



Universidade Federal do Rio Grande do Norte
Departamento de Informática e Matemática Aplicada

Relatório da 1ª Unidade - Grafos

Alexandre Dantas, Andriel Vinicius, Gabriel Carvalho, Maria Paz e
Vinicius de Lima

Professor: Matheus Menezes

29 de setembro de 2025

Abstract

Esse relatório propões uma implementação para os algoritmos discutidos no conteúdo abordado na 1ª Unidade da matéria de Grafos, sob o Departamento de Informática Aplicada da UFRN.

Sumário

Lista de Figuras	iv
Lista de Tabelas	v
1 Introdução	1
1.1 Teoria dos grafos	1
1.2 Problem statement	2
1.3 Aims and objectives	2
1.4 Solution approach	2
1.4.1 A subsection 1	2
1.4.2 A subsection 2	2
1.5 Summary of contributions and achievements	2
1.6 Organization of the report	3
2 Literature Review	4
2.1 Example of in-text citation of references in \LaTeX	4
2.1.1 Reference Resources	4
2.1.2 Changing Bibliography Styles	5
2.2 Avoiding unintentional plagiarism	5
2.3 Critique of the review	5
2.4 Summary	5
3 Methodology	6
3.1 Examples of the sections of a methodology chapter	6
3.1.1 Example of a software/Web development main text structure	7
3.1.2 Example of an algorithm analysis main text structure	7
3.1.3 Example of an application type main text structure	7
3.1.4 Example of a science lab-type main text structure	8
3.1.5 Ethical considerations	8
3.2 Example of an Equation in \LaTeX	9
3.3 Example of a Figure in \LaTeX	9
3.4 Example of an algorithm in \LaTeX	11
3.5 Example of code snippet in \LaTeX	11
3.6 Example of in-text citation style	12
3.6.1 Example of the equations and illustrations placement and reference in the text	12
3.6.2 Example of the equations and illustrations style	12
3.7 Summary	13

4	Results	14
4.1	A section	14
4.2	Example of a Table in \LaTeX	15
4.3	Example of captions style	15
4.4	Summary	15
5	Discussion and Analysis	16
5.1	A section	16
5.2	Significance of the findings	16
5.3	Limitations	16
5.4	Summary	16
6	Conclusions and Future Work	17
6.1	Conclusions	17
6.2	Future work	17
7	Reflection	18
	Referências Bibliográficas	19
	Appendices	20
A	An Appendix Chapter (Optional)	20
B	An Appendix Chapter (Optional)	21

Lista de Figuras

1.1	Um grafo com $V := \{1, 2, 3, 4\}$ e $A := \{(1, 2), (1, 4), (2, 3)\}$	1
1.2	Um grafo com $V := \{u, v\}$ e $A := \{(u, v)\}$	2
3.1	Example figure in \LaTeX	10

Lista de Tabelas

3.1	Undergraduate report template structure	6
3.2	Example of a software engineering-type report structure	7
3.3	Example of an algorithm analysis type report structure	7
3.4	Example of an application type report structure	8
3.5	Example of a science lab experiment-type report structure	8
4.1	Example of a table in \LaTeX	15

Capítulo 1

Introdução

Este projeto tem como objetivo implementar uma API de algoritmos relacionados a Grafos na linguagem de programação Rust. A especificação da API segue as funcionalidades descritas na definição da Avaliação 01 da disciplina de Grafos, sob o Departamento de Informática Aplicada (DIMAp) da Universidade Federal do Rio Grande do Norte (UFRN).

Este relatório visa discutir a teoria relacionada a implementação, isto é, revisar a teoria de grafos e discorrer sobre a estrutura e construção da API. O relatório também pretende documentar os testes de performance nos algoritmos sob as representações de grafos descritas na definição do projeto, com o objetivo de comparar as estruturas e evidenciar empiricamente suas principais diferenças.

1.1 Teoria dos grafos

As definições utilizadas no projeto foram em grande parte retiradas de [Diestel \(2025\)](#), mas com algumas modificações.

Definição 1 (Grafo). Um *grafo* é uma estrutura $G := \langle V, A \rangle$ tal que $A \subseteq V^2$ e V é um conjunto de um tipo qualquer. Os elementos de V são denominados vértices (em inglês *nodes*) e os elementos de A são denominados de arestas (em inglês *edges*). O jeito tradicional de visualizar um grafo é como uma figura composta de bolas e setas:

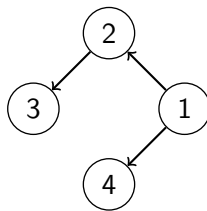


Figura 1.1: Um grafo com $V := \{1, 2, 3, 4\}$ e $A := \{(1, 2), (1, 4), (2, 3)\}$.

