TecnoCampus
Escola Superior
Politècnica

Centre adscrit a la
upf. Universitat
Pompeu Fabra
Barcelona

Grau en Enginyeria Informàtica de Gestió i Sistemes d'Informació

IMPLEMENTATION OF A VULNERABILITY ANALYSIS SERVICE OF PUBLIC HOSTS WITH SAAS MODALITY

Viability Study

ALEJANDRO COSTA RUEDA
TUTOR: LÉONARD JANER-GARCIA

2021-2022

TecnoCampus
Mataró-Maresme

Table of Contents

List of Figures

List of Tables

1. Introduction

This document consists of four parts, the introduction, the planning and the budget, the viability analysis, and the conclusions. In the planning and budget part, we compare the initial conditions of the project with the final result. In the viability analysis, we take into consideration the technical, economic and environmental viability while weighing different legal aspects. And finally, we end with an assessment of the project in the conclusions.

2. Planning and Budget

This chapter describes what has been the initial planning, what has been the actual planning, the budget, and the modifications that it has suffered.

2.1 Initial Planning

As it can be seen in the figure 2.1, the project consists of three main deliveries and a presentation. The hours of each phase are in the table 2.2

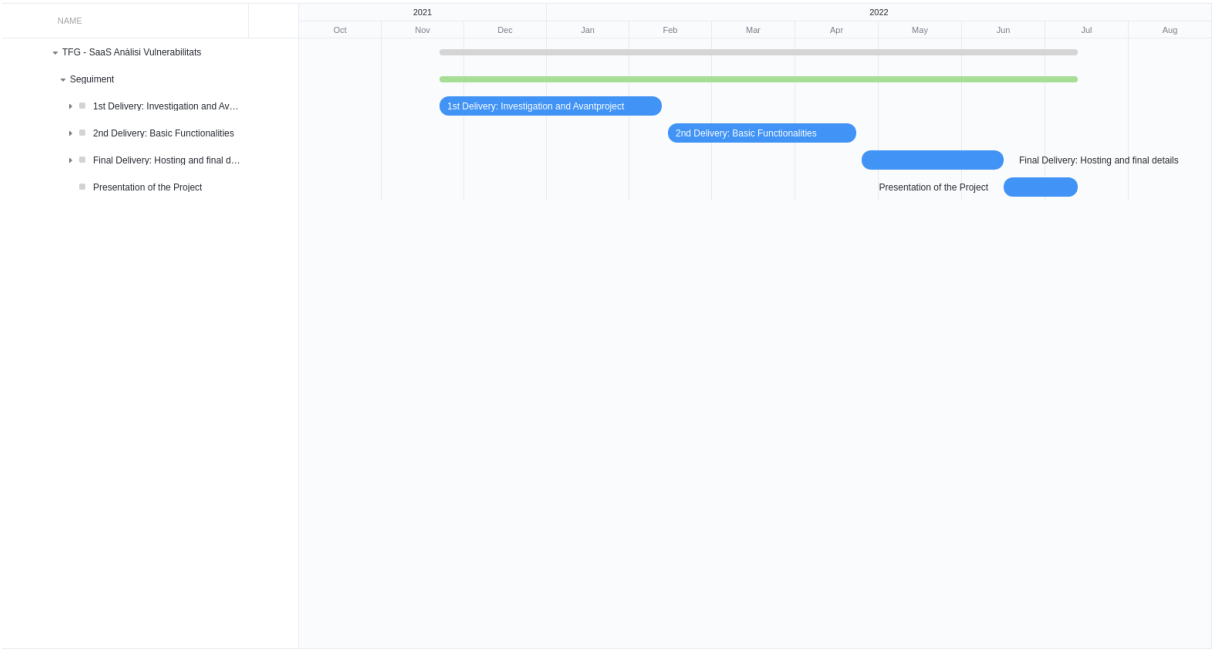

Figure 2.1: General planning

Hours	
Phase	Hours
1st Delivery: Investigation and Avantproject	75h
2nd Delivery: Basic Functionalities	220h
Final Delivery: Hosting and final details	180h
Presentation of the project	25h
Total Hours: **500 hours**	

Table 2.1: Table of hours

The detailed planning shows all the subtasks of each big group (Figure 2.2).

