

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Improving the Developer Experience of Dockerfiles

João Pereira da Silva Matos

WORKING VERSION



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Filipe Correia

January 1, 2023

Improving the Developer Experience of Dockerfiles

João Pereira da Silva Matos

Mestrado em Engenharia Informática e Computação

January 1, 2023

Contents

1	Introdução	1
1.1	Secção Exemplo	1
1.2	Outra Secção Exemplo	2
1.3	Estrutura da Dissertação	2
2	Repairing and Generating Dockerfiles (title is WIP)	3
2.1	Background	3
2.2	Speeding up Docker builds	3
2.3	Dockerfile Generation	4
2.4	Dockerfile Smells	6
2.5	Dockerfile Good Practices	8
2.6	Dockerfile Security	8
2.7	Dockerfile Repair	8
2.8	Dockerfile Bloat	8
2.9	Dockerfile Testing	8
2.10	General Discussion	8
3	Capítulo Exemplo	9
3.1	Introdução	9
3.2	Secção Exemplo	10
3.2.1	Exemplo de Figura	10
3.2.2	Exemplo de Tabela	10
3.3	Secção Exemplo	11
3.4	Resumo	12
4	Mais um Capítulo	13
4.1	Secção Exemplo	13
4.2	Mais uma Secção	14
4.3	Resumo ou Conclusões	15
5	Conclusões e Trabalho Futuro	16
5.1	Satisfação dos Objetivos	16
5.2	Trabalho Futuro	17
	References	18

A	 Lorem Ipsum	20
A.1	O que é o <i>Lorem Ipsum</i> ?	20
A.2	De onde Vem o Lorem?	20
A.3	Porque se usa o Lorem?	21
A.4	Onde se Podem Encontrar Exemplos?	21

List of Figures

3.1	Arquitectura da Solução Proposta	10
4.1	Two Figures side by side	14
	(a) UP at the left	14
	(b) UP at the right	14

List of Tables

2.1	Works about speeding up Docker builds	4
2.2	Works about generating Dockerfiles	5
2.3	Works about Dockerfile smells	7
2.4	Works about Dockerfile good practices	8
3.1	Tabela Exemplo	11

Listings

3.1 Python example 11

Abreviaturas e Símbolos

ADT	Abstract Data Type
ANDF	Architecture-Neutral Distribution Format
API	Application Programming Interface
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CORBA	Common Object Request Broker Architecture
UNCOL	UNiversal CCompiler-oriented Language
Loren	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vehicula lorem commodo dui
WWW	<i>World Wide Web</i>

Chapter 1

Introdução

Este documento ilustra o formato a usar em dissertações na Faculdade de Engenharia da Universidade do Porto. São dados exemplos de margens, cabeçalhos, títulos, paginação, estilos de índices, etc. São ainda dados exemplos de formatação de citações, figuras e tabelas, equações, referências cruzadas, lista de referências e índices. Este documento não pretende exemplificar conteúdos a usar.

Neste primeiro capítulo ilustra-se a utilização de citações e de referências bibliográficas.

1.1 Secção Exemplo

Para além de dar um exemplo de utilização de uma citação, o parágrafo seguinte introduz uma referência que pode ser consultada, entre muitas

“Like the Abstract, the Introduction should be written to engage the interest of the reader. It should also give the reader an idea of how the dissertation is structured, and in doing so, define the

elit. Sed eget nunc. Phasellus interdum, risus viverra mollis laoreet, felis justo iaculis ante, eget ornare purus augue non urna. Nam in magna. In a est. Phasellus a tellus vitae enim vehicula imperdiet. Etiam sit amet elit. In hac habitasse platea dictumst. Quisque eget turpis vel felis elementum tempus. Curabitur sit amet tortor id libero dapibus pretium. Integer mattis eros eu lorem. Duis erat tellus, porttitor sed, blandit eget, fringilla et, lacus. Phasellus tristique nibh nec orci. Mauris sed leo. Suspendisse fringilla tempor dolor. Donec sapien enim, congue in, porta et, sollicitudin in, quam. Curabitur semper, mauris ut vestibulum eleifend, diam ipsum tincidunt quam, et vestibulum velit mauris ut risus.

Sed eget libero. Nulla facilisi. Proin eget tortor. Morbi gravida. Donec arcu risus, blandit a, rutrum at, ornare ut, nisl. Etiam consectetur tortor eu odio. Etiam blandit molestie ligula. Nulla facilisi. Nam a augue non justo laoreet hendrerit. Nam aliquam, purus eu ultricies dictum, urna purus posuere neque, vel tempus tellus enim a arcu.