Definição 2 (Ordem e Tamanho). O número de vértices de um grafo G é chamado de *ordem* e é denotado por $|G|$ – o número de arestas é chamado de *tamanho* e é denotado por $\|G\|$. Por exemplo, na figura 1.1, $|G| = 4$ e $\|G\| = 3$.

Definição 3 (Adjacência). Dizemos que um vértice v é adjacente de um vértice u se somente se $(u, v) \in A$. Visualmente, enxergamos isso como:

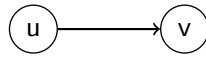


Figura 1.2: Um grafo com $V := \{u, v\}$ e $A := \{(u, v)\}$.

1.2 Problem statement

This section describes the investigated problem in detail. You can also have a separate chapter on “Problem articulation.” For some projects, you may have a section like “Research question(s)” or “Research Hypothesis” instead of a section on “Problem statement.”

1.3 Aims and objectives

Describe the “aims and objectives” of your project.

Aims: The aims tell a reader what you want/hope to achieve at the end of the project. The aims define your intent/purpose in general terms.

Objectives: The objectives are a set of tasks you would perform in order to achieve the defined aims. The objective statements have to be specific and measurable through the results and outcome of the project.

1.4 Solution approach

Briefly describe the solution approach and the methodology applied in solving the set aims and objectives.

Depending on the project, you may like to alter the “heading” of this section. Check with your supervisor. Also, check what subsection or any other section that can be added in or removed from this template.

1.4.1 A subsection 1

You may or may not need subsections here. Depending on your project’s needs, add two or more subsection(s). A section takes at least two subsections.

1.4.2 A subsection 2

Depending on your project’s needs, add more section(s) and subsection(s).

A subsection 1 of a subsection

The command `\subsubsection{}` creates a paragraph heading in \LaTeX .

A subsection 2 of a subsection

Write your text here...

1.5 Summary of contributions and achievements

Describe clearly what you have done/created/achieved and what the major results and their implications are.

1.6 Organization of the report

Describe the outline of the rest of the report here. Let the reader know what to expect ahead in the report. Describe how you have organized your report.

Example: how to refer a chapter, section, subsection. This report is organised into seven chapters. Chapter 2 details the literature review of this project. In Section 3...

Note: Take care of the word like “Chapter,” “Section,” “Figure” etc. before the \LaTeX command `\ref{}`. Otherwise, a sentence will be confusing. For example, In 2 literature review is described. In this sentence, the word “Chapter” is missing. Therefore, a reader would not know whether 2 is for a Chapter or a Section or a Figure. For more information on **automated tools** to assist in this work, see ??.

Capítulo 2

Literature Review

A literature review chapter can be organized in a few sections with appropriate titles. A literature review chapter might contain the following:

1. A review of the state-of-the-art (include theories and solutions) of the field of research.
2. A description of the project in the context of existing literature and products/systems.
3. An analysis of how the review is relevant to the intended application/system/problem.
4. A critique of existing work compared with the intended work.

Note that your literature review should demonstrate the significance of the project.

2.1 Example of in-text citation of references in L^AT_EX

The references in a report relate your content with the relevant sources, papers, and the works of others. To include references in a report, we *cite* them in the texts. In MS-Word, EndNote, or MS-Word references, or plain text as a list can be used. Similarly, in L^AT_EX, you can use the “thebibliography” environment, which is similar to the plain text as a list arrangement like the MS word. However, In L^AT_EX, the most convenient way is to use the BibTex, which takes the references in a particular format [see references.bib file of this template] and lists them in style [APA, Harvard, etc.] as we want with the help of proper packages.

These are the examples of how to *cite* external sources, seminal works, and research papers. In L^AT_EX, if you use “**BibTex**” you do not have to worry much since the proper use of a bibliographystyle package like “agsm for the Harvard style” and little rectification of the content in a BiBText source file [In this template, BibTex are stored in the “references.bib” file], we can conveniently generate a reference style.