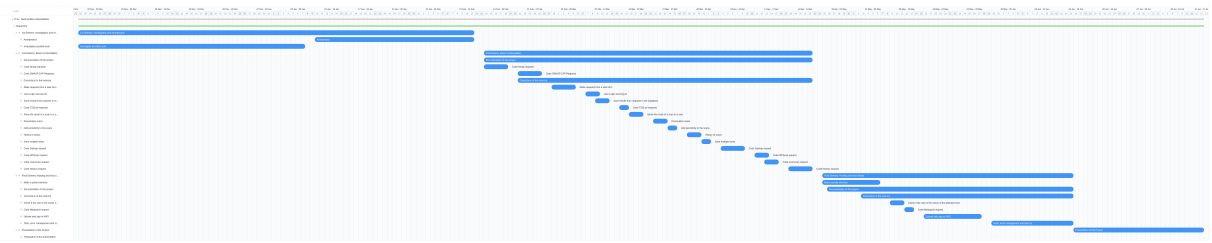

Figure 2.2: Detailed planning

2.1.1 First Delivery

The first delivery is the avantproject, and consists of an investigation phase and the writing of the avantproject (Figure 2.7).

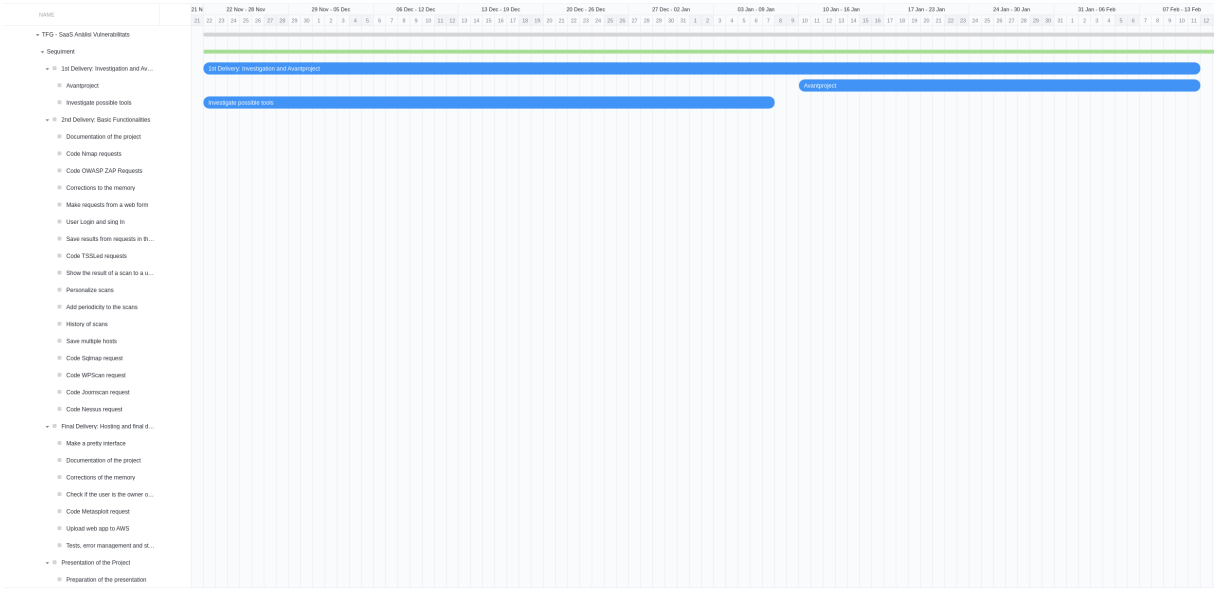

Figure 2.3: 1st delivery planning

2.1.2 Second Delivery

The second delivery focuses on the creation of the basic functionalities. At the end of this phase, the website is able to function and fulfills most of the basic requirements. It is the phase with more subtaks as it can be seen in the next figure 2.4:

1. **Documentation of the project:** Is the documentation of the processes followed to the implementation
2. **Code Nmap requests:** Code requests to Nmap and receive an output that can be saved
3. **Code OWASP ZAP requests:** Code requests to OWASP ZAP and receive an output that can be saved

4. **Corrections to the memory:** It starts after the delivery of the correction
5. **Make requests from web form:** Create web form to send request to Nmap and OWASP ZAP
6. **User login and sing in:** Create a form to register and login users
7. **Save results from requests in the database:** Save the results of the requests to the database
8. **Code TSSLed requests:** Integration of TSSLed into the webform of requests
9. **Show the results of a scan to a user:** Create a page to show results in an organized way
10. **Personalize scans:** Form to personalize scans
11. **Add periodicity to the scans:** Form to personalize the periodicty of the scans
12. **History of scans:** Shows the results of previous scans of the user
13. **Save multiple hosts:** A user can have more than one host saved
14. **Code Sqlmap request:** Integration of sqlmap to the requests
15. **Code WPScan request:** Integration of WPScan to the requests
16. **Code Joomscan request:** Integration of Joomscan to the requests
17. **Code Nessus request:** Integration of Nessus to the requests