Pellentesque congue sapien in ligula. Nulla nec mi sed augue congue tristique. Cras pretium. Pellentesque lobortis, libero id adipiscing auctor, orci massa vehicula nulla, vel ullamcorper tortor ipsum ut elit. Morbi rhoncus, dui sed tristique volutpat, lorem felis euismod lorem, vel tristique nibh arcu vel eros. Nunc tempor. Sed et erat a tortor fermentum consequat. Cras varius nisl accumsan libero. Aliquam faucibus, justo ut pharetra blandit, est nibh ultrices est, vel facilisis quam lectus ut enim. Aliquam convallis, nibh non bibendum pharetra, nisl velit lobortis orci, ac ultrices neque sem viverra massa. Donec malesuada. Aliquam ligula. Fusce in nisl. Etiam lacinia est quis velit. Maecenas massa. Maecenas sed pede. Nulla sodales. Etiam vitae erat. Duis tristique sem sit amet libero. Ut a libero nec ligula tempus mattis.

1.2 Outra Secção Exemplo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi sit amet nibh. Fusce faucibus, enim vel ultrices ornare, est mauris ultricies velit, vitae consequat sem erat vel nunc. Nam libero eros, mattis eget, sagittis nec, imperdiet at, sapien. Aliquam lacus. Aenean adipiscing nibh in orci. Aliquam vestibulum, elit at fringilla dignissim, metus diam lobortis urna, a laoreet nunc odio ac ipsum. Sed at urna. Integer vehicula fringilla augue. Nulla lacus eros, rhoncus sit amet, posuere ut, vehicula ac, nibh. Ut eleifend, eros eu placerat vehicula, justo turpis blandit dolor, eu tincidunt felis risus at ante. Aenean suscipit nisl eget eros. Ut laoreet libero eget enim. Cras tempus pellentesque felis. Vestibulum vitae erat ac nibh posuere eleifend.

Integer nec quam. Sed fermentum. Nunc vitae leo. Etiam sit amet quam. Nunc vestibulum massa in mauris. Duis eget nulla. Fusce ultricies arcu eu nibh volutpat feugiat. Maecenas urna pede, commodo quis, porta eu, bibendum elementum, pede. Sed eros massa, molestie eget, mattis non, rutrum ac, magna. Duis dui. Maecenas eget tortor ut dolor semper mattis. Maecenas auctor, tellus et ultricies tempor, elit est placerat lacus, in posuere mauris lorem et arcu.

Nulla nec eros et pede vehicula aliquam. Aenean sodales pede vel ante. Fusce sollicitudin sodales lacus. Maecenas justo mauris, adipiscing vitae, ornare quis, convallis nec, eros. Etiam laoreet venenatis ipsum. In tellus odio, eleifend ac, ultrices vel, lobortis sed, nibh. Fusce nunc augue, dictum non, pulvinar sed, consectetur eu, ipsum. Vivamus nec pede. Pellentesque pulvinar fringilla dolor. In sit amet pede. Proin orci justo, semper vel, vulputate quis, convallis ac, nulla. Nulla at justo. Mauris feugiat dolor. Etiam posuere fermentum eros. Morbi nisl ipsum, tempus id, ornare quis, mattis id, dolor. Aenean molestie metus suscipit dolor. Aliquam id lectus sed nisl lobortis rhoncus. Curabitur vitae diam sed sem aliquet tempus. Sed scelerisque nisi nec sem.

1.3 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais x capítulos. No Capítulo 2, é descrito o estado da arte e são apresentados trabalhos relacionados. No Capítulo 3, ipsum dolor sit amet, consectetur adipiscing elit. No Capítulo 4 praesent sit amet sem. No Capítulo 5 posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est.

Chapter 2

Repairing and Generating Dockerfiles (title is WIP)

This section covers existing literature and tools that are related to several components of the Dockerfile development experience and Docker in general. Section 2.1 goes over the background that is required to understand this paper. Section 2.2 goes over ways to accelerate Docker builds. Section 2.3 goes over works that cover the generation of Dockerfiles. Section 2.4 covers smells in Dockerfiles. Section 2.5 talks about good practices that should be taken into account when writing Dockerfiles. Section 2.6 tackles the topic of security in Dockerfiles. Section 2.7 covers the detection and repair of faults in Dockerfiles. Section 2.8 covers the detection and removal of bloat in Dockerfiles. Section 2.9 is about testing Dockerfiles. Section 2.10 concludes with a general discussion.

2.1 Background

...