Take a note of the commands `\cite{}` and `\citep{}`. The command `\cite{}` will write like “Author et al. (2019)” style for Harvard, APA and Chicago style. The command `\citep{}` will write like “(Author et al., 2019).” Depending on how you construct a sentence, you need to use them smartly. Check the examples of **in-text citation** of sources listed here [This Department recommends the **Harvard style** of referencing.]:

2.1.1 Reference Resources

You can find additional referencing resources from the University Library:

- <https://libguides.reading.ac.uk/computer-science>
- <https://libguides.reading.ac.uk/citing-references/citationexamples>

2.1.2 Changing Bibliography Styles

While this report used name and date formatting in the Harvard style, you might also wish to use a numbered style like that from IEEE. To enable this change, you will need to edit the `CS_report.sty` file. Uncomment the 2 lines of Harvard settings under Bibliography/References settings, and enable the IEEE style:

```
% IEEE, Numbered Style
\usepackage[numbers]{natbib}
\bibliographystyle{IEEEtran}
```

Note that when making this change, you will need to modify the way in which you refer to authors by name, as this is no longer immediately automatic. Instead, you will need to additionally rely on `\citeauthor{}`.

2.2 Avoiding unintentional plagiarism

Using other sources, ideas, and material always bring with it a risk of unintentional plagiarism. **MUST**: do read the university guidelines on the definition of plagiarism as well as the guidelines on how to avoid

2.3 Critique of the review

Describe your main findings and evaluation of the literature.

2.4 Summary

Write a summary of this chapter

Capítulo 3

Methodology

We mentioned in Chapter 1 that a project report's structure could follow a particular paradigm. Hence, the organization of a report (effectively the Table of Content of a report) can vary depending on the type of project you are doing. Check which of the given examples suit your project. Alternatively, follow your supervisor's advice.

3.1 Examples of the sections of a methodology chapter

A general report structure is summarised (suggested) in Table 3.1. Table 3.1 describes that, in general, a typical report structure has three main parts: (1) front matter, (2) main text, and (3) end matter. The structure of the front matter and end matter will remain the same for all the undergraduate final year project report. However, the main text varies as per the project's needs.

Tabela 3.1: Undergraduate report template structure

Frontmatter	Title Page
	Abstract
	Acknowledgements
	Table of Contents
	List of Figures
	List of Tables
	List of Abbreviations
Main text	Chapter 1 Introduction
	Chapter 2 Literature Review
	Chapter 3 Methodology
	Chapter 4 Results
	Chapter 5 Discussion and Analysis
	Chapter 6 Conclusions and Future Work
	Chapter 7 Refection
End matter	References
	Appendices (Optional)
	Index (Optional)

3.1.1 Example of a software/Web development main text structure

Notice that the “methodology” Chapter of Software/Web development in Table 3.2 takes a standard software engineering paradigm (approach). Alternatively, these suggested sections can be the chapters of their own. Also, notice that “Chapter 5” in Table 3.2 is “Testing and Validation” which is different from the general report template mentioned in Table 3.1. Check with your supervisor if in doubt.

Tabela 3.2: Example of a software engineering-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Requirements specifications
		Analysis
		Design
		Implementations
Chapter 4	Testing and Validation	
Chapter 5	Results and Discussion	
Chapter 6	Conclusions and Future Work	
Chapter 7	Reflection	

3.1.2 Example of an algorithm analysis main text structure

Some project might involve the implementation of a state-of-the-art algorithm and its performance analysis and comparison with other algorithms. In that case, the suggestion in Table 3.3 may suit you the best.

Tabela 3.3: Example of an algorithm analysis type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Algorithms descriptions
		Implementations
		Experiments design
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.1.3 Example of an application type main text structure

If you are applying some algorithms/tools/technologies on some problems/datasets/etc., you may use the methodology section prescribed in Table 3.4.

Tabela 3.4: Example of an application type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	Problems (tasks) descriptions Algorithms/tools/technologies/etc. descriptions Implementations Experiments design and setup
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.1.4 Example of a science lab-type main text structure

If you are doing a science lab experiment type of project, you may use the methodology section suggested in Table 3.5. In this kind of project, you may refer to the “Methodology” section as “Materials and Methods.”