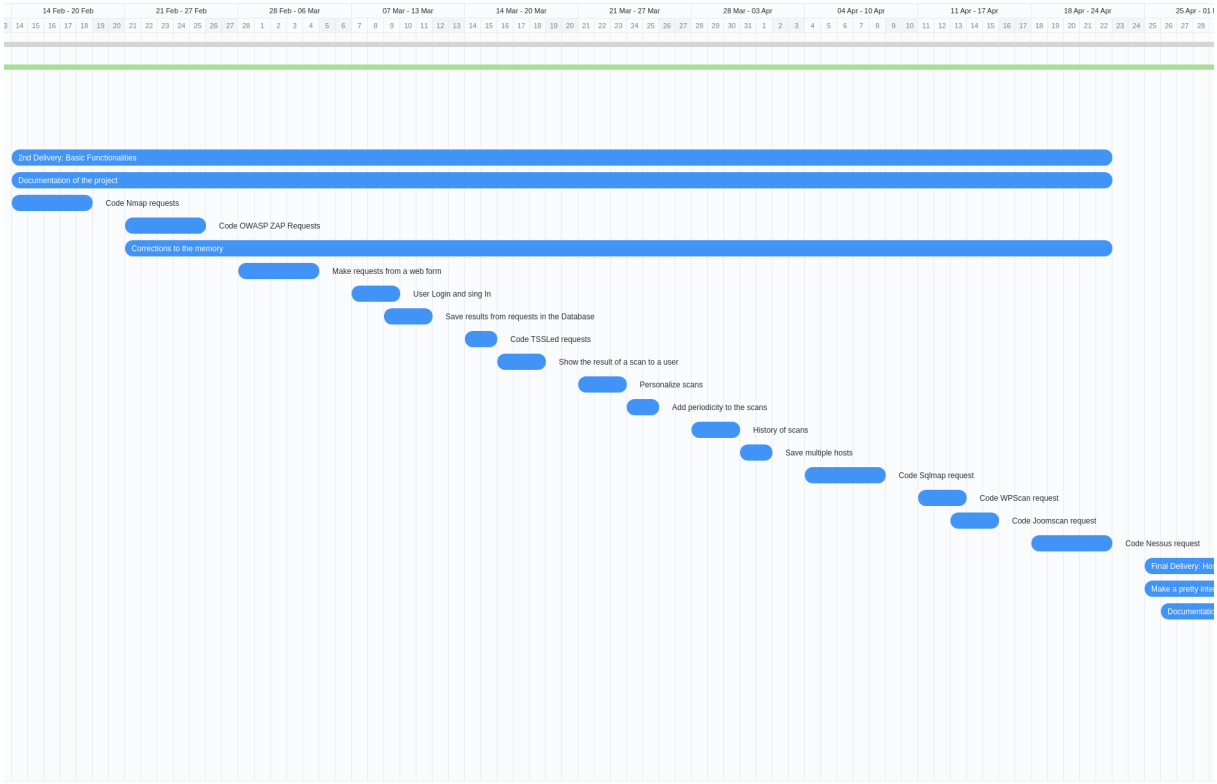

Figure 2.4: 2nd delivery planning

2.1.3 Final Delivery

The final delivery is about hosting the web application and giving the final details. There are less subtasks than in the previous phase (Figure 2.5):

1. **Make a pretty interface:** Give to the web page a good looking interface
2. **Documentation of the project:** Document the process of the implementation
3. **Corrections of the memory:** Make the necessary changes to the memory
4. **Check if the user is the owner of the attacked host:** Create a method to validate that the user is the owner of the hosts that it is wanted to be attacked
5. **Code Metasploit request:** Integration of Metasploit to the requests
6. **Upload web app top AWS** Upload the web application to the host and work in remote
7. **Tests, error management and start up** Test if there are any error left, solve problems if they are and start up the web application