2.2 Speeding up Docker builds

Building Docker images can take a considerable amount of time, especially when a large amount of files have to be fetched from the internet [8]. Therefore, we looked for ways to reduce the amount of time consumed by this activity. Our findings are summarized in Table 2.1.

Name	Speedup	Limitations	Implementation Complexity
A Code Injection Method for Rapid Docker Image Building [20]	Up to 100000x faster	Can only be used with interpreted languages, limited to modifications in the source code	High
FastBuild: Accelerating Docker Image Building for Efficient Development and Deployment of Container [11]	Up to 10x faster	Limited to network activity	High
Slacker: Fast Distribution with Lazy Docker Containers [8]	Up to 20x faster	Limited to network activity	High
Docker Buildx [4]	Unknown	Unknown	Low-Medium

Table 2.1: Works about speeding up Docker builds

Wang et al. [20] propose a technique that bypasses typical building procedures by injecting the code modifications directly in an image. The results are very promising. However, due to the nature of approach, it can only be used with interpreted languages and will not accelerate builds related to modifications in development artifacts that are not source code.

Huang et al. [11] and Harter et al. [8] address the network bottleneck in different ways. The former caches files locally and intercepts Docker’s network requests in order to serve files that have been stored locally. The latter proposes a new storage driver that lazily fetches files from the network. Both show promising results and do not address other inefficiencies in the Docker building process.

The works described so far interfere with the normal Docker build procedures, making them harder to implement. Another solution is offered by the Docker development team, Buildx [4]. Buildx makes use of a newer backend, BuildKit [3], which brings many features that can potentially accelerate docker builds. However, to our knowledge, an apples to apples time comparison has not been made. Implementing this in Dockerlive wouldn’t be very hard but because the output of the buildx command is different, some modifications would still be required.

2.3 Dockerfile Generation

Writing Dockerfiles is not an easy process [18]. Therefore, having a way to generate a Dockerfile for a given project can be very useful. In this section, we looked for works that showcased ways to accomplish this. Our findings are summarized in Table 2.2.

Name	Successful generation rate	Limitations	Implementation Complexity
Applying Model-Driven Engineering to Stimulate the Adoption of DevOps Processes in Small and Medium-Sized Development Organizations [19]	Unknown	An Open API Spec is required	Medium
Burner: Recipe Automatic Generation for HPC Container Based on Domain Knowledge Graph [24]	Up to 80%	A vast knowledge graph is required, focused on Singularity	High
Container-Based Module Isolation for Cloud Services [12]	Unknown	Requires the use of templates to generate the files	Medium-High
DockerGen: A Knowledge Graph based Approach for Software Containerization [23]	Up to 73%	A vast knowledge graph is required	High
DockerizeMe: Automatic Inference of Environment Dependencies for Python Code Snippets [10]	Up to 30%	Limited to Python, requires a knowledge base	High
ExploitWP2Docker: a Platform for Automating the Generation of Vulnerable WordPress Environments for Cyber Ranges [7]	Up to 39%	Limited to security testing scenarios	Low-Medium
MAKING CONTAINERS EASIER WITH HPC CONTAINER MAKER [15]	Unknown	Requires Python code to generate the files	Medium-High

Table 2.2: Works about generating Dockerfiles

Zhong et al. [24], Ye et al. [23] and Horton et al. [10] present solutions that require pre-existing

knowledge bases in order to generate the files, making them hard to implement in a project like ours, which is not completely focused on file generation. DockerizeMe is also limited to Python environments, while Burner is more focused on Singularity [2], a containerization tool similar to Docker but focused on HPC (High Performance Computing).

Caturano et al. [7] propose a tool that uses Docker to generate security testing environments from exploit descriptions. Sorgalla et al. [19]’s work can generate Dockerfiles from models, which are generated from Open API Specifications (like Swagger [5]). Kehrer et al. [12] use Apache FreeMarker [1] to generate Dockerfiles from templates. McMillan et al. [15] offer a tool that allows developer to use Python code to define the information required to generate Dockerfiles.

All these works show varying degrees of success and some of them are not even focused on the Dockerfile generation aspect.

2.4 Dockerfile Smells

Smells are commonly found in Dockerfiles [21], making it important to create ways of detecting and, if possible, remove them. This section covers works related to this. Our findings are summarized in Table 2.3.