Tabela 3.5: Example of a science lab experiment-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Materials and Methods	Problems (tasks) description Materials Procedures Implementations Experiment set-up
Chapter 4	Results	
Chapter 5	Discussion and Analysis	
Chapter 6	Conclusion and Future Work	
Chapter 7	Reflection	

3.1.5 Ethical considerations

This section addresses ethical aspects of your project. This may include: informed consent, describing how participants will be informed about the study’s purpose, procedures, risks, and benefits. You should detail the process used for obtaining consent and ensuring participants understand their rights.

- **Informed Consent:** If data was collected from participant, detail the process for obtaining consent and ensuring participants understand their rights.
- **Confidentiality and Privacy:** Explain measures taken to protect participants’ data and maintain confidentiality. Discuss how data is stored, who will have access, and how anonymity will be preserved.

- **Risk Assessment:** Identify potential risks to participants and outline strategies to minimize them.
- **Vulnerable Populations:** If applicable, address how you will protect vulnerable groups (e.g., children, elderly, or marginalized communities) involved in your project.
- **Research Integrity:** Highlight your commitment to honesty and transparency in research. Discuss how you will avoid plagiarism, fabrication, and falsification of data.
- **Compliance with Regulations:** Mention relevant ethical guidelines and regulations that your project will adhere to.
- **Impact on Society:** Reflect on the broader implications of your project. Discuss how the outcomes may affect communities, stakeholders, or the environment, and how you plan to address any potential negative consequences.
- **Feedback Mechanisms:** Describe how you incorporate feedback from participants and stakeholders to improve the ethical conduct of the project throughout its duration.

3.2 Example of an Equation in \LaTeX

Eq. 3.1 [note that this is an example of an equation's in-text citation] is an example of an equation in \LaTeX . In Eq. (3.1), s is the mean of elements $x_i \in \mathbf{x}$:

$$s = \frac{1}{N} \sum_{i=1}^N x_i. \quad (3.1)$$

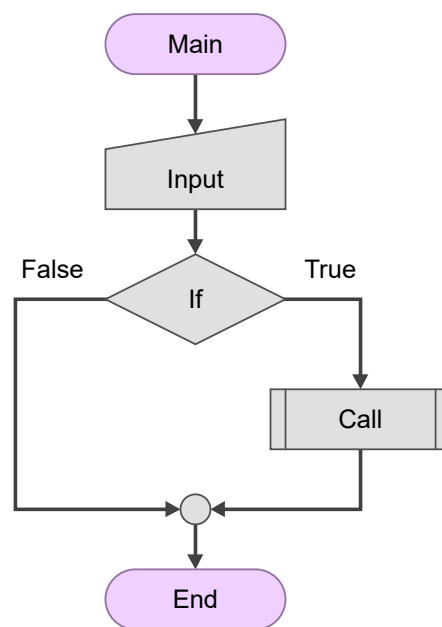
Have you noticed that all the variables of the equation are defined using the **in-text** maths command $\$$, and Eq. (3.1) is treated as a part of the sentence with proper punctuation? Always treat an equation or expression as a part of the sentence.

3.3 Example of a Figure in \LaTeX

Figure 3.1 is an example of a figure in \LaTeX . For more details, check the link:

wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions.

Keep your artwork (graphics, figures, illustrations) clean and readable. At least 300dpi is a good resolution of a PNG format artwork. However, an SVG format artwork saved as a PDF will produce the best quality graphics. There are numerous tools out there that can produce vector graphics and let you save that as an SVG file and/or as a PDF file. One example of such a tool is the “Flow algorithm software”. Here is the link for that: flowgorithm.org.

Figura 3.1: Example figure in \LaTeX .

3.4 Example of an algorithm in \LaTeX

Algorithm 1 is a good example of an algorithm in \LaTeX .

Algorithm 1 Example caption: sum of all even numbers

Input: $\mathbf{x} = x_1, x_2, \dots, x_N$

Output: *EvenSum* (Sum of even numbers in \mathbf{x})

```

1: function EvenSummation( $\mathbf{x}$ )
2:   EvenSum  $\leftarrow$  0
3:    $N \leftarrow \text{length}(\mathbf{x})$ 
4:   for  $i \leftarrow 1$  to  $N$  do
5:     if  $x_i \bmod 2 == 0$  then                                 $\triangleright$  Check whether a number is even.
6:       EvenSum  $\leftarrow$  EvenSum +  $x_i$ 
7:     end if
8:   end for
9:   return EvenSum
10: end function

```

3.5 Example of code snippet in \LaTeX

Code Listing 3.1 is a good example of including a code snippet in a report. While using code snippets, take care of the following:

- do not paste your entire code (implementation) or everything you have coded. Add code snippets only.
- The algorithm shown in Algorithm 1 is usually preferred over code snippets in a technical/scientific report.
- Make sure the entire code snippet or algorithm stays on a single page and does not overflow to another page(s).