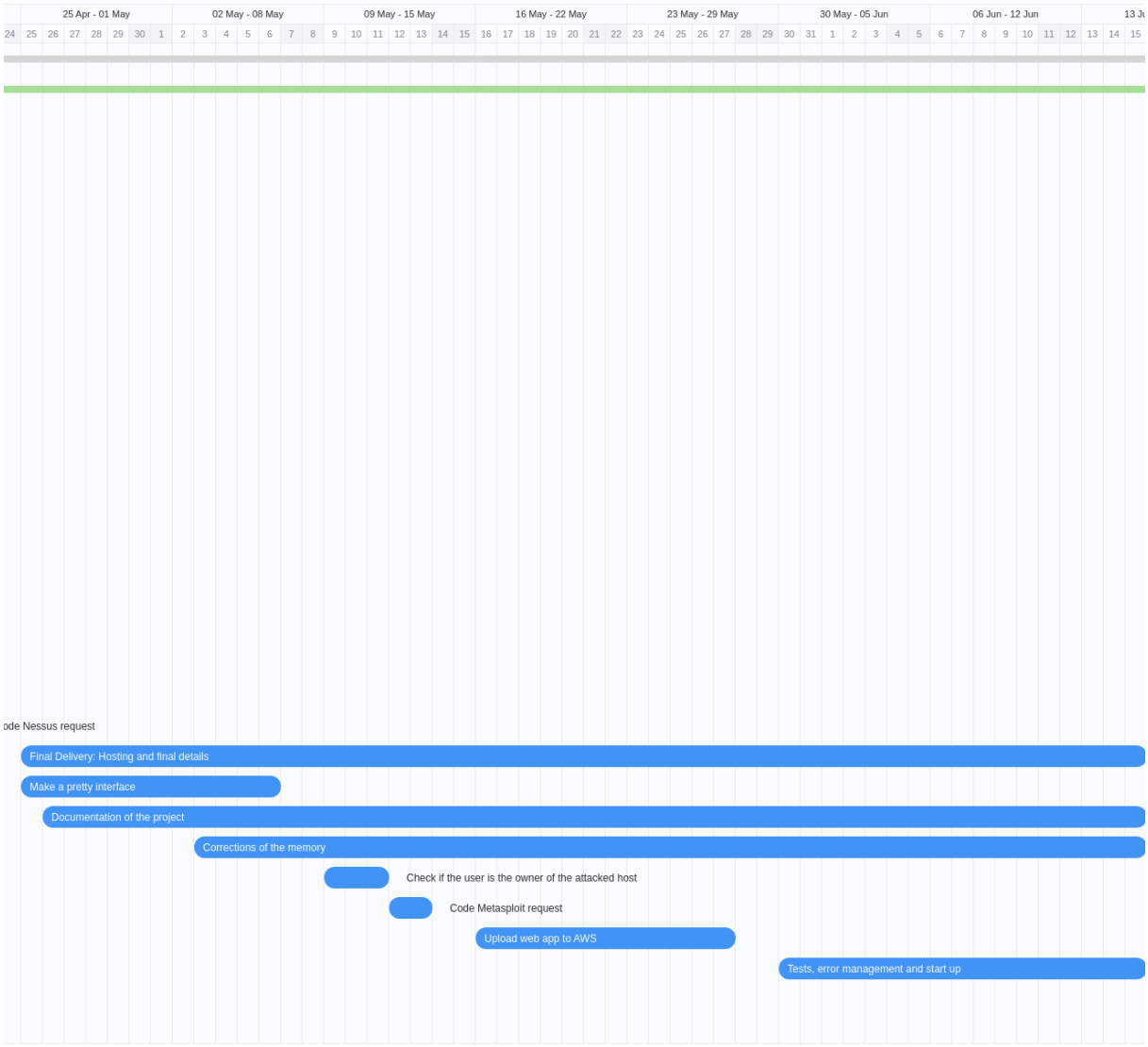

Figure 2.5: Final delivery planning

2.1.4 Presentation

The last part is the presentation and it consists of the preparation of the presentation (Figure 2.10).

