata
- Eurostat
- Anacom
- knoema

Depois de executada a pesquisa para fontes de dados fez-se uma comparação entre fontes de dados mais relevantes de maneira a perceber a veracidade das mesmas.

Posto isto fez-se também uma pesquisa sobre como utilizar a API do INE que se baseou nos documentos fornecidos pelo próprio INE, neste caso eles fornecem um documento de ajuda para extrair dados ou meta dados da sua base de dados em json.

### Semana 4

Visto o INE ser a única fonte com uma API e a qual os dados aparentemente serem fidedignos decidiu-se utilizar esta fonte para o trabalho de obter os dados.

Começamos então por analisar os dados fornecidos em json assim como maneiras de obtê-los pela API.

Estive ainda a fazer uma pesquisa referente a possíveis cidades com características semelhantes a Maia fora do país.

## **Semana 5**

Começou-se a conceptualizar os ficheiros de código assim como lógica e metodologia do mesmo chegando a conclusão de que:

Por questão de boas práticas vão ser divididos as funções em vários ficheiros contendo cada um deles funções referentes ao que tratam, por exemplo existirá um ficheiro que albergará todas as funções referentes a obtenção dos dados da API em json que por questão de lógica será o 1º ficheiro a terminar posto isto passarei para o ficheiro que irá tratar de ler os Json obtidos e transformar os dados em objetos.

Visto também existir dados que não são necessários passar para a base de dados, tenciono então usar programação orientada a objetos para facilitar essa filtragem.

Tendo a lógica relativamente resolvida passei então a tratar do primeiro ficheiro sendo este o que trata de ir buscar dados á API, fiz alguma pesquisa e consegui resolver um dos primeiros problemas sendo este o pedido que vai ser feito, neste caso através de um httprequest.

## **Semana 6**

Continuou-se a desenvolver o ficheiro de download fazendo algumas experiências com os pedidos e criando as respetivas funções por exemplo percebemos que era possível fazer o download do ficheiro do ano mais recente se não indicarmos o ano no pedido à API e daí criamos uma função chamada GetLast.

Entre esta criei outras funções de coisas que já tínhamos conhecimento como por exemplo uma função que faz um pedido de um ano específico e uma função que faz o pedido a um conjunto de anos (sendo que é feito 1 pedido de cada vez por limitação da API). Para terminar criei uma função que iria guardar os dados enviados pela API para um ou vários ficheiros locais. Ainda esta semana comecei a tratar do ficheiro que trata de ler os dados dos ficheiros locais inserindo uma função simples de leitura.

## **Semana 7**

Esta semana tratei de focar no ficheiro que trata da leitura dos ficheiros locais criando mais funções que leem os ficheiros e criam objetos dos dados contidos nos mesmos como podemos observar o exemplo da função que transforma todos os locais num objeto e o envia para uma lista.

Posto isto criei uma função que faz algo semelhante para os dados e para o indicador de maneira que futuramente se necessário esta função possa ser reaproveitada para futuros trabalhos com outros indicadores e dados que sejam da mesma fonte (INE).

Ainda aprimorei o código do ficheiro que trata dos pedidos à API tornando mais dinâmico para que, como referido anteriormente, possa ser utilizado para outros indicadores da mesma fonte de dados.

## **Semana 8**

Tendo os ficheiros de leitura e pedido a API comecei então a criar um ficheiro que tratasse da ligação à base de dados. Para testar a ligação decidi criar a base de dados localmente e testar a ligação, por questões académicas já tinha localmente uma base de dados criada no programa Microsoft SQL Server infelizmente devido a falta de suporte nativo de Node para este tipo de base de dados (TSQL) decido então fazer uma base de dados usando o software popular XAMPP que usa MariaDb uma base de dados MySQL.:

Foi então desenvolvida uma nova base de dados em equipa com um colega que está a utilizar a mesma fonte de dados e, portanto, de maneira a poder usar a mesma tabela para ambos os trabalhos decidimos criar em conjunto a base de dados.

Criada a base de dados decidi testar o software que depois de alguma correção de bugs estava efetivamente a enviar os dados para a base de dados através de algumas queries simples.