Here are three examples of code snippets for three different languages (Python, Java, and CPP) illustrated in Listings 3.1, 3.2, and 3.3 respectively.

```

1 import numpy as np
2
3  $\mathbf{x}$  = [0, 1, 2, 3, 4, 5] # assign values to an array
4 evenSum = evenSummation( $\mathbf{x}$ ) # call a function
5
6 def evenSummation( $\mathbf{x}$ ):
7     evenSum = 0
8      $n = \text{len}(\mathbf{x})$ 
9     for  $i$  in range( $n$ ):
10         if np.mod( $\mathbf{x}[i]$ ,2) == 0: # check if a number is even?
11             evenSum = evenSum +  $\mathbf{x}[i]$ 
12     return evenSum

```

Listing 3.1: Code snippet in \LaTeX and this is a Python code example

Here we used the “\clearpage” command and forced-out the second listing example onto the next page.

```

1 public class EvenSum{
2     public static int evenSummation(int[] x){
3         int evenSum = 0;
4         int n = x.length;
5         for(int i = 0; i < n; i++){
6             if(x[i]%2 == 0){ // check if a number is even?
7                 evenSum = evenSum + x[i];
8             }
9         }
10        return evenSum;
11    }
12    public static void main(String[] args){
13        int[] x = {0, 1, 2, 3, 4, 5}; // assign values to an array
14        int evenSum = evenSummation(x);
15        System.out.println(evenSum);
16    }
17 }

```

Listing 3.2: Code snippet in \LaTeX and this is a Java code example

```

1 int evenSummation(int x[]){
2     int evenSum = 0;
3     int n = sizeof(x);
4     for(int i = 0; i < n; i++){
5         if(x[i]%2 == 0){ // check if a number is even?
6             evenSum = evenSum + x[i];
7         }
8     }
9     return evenSum;
10 }
11
12 int main(){
13     int x[] = {0, 1, 2, 3, 4, 5}; // assign values to an array
14     int evenSum = evenSummation(x);
15     cout<<evenSum;
16     return 0;
17 }

```

Listing 3.3: Code snippet in \LaTeX and this is a C/C++ code example

3.6 Example of in-text citation style

3.6.1 Example of the equations and illustrations placement and reference in the text

Make sure whenever you refer to the equations, tables, figures, algorithms, and listings for the first time, they also appear (placed) somewhere on the same page or in the following page(s). Always make sure to refer to the equations, tables and figures used in the report. Do not leave them without an **in-text citation**. You can refer to equations, tables and figures more than once.

3.6.2 Example of the equations and illustrations style

Write **Eq.** with an uppercase “Eq” for an equation before using an equation number with (`\eqref{.}`). Use “Table” to refer to a table, “Figure” to refer to a figure, “Algorithm” to refer to an algorithm and “Listing” to refer to listings (code snippets). Note that, we do not use

the articles “a,” “an,” and “the” before the words Eq., Figure, Table, and Listing, but you may use an article for referring the words figure, table, etc. in general.

For example, the sentence “A report structure is shown in **the** Table 3.1” should be written as “A report structure is shown **in** Table 3.1.”

3.7 Summary

Write a summary of this chapter.

Note: In the case of **software engineering** project a Chapter “**Testing and Validation**” should precede the “Results” chapter. See Section 3.1.1 for report organization of such project.

Capítulo 4

Results

The results chapter tells a reader about your findings based on the methodology you have used to solve the investigated problem. For example:

- If your project aims to develop a software/web application, the results may be the developed software/system/performance of the system, etc., obtained using a relevant methodological approach in software engineering.
- If your project aims to implement an algorithm for its analysis, the results may be the performance of the algorithm obtained using a relevant experiment design.
- If your project aims to solve some problems/research questions over a collected dataset, the results may be the findings obtained using the applied tools/algorithms/etc.

Arrange your results and findings in a logical sequence.

4.1 A section

...