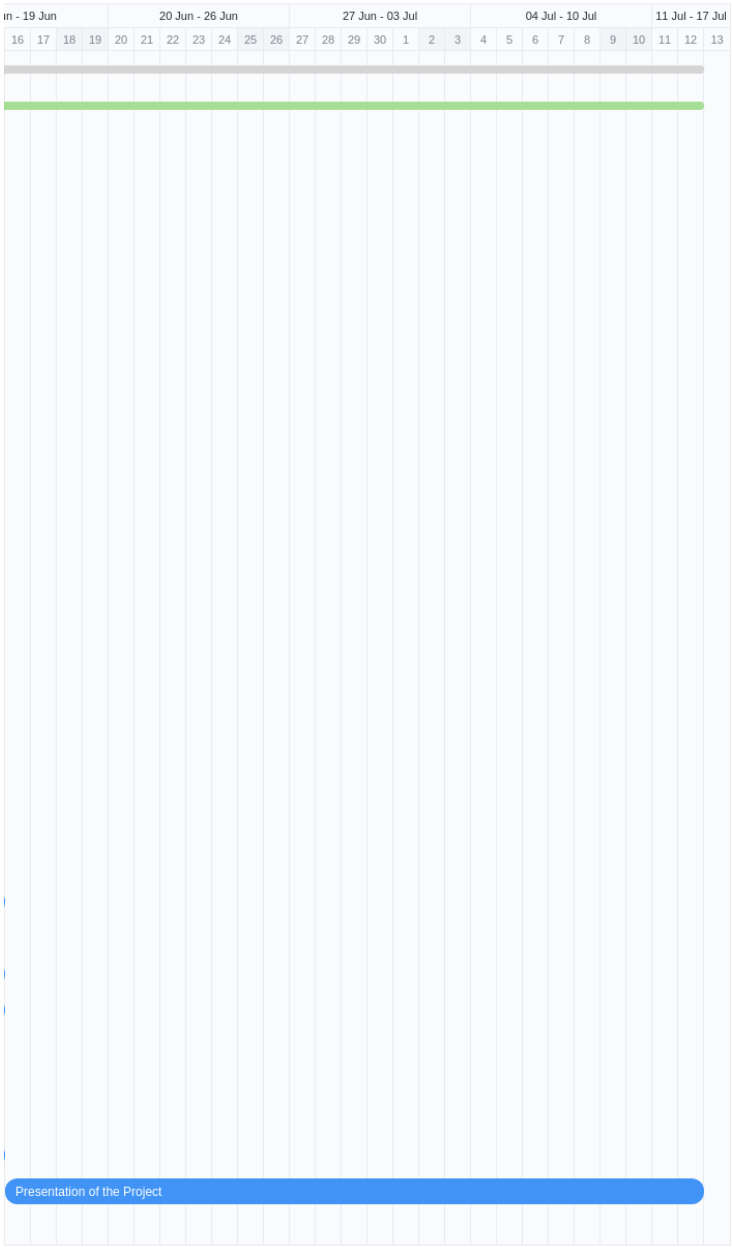

Figure 2.6: Presentation planning

2.2 Result of the Planning

The final planning has maintained the same structure of three phases and a presentation. But during the development, some needs have changed, and the planning has had to adapt accordingly. We can see major differences in the sub-tasks of each phase.

The final worked hours are:

Hours	
Phase	Hours
1st Delivery: Investigation and Avantproject	75h
2nd Delivery: Basic Functionalities	193h
Final Delivery: Hosting and final details	154h
Presentation of the project	25h
Total Hours: **447 hours**	

Table 2.2: Table of hours

2.2.1 First Delivery

The first delivery has remained the same and has no differences because it was programmed and done, almost at the same time.(Figure 2.7).

Figure 2.7: 1st delivery planning

2.2.2 Second Delivery

In this second delivery, we can see differences between the number of tasks and the days that it takes to finish them. (figure 2.8) The sub-tasks that were completed in that phase are:

1. **Code Nmap requests:** Code requests to Nmap and receive an output that can be saved.
2. **Code OWASP ZAP requests:** Code requests to OWASP ZAP and receive an output that can be saved.
3. **Make requests from web form:** Create web form to send request to Nmap and OWASP ZAP.
4. **User login and sing in:** Create a form to register and login users.
5. **Save results from requests in the database:** Save the results of the requests to the database.

6. **Show the results of a scan to a user:** Create a page to show results in an organized way.
7. **Create Attacks for host:** Creation of a class host to save the possible machines to attack.
8. **Save multiple hosts:** A user can have more than one host saved.
9. **Analyse results of a scan:** Creation of a class to analyze the results and present them to the user.
10. **Create classes to define types of scans:** Creation of classes to provide the user with different types of scans.
11. **Integration as a provider:** Create the API to communicate as a Provider with the Saas application.
12. **Write memory:** Is the documentation of the processes followed to the implementation