## **Semana 9**

Nesta semana decidi tratar então de começar a aprimorar o código em termos de segurança especialmente no que toca as Query enviadas para base de dados. Comecei por criar um método que faz uma Query de verificação a base de dados de maneira que se esta já tiver dados relativos aquele objeto esta vai então fazer um update se o objeto ainda não for existente então vai ser criada na base de dados.

## **Semana 10**

Posto ter um código mais robusto decidi então fazer algumas alterações a base de dados deixando mais simples pois nem todas as informações que estavam a ser guardadas eram necessárias portanto decidi refazer a base de dados para algo muito mais simples.

De momento a nova base de dados apenas guarda o nome da localidade, valor total de Ligações internet nessa localidade e o ano correspondente aos dados.

Também em termos de código criei uma nova função que preenche os dados desta tabela apenas com as localidades da Zona do Porto.

## **Semana 11**

Encontrados problemas com a ligação à base de dados da entidade o que levaram a várias horas de pesquisa e debugging de código. Encontrado o problema foi necessário estudo de documentação em relação a base de dados MSSQL.

## **Semana 12**

Alterações feitas a nível da conexão à base de dados para que possa ser estabelecida a conexão para MsSQL, ainda tiverem de ser realizadas alterações profundas a nível do código para que possa compactar com uma base de dados MsSQL ou invés de MySQL.

## **Semana 13**

Feitos testes de performance ao código e consequentemente alguns ajustes ao mesmo de maneira que a performance seja a melhor possível.

## **Semana 14**

Feitos novos testes de performance ao código e algum refactoring e ajustes ao mesmo.

## Pesquisa e obtenção de dados

Escolhi integrar na equipa de *Data Engineer* do projeto Baze o que fez com que as minhas funções resumidamente fossem compostas pela criação, design e construção dos ambientes de dados, incluído o seu sistema de processamento e armazenamento.

Portanto, inicialmente para executar esta função foi-me dada a opção de que dados queria tratar, sendo que, depois de um estudo feito aos documentos relevantes optei por tratar dos dados referentes às ligações de internet do concelho da Maia, tendo escolhido o tema passei a escolha da ferramenta a qual me foi dada liberdade para escolher a mesma, decidi portanto escolher JavaScript e Node pois seriam uma mais valia para o meu currículo assim como são bastante requeridas no contexto de mercado de trabalho como se pode ver no anexo 1.

Posto estarem feitas as minhas escolhas em termos de ferramentas e tema decidi então começar a fazer uma pesquisa extensiva sobre todas as possíveis fontes de dados sendo que resultado dessa pesquisa encontrei várias fontes de dados sendo estas:

- INE- dados que podem ser obtidos através de uma API, são fornecidos via ficheiro json e contém dados referentes a cada concelho.
- Pordata- dados provenientes do INE
- Eurostat- Apenas contém dados referentes ao país como um todo e, portanto, não é relevante para o que é pretendido neste momento.
- Anacom - Os dados podem ser obtidos via xls ou via pdf sendo que são apenas referentes ao país.
- knoema - Tem dados referentes apenas a países e os dados apenas podem ser obtidos em xls e outros formatos menos relevantes.

Depois de terminada a pesquisa decidi fazer então uma comparação entre as fontes que contém os dados relevantes para perceber a veracidade das mesmas tendo em conta que para isto posso usar os dados fornecidos pela Câmara da Maia no ficheiro REOT.

	INE	Pordata	REOT
2019	138 971	138 349	
2018	137 727		
2017	136 769		
2016	136 011		136 011
2015	135 678		135 678
2014	135 817		135 817
2013	135 924		135 924
2012	136 017		136 017
2011	135 937		135 306
2010		134 759	

Posto isto decidi então usar a base de dados do INE pelo facto de os dados serem fidedignos e por ter a maior quantidade de dados das 3 fontes possíveis.