4.2 Example of a Table in \LaTeX

Table 4.1 is an example of a table created using the package \LaTeX “booktabs.” do check the link: wikibooks.org/wiki/LaTeX/Tables for more details. A table should be clean and readable. Unnecessary horizontal lines and vertical lines in tables make them unreadable and messy. The example in Table 4.1 uses a minimum number of liens (only necessary ones). Make sure that the top rule and bottom rule (top and bottom horizontal lines) of a table are present.

Tabela 4.1: Example of a table in \LaTeX

Bike		
Type	Color	Price (£)
Electric	black	700
Hybrid	blue	500
Road	blue	300
Mountain	red	300
Folding	black	500

4.3 Example of captions style

- The **caption of a Figure (artwork)** goes **below** the artwork (Figure/Graphics/illustration). See example artwork in Figure 3.1.
- The **caption of a Table** goes **above** the table. See the example in Table 4.1.
- The **caption of an Algorithm** goes **above** the algorithm. See the example in Algorithm 1.
- The **caption of a Listing** goes **below** the Listing (Code snippet). See example listing in Listing 3.1.

4.4 Summary

Write a summary of this chapter.

Capítulo 5

Discussion and Analysis

Depending on the type of project you are doing, this chapter can be merged with “Results” Chapter as “ Results and Discussion” as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

5.1 A section

The Discussion and Analysis chapter evaluates and analyses the results. It interprets the obtained results.

5.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader’s understanding of the investigated problem

5.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

5.4 Summary

Write a summary of this chapter.

Capítulo 6

Conclusions and Future Work

6.1 Conclusions

Typically a conclusions chapter first summarizes the investigated problem and its aims and objectives. It summarizes the critical/significant/major findings/results about the aims and objectives that have been obtained by applying the key methods/implementations/experiment set-ups. A conclusions chapter draws a picture/outline of your project's central and the most significant contributions and achievements.

A good conclusions summary could be approximately 300–500 words long, but this is just a recommendation.

A conclusions chapter followed by an abstract is the last things you write in your project report.

6.2 Future work

This section should refer to Chapter 4 where the author has reflected their criticality about their own solution. Concepts for future work are then sensibly proposed in this section.

Guidance on writing future work: While working on a project, you gain experience and learn the potential of your project and its future works. Discuss the future work of the project in technical terms. This has to be based on what has not been yet achieved in comparison to what you had initially planned and what you have learned from the project. Describe to a reader what future work(s) can be started from the things you have completed. This includes identifying what has not been achieved and what could be achieved.

A good future work summary could be approximately 300–500 words long, but this is just a recommendation.

Capítulo 7

Reflection

Write a short paragraph on the substantial learning experience. This can include your decision-making approach in problem-solving.

Some hints: You obviously learned how to use different programming languages, write reports in \LaTeX and use other technical tools. In this section, we are more interested in what you thought about the experience. Take some time to think and reflect on your individual project as an experience, rather than just a list of technical skills and knowledge. You may describe things you have learned from the research approach and strategy, the process of identifying and solving a problem, the process research inquiry, and the understanding of the impact of the project on your learning experience and future work.

Also think in terms of:

- what knowledge and skills you have developed
- what challenges you faced, but was not able to overcome
- what you could do this project differently if the same or similar problem would come
- rationalize the divisions from your initial planed aims and objectives.

A good reflective summary could be approximately 300–500 words long, but this is just a recommendation.

Note: The next chapter is “**References**,” which will be automatically generated if you are using BibTeX referencing method. This template uses BibTeX referencing. Also, note that there is difference between “References” and “Bibliography.” The list of “References” strictly only contain the list of articles, paper, and content you have cited (i.e., refereed) in the report. Whereas Bibliography is a list that contains the list of articles, paper, and content you have cited in the report plus the list of articles, paper, and content you have read in order to gain knowledge from. We recommend to use only the list of “References.”

Referências Bibliográficas

Diestel, R. (2025), *Graph theory*, Vol. 173, Springer Nature.

Apêndice A

An Appendix Chapter (Optional)

Some lengthy tables, codes, raw data, length proofs, etc. which are **very important but not essential part** of the project report goes into an Appendix. An appendix is something a reader would consult if he/she needs extra information and a more comprehensive understating of the report. Also, note that you should use one appendix for one idea.

An appendix is optional. If you feel you do not need to include an appendix in your report, avoid including it. Sometime including irrelevant and unnecessary materials in the Appendices may unreasonably increase the total number of pages in your report and distract the reader.

Apêndice B

An Appendix Chapter (Optional)

...