Name	Smells	Findings	Possible repair implementation complexity
An Empirical Case Study on the Temporary File Smell in Dockerfiles [14]	Temporary File	The smell is quite common and can be divided into 4 different types, 3 of which can be detected through the proposed methods	Medium
An empirical study on self-admitted technical debt in Dockerfiles [6]	SATD	Shows that a type of smell appears in Dockerfiles and can be divided into several classes and subclasses	High
Characterizing the Occurrence of Dockerfile Smells in Open-Source Software: An Empirical Study [21]	All	Smells appear commonly in projects and the frequency with which they appear varies according to several metrics	High
Dockerfile TF Smell Detection Based on Dynamic and Static Analysis Methods [22]	Temporary File	It's possible to detect the smell with high accuracy using static and dynamic analysis	Medium

Table 2.3: Works about Dockerfile smells

Lu et al. [14] and Xu et al. [22] have focused on the temporary file smell and propose ways to detect this smell. A repair to deal with this smell could be implemented using the information provided by these works.

Azuma et al. [6] focus on a variation of smells they call SATD (self-admitted technical debt) which can be detected in comments written in the Dockerfiles. Due to the nature of these SATDs, implementing repairs to eliminate them could be very complicated.

Wu et al. [21] analyzed a large amount of open-source projects and found that Dockerfile smells are very common and their frequency changes according to several factors like the programming language used by the project or the project's age. Due to the variety of smells covered

by this study implementing repairs to deal with all of them would be difficult.

2.5 Dockerfile Good Practices

To prevent the creation of smells like the ones mentioned in Section 2.4, a developer should follow good practices. This section goes over works that cover these practices. Our findings are summarized in Table 2.4.

Name	Practices	Limit
Learning from, Understanding, and Supporting DevOps Artifacts for Docker [9]	Related to usage of package managers and command line utilities (gold rules)	No i func
Security Misconfigurations Detection and Repair in Dockerfile [17]	???	???
Ten simple rules for writing Dockerfiles for reproducible data science [16]	???	???

Table 2.4: Works about Dockerfile good practices

2.6 Dockerfile Security

2.7 Dockerfile Repair

2.8 Dockerfile Bloat

2.9 Dockerfile Testing

2.10 General Discussion

Ksontini et al. [13] focused on refactorings (a concept which is closely tied to smells) and found that developers' main motivations for performing refactorings were tied to maintainability and image size among others. Implementing some of these refactorings as repairs would be useful, although implementing all of them would be challenging.

Chapter 3

Capítulo Exemplo

Neste capítulo apresentam-se exemplos de formatação de figuras e tabelas, equações e referências cruzadas.

Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

3.1 Introdução

Apresenta-se de seguida um exemplo de equação¹, completamente fora do contexto²:

$$CIF_1: \quad F_0^j(a) = \frac{1}{2\pi i} \oint_{\gamma} \frac{F_0^j(z)}{z-a} dz \quad (3.1)$$

$$CIF_2: \quad F_1^j(a) = \frac{1}{2\pi i} \oint_{\gamma} \frac{F_0^j(x)}{x-a} dx \quad (3.2)$$

Na Equação 3.2 lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse tincidunt viverra elit. Donec tempus vulputate mauris. Donec arcu. Vestibulum condimentum porta justo. Curabitur ornare tincidunt lacus. Curabitur ac massa vel ante tincidunt placerat. Cras vehicula semper elit. Curabitur gravida, est a elementum suscipit, est eros ullamcorper quam, sed cursus velit velit tempor neque. Duis tempor condimentum ante. Nam sollicitudin. Vestibulum adipiscing, orci eu tempor dapibus, risus sapien porta metus, et cursus leo metus eget nibh.

Pellentesque rutrum, sapien at viverra facilisis, metus eros blandit sem, quis dictum erat metus eget erat. Vivamus malesuada dapibus nulla. Maecenas nec purus. Suspendisse auctor mattis augue. Phasellus enim nisi, iaculis sit amet, pellentesque a, iaculis in, dui. Integer risus.