Comecei então a fazer alguma investigação sobre a base de dados do INE e a ler a sua documentação para que a pudesse utilizar, apesar de existir documentação da mesma tive alguma dificuldade em encontrar os dados que se pretendia, visto que até este momento, não encontrei uma tabela onde facilmente possa encontrar os valores necessários para fazer o preenchimento do pedido à API. Para solucionar este problema recorreu-se a tentativa/erro até que depois de algumas horas chegou-se a conclusão que se podia utilizar o site do INE para encontrar os códigos necessários à API pois ao utilizar o site “normal” do INE os *url* contém os códigos necessários como se podem ver nas imagens seguintes



## API access and extraction

{host\_url}/ine/json\_indicador/pindica.jsp?op=2&varcd={varcd\_cod}&Dim1={dim1\_cod}&Dim2={dim2\_cod}&Dim??={dim??\_cod}&lang={lang}

- **host\_url** -> Host address of INE's website, (<https://www.ine.pt>)
- **varcd\_cod** -> Code of the indicator
- **dim1\_cod** -> Code of the time dimension, (example: S7A2015)
- **dim2\_cod** -> Code of the geographic dimension, (example: 11)
- **dim??\_cod** -> Any other dimension, as long as it is available in the chosen indicator
- **lang** -> Language intended for the extraction result ("PT" or "EN")

Figura 1 Excerto da documentação da API

Esta imagem representa os códigos necessários para fazer um pedido à API do INE sendo que não precisamos que todos sejam preenchidos, precisamos exclusivamente para que funcione de fornecer o código do indicador, ano ao qual pertencem os dados e a linguagem que queremos. Como no ano de língua temos um exemplo é fácil testar, mas sem código de indicador a API não funciona corretamente e, portanto, depois de alguma pesquisa, como já mencionado anteriormente, conseguimos perceber que podias ver o código no *url* do INE como se pode observar na imagem seguinte.

[https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine\\_indicadores&contexto=pi&ndOcorrCod=0008273&selTab=tab0](https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_indicadores&contexto=pi&ndOcorrCod=0008273&selTab=tab0)

The screenshot shows the INE website interface. At the top, there is a navigation bar with the INE logo and the text 'INSTITUTO NACIONAL DE ESTATÍSTICA STATISTICS PORTUGAL'. Below this, there is a search bar and a navigation menu with options like 'Estatísticas', 'Produtos', 'Webinq', etc. The main content area shows a breadcrumb trail: 'Início / Produtos / Base de Dados'. Below this, there are tabs for 'Principais indicadores', 'Principais quadros', 'Base de dados' (selected), 'Microdados', and 'Estatísticas'. The 'Base de dados' tab is active, showing a table of data. The table has columns for 'Localização geográfica (NUTS - 2013) (1)', 'Período de referência dos dados (2)', and 'Tipo de consumo (3)'. The table contains data for various locations and years, with columns for 'Total', 'Doméstico', and 'Não Doméstico'.

Figura 2 Excerto de página o INE

Obtendo assim o código do nosso tema podemos então passar à análise dos dados fornecidos para que se possa adotar planear a filtragem dos mesmos.

Analisando os dados fornecidos em json assim como maneiras de obtê-los pela API em que chegou-se às seguintes conclusões:

- É apenas possível ou pedir os dados de todos o país ou de uma localização específica o que não nos deixa fazer o pedido apenas da área que queremos neste caso Área metropolitana do Porto.
- Existem 3 tipos de valor (valor total, valor ligação residencial e valor não residencial) que são referentes a ligações empresariais ou a residências, assim como a soma das duas.
- Só conseguimos fazer os pedidos de 1 ano de cada vez.
- Temos informação referente ao indicador, locais, data extração e data da última atualização daquele indicador

Estas conclusões podem ser maioritariamente vistas no Anexo 2 que contém um excerto de um ficheiro. Ainda a pedido estive a fazer uma breve procura sobre cidades que possam ser semelhantes a cidade da maia sendo que dessa pesquisa encontrei as seguintes cidades:

- Alemanha - Würzburg e Regensburg
- Polónia - Ruda Śląska e Tychy
- Japão - Kuki

estas cidades aproximam-se a Maia em termos de população total e densidade populacional, apesar de que não consegui encontrar base de dados que forneçam os mesmos dados que procura para a maia para essas cidades e, portanto, esta parte ficou em “stand by”.

## Desenvolvimento

Antes de começar qualquer tipo de desenvolvimento comecei por elaborar os passos que queria seguir, ou seja, por onde iria começar. Decidi então elaborar o seguinte caminho de desenvolvimento:

1. Tratar de obter os dados da API da melhor maneira.
2. Tentar filtrar os dados de maneira que não hajam dados desnecessários ou redundantes.
3. Criar uma Base de dados local em que possa testar o programa.
4. Fazer *refactoring* e melhorias de performance.