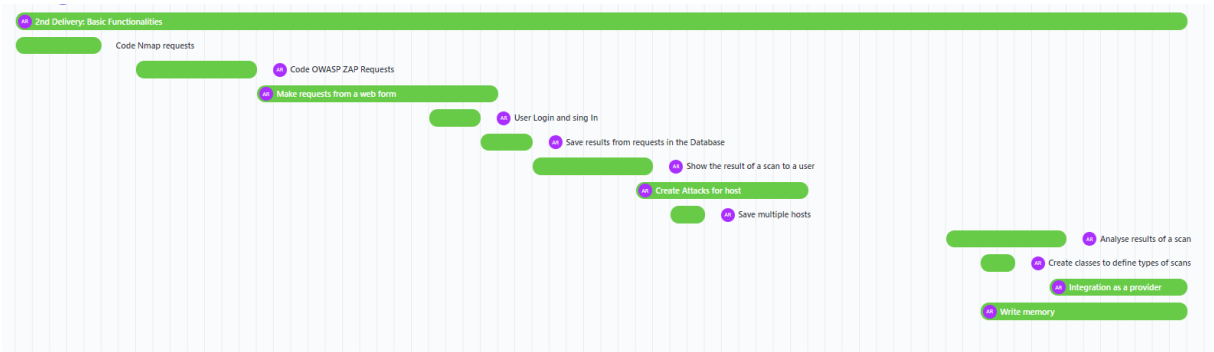

Figure 2.8: 2nd delivery planning

Some of the tasks like ”Make requests from a web form” took longer than expected and forced to move other tasks like, ”Code TLSSLed requests”, ”Code WPSCan Requests”, or ”Code Joomscan”. Another important thing that we can see is that there is a blank gap where there is no activity. The period of inactivity was due to other responsibilities that consumed a lot of time.

2.2.3 Final Delivery

In the final delivery we also have tasks removed or with different times than the expected ones. (Figure 2.9):

1. **Integration as a provider:** Continue with the integration as a provider.
2. **Code TLSSLed requests:** Integration of TSSLed into the webform of requests.
3. **Code SQLMap requests:** Integration of Metasploit to the requests.
4. **Code Joomscan request:** Integration of Joomscan to the requests.
5. **Code WPScan request:** Integration of WPScan to the requests.
6. **Put web app and scanner tool in the same server:** Change from a virtualized kali machine into a docker kali machine.
7. **Code Traceroute:** Integration of Traceroute to the requests.
8. **Personalize scans:** Adapt the resutls .
9. **Documentation of the API with swagger:** Document the API with swagger.
10. **Code Nessus request:** Integration of Nessus to the requests.
11. **Upload web app top AWS** Upload the web application to the host and work in remote
12. **Tests, error management and start up** Test if there are any error left, solve problems if they are and start up the web application

13. **Documentation of the project:** Document the process of the implementation

14. **Make a pretty interface:** Give to the web page a good looking interface

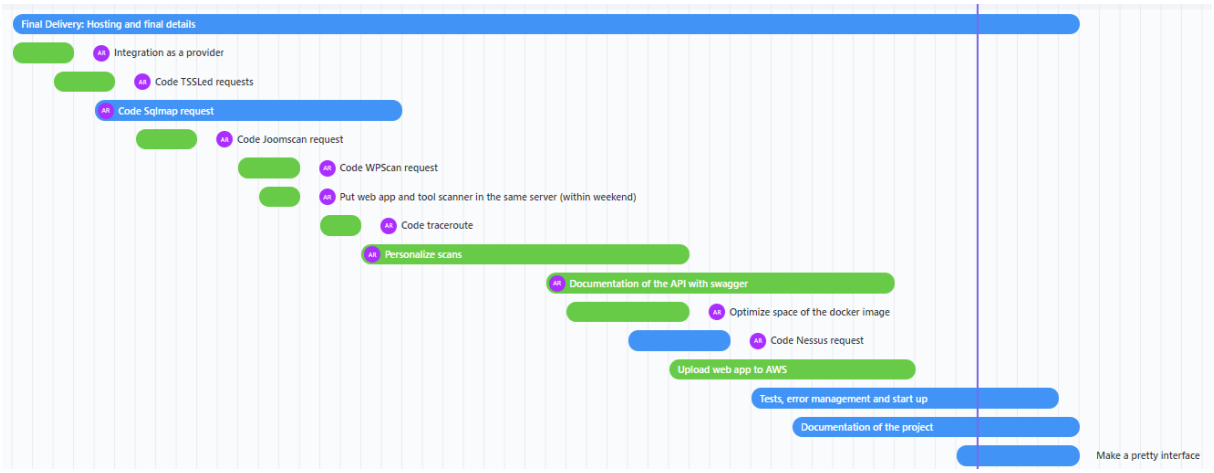

Figure 2.9: Final delivery planning

The tasks ”Code Sqlmap request” and ”Code Nessus requests” were not finished because of the time limit. They were not considered primary tools and given the time constraints, we decided to not lose more time with them, to have more time for more important things.

2.2.4 Presentation

The presentation continues with the same schedule, as we can see below: (Figure 2.10).

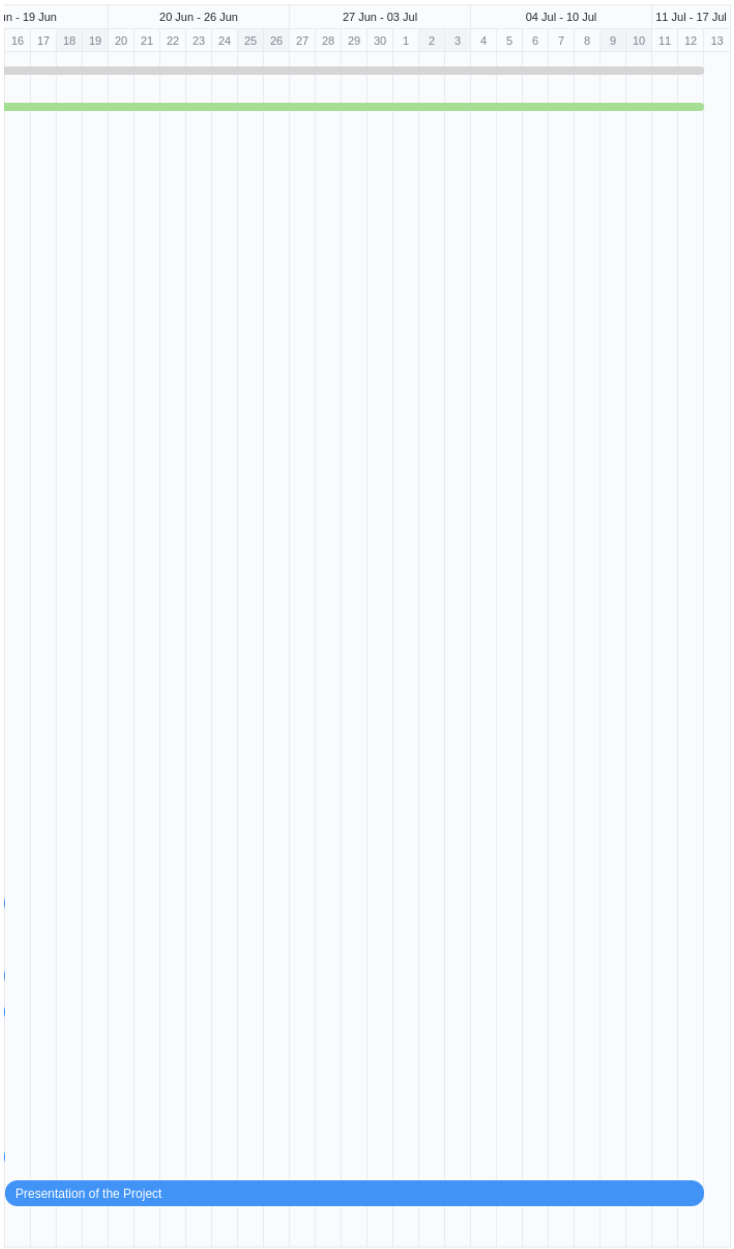

Figure 2.10: Presentation planning

2.3 Initial Budget

Given the initial planning, it is estimated that it will take:

- 75 hours of Investigation
- 150 hours of Writing
- 275 hours of Development

This estimation gives 500 hours of work in total. Taking into account that the different parts are paid 10, 9 and 12 euros per hour respectively. The total of the **gross wage perceived by the worker** is:

- Investigation: $75hours \times 10€ = 750€$
- Writing: $150hours \times 9€ = 1.350€$
- Development: $275hours \times 12€ = 3.300€$
- **TOTAL: 5.400€**

Then the employer has to add up the Social Security fee, which is approximately a 30% of the wage. **The final wage is** $5.400 + 30\% =$ **7.020€** (See table 2.3)

To develop the project is necessary a **laptop**, which has a price of 1.500€. It has an **amortised cost of 300€** per year (See table 2.4).

The costs of the **electricity, gas, water, Internet and space** are estimated in **1.680€** (See table 2.5).

This gives a **total of 9.000€** (See table 2.8)

Wages			
Area	Hours	Price/Hour	Total
Investigation	75h	10€	750€
Writing	150h	9€	1.350€
Development	275h	12€	3.300€
Total Amount Before Fees: **5.400€**			
Social Security Fee (30%)			
Total Amount After Fees: **7.020€**			

Table 2.3: Table of wages

Amortizations			
Name	Price	Years	Total
Laptop	1.500€	5	300€
Total: **300€**			

Table 2.4: Table of amortizations

Other costs			
Resource	Months	Price/Month	Total
Electricity	7	50€	350€
Gas	7	40€	280€
Water	7	15€	105€
Internet	7	35€	245€
Space	7	100€	700€
Total: **1.680€**			

Table 2.5: Table of other costs

Total cost	
Cost	Price
Wages	7.020€
Amortizations	300€
Other costs	1.680€
Total: **9.000€**	

Table 2.6: Table of total cost

2.4 Final Budget

With the final planning, we have different times, and thus, a different budgets:

- 75 hours of investigation
- 120 hours of Writing
- 252 hours of development

This estimation gives 447 hours of work in total. Taking into account that the different parts are paid 10, 9 and 12 euros per hour respectively. The total of the **gross wage perceived by the worker** is:

- Investigation: $75hours \times 10€ = 750€$
- Writing: $120hours \times 9€ = 1.080€$
- Development: $252hours \times 12€ = 3.024€$
- **TOTAL: 4.854€**

Then the employer has to add up the Social Security fee, which is approximately a 30% of the wage. **The final wage is** $4.854 + 30\% =$ **6.310€** (See table 2.7)

The rest of the costs have stayed the same (See table 2.4 for amortizations) (See table 2.5 for other costs), which gives a **total of 8.290€** (See table 2.8)

Wages			
Area	Hours	Price/Hour	Total
Investigation	75h	10€	750€
Writing	150h	9€	1.350€
Development	275h	12€	3.300€
Total Amount Before Fees: **5.400€**			
Social Security Fee (30%)			
Total Amount After Fees: **7.020€**			

Table 2.7: Table of wages

Total cost	
Cost	Price
Wages	6.310€
Amortizations	300€
Other costs	1.680€
Total: **8.290€**	

Table 2.8: Table of total cost

3. Viability Analysis

This chapter analyzes the technical viability, the economic perspectives of the project, the impact that this project can have on the environment, and the legal aspects involved with the development of the project.

3.1 Technical Viability Analysis

This web application does not use new or experimental technologies. It focuses on well-established technologies that are used daily by professionals. All these technologies are well documented and tested by multiple users. This makes it relatively easy to find a solution on the Internet to each type of problem. As we see it, it is a viable project from a technical perspective.

3.2 Economical Viability Analysis

This project is part of a larger Software as a Service (SaaS) web, so it doesn't make any income on its own. The user of this service does not pay to use it because he is the owner. The benefits of this project come from the ability of the SaaS web to convert free users using this service to paid users.

Even if the project does not achieve a high enough conversion rate, it can provide a solid ground to develop better attack frameworks. Therefore, saving time, and making this investment valuable to the company.

3.3 Environmental Viability Analysis

We used to think that hosting a web application on a server wouldn't have any effect on the environment, but recent studies have shown that data centres and servers have grown to account for an estimated 3.7 % of global carbon emissions [1].

Once understood that this service has an environmental impact, it is relevant to mention that there exist approximately 200 million web pages on the Internet [2]. Assuming

that all of them have the same environmental footprint, each one of these websites is contributing a $1,85 \times 10^{-8}$ % of the global carbon emissions.

Even though this project would have a minimal impact on the environment, as seen earlier, it is necessary to have in mind that we should fight to reduce this impact by using servers that use energy-efficient technologies and renewable energies.

3.4 Legal Aspects

It is important to keep in mind that this project may conflict with some of the laws that are in effect. As it is a service which analyses public hosts for vulnerabilities and stores the data collected, it has to respect several regulations, some of the most important are:

- The Organic Law 3/2018 of 5 December on the Protection of Personal Data and granting the digital rights. [3]
- Regulation (EU) 2016/679 (General Data Protection Regulation) [4]
- Intellectual Property Act No 1/1996 [5]
- ISO/IEC 27032:2012[6]

4. Conclusions

The viability analysis shows that the planning has had to change due to lack of time and the lack of knowledge in a few areas. We have had to discard tools like Sqlmap and Nessus, and we have had to implement unexpected things like transforming the virtual kali machine into a docker-machine. In the end, the base tools are integrated and there is a solid structure to build more tools on top.

The budget has changed due to that 50 hours difference, but the differences are 610€ less, which is not a great deviation from the initial budget.

We believe that this project continues to be a viable project, in each of the technical, economic and environmental aspects, and it is worth continuing to invest in it to gain bigger benefits.

5. Bibliography

[1] Website performance and environmental impact — digital communications team blog. https://digitalcommunications.wp.st-andrews.ac.uk/2020/03/30/website-performance-and-the-hidden-environmental-impact-of-the-internet/.

[2] How many websites are there in the world? [2022] - siteefy. https://siteefy.com/how-many-websites-are-there/.

[3] Boe.es - boe-a-2018-16673 ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673.

[4] Eur-lex - 32016r0679 - en - eur-lex. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679.

[5] Boe.es - boe-a-1996-8930 real decreto legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la ley de propiedad intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.

[6] Information technology-security techniques-guidelines for cybersecurity copyright protected document.