¹<https://en.wikipedia.org/wiki/Equation>

²https://www.overleaf.com/learn/latex/Mathematical_expressions

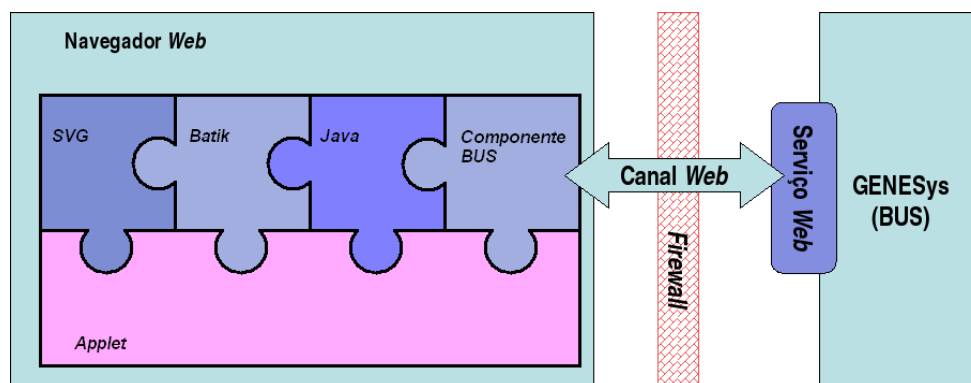


Figure 3.1: Arquitetura da Solução Proposta

3.2 Secção Exemplo

A arquitectura do visualizador assenta sobre os seguintes conceitos

- **Componentes** — Suspendisse auctor mattis augue *push*;
- **Praesent** — Sit amet sem maecenas eleifend facilisis leo;
- **Pellentesque** — Habitant morbi tristique senectus et netus.

3.2.1 Exemplo de Figura

É apresentado na Figura 3.1 da página 10 um exemplo de figura flutuante que deverá ficar no topo da página.

Loren ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed, luctus in.

3.2.2 Exemplo de Tabela

É apresentado na Tabela 3.1 um exemplo de tabela flutuante que deverá ficar no topo da página.

Loren ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

Table 3.1: Tabela Exemplo

k	Iteração k de $f(x_n)$			comentários
	x_1^k	x_2^k	x_3^k	
0	-0.3	0.6	0.7	-
1	0.47102965	0.04883157	-0.53345964	$\delta < \varepsilon$
2	0.49988691	0.00228830	-0.52246185	$\delta < \varepsilon$
3	0.49999976	0.00005380	-0.523656	N
4	0.5	0.00000307	-0.52359743	
\vdots	\vdots	\ddots	\vdots	
7	0.5	0.0	-0.52359878	$\delta < 10^{-8}$

Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed, luctus in.

3.3 Secção Exemplo

Loren ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

```

1 # take the users input
2 words = input("Enter the text to translate to pig latin: ")
3 print(f"You entered: {words}")
4
5 # now, break apart the words into a list
6 words = words.split(' ')
7
8 # let's use the list to translate words greater than 3 characteres
9 for i in words:
10     if len(i) >= 3:
11         i = i + "%say" % (i[0])
12         i = i[1:]
13         print(i)
14     else:
15         pass

```

Listing 3.1: Python example

Duis eget diam. In est justo, tristique in, lacinia vel, feugiat eget, quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed, luctus in.

3.4 Resumo

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Fusce feugiat, elit ac placerat fermentum, augue nisl ultricies eros, id fringilla enim sapien eu felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed dolor mi, porttitor quis, condimentum sed, luctus in.

Chapter 4

Mais um Capítulo

Neste capítulo mostra-se apenas o formato da dissertação.

Ipsum dolor sit amet, consectetur adipiscing elit. Praesent sit amet sem. Maecenas eleifend facilisis leo. Vestibulum et mi. Aliquam posuere, ante non tristique consectetur, dui elit scelerisque augue, eu vehicula nibh nisi ac est. Suspendisse elementum sodales felis. Nullam laoreet fermentum urna.

4.1 Secção Exemplo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer hendrerit commodo ante. Pellentesque nibh libero, aliquam at, faucibus id, commodo a, velit. Duis eleifend sem eget leo. Morbi in est. Suspendisse magna sem, varius nec, hendrerit non, tincidunt quis, quam. Aenean congue. Vivamus vel est sit amet sem iaculis posuere. Cras mollis, enim vel gravida aliquam, libero nunc ullamcorper dui, ullamcorper sodales lectus nulla sed urna. Morbi aliquet porta risus. Proin vestibulum ligula a purus. Maecenas a nulla. Maecenas mattis est vitae neque auctor tempus. Etiam nulla dui, mattis vitae, porttitor sed, aliquet ut, enim. Cras nisl magna, aliquet et, laoreet at, gravida ac, neque. Sed id est. Nulla dapibus dolor quis ipsum rhoncus cursus.

Etiam nisi est, dignissim sodales, fermentum id, pulvinar ac, eros. Duis id orci. Nam pretium nisl ac augue. Ut adipiscing magna eget est. Curabitur varius. Nulla facilisi. Pellentesque sit amet neque ac dui accumsan blandit. Donec mauris felis, egestas sit amet, convallis ac, dignissim quis, dolor. Maecenas cursus tortor vel leo. Quisque tristique. Nunc augue odio, tincidunt in, dapibus sed, ultricies sit amet, lorem. In hac habitasse platea dictumst. Praesent iaculis, lacus hendrerit tempor sodales, libero tellus aliquet orci, ut rhoncus massa lectus quis erat. Pellentesque quis dolor nec tortor rhoncus convallis. Aliquam erat volutpat. Fusce placerat, magna eu imperdiet lobortis, augue massa blandit turpis, a consectetur quam arcu sit amet risus. Suspendisse potenti. Praesent sapien metus, interdum vitae, fermentum id, faucibus ut, lorem. Nunc iaculis purus id tortor. Aenean risus pede, laoreet ac, tristique sed, lobortis in, turpis (see Figure 4.1b).