Tendo então delineado o caminho a seguir comecei então por tratar da obtenção dos dados para isto criei um ficheiro em JavaScript com este intuito. Neste ficheiro criei então uma função que executa um pedido *httprequest* como se pode ver na imagem seguinte:



```
function Get(urlrequest){  
    var Httpreq = new XMLHttpRequest();  
    Httpreq.open("GET",urlrequest,false);  
    Httpreq.send(null);  
    return Httpreq.responseText;  
}
```

Figura 3 pedido HTTP

Posto isto uso esse método num outro método que pega na resposta a esse pedido e escreve a resposta ao mesmo em ficheiros *json* usando. Como cada pedido da API apenas envia os dados referentes apenas a 1 ano tive então de fazer com que este processo ocorresse por cada ano que fosse necessário gerando então assim vários ficheiros (neste caso em específico ficheiros referentes aos anos 2011 ate 2019), como nome para o ficheiro usei o ano a que correspondem de maneira a serem facilmente identificáveis usando também a extensão *.json* para que sejam ficheiros desse mesmo tipo como se pode observar no Anexo 3, também neste anexo podemos observar que criei outras funções que poderia mais tarde serem necessários sendo que um apenas iria retirar dados do ano mais recente assim como outro que iria retirar os dados de um ano específico dado pelo utilizador. Todos estes funções foram inicialmente criados num ficheiro separado do resto do programa para que mais tarde quando tivesse uma melhor noção do que fosse preciso poderia só os que precisasse.

Tento então tratado da obtenção de dados comecei então a tratar da leitura e filtragem desses ficheiros e para isso decidi então criar uma classe para que fosse mais fácil saber quais os parâmetros desses dados como podemos observar na imagem seguinte:

```

class Dados {
  constructor(id,valor,ano,local,indicador,fonte,tipo) {
    this.id= id;
    this.valor = valor;
    this.ano = ano;
    this.local = local;
    this.indicador= indicador
    this.fonte = fonte;
    this.tipo = tipo;
  }
}
exports.Dados = Dados

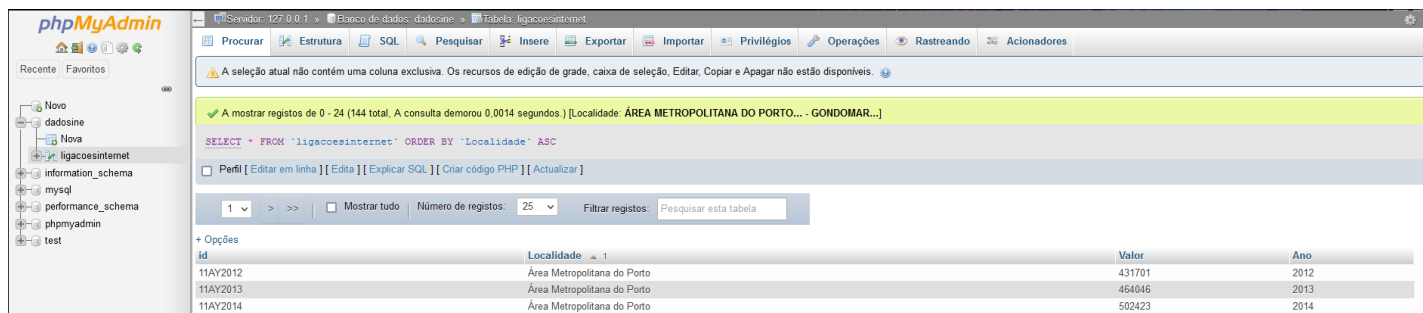
```

Figura 4 classe Dados

Tendo então ficheiros com os dados comecei a tratar de ler esses dados primeiro decidi criar uma função que lê um ficheiro json e outra função que usa esse ficheiro para adicionar vários objetos do tipo “Dados” a uma lista utilizando apenas os valores necessários como podemos observar no anexo 4.