Vestibulum et lorem in ligula viverra pharetra. Curabitur quis purus in urna facilisis bibendum. Pellentesque at arcu accumsan velit bibendum ornare. Praesent massa. Quisque dolor. In libero.



Figure 4.1: Two Figures side by side

Vestibulum ac diam id leo feugiat blandit. Donec porta, tellus ac pellentesque molestie, felis mauris viverra lacus, sed dignissim purus justo eu justo. Proin iaculis, nunc eu volutpat volutpat, libero purus rutrum enim, id euismod lacus lorem nec augue. Donec hendrerit lacinia ante. Integer mollis vulputate orci. In pellentesque, metus pharetra elementum pharetra, est purus bibendum turpis, eu pretium sapien libero convallis odio. Cras sodales bibendum risus. Sed mattis nulla non leo. Nulla nunc. Phasellus egestas sodales massa. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Etiam mi.

4.2 Mais uma Secção

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque purus sapien, interdum ut, vestibulum a, accumsan ullamcorper, erat. Mauris a magna ut leo porta imperdiet. Donec dui odio, porta in, pretium non, semper quis, orci. Quisque erat diam, pharetra vel, laoreet ac, hendrerit vel, enim. Donec tristique luctus risus. Fusce dolor est, eleifend id, elementum sit amet, varius vitae, neque. Morbi at augue. Ut sem ligula, auctor vitae, facilisis id, pharetra non, lectus. Nulla lacus augue, aliquam eget, sollicitudin sed, hendrerit eu, leo. Suspendisse ac tortor. Mauris at odio. Etiam vehicula. Nam lacinia purus at nibh. Aliquam fringilla lorem ac justo. Ut nec enim. Nunc ornare, eros eu facilisis tristique, nisl lorem lacinia risus, non ullamcorper tellus urna et eros. Quisque eleifend tempus metus. Nunc ipsum.

Phasellus ullamcorper justo id risus. Nunc in leo. Mauris auctor lectus vitae est lacinia egestas. Nulla faucibus erat sit amet lectus varius semper. Praesent ultrices vehicula orci. Nam at metus. Aenean eget lorem nec purus feugiat molestie. Phasellus fringilla nulla ac risus. Aliquam elementum aliquam velit. Aenean nunc odio, lobortis id, dictum et, rutrum ac, ipsum. Aenean tellus magna, lacinia eget, bibendum ut, interdum sit amet, ipsum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Mauris felis lacus, dapibus sit amet, pretium feugiat, aliquet non, purus. Aliquam elementum, diam quis porttitor gravida, sem sapien iaculis nulla, ut pharetra odio felis a metus. Nulla lacus ipsum, tristique ut, dapibus sed, mollis et, justo. Vivamus non ipsum sed ligula placerat ultrices. Maecenas dictum leo adipiscing mauris. Vestibulum tristique, lacus a consequat suscipit, nunc dui sollicitudin arcu, non interdum libero est eget tortor. Ut eget neque quis leo tempor dictum.

Quisque ullamcorper. Aliquam vel magna. Sed pulvinar dictum ligula. Sed ultrices dolor ut turpis. Vivamus sagittis orci malesuada arcu venenatis auctor. Proin vehicula pharetra urna.

Aliquam egestas nunc quis nisl. Donec ullamcorper. Nulla purus. Ut suscipit lacus vitae dui. Mauris semper. Ut eget sem. Integer orci. Nam vitae dui eget nisi placerat convallis.

Sed id lorem. Proin gravida bibendum lacus. Sed molestie, urna quis euismod laoreet, diam dolor dictum diam, vitae consectetur leo ipsum id ante. Integer eu lectus non mauris pharetra viverra. In feugiat libero ut massa. Morbi cursus, lorem sollicitudin blandit semper, felis magna pellentesque lacus, ut rhoncus leo neque at tellus. Sed mattis, diam eget eleifend tincidunt, ligula eros tincidunt diam, vitae auctor turpis est vel nunc. In eu magna. Donec dolor metus, egestas sit amet, ultrices in, faucibus sed, lectus. Etiam est enim, vehicula pharetra, porta non, viverra vel, nunc. Ut non sem. Etiam nec neque. Sed rhoncus, justo id imperdiet pharetra, mi tellus accumsan neque, vitae volutpat tortor enim in odio. Nunc porta justo a lorem. Nulla hendrerit odio vitae dolor. Suspendisse eu nisl.

4.3 Resumo ou Conclusões

Proin vehicula pharetra urna. Aliquam egestas nunc quis nisl. Donec ullamcorper. Nulla purus. Ut suscipit lacus vitae dui. Mauris semper. Ut eget sem. Integer orci. Nam vitae dui eget nisi placerat convallis.

Chapter 5

Conclusões e Trabalho Futuro

Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus quam lacus, pharetra vel, aliquam vel, volutpat sed, nisl.

5.1 Satisfação dos Objetivos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam non felis sed odio rutrum ultrices. Donec tempor dolor. Vivamus justo neque, tempus id, ullamcorper in, pharetra non, tellus. Praesent eu orci eu dolor congue gravida. Sed eu est. Donec pulvinar, lectus et eleifend volutpat, diam sapien sollicitudin arcu, a sagittis libero neque et dolor. Nam ligula. Cras tincidunt lectus quis nunc. Cras tincidunt congue turpis. Nulla pede velit, sagittis a, faucibus vitae, porttitor nec, ante. Nulla ut arcu. Cras eu augue at ipsum feugiat hendrerit. Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vivamus quam lacus, pharetra vel, aliquam vel, volutpat sed, nisl.

Nullam erat est, vehicula id, tempor non, scelerisque at, tellus. Pellentesque tincidunt, ante vehicula bibendum adipiscing, lorem augue tempor felis, in dictum massa justo sed metus. Suspendisse placerat, mi eget molestie sodales, tortor ante interdum dui, ac sagittis est pede et lacus. Duis sapien. Nam ornare turpis et magna. Etiam adipiscing adipiscing ipsum. Fusce sodales nisl a arcu. Cras massa leo, vehicula facilisis, commodo a, molestie faucibus, metus. Suspendisse potenti. Duis sagittis. Donec porta. Sed urna. Maecenas eros. Vivamus erat ligula, pharetra sit amet, bibendum et, fermentum sed, dolor. Nullam eleifend condimentum nibh. Integer leo nibh, consequat eget, mollis et, sagittis ac, felis. Duis viverra pede in pede. Phasellus molestie placerat leo. Praesent at tellus a augue congue molestie. Proin sed justo eu sapien eleifend elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

5.2 Trabalho Futuro

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam tempor tristique risus. Suspendisse potenti. Fusce id eros. In eu enim. Praesent commodo leo. Nullam augue. Pellentesque tellus. Integer pulvinar purus a dui convallis consectetur. In adipiscing, orci vitae lacinia semper, sapien elit posuere sem, ac euismod ipsum elit tempus urna. Aliquam erat volutpat. Nullam suscipit augue sed felis. Phasellus faucibus accumsan est.

Aliquam felis justo, facilisis sit amet, bibendum ut, tempus ac, dolor. Sed malesuada. Nunc non massa. In erat. Nulla facilisi. Phasellus blandit, est in accumsan cursus, libero augue elementum leo, vitae auctor mauris nisl ac tortor. Cras porttitor ornare elit. Fusce at lorem. Sed lectus tortor, vestibulum id, varius a, condimentum nec, lectus. Maecenas in nisi et magna pretium aliquam. Pellentesque justo elit, feugiat nec, tincidunt a, dignissim vel, ipsum. Sed nunc. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam tempus rhoncus leo. Donec neque quam, cursus sit amet, ultricies varius, semper non, pede. Donec porttitor. Sed aliquet feugiat elit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus tellus pede, auctor ut, tincidunt a, consectetur in, felis. Mauris quis dolor et neque accumsan pellentesque. Donec dui magna, scelerisque mattis, sagittis nec, porta quis, nulla. Vivamus quis nisl. Etiam vitae nisl in diam vehicula viverra. Sed sollicitudin scelerisque est. Nunc dapibus. Sed urna. Nulla gravida. Praesent faucibus, risus ac lobortis dignissim, est tortor laoreet mauris, dictum pellentesque nunc orci tincidunt tellus. Nullam pulvinar, leo sed vestibulum euismod, ante ligula elementum pede, sit amet dapibus lacus tortor ac nisl. Morbi libero. Integer sed dolor ac lectus commodo iaculis. Donec ut odio.