Testando este código e tendo tudo funcional comecei então assim a criação de uma base de dados local onde pudesse se testar a parte de ligação a uma base de dados, para isto decidi então usar o software XAMP para Windows que me permite criar uma base de dados MySQL. Para isto criei então um ficheiro o qual chamei *Server.js* que iria servir de template para a possível base de dados real e esse ficheiro pode ser observado no anexo 5:

Posto alguns ajustes ao código a base de dados e algum debugging esta tudo funcional e a base de dados estava a ser preenchida pelos dados



id	Localidade	Valor	Ano
11AY2012	Área Metropolitana do Porto	431701	2012
11AY2013	Área Metropolitana do Porto	464046	2013
11AY2014	Área Metropolitana do Porto	502423	2014

Figura 5 Base de dados local

Depois disto foi-me proposto então simplificar a base de dados assim como obtive acesso a base de dados da entidade assim comecei então a trabalhar no refactoring assim como na ligação a base de dados.

## Refactoring

Posto o código estar funcional comecei então por refatorar o mesmo sendo que iniciei esta tarefa remover os vários ficheiros que tinha criado e passado todas as funções para 1 só ficheiro que continha tudo. Para complementar o refactoring também decidi mudar a minha *Naming convention* para algo que fosse mais correto e mais legível. Antes de começar com o refactoring decidi ajustar a classe para a base de dados

que me foi proposta sendo que esta continha apenas 3 simples campos o que servir de base para a nova classe como se pode observar na imagem seguinte:

```
class dataIndicator {
  constructor(locationName, Value, year) {
    this.locationName = locationName;
    this.Value = Value;
    this.year = year;
  }
}
```

Figura 6 Nova classe

Tendo então a classe criada passei então para o refactoring do código sendo que comecei da mesma maneira que o criei, ou seja, pela função que faz o pedido e os escreve em ficheiros json.

Nesta função não houve grandes alterações profundas todas as alterações foram maioritariamente *Naming convention* para que esta seja mais legível sendo que também adicionei uma mensagem para a consola em caso de sucesso o que não acontecia anteriormente.

```
function writteDataToFile(year,indicatorCode) {
  for (let i = 0; i < year.length; i++) {
    let fullUrl = ineBaseUrl+"varcd="+indicatorCode+"&Dim1=S7A"+year[i]+"&lang=PT"
    filewriter.writeFileSync( path: year[i]+".json", Getdata(fullUrl), options: async function(err) {
      if (err) {
        console.error(err);
      }
      else
      {
        console.log("File"+year[i]+"written successfully")
      }
    });
  }
}
```

Figura 7 função de escrita pra ficheiros

Na função de leitura e filtragem dos dados já houve algumas alterações mais profundas sendo que fizemos filtros muito mais específicos que anteriormente de maneira que a lista de objetos apenas contivesse dados referentes a zona metropolitana do Porto e apenas isso porque apenas estes eram dados relevantes para o projeto.

```
function writteDataToFile(year,indicatorCode) {...}
function filetoRead(nameOfFile) {
  let file = filewriter.readFileSync( path: nameOfFile+".json", options: 'utf8');
  return JSON.parse(file);
}
function filterData(parsedjason,indicatorData,year) {
  for (let i = 0; i < parsedjason[0].Dados[year].length; i++) {
    if (parsedjason[0].Dados[year][i].geocod.substring(0,3)=="11A" && parsedjason[0].Dados[year][i].dim_3 === "T" && parsedjason[0].Dados[year][i].geocod.length > 3)
    {
      let data = new dataIndicator(parsedjason[0].Dados[year][i].geodsg, parsedjason[0].Dados[year][i].valor, year);
      indicatorData.push(data);
    }
  }
}
```

Figura 8 função de criação de objetos

Posto então feita a filtragem do dado passei para o ajuste para a nova base de dados e este foi sem dúvida o maior desafio em termos de desenvolvimento pois a base de dados da entidade não era MySQL, mas sim MsSQL o que era algo que não tinha tanto a vontade enquanto desenvolvedor e, portanto, tive algumas dificuldades com as alterações mas nada que alguma dedicação e estudo da documentação não resolvesse e portanto decidi começar por perceber como fazer a ligação ao servidor. Ao contrário do que acontece com MySQL em MsSQL para que seja feita a ligação ao servidor é necessário dar mais parâmetros do que apenas uma utilização e a localização do servidor com podemos observar na figura seguinte:

```
const sqlConnId = {
  user: '...',
  password: '...',
  server: '...',
  port: 1433,
  pool: {
    max: 10,
    min: 0,
    idleTimeoutMillis: 2000
  },
  options: {
    encrypt: true,
    trustServerCertificate: true
  }
};
```

Figura 9 Ligação MsSQL

Como podemos observar para além de dar o utilizador e localização do servidor temos de dar as informações da *pool* que são as ligações feitas sendo que neste caso dizemos que no máximo podemos ter 10 ligações em simultâneo e que cada ligação é desligada se tiver inativa por mais de 2 segundos, também ativamos a encriptação e aceitamos o certificado do servidor que estamos a ligar, esta última parte é necessária para ligar a qualquer servidor que não seja oficial da Azure (Microsoft).

Posto ter a ligação funcional comecei a tratar das do envio de dados para a base de dados e para isso reformulei então a função que trata disto mesmo. Como podemos ver no anexo 6 existem alterações profundas nomeadamente como fazemos a ligação, por exemplo em MySQL não existe necessidade de fazer a gestão de ligações e quanto em MsSQL é necessário fazer um pedido novo assim como para enviar qualquer tipo de variável junto com a *Query* em MsSQL é necessário fazer a passagem do seu tipo de JavaScript para o tipo de SQL e ser enviado como um input especial como podemos ver na imagem seguinte:

```
let getid = await pool.request()
  .input( name: 'Concelho', sql.NVarChar, data[i].locationName)
  .query( command: `select Concelho.Id FROM Concelho WHERE Concelho = @Concelho`)
```

Figura 10 exemplo de pedido MsSQL

É de também mencionar que esta toda a ligação tem de ser feita de maneira assíncrona, ou seja, para que próxima conexão seja feita a operação anterior tem de ser terminada primeiro pois como existem verificações que são necessárias fazer nomeadamente se os dados já existem assim como verificar qual o id correspondente ao concelho.

Ao testar o programa percebi que havia um problema com a base de dados sendo que foram necessárias alterações á mesma para que o programa funcionasse corretamente. Este problema tinha haver com o uso do *text* ao invés do *varchar*.

Para terminar fiz medidas de performance sendo que medi o tempo de execução necessário assim como a utilização da memória do programa como se pode ver na figura seguinte:

```
function startProgram(){
    let t0= performance.now()
    writteDataToFile(indicatorYears,indicatorCode);
    for (let i = 0; i < indicatorYears.length; i++) {
        filterData(filetoRead(indicatorYears[i]),data,indicatorYears[i])
    }
    dataDatabase(data, sqlConfig).then(() => {
        let t1= performance.now()
        console.log('The time of execution is '+ (t1-t0)/1000 +' |sec');
        let used = process.memoryUsage().heapTotal / 1024 / 1024;
        console.log(`The program uses ${Math.round(used * 100) / 100} MB of Memory`);
    });
}
```

Figura 11 medida de performance

Em termos de performance o programa demora cerca de 70 segundos a executar de princípio a fim e usa cerca de 30 Mb de memória RAM sendo que um exemplo de um teste se encontra na imagem seguinte:

```
"C:\Program Files\nodejs\node.exe" D:\estagio\Exec.js
The time of execution is 71.18116600000859 sec
The program uses 33.52 MB of Memory
```

Figura 12 teste de performance

Posto isto tentei fazer alguns ajustes de maneira que pudesse ter a melhor performance do mesmo mas não tive sucesso significativo.

## Conclusão

O estágio é um período bastante agitado, de trabalho intenso, obrigando os estagiários a esforços e dedicação acrescido para se certificarem do desenvolvimento de todos os requisitos necessários para a sua profissão futura. Com o decorrer da experiência real de trabalho enquanto estagiário, foi-me possível sentir um crescimento contínuo da minha capacidade assim como a minha segurança perante essa capacidade. Todas as tarefas propostas foram algo desafiadoras, mas nunca demasiado complicadas sendo que todas estavam ao nosso alcance. Tudo o que se aprendeu neste estágio será útil para um futuro a curto e longo prazo. Em suma e fazendo uma análise crítica do percurso como estagiário, importa referir a persistência nas tarefas, bem como a melhoria sentida especialmente na área do desenvolvimento em JavaScript.

## Referências

Baze Conceito, C. d. (19 de 06 de 2021). Obtido de <https://www.cm-maia.pt/baze/conceito>

Baze, C. d. (06 de 18 de 2021). Obtido de <https://www.cm-maia.pt/baze>

Google. (19 de 06 de 2021). Obtido de <https://www.google.com/intl/pt-PT/drive/>

Objectivos Especificos, C. d. (18 de 06 de 2021). Obtido de <https://www.cm-maia.pt/baze/objetivos/objetivos-especificos>

Objectivos Gerais, C. d. (18 de 06 de 2021). Obtido de <https://www.cm-maia.pt/baze/objetivos/objetivos-gerais>



# Anexos

# Anexo 1 - Mercado de trabalho para JavaScript

Pesquisa feita em <https://pt.indeed.com/> a 13/06/2021

nova

**SOFTWARE ENGINEER - JAVASCRIPT, HTML, CSS - PORTO**  
Kelly Services Portugal **3,8★**  
Porto

- Working within a team to design, code, test and deliver new functionality and improve the code base of our existing range of products;

há 4 dias

nova

**.NET Developer**  
ITSector - Sistemas de Informação, SA  
Porto

- Ambitious people with a passion for this area;
- Degree in Computer Engineering or similar;
- Minimum 1-2 years of experience in .NET MVC and/or Webforms);

há 1 dia · [mais...](#)

[Visualizar todas as Ofertas de emprego de ITSector - Sistemas de Informação, SA - empregos: Porto](#)

**JUNIOR C# / .NET ENGINEER**  
Critical Manufacturing **5,0★**  
Porto

- Build, improve and extend our products for advanced manufacturing environments.
- Incorporate the needs of our customers to help them increase the performance of...

há 30+ dias · [mais...](#)

nova

**Junior Developer Javascript**  
Oddacy  
Porto

- Licenciatura em Engenharia Informática ou similar;
- Conhecimento de SQL para interação com bases de dados MySQL e formato JSON.

há 3 dias · [mais...](#)

## Anexo 2 - Excerto de ficheiro INE

```
[ {
  "IndicadorCod" : "0008450",
  "IndicadorDsg" : "Acessos à Internet em banda larga em local fixo (N.) por Localização geográfica (NUTS - 2013) e Segmento de acesso; Anual - INE",
  "MetaInfUrl" : "https://www.ine.pt/bddXplorer/htdocs/minfo.jsp?var_cd=0008450&lingua=PT",
  "DataExtracao" : "2021-05-02T22:14:00.165+01:00",
  "DataUltimoAtualizacao" : "2020-12-15",
  "UltimoPref" : "2019",
  "Dados" : {
    "2018" : [ {
      "geocod" : "185",
      "geodsg" : "Lezíria do Tejo",
      "dim_3" : "01",
      "dim_3_t" : "Residencial",
      "valor" : "64866"
    }, {
      "geocod" : "1851103",
      "geodsg" : "Azambuja",
      "dim_3" : "01",
      "dim_3_t" : "Residencial",
      "valor" : "5804"
    }, {
      "geocod" : "1851403",
      "geodsg" : "Almeirim",
      "dim_3" : "01",
      "dim_3_t" : "Residencial",
      "valor" : "5857"
    }, {
      "geocod" : "1851404",
      "geodsg" : "Alpiarça",
      "dim_3" : "01",
      "dim_3_t" : "Residencial",
      "valor" : "1756"
    }
  ]
}
```

## Anexo 3 – Obtenção de dados

```
var XMLHttpRequest = require("xmlhttprequest").XMLHttpRequest;
var xhr = new XMLHttpRequest();
var fs = require('fs');
var baseUrl = "https://www.ine.pt/ine/json_indicador/pindica.jsp?op=2&";
function Get(urlrequest){...}
//Pt: Obtem ultimo ficheiro adicionado pelo INE
function GetLast(codigo) {...}
//PT: Obtem um ficheiro com dados de um ano especifico
function GetAno(ano,codigo) {...}
//Pt: Obter varios anos
function GetAnos(anos,codigo) {
    for (let i = 0; i < anos.length; i++) {
        var url = baseUrl+"varcd="+codigo+"&Dim1=S7A"+anos[i]+"&lang=PT"
        fs.writeFileSync(anos[i]+".json", Get(url), async function(err) {
            if (err) {
                console.log(err);
            }
        });
    }
}
// exportar como modulo
module.exports = { GetAnos, GetLast, GetAno , Get};
```

## Anexo 4 – Ficheiro de leitura

```
var fs = require('fs');

//PT: escolher que ficheiro ler
function chosenfile(nome)
{
    var ficheiro = fs.readFileSync(nome+".json", 'utf8')
    var fdados = JSON.parse(ficheiro);
    return fdados
}

//Pt:Importar as classes necessarias
const {Dados} = require('./c_dados.js')
//Funcao que vai buscar o ano do ficheiro
function getAno(parsedjson)
{...}

//PT: Coloca as informacoes dos dados em objectos
function getDados(parsedjson,listadados,listalocais,ano) {
    for (let i = 0; i < parsedjson[0].Dados[ano].length; i++) {
        listadados.push(new Dados(ano+listalocais[i].id+parsedjson[0].Dados[ano][i]
    }
}

//Exportar como modulo
module.exports = { getAno, getLocais, getIndicador,chosenfile,getDados};
```

## Anexo 5 – Conexão ao servidor MySQL

```
var mysql = require('mysql');
const { Console } = require('node:console');
const con = mysql.createConnection({
  multipleStatements: true,
  host: "localhost",
  user: "fabio",
  password: "teste",
  database: "dadosIine"
});

con.connect(function(err) {
  if (err) throw err
});

// função que trata dos valores
function valorquery(valorArray) {

  for (let i = 0; i < valorArray.length; i++) {
    const valores = {idValores: valorArray[i].id , Dim: valorArray[i].tipo, valor: valorArray[i].valor, Data:va
    const check = valorArray[i].id

    con.query('SELECT * FROM valores WHERE idValores= ?;',check, async(err,result) => {
      if(err) {
        throw err;
      } else {
        if (result && result.length ) {
          con.query('UPDATE valores SET ? WHERE idValores = ?',[valores,check], async(err,result) => {
            if(err) throw err;
          });
        } else {
          console.log('No case row was found :( !');
          con.query('INSERT INTO valores SET ?',[valores, async(err,result) => {
            if(err) throw err;
          });
        }
      }
    });
  }
});
}
```

## Anexo 6 – Conexão ao servidor MsSQL

```
async function dataDatabase(data ,config) {
  await (async () => {
    try {
      let pool = await sql.connect(config)
      for (let i = 0; i < data.length; i++)
      {
        let getid = await pool.request()
          .input( name: 'Concelho', sql.NVarChar, data[i].locationName)
          .query( command: 'Select Concelho.Id FROM Concelho WHERE Concelho = @Concelho')

        let check = await pool.request()
          .input( name: 'ano', data[i].year)
          .input( name: 'Conc_id', getid.recordset[0]['Id'])
          .input( name: 'NLigacao', data[i].Value)
          .query( command: 'SELECT * FROM LigacaoInternet WHERE ano = (@ano) AND Conc_id = (@Conc_id)')
        if (check.recordset.length === 1 )
        {
          pool.request()
            .input( name: 'ano', data[i].year)
            .input( name: 'Conc_id', getid.recordset[0]['Id'])
            .input( name: 'NLigacao', data[i].Value)
            .query( command: 'UPDATE LigacaoInternet SET ano = (@ano), Conc_id = (@Conc_id),NLigacao= (@NLigacao) WHERE ano= (@ano) AND Conc_id= (@Conc_id)');
        }else {
          pool.request()
            .input( name: 'ano', data[i].year)
            .input( name: 'Conc_id', getid.recordset[0]['Id'])
            .input( name: 'NLigacao', data[i].Value)
            .query( command: 'INSERT INTO LigacaoInternet(ano,Conc_id,NLigacao) values (@ano,@Conc_id,@NLigacao)');
        }
      }
    } catch (err) {
      console.log(err);
    }
  })()
}
```