References

- [1] Apache FreeMarker.
- [2] Apptainer.
- [3] BuildKit.
- [4] Docker buildx.
- [5] Swagger.
- [6] Hideaki Azuma, Shinsuke Matsumoto, Yasutaka Kamei, and Shinji Kusumoto. An empirical study on self-admitted technical debt in Dockerfiles. *27(2)*:49.
- [7] Francesco Caturano, Nicola d’ Ambrosio, Gaetano Perrone, Luigi Previdente, and Simon Pietro Romano. ExploitWP2Docker: A Platform for Automating the Generation of Vulnerable WordPress Environments for Cyber Ranges. In *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–7.
- [8] Tyler Harter, Brandon Salmon, Rose Liu, Andrea C Arpaci-Dusseau, and Remzi H Arpaci-Dusseau. Slacker: Fast Distribution with Lazy Docker Containers.
- [9] Jordan Henkel, Christian Bird, Shuvendu K. Lahiri, and Thomas Reps. Learning from, Understanding, and Supporting DevOps Artifacts for Docker. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 38–49.
- [10] Eric Horton and Chris Parnin. DockerizeMe: Automatic Inference of Environment Dependencies for Python Code Snippets. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 328–338.
- [11] Zhuo Huang, Song Wu, Song Jiang, and Hai Jin. FastBuild: Accelerating Docker Image Building for Efficient Development and Deployment of Container. In *2019 35th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 28–37.
- [12] Stefan Kehrer, Florian Riebandt, and Wolfgang Blochinger. Container-Based Module Isolation for Cloud Services. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 177–17709.
- [13] Emna Ksontini and Marouane Kessentini. Refactorings and Technical Debt for Docker Projects.
- [14] Zhigang Lu, Jiwei Xu, Yuewen Wu, Tao Wang, and Tao Huang. An Empirical Case Study on the Temporary File Smell in Dockerfiles. *7*:63650–63659.

- [15] Scott McMillan. MAKING CONTAINERS EASIER WITH HPC CONTAINER MAKER. page 47.
- [16] Daniel Nüst, Vanessa Sochat, Ben Marwick, Stephen J. Eglen, Tim Head, Tony Hirst, and Benjamin D. Evans. Ten simple rules for writing Dockerfiles for reproducible data science. 16(11):e1008316.
- [17] Paolo Ernesto Prinetto, Dott Riccardo Bortolameotti, and Giuseppe Massaro. Security Misconfigurations Detection and Repair in Dockerfile. page 78.
- [18] David Reis, Bruno Piedade, Filipe F. Correia, João Pedro Dias, and Ademar Aguiar. Developing Docker and Docker-Compose Specifications: A Developers' Survey. 10:2318–2329.
- [19] Jonas Sorgalla, Philip Wizenty, Florian Rademacher, Sabine Sachweh, and Albert Zündorf. Applying Model-Driven Engineering to Stimulate the Adoption of DevOps Processes in Small and Medium-Sized Development Organizations: The Case for Microservice Architecture. 2(6):459.
- [20] Yujing Wang and Qinyang Bao. A Code Injection Method for Rapid Docker Image Building.
- [21] Yiwen Wu, Yang Zhang, Tao Wang, and Huaimin Wang. Characterizing the Occurrence of Dockerfile Smells in Open-Source Software: An Empirical Study. 8:34127–34139.
- [22] Jiwei Xu, Yuewen Wu, Zhigang Lu, and Tao Wang. Dockerfile TF Smell Detection Based on Dynamic and Static Analysis Methods. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 185–190.
- [23] Hongjie Ye, Jiahong Zhou, Wei Chen, Jiaxin Zhu, Guoquan Wu, and Jun Wei. DockerGen: A Knowledge Graph based Approach for Software Containerization. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 986–991.
- [24] Shuaihao Zhong, Duoqiang Wang, Wei Li, Feng Lu, and Hai Jin. Burner: Recipe Automatic Generation for HPC Container Based on Domain Knowledge Graph. 2022:e4592428.

Appendix A

Lorem Ipsum

Depois das conclusões e antes das referências bibliográficas, apresenta-se neste anexo numerado o texto usado para preencher a dissertação.

A.1 O que é o *Lorem Ipsum*?

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like

A.2 De onde Vem o Lorem?

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, *consectetur*, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of “*de Finibus Bonorum et Malorum*” (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, “*Lorem ipsum dolor sit amet...*”, comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from “*de Finibus Bonorum et Malorum*” by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.

A.3 Porque se usa o Lorem?

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using “Content here, content here”, making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for “lorem ipsum” will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

A.4 Onde se Podem Encontrar Exemplos?

There are many variations of passages of Lorem Ipsum available, but the majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc.