Online privacy: Analyzing the use of cookies in web pages

Master Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Meritxell Basart Dotras

In partial fulfillment
of the requirements for the master in
Cybersecurity **ENGINEERING**

Advisor: Pere Barlet Ros
Barcelona, 30th June 2021

Contents

List of Figures

Listings

List of Tables

Revision history and approval record

Revision	Date	Purpose
0	15/05/2021	Document creation
1	18/06/2021	Document revision
2	28/06/2021	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Meritxell Basart Dotras	meritxell.basart@estudiantat.upc.edu
Pere Barlet Ros	pbarlet@ac.upc.edu
Ismael Castell Uroz	ismael.castell@upc.edu

Written by:		Reviewed and approved by:	
Date	18/06/2021	Date	30/06/2021
Name	Meritxell Basart Dotras	Name	Pere Barlet Ros
Position	Project Author	Position	Project Supervisor

Acknowledgments

First of all, I would like to thank my advisor, Pere Barlet Ros, for proposing the project to me and for supporting me throughout the process. I also want to thank Ismael Castell Uroz, for helping me with all the difficulties I have had.

On the other hand, I would like to thank Ismael Douha Prieto for the teamwork during this process. Also, all the teammates in the group that are doing their final projects and together it has been possible to develop the observatory.

I want to thank Aleix Galan and Javier De Muniategui for being part of this project in its beginnings in the subject of TMA.

Finally, my parents and family to accompany me on this journey.

Abstract

General Data Protection Regulation (GDPR) [5] is the European regulation on the protection of individuals regarding the processing of their data. One point of the GPDR says that all websites must ask their EU users for consent to have their data processed.

In this project several methods are developed to interact with web pages and analyze if websites are compliant with GDPR, analyzing the usage of cookies.

The first objective is to ignore cookies. Using ORM [2] we can identify the number of cookies that each domain sets by default. As we can see, a large number of websites sets cookies by default.

The second objective is the interaction with web pages to accept cookies. To do so, we use two different methods. The first one is a modified ORM version using Selenium and the second one also is a modified ORM version that uses Computer Vision. Both methods have shown that more than 50% of the websites visited do not ask for user consent.

Finally, the third objective of the project is to find out what would happen if we blocked cookies and how this changed the operation of the web. We show that the cookies already inserted by default will remain and very few cookies are blocked.

After we do all the methods and generate statistics, they are integrated into the ePrivacy Observatory [12], which is an observatory that provides the function of determining the level of tracking of each domain and different tracking methods.

1 Introduction

General Data Protection Regulation (GDPR) [5] is the European regulation on the protection of individuals regarding the processing of their data and the free movement of such data. One point of the GDPR in Europe is that all web pages must ask their EU customers for consent to have their data processed.

Not all websites comply with GDPR, because they are using different web tracking techniques. Web tracking is the practice in which trackers integrated into a website identification of its users while they are browsing the internet, collecting information about the sites they visit, and analyzing their behavior. With this information, trackers can implement personalized advertisements and other practices.

A cookie is a data file that a web page sends to your device when you are visiting a website. They can be essential cookies, implemented for the good operation of the website. However, trackers use cookies as a tracking technique, because it is possible to collect more information and make a user profile.

So the usage of cookies is not always for good reasons, and internet users should be worried about them, because trackers are obtaining personal information, sometimes without our consent.

In this project, we will verify if web pages ask consent from their users to have their data processed as says GDPR. Doing a cookies study, we will provide a tool for internet users to have a better knowledge of what are doing websites with cookies and this will improve the privacy of the users because they will be able to choose if they want to use this type of websites.

To do so, we will use Online Resource Mapper (ORM) [2] developed by Ismael Castell Uroz and Pere Barlet Ros, to open a web page from the most visited websites by the Alexa ranking [7] in an automated way and save different information related to resources, URL headers, etc. in a database.

Furthermore, we use different approaches to obtain distinct interactions with web pages. The first approach will consist of running ORM without any change and obtain the number of cookies added without user consent. Then we use tools like Selenium to atomize the click on the "Accept cookies" button and compare the number of cookies added doing this approach and the first one. Also, we will carry out the same approach with Computer Vision and artificial intelligence to click the "accept cookies" button developed by Pablo Fonoll Soto. Accepting cookies EU costumers provide their consent to web pages.

The next approach is adding a plugin to ORM. With the previous approaches, we can accept cookies, however, we can not decline them. So we will use the Ninja Cookies plugin [1] to decline the cookies. Finally, we will determine the country of each domain to see if it is an EU country.

Once we have done the study about compliance, we will add the results in the ePrivacy Observatory [12], developed by Pol Mesegué Molina, which will provide the capacity for users to check in real-time different parameters of web pages related to the compliance of

GDPR. Also, the idea of this observatory is to see which web pages are not following the rules, and could be useful also for governments to detect who is not following the law.

The following schema shows the different modules that ePrivacy Observatory has. The first two modules, **HTTP cookies** and **JS cookies**, base these modules on data that a web page sends to your device when you are visiting a website and it is stored. Another module is **Canvas fingerprinting**, they based this method on how a browser renders an image and gets information. **Font fingerprinting** it is based on getting an identifier in the function of the fonts that users have. The next one is **Mouse fingerprinting**, which can identify the user based on the mouse movement. The last one is **WebGL fingerprinting**, which is a technique that identifies your browser based on rendering an image in the GPU. It obtains hardware and software information. All these modules are developed by Ismael Douha Prieto. Mouse tracking was developed by Álvaro Macias López and Ismael Douha Prieto added it into a module.

Finally, the last module is **Web page interactions**, which has four sub-modules. These four modules are the ones explained at the beginning. The first one is to run ORM without any plugin or modification and extract the number of cookies added to each web page without consent. Then the Ninja Cookie plugin should decline cookies. Finally, the last two modules are different approaches to click the button ”accept cookies”.

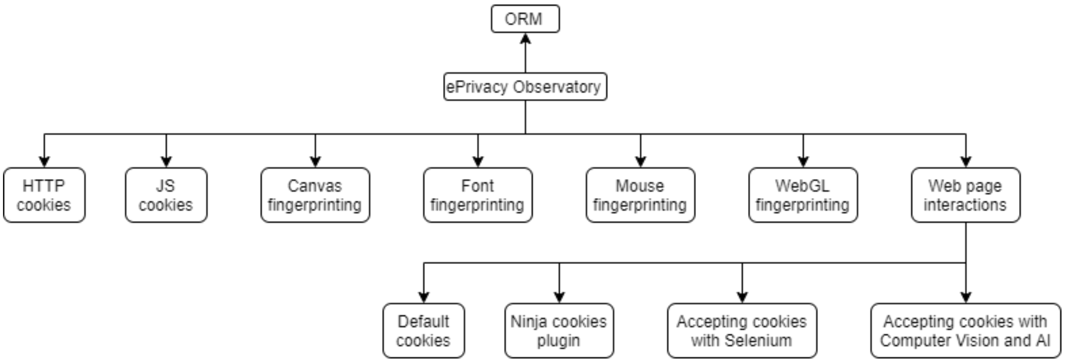


Figure 1: Global Scheme.

This project contains the following parts, first of all, an introduction where there are defined the different objectives, the work plan, and deviations. Section two is possible to find the state of the art of technology used or applied in this thesis. Section three contains the methodology and project development. In section 4 we can find the results. Section 5 contains the budget. Section 6 describes the environmental impact. Finally, section 7 contains the conclusions and future development.

1.1 Statement of purpose

In this section we define the main objectives that we want to achieve with this work.

- Verify if web pages are compliant with the GDPR point related to ask consent to their EU costumers to have their data processed.
- Use ORM to obtain information of web pages.
- Use different modules and plugins to be able to accept and reject cookies.
- Integrate results with ePrivacy Observatory to provide internet users the capacity to obtain this information.

1.2 Requirements and specifications

This project should accomplish the following requirements:

- Running ORM and collecting web pages data.
- Install Ninja Cookie plugin and should prevent the installation of cookies.
- Modifying ORM using Selenium to click the ”accept cookies” button should add more cookies after clicking the button.

1.3 Methods and procedures

This work uses Online Resource Mapper (ORM) [2], which is a tool that maps the relation between each URL with the corresponding resources loaded when they open the webpage. The tool is developed by Ismael Castell Uroz and Pere Barlet Ros. More precisely, in this project, we will use this tool to inspect the number of loaded cookies.

On the other hand, once we will obtain the results, we add the information in ePrivacy Observatory, which is an observatory that provides in real-time information related to web pages to know different specifications and if they are compliant with GDPR or not.

1.4 Work plan

In this section we describe the different tasks that we will realize in this thesis, the planning to carry out the tasks in the corresponding time, milestones, and finally a Gantt diagram.

- **Initial date:** 9th February 2021
- **Final date:** 30th June 2021
- **Duration:** 360 hours
- **Expected lecture date:** 7th July 2021
- **Daily dedication:** 3.5 hours

1.4.1 Tasks description

- **G1: Project management**
  - *Control meetings (20h):* Meetings every 2 weeks to monitor the progress of the project.
- **G2: Learning**
  - *State of the art (40 h):* Search information and papers in order to learn how the topic is nowadays.
  - *ORM (25h):* Understand and install the code.
  - *Selenium (15 h):* Do some tutorials about Selenium.[4]
  - *Python review (20h):* Review some specific things about python.
- **G3: Development**
  - *Modifying ORM (30 h):* Adding some modifications to ORM in order to accept cookies in the URLs.
  - *Add plugins (20 h):* Adding "Ninja Cookie"[1] and "I don't care about cookies"[3] plugins in ORM in order to collect information with this plugins activated.
  - *Determining the country of each domain (20 h):* After executing ORM, execute another script to determine the country of each domain.
  - *Generating results (60 h):* Generating plots to see the results obtained about cookies.
- **G4: Documentation**
  - *Memory write (70h):* Develop a document with information and procedures.
  - *Lecture preparation (40h):* Prepare the presentation.

1.4.2 Gantt Diagram

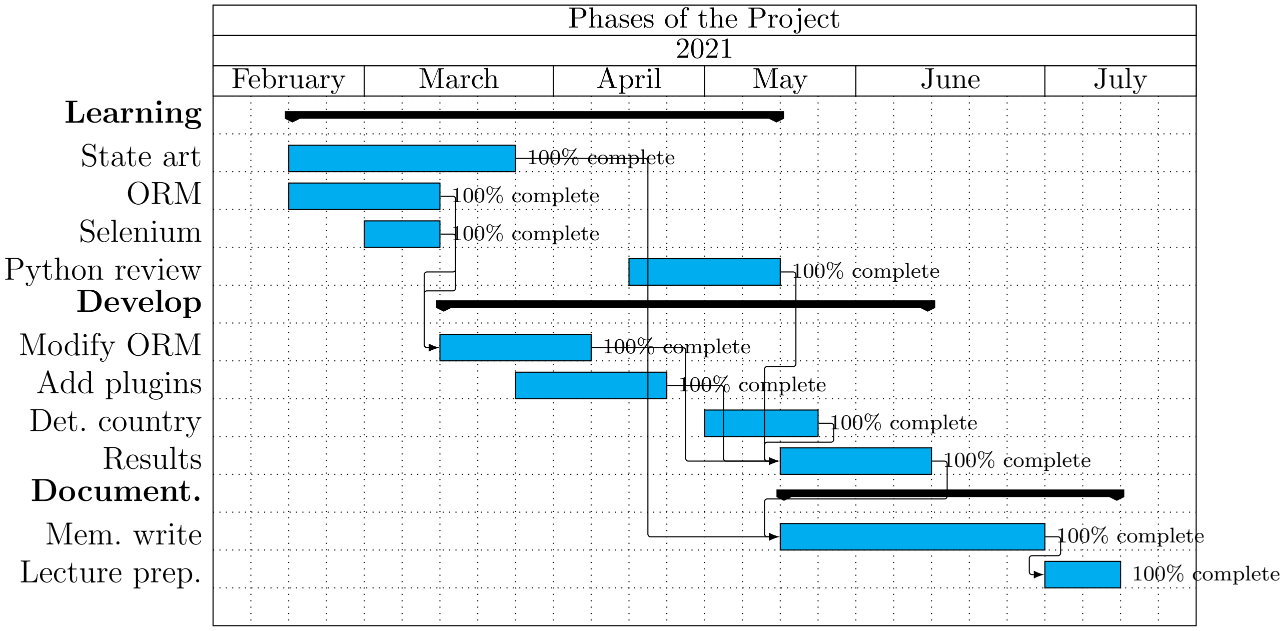


Figure 2: Gantt diagram of the project

Work plan in the Gantt Diagram format.

1.5 Deviations from the initial plan

The initial plan was to integrate this project with ORM codesets, which is the newest branch of this project, but for plugin compatibility reasons it has not been possible. It means modifying lots of things of the original project to integrate it and some time. We extract interesting results, in the future could be interesting to try to integrate these results with the actual ORM version.

2 State of the art

In this section, we do a review of the actual state of the art. We define GDPR, more precisely what says about web pages that must ask consent from their EU customers. Also, we do research related to if web pages are compliant with the law nowadays.

2.1 GDPR

In this section, we define what is GDPR (General Data Protection Regulation). GDPR is a law that was put in effect on May 25, 2018, and defines that all the data collected from European Union (EU) people must comply with laws related to privacy and security. If websites violate these laws, they apply fines of millions of euros. [5]

GDPR applies to all enterprises that collect data from EU people, this means that even though the company is not in an EU country, also must follow GDPR laws.

This project focuses on a specific point of GDPR, the one related to consent. More precisely, the web page *"What is GDPR, the EU's new data protection law"* [5] says the following related to consent to processing the information of a person:

- *Consent must be "freely given, specific, informed and unambiguous."*
- *Requests for consent must be "clearly distinguishable from the other matters" and presented in "clear and plain language."*
- *Data subjects can withdraw previously given consent whenever they want, and you have to honor their decision. You can't simply change the legal basis of the processing to one of the other justifications.*
- *Children under 13 can only give consent with permission from their parent.*
- *You need to keep documentary evidence of consent.*

Furthermore, another important point that mentions on the same web page, is the one related to when GDPR allows a web page to process data [5]:

Article 6 lists the instances in which it's legal to process person data. Don't even think about touching somebody's personal data — don't collect it, don't store it, don't sell it to advertisers — unless you can justify it with one of the following:

1. *The data subject gave you specific,* ***unambiguous consent*** *to process the data.*
2. *Processing is necessary to execute or to prepare* ***to enter into a contract*** *to which the data subject is a party.*
3. *You need to process it* ***to comply with a legal obligation*** *of yours.*
4. *You need to process the data* ***to save somebody's life.***
5. *Processing is necessary* ***to perform a task in the public interest*** *or to carry out some official function.*

6. *You have a **legitimate interest** to process someone's personal data. This is the most flexible lawful basis, though the "fundamental rights and freedoms of the data subject" always override your interests, especially if it's a child's data.*

Once you've determined the lawful basis for your data processing, you need to document this basis and notify the data subject (transparency!). And if you decide later to change your justification, you need to have a good reason, document this reason, and notify the data subject.

In this work, we want to see if web pages comply with these points, if they clearly ask the consent of their users and they give it freely, and they not collect the information and store it before.

2.2 Cookies

A cookie is a data file that a web page sends to your device when you are visiting a website. Your web browser will process and store them. Also, it is easy to delete and view the cookies.

There are essential cookies used to navigate the website and give us the possibility to use the necessary features of that page. The problem with the cookies is that they can store enough data to identify the user without their consent.

Usually, advertisers use cookies to track the online activity of their users and make a profile, after that with this information will show personalized advertisements. This information can be considered personal data so it is a point that GDPR takes into account and have some rules that cookies have to follow.

In *"Cookies, the GDPR, and the ePrivacy Directive"* [6] GDPR and ePrivacy Directive says the following about cookies compliance:

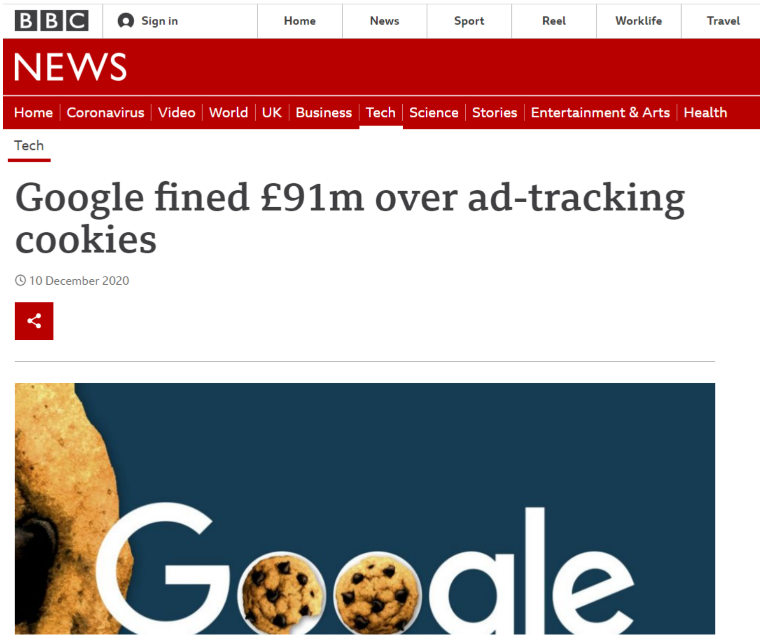
- *Receive users' consent before you use any cookies except strictly necessary cookies.*
- *Provide accurate and specific information about the data each cookie tracks and its purpose in plain language before consent is received.*
- *Document and store consent received from users.*
- *Allow users to access your service even if they refuse to allow the use of certain cookies*
- *Make it as easy for users to withdraw their consent as it was for them to give their consent in the first place.*

Given the knowledge of cookies compliance, in this project using ORM and different plugins and modifications of ORM we will obtain the cookies of each domain of the Alexa Ranking [7] and we will analyze if web pages comply with the rules mentioned before.

2.3 News

In this section, we search news related to fines that some companies paid because they have not complied with GDPR and the points presented in the sections before.

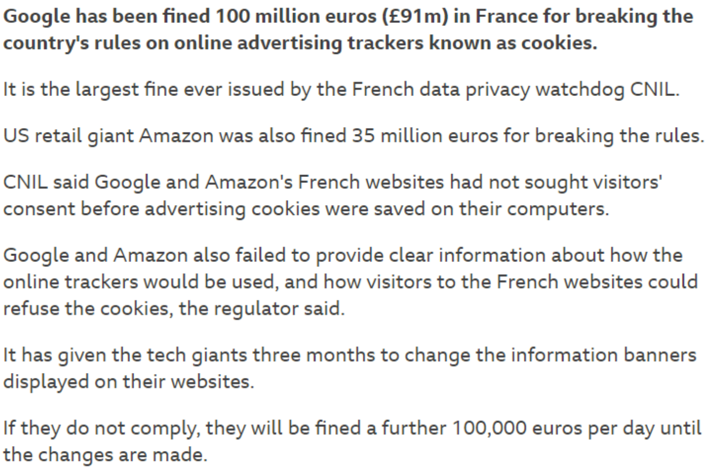
In BBC news we can see the following fine that Google paid in France because breaks the rules related to the use of cookies. Also, we can see in the same news that Amazon paid an important fine too: [8]



Google fined £91m over ad-tracking cookies

10 December 2020

Figure 3: BBC news Google fine.[8]



Google has been fined 100 million euros (£91m) in France for breaking the country's rules on online advertising trackers known as cookies.

It is the largest fine ever issued by the French data privacy watchdog CNIL.

US retail giant Amazon was also fined 35 million euros for breaking the rules.

CNIL said Google and Amazon's French websites had not sought visitors' consent before advertising cookies were saved on their computers.

Google and Amazon also failed to provide clear information about how the online trackers would be used, and how visitors to the French websites could refuse the cookies, the regulator said.

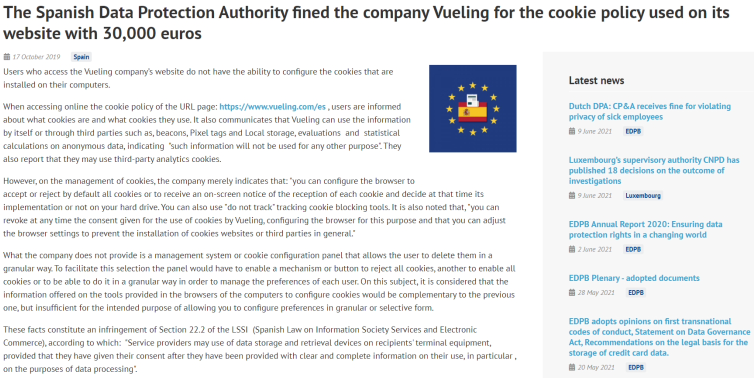
It has given the tech giants three months to change the information banners displayed on their websites.

If they do not comply, they will be fined a further 100,000 euros per day until the changes are made.

Figure 4: BBC news text.[8]

Also, we can see another one in European Data Protection Board, this one is a fine that

Vueling paid because they announced cookies, however, it was not possible to manage the cookies properly. In figure 5 we can see all the information related to the fine of Vueling.



The Spanish Data Protection Authority fined the company Vueling for the cookie policy used on its website with 30,000 euros

17 October 2019 Spain

Users who access the Vueling company's website do not have the ability to configure the cookies that are installed on their computers.

When accessing online the cookie policy of the URL page: https://www.vueling.com/es , users are informed about what cookies are and what cookies they use. It also communicates that Vueling can use the information by itself or through third parties such as, beacons, Pixel tags and Local storage, evaluations and statistical calculations on anonymous data, indicating "such information will not be used for any other purpose". They also report that they may use third-party analytics cookies.

However, on the management of cookies, the company merely indicates that: "you can configure the browser to accept or reject by default all cookies or to receive an on-screen notice of the reception of each cookie and decide at that time its implementation or not on your hard drive. You can also use "do not track" tracking cookie blocking tools. It is also noted that, "you can revoke at any time the consent given for the use of cookies by Vueling, configuring the browser for this purpose and that you can adjust the browser settings to prevent the installation of cookies websites or third parties in general."

What the company does not provide is a management system or cookie configuration panel that allows the user to delete them in a granular way. To facilitate this selection the panel would have to enable a mechanism or button to reject all cookies, another to enable all cookies or to be able to do it in a granular way in order to manage the preferences of each user. On this subject, it is considered that the information offered on the tools provided in the browsers of the computers to configure cookies would be complementary to the previous one, but insufficient for the intended purpose of allowing you to configure preferences in granular or selective form.

These facts constitute an infringement of Section 22.2 of the LSSI (Spanish Law on Information Society Services and Electronic Commerce), according to which: "Service providers may use of data storage and retrieval devices on recipients' terminal equipment, provided that they have given their consent after they have been provided with clear and complete information on their use, in particular , on the purposes of data processing".

Latest news

Dutch DPA: CP&A receives fine for violating privacy of sick employees
9 June 2021 EDPB

Luxembourg's supervisory authority CNPD has published 18 decisions on the outcome of investigations
9 June 2021 Luxembourg

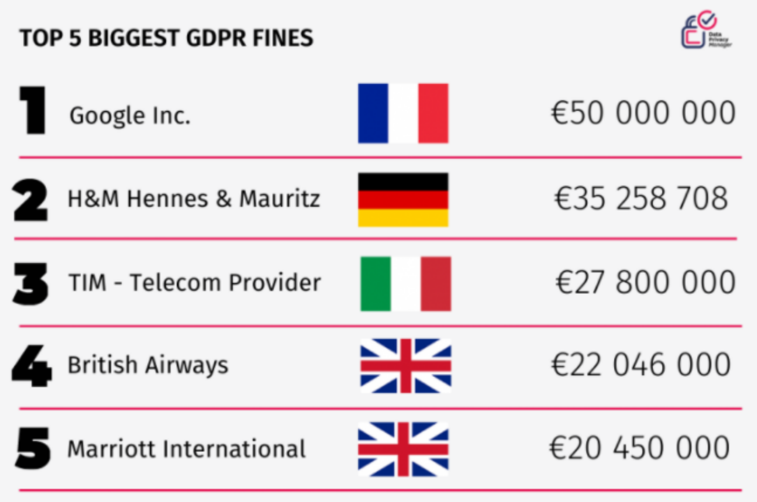
EDPB Annual Report 2020: Ensuring data protection rights in a changing world
2 June 2021 EDPB

EDPB Plenary - adopted documents
28 May 2021 EDPB

EDPB adopts opinions on first transnational codes of conduct, Statement on Data Governance Act, Recommendations on the legal basis for the storage of credit card data.
20 May 2021 EDPB

Figure 5: Fine of Vueling. [9]

Finally, two more interesting web pages are dataprivacymanager.net [10] and gdpr-fines.inplp.com [11]. The first page shows different information related to the 5 biggest GDPR fines as it is possible to see in Figure 6 and a detailed explanation of the reason for that fines. Furthermore, there is more information related to the number of GDPR fines by country, the top country is Spain with 212 fines, and the total amount by country, the top country is Italy with 76.065.307 euros [10]. The second web page, [11] is a database of GDPR fines, and it is possible to see a list with detailed information of the fines.



TOP 5 BIGGEST GDPR FINES

1	Google Inc.	🇫🇷	€50 000 000
2	H&M Hennes & Mauritz	🇩🇪	€35 258 708
3	TIM - Telecom Provider	🇮🇹	€27 800 000
4	British Airways	🇬🇧	€22 046 000
5	Marriott International	🇬🇧	€20 450 000

Figure 6: 5 biggest GDPR fines. [10]

After this brief section, we can see that although the GDPR rules, lots of companies do not comply with the corresponding rules. Nowadays there are lots of businesses that do not ask properly for consent from their customers and save personal data.

2.4 Articles

In this section, we analyze some papers to have a deeper knowledge of what is done related to the study of the compliance of GDPR and cookie banners and how they ask consent from their customers.

In the paper *"Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence"* [13] they analyze two things, the first one is how are the design of the CMPs (consent management platforms) and the prevalence of the elements that do not comply with GDPR. The second thing that they analyze is how the design affects the consent of the users.

To answer the first question, they use a web scraper with Python library Scrapy and JavaScript rendering service Splash from the top 10.000 UK sites according to web-traffic service Alexa. They determine the compliance as:

- *Consent must be explicit*
- *Accepting all is as easy as rejecting all*
- *No pre-ticked boxes*

They determine that only 11.8% of the sites designs of the CMPs complies the 3 points as it is possible to see in figure 7.

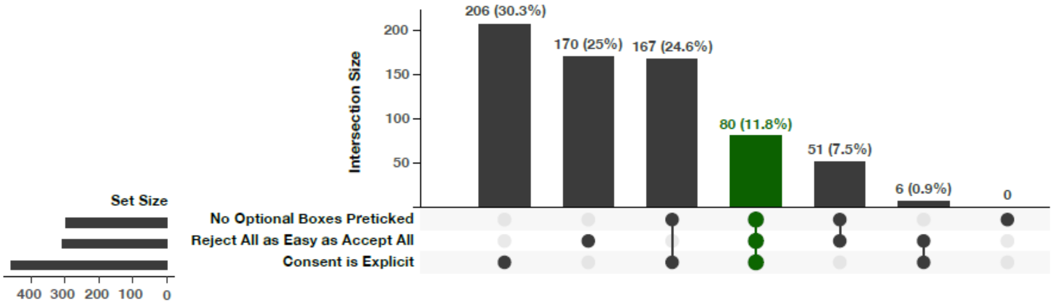
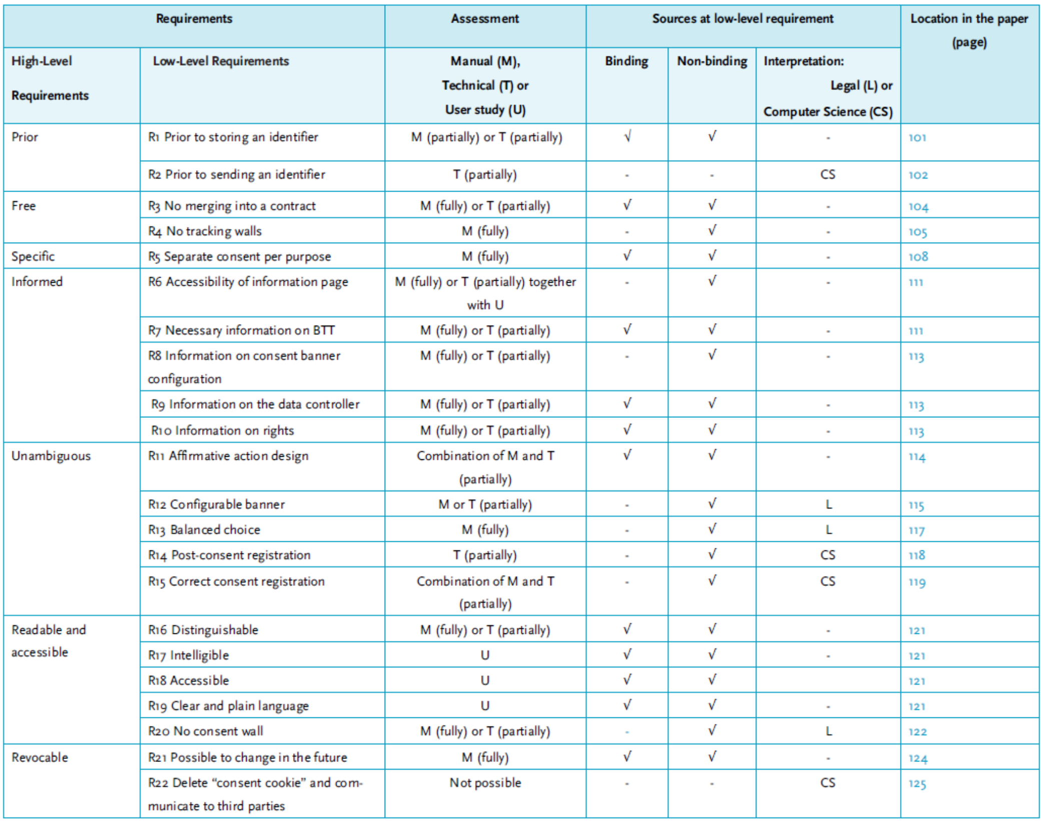


Figure 7: Designs of CMPs that comply with EU law. [13]

Related to how the design affects the answers, they did different designs and install it as an extension on Chrome and inject a pop-up every fourth URL visited. They had 40 participants from the US. The result was that most of the participants after knowing the results they said that do not match their ideal settings.

In the article *"Are cookie banners indeed compliant with the law?"* [14] they identify 22 legal-technical requirements for a valid consent cookie banner design based on the requirements of GDPR, Justice of the EU (CJEU), and their interpretation. In Figure 8 we can see the 22 requirements:



Requirements		Assessment	Sources at low-level requirement			Location in the paper (page)
High-Level Requirements	Low-Level Requirements	Manual (M), Technical (T) or User study (U)	Binding	Non-binding	Interpretation: Legal (L) or Computer Science (CS)	
Prior	R1 Prior to storing an identifier	M (partially) or T (partially)	√	√	-	101
	R2 Prior to sending an identifier	T (partially)	-	-	CS	102
Free	R3 No merging into a contract	M (fully) or T (partially)	√	√	-	104
	R4 No tracking walls	M (fully)	-	√	-	105
Specific	R5 Separate consent per purpose	M (fully)	√	√	-	108
Informed	R6 Accessibility of information page	M (fully) or T (partially) together with U	-	√	-	111
	R7 Necessary information on BTT	M (fully) or T (partially)	√	√	-	111
	R8 Information on consent banner configuration	M (fully) or T (partially)	-	√	-	113
	R9 Information on the data controller	M (fully) or T (partially)	√	√	-	113
	R10 Information on rights	M (fully) or T (partially)	√	√	-	113
Unambiguous	R11 Affirmative action design	Combination of M and T (partially)	√	√	-	114
	R12 Configurable banner	M or T (partially)	-	√	L	115
	R13 Balanced choice	M (fully)	-	√	L	117
	R14 Post-consent registration	T (partially)	-	√	CS	118
	R15 Correct consent registration	Combination of M and T (partially)	-	√	CS	119
Readable and accessible	R16 Distinguishable	M (fully) or T (partially)	√	√	-	121
	R17 Intelligible	U	√	√	-	121
	R18 Accessible	U	√	√		121
	R19 Clear and plain language	U	√	√	-	121
	R20 No consent wall	M (fully) or T (partially)	-	√	L	122
Revocable	R21 Possible to change in the future	M (fully)	√	√	-	124
	R22 Delete "consent cookie" and communicate to third parties	Not possible	-	-	CS	125

Figure 8: Requirements for a valid design. [14]

These papers focused on cookie banner design, they describe the points that do not comply and which points should have to follow the law. Although, we are interested in obtaining information on websites, more specifically without interacting with the banner and after interacting with it. We want to know if there is a banner and in that case the number of cookies that added.

The paper *"Online Tracking: A 1-million-site Measurement and Analysis"* [16] consist of a large-scale analysis of web pages where the tracking methods are analyzed and elements that affect the privacy of users. The scope of the paper is on the top 1 million most visited websites in the Alexa ranking.

To do the analysis they develop a platform for website tracking analysis called OpenWPM. Using this method they visited 1 million websites. OpenWPM only visits the home web page. We will focus on the different parts that have, to understand how to develop a platform for website tracking analysis.

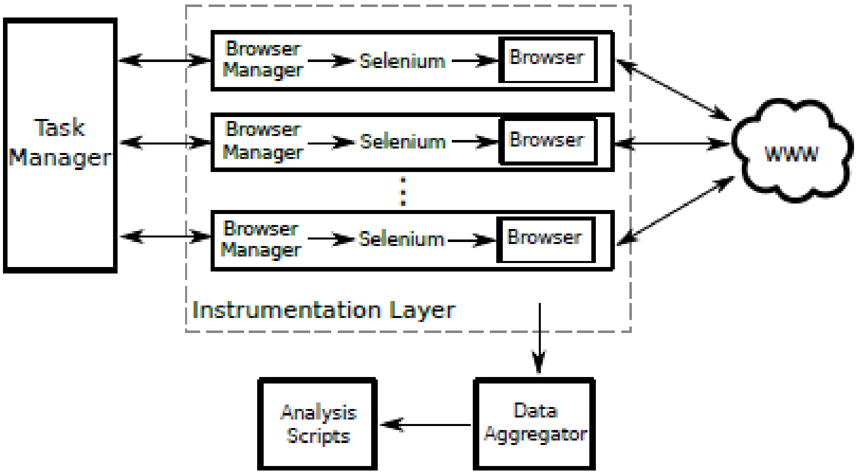


Figure 9: OpenWPM. [16]

In Figure 9 it is possible to see the core parts of OpenWPM which we will explain in the following lines. [16]

- **Browser driver:** The one at a charge to order the web browser to access the website and if needed interact with it. It uses Selenium and a full browser such as Firefox, Chrome, or IE. It allows using plugins to test their effectiveness and also provides all the features.
- **Browser managers:** Selenium is powerful however it has a really bad drawback and that is its instability. Each Browser Manager instantiates a Selenium instance with a specific configuration. It translates the commands given by the Task Manager in Selenium orders and it encapsulates each browser state. To achieve independence between them each Browser Manager is a process.
- **Task manager:** The Task Manager allows the control of the Browser Managers by providing a command-line interface. It will dispatch those commands to the Browser Managers. Each command is launched in a thread associated with a Browser Manager. Each execution thread handles the crashes of its corresponding Browser Manager, so a crash in a Browser Manager or a time out will be caught by the execution thread and it will enter in recovery mode.
- **Data Aggregator:** It is the one in charge to log the data in a standardized format to be able to share scripts and data. It has 2 data aggregators, a structured SQLite aggregator for storing the majority of measurements such as the data collected by the proxy or the extension. The other aggregator is a LevelDB aggregator that stores the HTML and JS in a de-duplicated compressed way. It checks if the hash is in the DB and if not it adds the content.
- **Instrumentation:** The Instrumentation provides data access to Raw data on disk, Network level with HTTP proxy, and at JavaScript level with a Firefox extension. So it provides all the information when the browser interacts with the web.

After these articles review, we have seen how is the state of the art related to the cookies banners, and also we analyzed a platform for website tracking.

2.5 Online Resource Mapper (ORM)

Finally, the last section is a brief description of Online Resource Mapper (ORM) [2] developed by Ismael Castell Uroz. It is the main framework that we will use in this project. In a similar way to OpenWPM, ORM also automatizes the process of opening a website and collects the resources loaded. It can collect different information like scripts, files, documents, etc., and save it into a database. ORM has different parts also, as OpenWPM, that will be explained in the following lines as details in *"TrackSign: Guided Web Tracking Discovery"* section 4 [17].

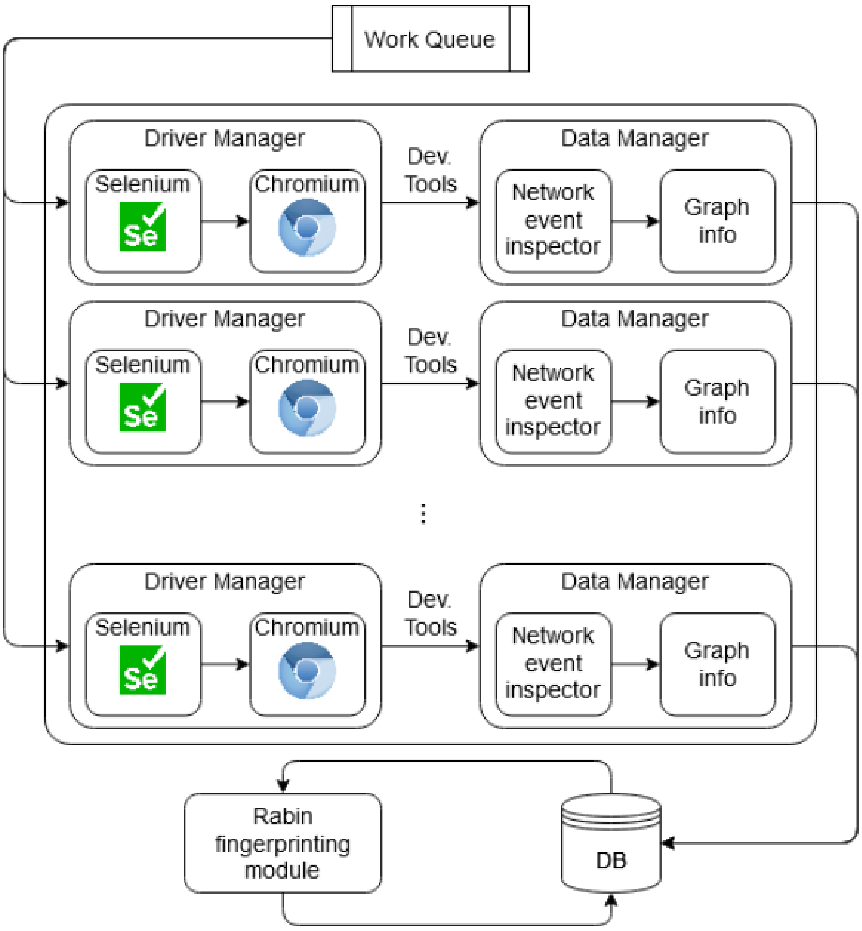


Figure 10: ORM diagram. [17]

As it is possible to see in Figure 10, ORM has de following parts, and it is the main order that will follow the process of executing ORM, determining the domains that we want to analyze from the Alexa ranking until it saves all the resources into the database to analyze later and extract results and conclusions:

- **Work queue:** First of all, we introduce several websites from the Alexa ranking. The driver manager will open all these websites following order, and they will be in the queue while the Driver Manager does not open them. Once Driver Manager opens and closes all the websites, the queue will be empty. It is possible to open as many sites as the number of threads introduced.
- **Driver manager:** It opens the websites that there are in the queue in an automated way and executes Selenium, which will interact with the website and configure what is necessary. As we can see in Figure 10, there are many Driver Managers, and each one executes an instance of Selenium. Also, it executes Chromium, the free version of Google Chrome. This web browser opens the websites and Chromium supplies also the usage of Dev. Tools protocols from Google that provides the capacity to access the internal communications and parameters that the browser has.

  Also, sometimes Selenium crashes, so it is not possible to open the desired web page. However, thanks to the Driver Manager that provides stability, it skips this website, and it executes the Selenium driver again to be able to open the next web page.
- **Data manager:** Once Driver Manager opens the web page, the Data Manager will collect and process the corresponding data and resources received from Driver Manager, and store them into the database. It saves different relations between the domain and their corresponding resources and also the URL and the resource.
- **Database:** Finally the Data manager stores all the data into the database to later we can do any type of analysis with this data and extract conclusions. It is possible to generate analysis scripts that access the database and obtain a lot of information about the different resources. Also, it is possible to use MySQL Workbench to see the saved information.

ORM is the main tool used to develop this project. The reason for the usage of ORM is because the ePrivacy Observatory uses ORM, so it will be easy to integrate the different modifications to the observatory.

3 Methodology and project development

This section contains detailed information on the project development, as well as the software and hardware that we have used and the methodology that we have followed to develop this project.

3.1 Methodology

During the process, there we have carried out meetings every two weeks. During the meeting, we presented the work done and the possible problems and we defined the work for the next meeting. So we have followed a kind of agile methodology. More or less at the begging of the process, we defined the project, during this time there have been changes.

3.2 Requirements

3.2.1 Software

We have been using the following software, described below:

- **Python [18]:** It is a programming language that in this project we use it to develop ORM and extract results. Python has different libraries and packages. Most of the different parts of the ePrivacy Observatory, the different modules that contain the observatory are programmed using python for compatibility reasons.
- **MySQL Workbench [19]:** It is a tool that provides in a visual way the management of the databases. In this project, we use it to see the stored information obtained after running ORM, and then we can see the different tables and make queries to obtain specific information.
- **Chromium [20]:** It is an open-source browser. We use it during the execution of ORM, in that process, Chromium opens each domain of the Alexa ranking and remains open for some seconds to collect the corresponding data.
- **VirtualBox [21]:** The native operating system of my laptop is Windows 10. Since one of the main requirements to run ORM is the usage of Ubuntu, I decided to use a virtual machine with Ubuntu. So VirtualBox is a useful tool to develop this project since we can run the project in another OS without installing it on the laptop.
- **Slack [22]:** During all the project we have been using Slack, that is a tool that provides the possibility to create different channels and facilitates the communication between the different members of the group. Using Slack we can send written messages, documents, pictures, etc.
- **Ubuntu [23]:** It is the operating system used to develop this project. To run ORM and install all the requirements, it is necessary to use this OS. Ubuntu is based on Linux and it is a free software.

- **Overleaf [24]:** It is a free LaTeX editor online. I used it to write the memory of the project. Furthermore, it is useful because with an account you can access your documents from any device.
- **GitHub [25]:** During all the projects we develop different codes. We use GitHub in order to generate repositories and save the codes there. Furthermore, it is an easy way to update the codes and download them if we need another code that has developed a teammate.
- **Visual Studio Code [26]:** It is a free code editor. I have been using it during the project to write and visualize the different codes developed. Also provides some useful functionalities like debugging or Git control.
- **ORM [2]:** It is the most important technology used during this project. It provides the possibility to open the domains of the Alexa ranking, analyze it and obtain useful information such as the headers that contain the cookies and save it into a database.
- **Google Meet [27]:** It provides free calls with multiple people at the same time. During these months we have been using it every two weeks to do the meetings of the group.
- **Tableau [28]:** It is a useful tool to plot data of a .csv file in a world map. This tool is used to produce results and show the values in a different form.
- **TeamViewer [29]:** It provides the possibility to control remotely another computer. During the process of collecting data to generate the results, I used another computer to obtain that information because I leave it running some days, and I controlled it remotely.

3.2.2 Hardware

There is not any specific hardware to develop this project. It is only necessary a laptop, in this case, I use my personal computer. It has the following characteristics:

- Windows 10 Pro
- **Processor:** Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz 2.30 GHz
- **RAM:** 32 GB
- **System type:** 64 bits operating system
- **Model:** MSI GS66 Stealth 10SE

3.3 Cookies study

In this section, we want to analyze web behavior in different situations. As we can see in Figure 11, the situations are ignoring cookies, accepting cookies, and rejecting cookies. Also, we will determine the country of each domain, to see if the domain is an EU country or not. All the domains, independently of the country, must comply with GDPR law. However, the EU countries will visit more frequently EU domains.

To accept cookies, we will follow two different approaches. We will use Selenium to search a keyword and click the button. The other approach is the usage of Computer Vision to search the button.

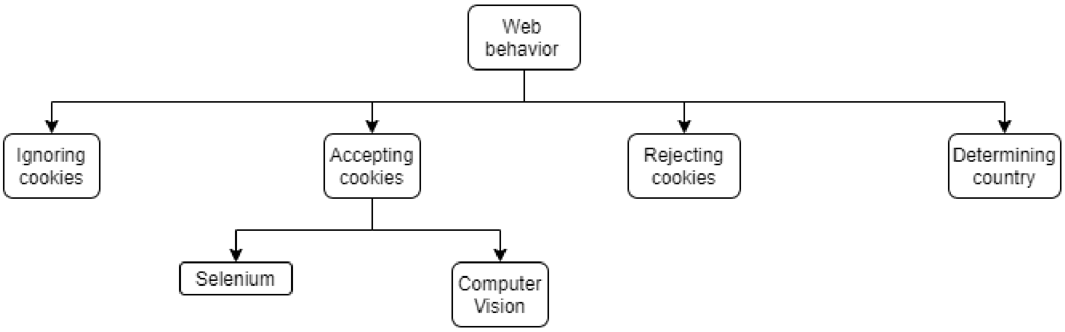


Figure 11: Global view of cookies study.

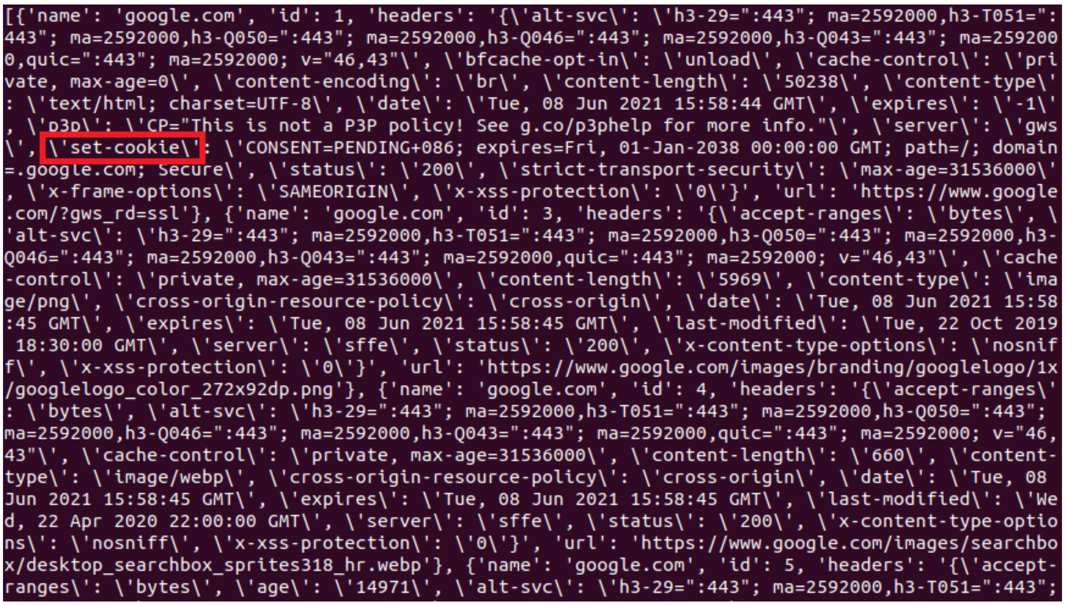
We describe in detail all the different approaches in the following sections to obtain the number of cookies in the different situations, depending on the behavior of the web.

3.3.1 Ignoring cookies

In the "State of the art" section, we explain in detail how the ORM works and the different parts of ORM. In this section, we are focused on its use.

As we have explained, ORM opens web pages in an automated way and saves different data into a database. In this project, we are focused on determining if websites comply with GDPR and if they ask for the consent of their users before storing data and the management of cookies.

So we run ORM, we collect data from the top Alexa ranking, and after that we extract results. To detect the presence of cookies we analyzed the HTTP headers collected searching for the "Set-Cookie" and "set-cookie" header as we can see in the red square in Figure 12.



```
[{'name': 'google.com', 'id': 1, 'headers': '{\'alt-svc\': \'h3-29=":443"; ma=2592000,h3-T051=":
443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=259200
0,quic=":443"; ma=2592000; v="46,43"\', \'bfcache-opt-in\': \'unload\', \'cache-control\': \'pri
vate, max-age=0\', \'content-encoding\': \'br\', \'content-length\': \'50238\', \'content-type\'
: \'text/html; charset=UTF-8\', \'date\': \'Tue, 08 Jun 2021 15:58:44 GMT\', \'expires\': \'-1\'
, \'p3p\': \'CP="This is not a P3P policy! See g.co/p3phelp for more info."\', \'server\': \'gws
\', \'set-cookie\': \'CONSENT=PENDING+086; expires=Fri, 01-Jan-2038 00:00:00 GMT; path=/; domain
=.google.com; Secure\', \'status\': \'200\', \'strict-transport-security\': \'max-age=31536000\'
, \'x-frame-options\': \'SAMEORIGIN\', \'x-xss-protection\': \'0\'}', 'url': 'https://www.google
.com/?gws_rd=ssl'}, {'name': 'google.com', 'id': 3, 'headers': '{\'accept-ranges\': \'bytes\', \
'alt-svc\': \'h3-29=":443"; ma=2592000,h3-T051=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-
Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"\', \'cache
-control\': \'private, max-age=31536000\', \'content-length\': \'5969\', \'content-type\': \'ima
ge/png\', \'cross-origin-resource-policy\': \'cross-origin\', \'date\': \'Tue, 08 Jun 2021 15:58
:45 GMT\', \'expires\': \'Tue, 08 Jun 2021 15:58:45 GMT\', \'last-modified\': \'Tue, 22 Oct 2019
 18:30:00 GMT\', \'server\': \'sffe\', \'status\': \'200\', \'x-content-type-options\': \'nosnif
f\', \'x-xss-protection\': \'0\'}', 'url': 'https://www.google.com/images/branding/googlelogo/1x
/googlelogo_color_272x92dp.png'}, {'name': 'google.com', 'id': 4, 'headers': '{\'accept-ranges\'
: \'bytes\', \'alt-svc\': \'h3-29=":443"; ma=2592000,h3-T051=":443"; ma=2592000,h3-Q050=":443";
ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,
43"\', \'cache-control\': \'private, max-age=31536000\', \'content-length\': \'660\', \'content-
type\': \'image/webp\', \'cross-origin-resource-policy\': \'cross-origin\', \'date\': \'Tue, 08
Jun 2021 15:58:45 GMT\', \'expires\': \'Tue, 08 Jun 2021 15:58:45 GMT\', \'last-modified\': \'We
d, 22 Apr 2020 22:00:00 GMT\', \'server\': \'sffe\', \'status\': \'200\', \'x-content-type-optio
ns\': \'nosniff\', \'x-xss-protection\': \'0\'}', 'url': 'https://www.google.com/images/searchbo
x/desktop_searchbox_sprites318_hr.webp'}, {'name': 'google.com', 'id': 5, 'headers': '{\'accept-
ranges\': \'bytes\', \'age\': \'14971\', \'alt-svc\': \'h3-29=":443"; ma=2592000,h3-T051=":443";
```

Figure 12: Header with set-cookie.

As a result, we detect that a large number of web pages do not comply with cookies rules, because they are adding cookies before that the user has had any interaction with the cookie banner.

3.3.2 Accepting cookies

Once basic ORM information is collected, now we want to interact with the website, doing click on the "Accept cookies" button and compare it with the case without doing any interacting. Supposedly it should be more cookies after accepting it.

We will follow two different approaches, the first one is the use of Selenium to iterate through iframes and search-specific keyboards. The second approach is the use of Computer Vision developed by Pablo Fonoll Soto.

3.3.2.1 Modified ORM using Selenium

In order to interact with the website and be able to click on the button, we will use Selenium to emulate a user in a web browser. We developed a Selenium script that is integrated in ORM which will click the accept button [31].

As we can see in Figure 13, the Driver Manager will open the websites in an automated way and executes Selenium. After that, ORM will wait 10 seconds to load the website, and then ORM executes the new script that will click the "accept cookies" button. This script searches keyboards. If it finds a word, it will click the button and will wait 10 seconds more. Then the process is the same as non-modified ORM, the Data Manager will save data and resources into the database.

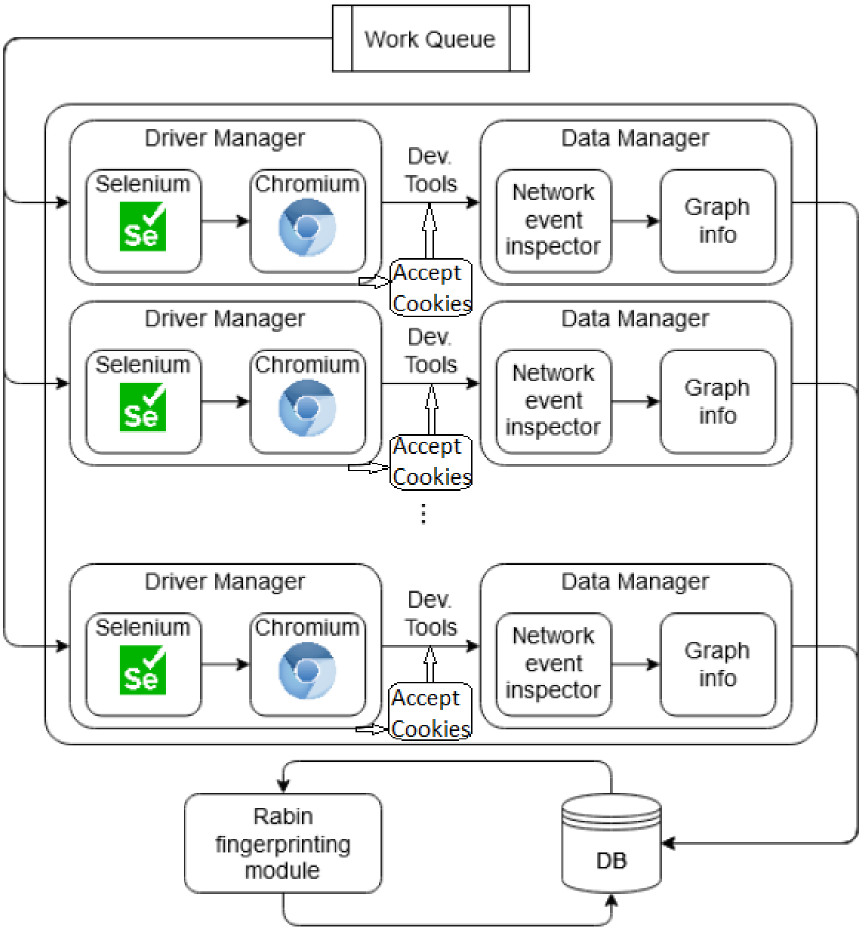


Figure 13: Modified ORM using Selenium.

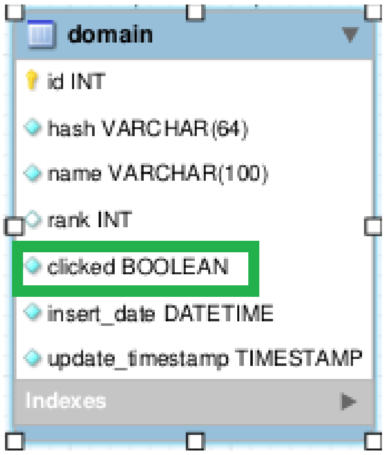
To determine which button has to click, we inspect some websites manually and we make a list of all the possible words that means "accept cookies" in different languages. The list that we elaborated is the following one:

- enable all
- accept
- accept all
- got it!
- acepto
- Yes, I agree
- accept cookies
- aceptar cookies
- consent

- I accept
- agree
- aceptar todas las cookies
- cookies akzeptieren
- agree & continue
- prosseguir
- accept all cookies
- I agree
- OK
- Yes, I'm happy

It iterates through all iframes searching for the specified keywords in multiple languages. To do so, we use the Selenium method find_elements_by_xpath(). This method will return a list with all the elements that contain the keywords, that we have specified. Then we inspect if the element is a button type. If found it will click the button executing the "click()" method. To do so, the driver_manager.py file is modified.

Also, it is necessary to modify the database model. The basic ORM has different tables. One table is called domain, which contains information about the domain name, domain id, and other parameters. In this table we will do a modification, the "clicked" field of type boolean will be added as we can see in Figure 14. So if the button is clicked this field will be true, otherwise will be false.



domain

id INT
hash VARCHAR(64)
name VARCHAR(100)
rank INT
clicked BOOLEAN
insert_date DATETIME
update_timestamp TIMESTAMP

Figure 14: Modified domain table.

During the process, we found several problems. Very few websites respect the HTML good practices, and we found lots of supposed "buttons" that were Span, Buttons, Anchors, Input, etc. Also, another problem that we had was that the "click()" method sometimes was not working on some found elements because it was the parent or grandparent the one listening. The last problem that we found was that there are cookie messages inside iframes.

Developing this script was a difficult task because we detected some problems as described and not always work well. Although the results are good and it was possible to click the button in most cases.

Also, another objective is to click the ”reject cookies” button, however, we realize that it is a hard task because usually this button is not on the main page and it is necessary more interaction. Furthermore, the reject cookies option is different on most web pages. For this reason, we decided to use a plugin, that could do the work of ”reject cookies”.

3.3.2.2 Modified ORM using Computer Vision

Finally, the last method to accept cookies is the usage of computer vision. Pablo Fonoll Soto developed it, and we will integrate this method with ORM to obtain the cookies results and compare them with the other methods. In this section, it is explained in more detail how it works the computer vision method to click the ”accept cookies” button.

With computer vision, he generates the labeled dataset which he uses to train the neural network model. The dataset is about 2500 images. This neural network is the one used to identify the accept button. If it is not able to identify it, the same image is evaluated again with computer vision techniques that helped to create the labeled dataset.

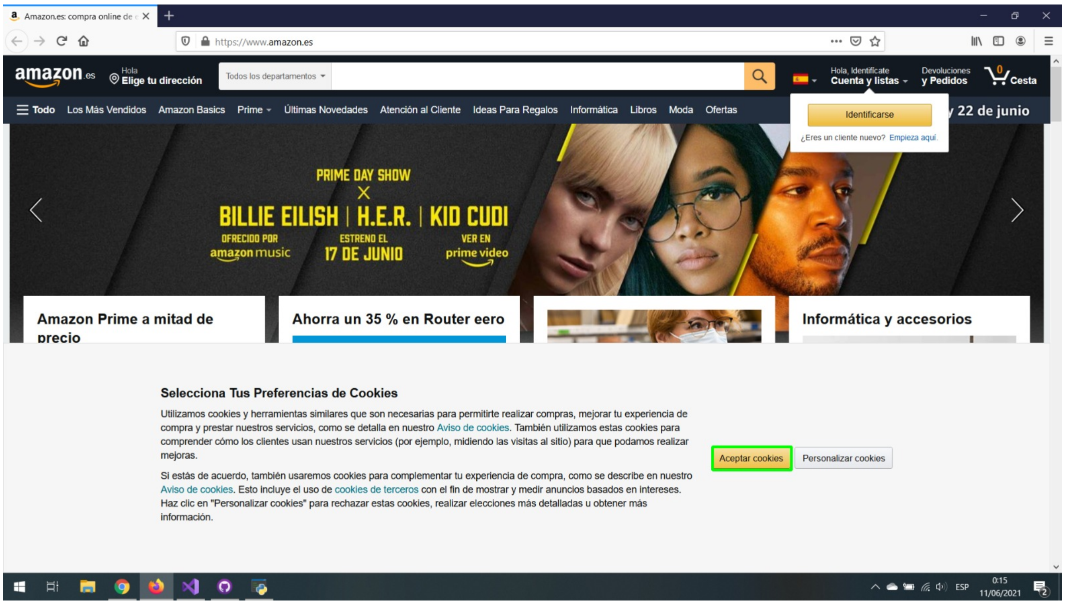
In the computer vision part, he applied two techniques: OCR and Canny's curve detection algorithm. OCR is Optical Character Recognition. It recognizes and locates the texts of the images. It looks for keywords usually found inside the accept cookies button, such as the words ”OK” and ”Close”. This OCR passes it over to different versions of the image, processed differently to facilitate text recognition. Once the keywords are found, to avoid fake candidates, it is analyzed if these words are contained in a button, this is possible thanks to Canny's algorithm, which detects shapes within an image and goes very well to detect the buttons as they usually stand out over the rest of the text.

The Computer Vision part is not 100% effective so, to create the dataset, he performed a manual review to check and correct the erroneous cases.

As we can see in Figure 15 and Figure 16, the Computer Vision technique works properly and detect the ”Accept cookies” button. In this case, to indicate that it has detected the button, it makes a green square around the button.

On the other hand, in Figure 17, the button is not well detected, as we can see the horizontal rectangle is a little longer. That is because the neural network has been trained to see many buttons and tries to classify the case it has to analyze from all the ones it has seen. Because not in all cases the button has the same size, sometimes it does not do 100% well. In this case, many cases will influence the button detection. He has seen previous examples with longer buttons.

The important point in the three Figures is that the center of the painted rectangle is inside the button because will be the clicked point.



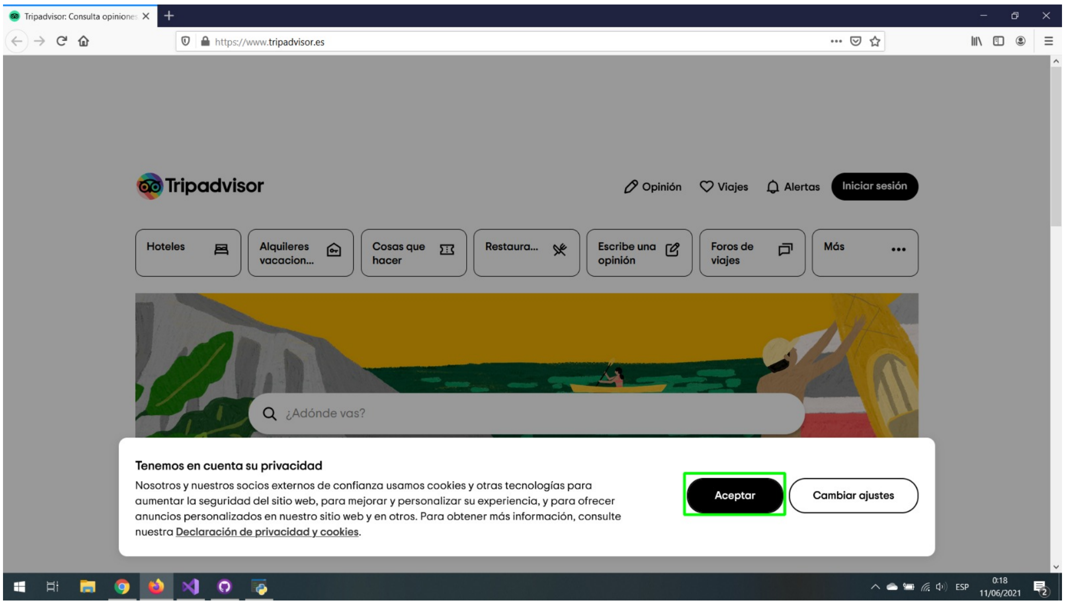
Selecciona Tus Preferencias de Cookies

Utilizamos cookies y herramientas similares que son necesarias para permitirte realizar compras, mejorar tu experiencia de compra y prestar nuestros servicios, como se detalla en nuestro Aviso de cookies. También utilizamos estas cookies para comprender cómo los clientes usan nuestros servicios (por ejemplo, midiendo las visitas al sitio) para que podamos realizar mejoras.

Si estás de acuerdo, también usaremos cookies para complementar tu experiencia de compra, como se describe en nuestro Aviso de cookies. Esto incluye el uso de cookies de terceros con el fin de mostrar y medir anuncios basados en intereses. Haz clic en "Personalizar cookies" para rechazar estas cookies, realizar elecciones más detalladas u obtener más información.

Aceptar cookies | Personalizar cookies

Figure 15: Detecting the button Amazon.

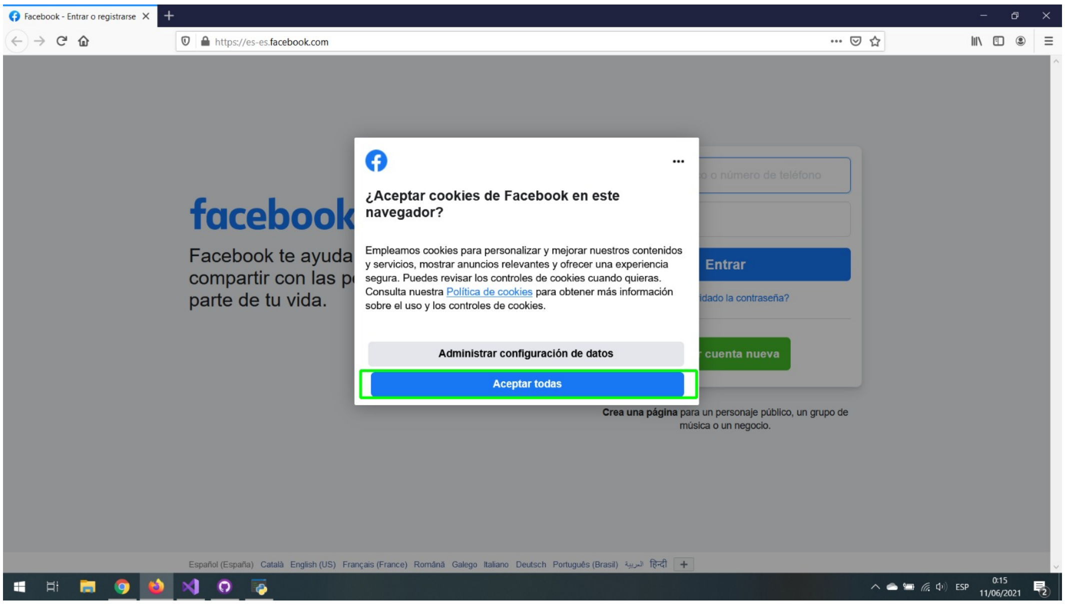


Tenemos en cuenta su privacidad

Nosotros y nuestros socios externos de confianza usamos cookies y otras tecnologías para aumentar la seguridad del sitio web, para mejorar y personalizar su experiencia, y para ofrecer anuncios personalizados en nuestro sitio web y en otros. Para obtener más información, consulte nuestra Declaración de privacidad y cookies.

Aceptar | Cambiar ajustes

Figure 16: Detecting the button Tripadvisor.


Facebook - Entrar o registrarse
https://es-es.facebook.com
facebook
Facebook te ayuda
compartir con las p
parte de tu vida.
o número de teléfono
Entrar
idado la contraseña?
cuenta nueva
Administrar configuración de datos
Aceptar todas
Español (España) Català English (US) Français (France) Română Galego Italiano Deutsch Português (Brasil) العربية हिन्दी
ESP
0:15
11/06/2021

¿Aceptar cookies de Facebook en este navegador?

Empleamos cookies para personalizar y mejorar nuestros contenidos y servicios, mostrar anuncios relevantes y ofrecer una experiencia segura. Puedes revisar los controles de cookies cuando quieras. Consulta nuestra Política de cookies para obtener más información sobre el uso y los controles de cookies.

Crea una página para un personaje público, un grupo de música o un negocio.

Figure 17: Detecting the button Facebook.

3.3.3 Rejecting cookies

In this section, we try different plugins to interact with websites and accept cookies and reject them. We try 2 different plugins, ”Ninja Cookie” [1] plugin to reject cookies and also we try ”I don’t care about cookies” [3]. This second one could be another option to accept cookies.

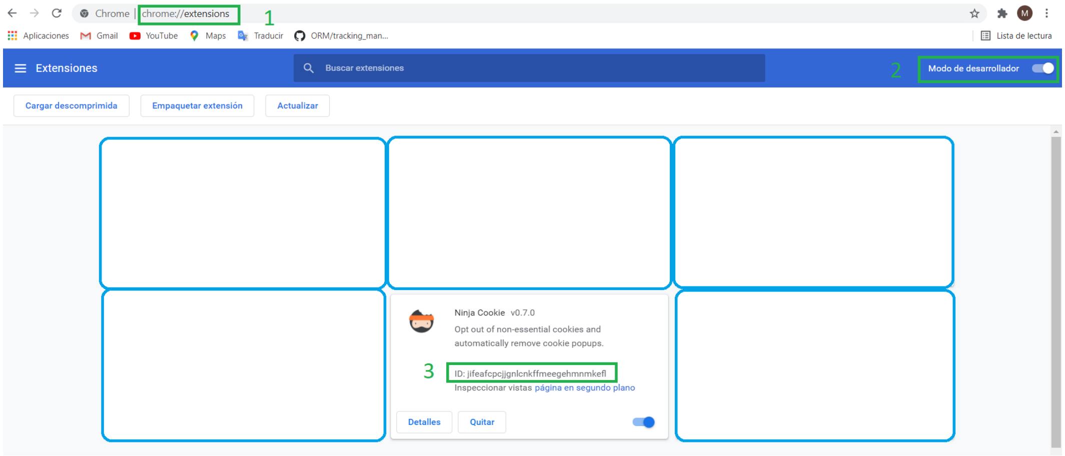
3.3.3.1 Adding Ninja Cookie plugin to ORM

Ninja Cookie plugin [1] will be used to decline cookies on web pages. Basically, Ninja Cookie will try to block all the possible cookies that could be added from banners. Furthermore, it is a useful tool because if a user adds this plugin to the browser, the block function is done automatically and there is no need to interact with the cookie banner. So the user can save time navigating websites and furthermore it is not giving consent to all cookies.

To add this plugin in ORM, the process is the following:

- First of all we install Ninja Cookie plugin in a web browser such as Chrome.
- Then we open the extension panel writing the following direction ”chrome://extensions”.
- Once in the page, we activate developer mode in the top right.
- We identify the ID of the extension.

In Figure 18 is possible to see these steps.



Ninja Cookie v0.7.0

Opt out of non-essential cookies and automatically remove cookie popups.

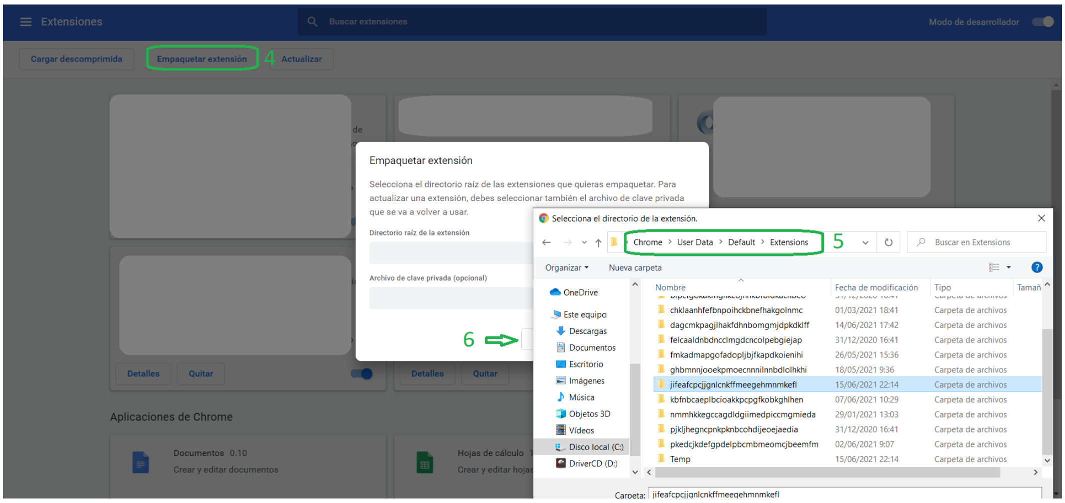
ID: jifeafcpcjjgnlcnkffmeegehmnmkefl

Inspeccionar vistas página en segundo plano

Figure 18: Ninja Cookie Chrome extension.

- The next step is click over the button ”packet extension”.
- Then we click over inspect and we search the mentioned ID in step 3 in the following route "`C:\Users\UserName\AppData\Local\Google\Chrome\UserData\Default\Extensions\`"
- Finally we click the button ”pack extension”.

In Figure 19 is possible to see the last steps.



Empaquetar extensión

Selecciona el directorio raíz de las extensiones que quieras empaquetar. Para actualizar una extensión, debes seleccionar también el archivo de clave privada que se va a volver a usar.

Directorio raíz de la extensión

Archivo de clave privada (opcional)

Selecciona el directorio de la extensión.

Nombre	Fecha de modificación	Tipo	Tamañ
chklaanhfefbnpoihckbnefhakgolnmc	01/03/2021 18:41	Carpeta de archivos	
dagcmkpagjlhakfdhnbomgmjdpkdklff	14/06/2021 17:42	Carpeta de archivos	
felcaaldnbdncclmgdcncolpebgiejap	31/12/2020 16:41	Carpeta de archivos	
fmkadmapgofadopljbjfkapdkoienihi	26/05/2021 15:36	Carpeta de archivos	
ghbmnnjooekpmoecnnnilnnbdlolhkhi	18/05/2021 9:36	Carpeta de archivos	
jifeafcpcjjgnlcnkffmeegehmnmkefl	15/06/2021 22:14	Carpeta de archivos	
kbfnbcaeplbcioakkpcpgfkobkghlhen	07/06/2021 10:29	Carpeta de archivos	
nmmhkkegccagdldgiimedpiccmgmieda	29/01/2021 13:03	Carpeta de archivos	
pjkljhegncpnkpknbcohdijeoejaedia	31/12/2020 16:41	Carpeta de archivos	
pkedcjkdefgpdelpbcmbmeomcjbeemfm	02/06/2021 9:07	Carpeta de archivos	
Temp	15/06/2021 22:14	Carpeta de archivos	

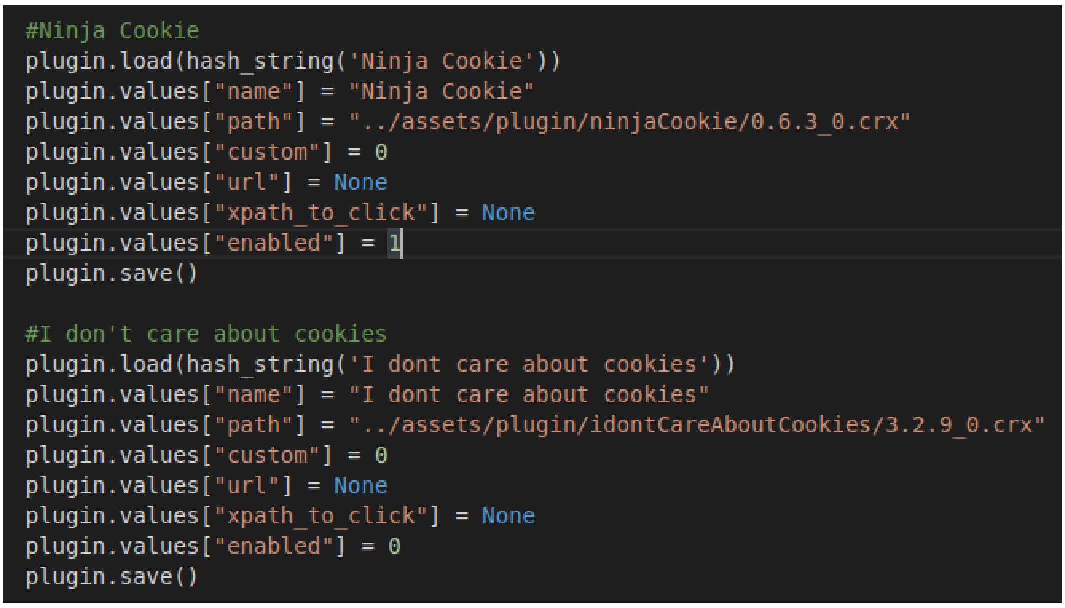
Carpeta: jifeafcpcjjgnlcnkffmeegehmnmkefl

Figure 19: Ninja Cookie Chrome extension last steps.

Inside the folder "`C:\Users\UserName\AppData\Local\Google\Chrome\UserData\Default\Extensions\pluginID\`" two new files are created, *.crx file and *.pem file. We copy these two files inside the ORM folder `ORM\assets\plugin\ninjaCookie\`.

Once all these steps are done, some modifications to the ORM code are necessary.

First of all, we modify db_initializer.py, adding the path to the plugin and enabling it. So when the database will be initialized, this plugin will be enabled. Then, while the ORM will be running, also the plugin will appear in each website blocking the cookies.



```
#Ninja Cookie
plugin.load(hash_string('Ninja Cookie'))
plugin.values["name"] = "Ninja Cookie"
plugin.values["path"] = "../assets/plugin/ninjaCookie/0.6.3_0.crx"
plugin.values["custom"] = 0
plugin.values["url"] = None
plugin.values["xpath_to_click"] = None
plugin.values["enabled"] = 1
plugin.save()

#I don't care about cookies
plugin.load(hash_string('I dont care about cookies'))
plugin.values["name"] = "I dont care about cookies"
plugin.values["path"] = "../assets/plugin/idontCareAboutCookies/3.2.9_0.crx"
plugin.values["custom"] = 0
plugin.values["url"] = None
plugin.values["xpath_to_click"] = None
plugin.values["enabled"] = 0
plugin.save()
```

Figure 20: Adding Ninja Cookie plugin to ORM.

Also, it is necessary to create a new database in order to save the values collected in this case in order to compare it with default ORM and ORM that click the button to accept cookies. So we will modify the config.py file and change the MYSQL_DB name to ORMninja.

In MySQLWorkbench we have to open the corresponding model to save the data.

After obtaining the results, in very few cases, we detect something that is not normal, that the number of cookies has increased instead of decreasing compared with the number of cookies in the default ORM. We inspect the headers of these websites manually, to see the "set-cookie" value. Some examples of websites that we have detected this problem are:

- amazon.it
- oracle.com
- marketwatch.com
- cnn.com
- alwafd.news
- amazon.com
- intuit.com

After analyzing it manually, we detect that in all of these websites the value "set-cookie" is repeated. So we can conclude that the Ninja Cookie plugin is working properly, however, we deduce that it is blocking many times and for this reason the "set-cookie" value is repeated.

3.3.3.2 Adding I don't care about cookies plugin to ORM

We use another plugin, called "I don't care about cookies" [3] to have another method to accept cookies. Basically, the function of this plugin is to hide the cookie banner and accept cookies when it is necessary. So it could be interesting to compare this method with the one that we have developed to click the "accept cookies" button.

In this case, we will follow the same procedure as Ninja Cookie plugin, installing the plugin in a web browser, then generating the *.crx and *.pem files and finally adding these files into a folder `ORM\assets\plugin\idontCareAboutCookies\`.

Also, some modifications of the code are necessary. In Figure 20 we can see that under the Ninja Cookie modifications, are the corresponding modifications for the "I don't care about cookies" plugin. In this case we add the corresponding path and we desable "Ninja Cookie" plugin and we enabled "I don't care about cookies" plugin.

It is necessary to create a new database in order to save the values collected in this case in order to compare it with default ORM and ORM that click the button to accept cookies. So we will modify the config.py file and change the MYSQL_DB name to ORMdontCare.

In MySQLWorkbench we have to open the corresponding model to save the data.

Supposedly, we have used this plugin to do the function of accepting cookies. However, we spotted something we did not like. We thought that more cookies should be added, so should be something similar to the case of clicking the "accept cookies" button. Nevertheless, in some cases, more cookies are added compared with the number of the default ORM cookies, but in other cases, the number of cookies is lower than the default ORM case.

We can conclude that the result is not the ones that we were expecting, for this reason, we will not use this plugin to generate results, because sometimes add more cookies but sometimes it blocks cookies, and it is not possible to get a clear behavior of this plugin.

3.3.4 Determining the country of each domain

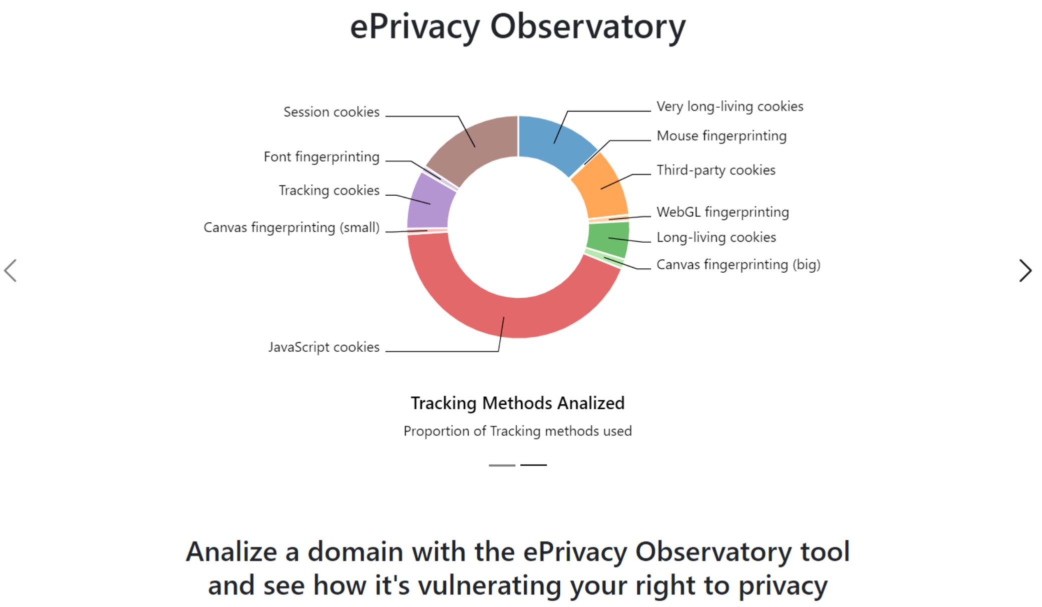
After obtaining different methods to interact with websites and obtain the number of cookies in each case, we inspect also the country of each domain. We detect that there are different countries for every domain, this is because there are subdomains. Furthermore, we analyze the countries of each domain and verify which country corresponds to the main domain, called the first party, and the other domains that are third parties.

To obtain the country values of each domain, first of all, we run ORM for the desired number of websites that are in the Alexa ranking. After that, we execute geo_checker.py that will load the geolocation database and using the ”remote IP address” value from the URL table, will check the location. We save the results in a table called location, which contains the information of country code, is EU, latitude, longitude, etc. for each URL.

3.4 ePrivacy Observatory

The last step is to integrate all the results in the ePrivacy Observatory [12]. As we mention in the introduction, different modules integrate ePrivacy Observatory, which provides the users the possibility to see different information about how any webpage is vulnerating the right to privacy.

In the first image of the ePrivacy Observatory is possible to see some statistics related to tracking methods, the percentage of domains using tracking, etc. In Figure 21 it is possible to see an example of the statistics.



ePrivacy Observatory

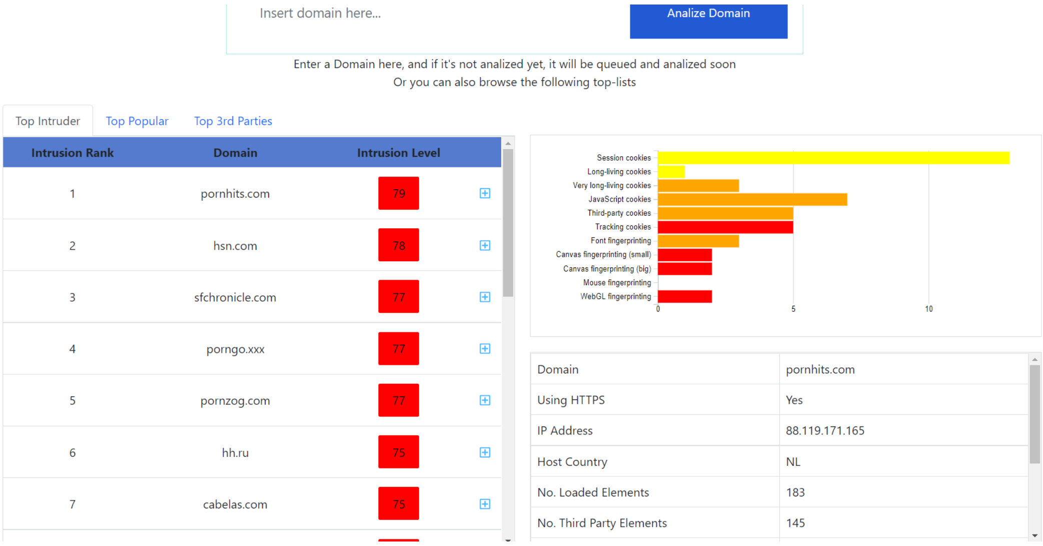
Tracking Methods Analized

Proportion of Tracking methods used

Analize a domain with the ePrivacy Observatory tool and see how it's vulnerating your right to privacy

Figure 21: ePrivacy Observatory statistics. [12]

In Figure 22 is possible to see a list with the top intrusion level on the left side. On the right side, specifically for a domain, in this case, ”pornhists.com” it is possible to see the different tracking methods. We can see HTTP cookies and JS cookies, canvas fingerprinting, font fingerprinting, mouse fingerprinting, and webGL fingerprinting. Specifically, this domain has all the types of tracking methods except mouse fingerprinting. So it is a useful tool to see which type of tracking methods are used in each domain. Also, it is possible to analyze a domain on demand.



Insert domain here... | Analize Domain

Enter a Domain here, and if it's not analized yet, it will be queued and analized soon
Or you can also browse the following top-lists

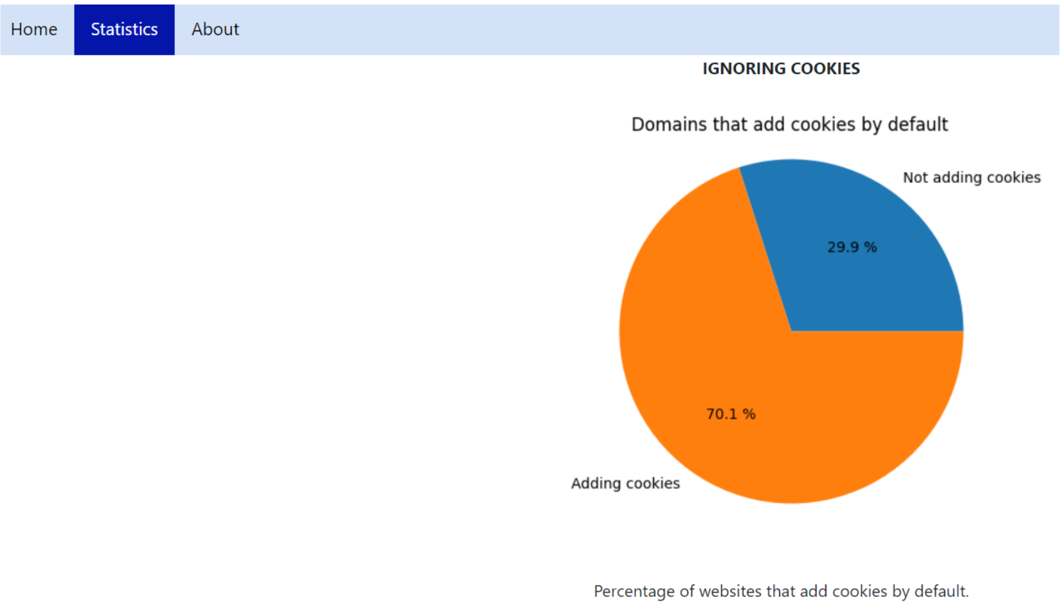
Top Intruder | Top Popular | Top 3rd Parties

Intrusion Rank	Domain	Intrusion Level
1	pornhits.com	79
2	hsn.com	78
3	sfchronicle.com	77
4	porngo.xxx	77
5	pornzog.com	77
6	hh.ru	75
7	cabelas.com	75

Domain	pornhits.com
Using HTTPS	Yes
IP Address	88.119.171.165
Host Country	NL
No. Loaded Elements	183
No. Third Party Elements	145

Figure 22: ePrivacy Observatory modules. [12]

The cookie statistics generated after interacting with websites are added to a new page, as we can find the results for ignoring cookies, accepting cookies, and rejecting cookies, as we can see in Figure 23 the statistics for ignoring cookies.



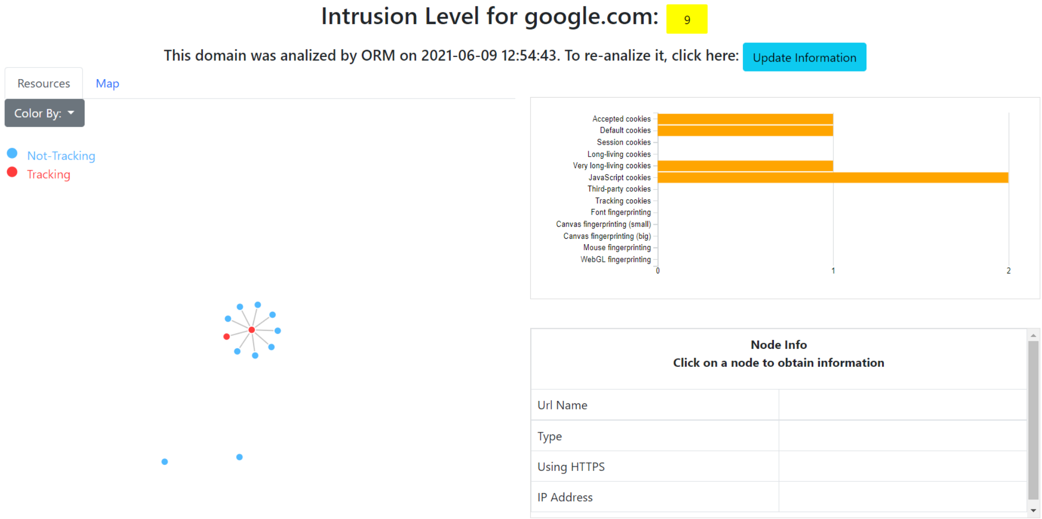
IGNORING COOKIES

Domains that add cookies by default

Percentage of websites that add cookies by default.

Figure 23: ePrivacy Observatory statistics. [12]

Also, another application of cookies study is the possibility to see in each domain the number of cookies by default and the number of cookies after accepting it. As we can see in Figure 24, there are two values on the right side list, that are called ”Accepted cookies” and ”Default cookies”, showing that Google before accepting cookies is adding one cookie by default and after that has one cookie too.



Intrusion Level for google.com: 9

This domain was analized by ORM on 2021-06-09 12:54:43. To re-analize it, click here: Update Information

Node Info
Click on a node to obtain information

Url Name	
Type	
Using HTTPS	
IP Address	

Figure 24: ePrivacy Observatory Google statistics. [12]

4 Results

In this section, we present the corresponding results after analyzing the different methods that interact with websites to see what is happening with cookies. To obtain the data, after running ORM or the modified ORM, we store the data in the database. Then we do some queries into the database to obtain the desired information [30]. We inspect HTTP headers to obtain the number of cookies that correspond to each domain, and we make the graphs that we will see below.

We generate the results after analyzing 10000 domains. We take a subset of 500 domains to see how the results changed if we used Computer Vision. The data collection was from the 24th of May of 2021 until the 30th of May of 2021 and from the 7th of July of 2021 until the 17th of July of 2021.

4.1 Cookies by default

In section 3, we inspect what happens in the case of ”ignoring cookies”, which means run ORM without any modification. After that, we inspect the number of cookies added by default. As we can see in Figure 25, 70 % of websites are adding cookies by default.

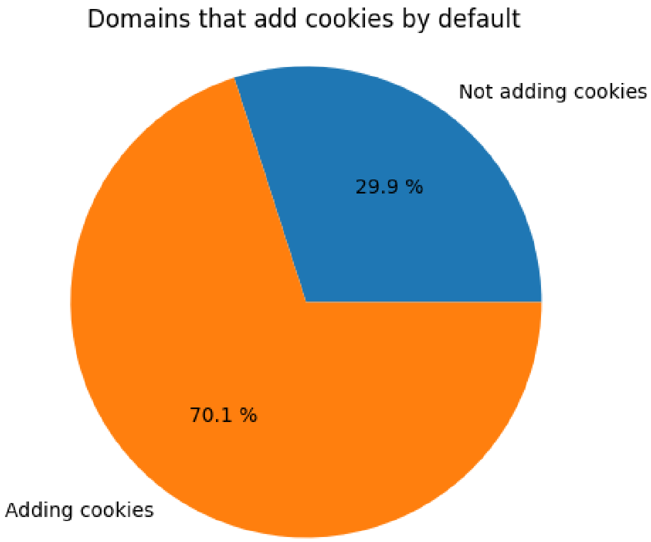


Figure 25: Cookies by default.

If we compare Figure 25 and Figure 26, the first one is the 500 domains analysis, and the second one is the 10000 domains study. Both of them have 70% of the websites that add cookies by default.

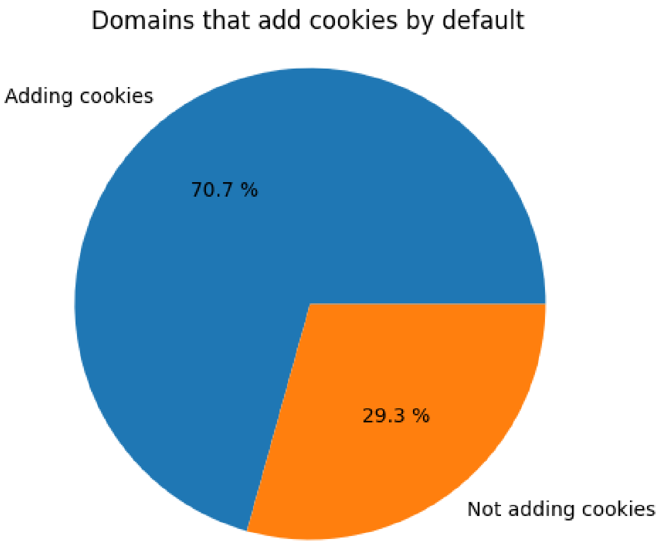


Figure 26: Cookies by default 10000 domains.

4.2 Pages it was possible to click on

The first result obtained in the case of accepting cookies is the percentage of web pages that it was possible to click on the button ”Accept cookies”, for the case of the modified ORM using Selenium to iterate in the iframes and search the keywords.

We can see that it was only possible to click the ”accept button” on 39.2% of the sites. So lots of websites have not a banner asking EU users for consent to store their data.

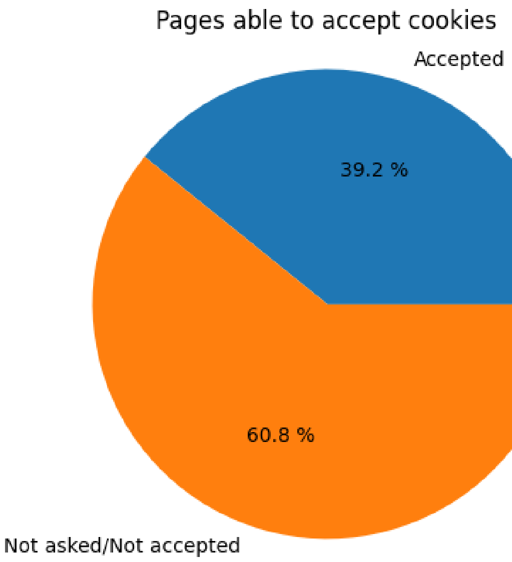


Figure 27: Pages able to accept cookies using modified ORM.

In Figure 28 we can see that it is the same type of graph. The main difference is that the results of this graph are generated after executing ORM with Computer Vision. We can see that this technique only was able to click the "accept cookies" button on 19.8% of the websites. As same as before, the number of web pages that were not possible to accept cookies is bigger than the number of possibilities to accept.

Furthermore, as it is mentioned in section 3.8, Computer Vision is not 100% effective so we can consider that this approach has not detected all the "accept cookies" buttons.

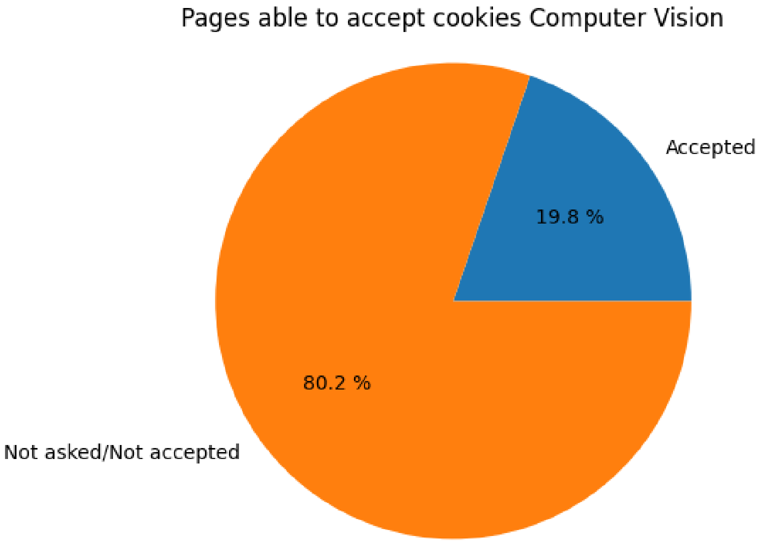


Figure 28: Pages able to accept cookies using CV.

Finally, in Figure 29 we can see the percentage of websites that were possible to accept cookies for 10000 domains. Comparing with Figure 27, the percentage for the 10000 domains analysis is lower but higher than 30%. So we can conclude that with a higher domains analysis, the percentage of acceptance decrease.

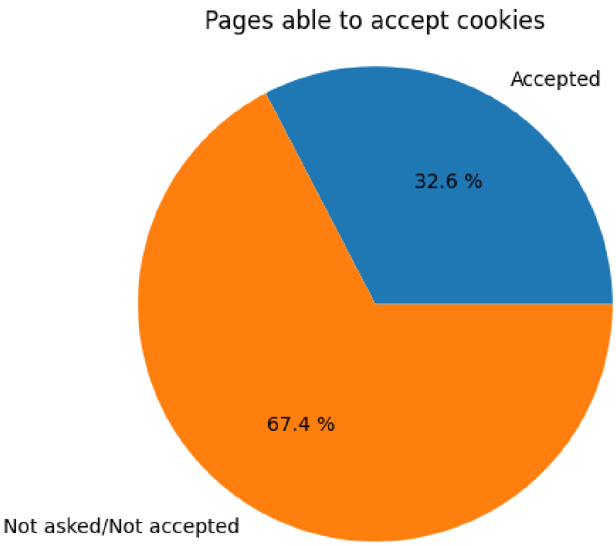


Figure 29: Pages able to accept cookies using modified ORM for 10000 domains.

4.3 Percentage of success with modified ORM

After analyzing the percentage of websites that it was possible to click on the "Accept cookies" button in both methods, the Selenium one and the Computer Vision one, we analyze the percentage of success of these methods.

In this section manual verifications are done. On one hand, a list of 100 random sites is done. Then we check manually the site, so the site is opened in a web browser and is inspected if the cookie banner exists with a button to accept cookies.

Then in the database, in the domain table, it is verified if the field clicked is a "True" or "False". If it is true, and on the website was possible to click on the button, we call this True Positive (TP). If the database has a False, and it was not possible to click the button, because there was not any banner neither the button, then is a True Negative (TN).

There are two more possibilities, False Negative (FN) and False Positive (FP). The first one is given when there is not any button to accept cookies and the modified ORM clicks something and saves a True into the database. The second one is given when there is a button to click and the modified ORM does not click anything and save into the database a False.

In Table 1 is possible to see an example of a table with all the cases. [15]

Table 1: Percentage of success with modified ORM, case Selenium.

	Yes ORM Click	**No ORM Click**
Yes Real Click	TP	FP
No Real Click	FN	TN

After all this detailed explanation, in Table 2 and Table 3 it is possible to see the corresponding results of True Positives, True Negatives, False Positives and False Negatives.

With the values of this tables it is possible to calculate the following parameters [15]:

- **Accuracy:** It is the number of positive predictions that were correct.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision:** It is the percentage of positive cases detected.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall:** It is the proportion of positive cases that were correctly identified by the algorithm.

$$Recall = \frac{TP}{TP + FN}$$

In Table 2 we can see the corresponding results for modified ORM using Selenium to click the ”accept cookies” button. Also, the following parameters are calculated:

- Accuracy = 0,78
- Precision = 0,783
- Recall = 0,839

Table 2: Percentage of success with modified ORM, case Selenium.

	Yes ORM Click	**No ORM Click**
Yes Real Click	47	13
No Real Click	9	31

In Table 3 we can see the corresponding results for modified ORM using Computer Vision to click the ”accept cookies” button. Also, the following parameters are calculated:

- Accuracy = 0,61
- Precision = 0,45
- Recall = 0,81

Table 3: Percentage of success with modified ORM, case Computer Vision.

	Yes ORM Click	No ORM Click
Yes Real Click	27	33
No Real Click	6	34

We can conclude that comparing the values of accuracy, precision, and recall, modified ORM with Selenium is better in all cases. So this answer why in Section 4.1. the percentage of accepted cookies in websites was lower for the Computer Vision case than the Selenium case.

Also, it is possible to see that both methods have not high precision, so the percentage of accepted cookies in Section 4.1 should be higher in both cases.

4.4 Domains with the same cookies

Once the domains that it was possible to click are determined, in the modified ORM with Selenium is 39.2% and modified ORM with Computer Vision is 19.8%, we determine if the cookies after clicking the button are the same or more.

We can see in Figure 30 that only 29.1% of the websites were added more cookies. However, 70.9% had the same number of cookies. So there are lots of cookies by default and the accept button is not doing anything.

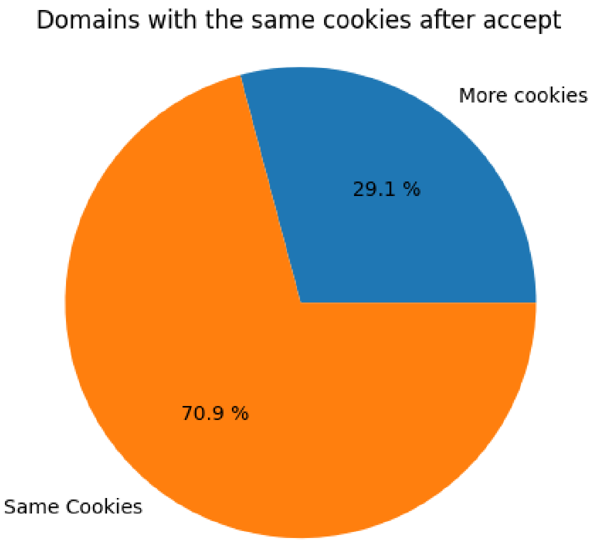


Figure 30: Domains with the same cookies after accepting.

Comparing Figure 30 and Figure 31, analyzing 10000 domains increase the percentage of added cookies, although the percentage of cookies that remain equal and do not add cookies is very high.

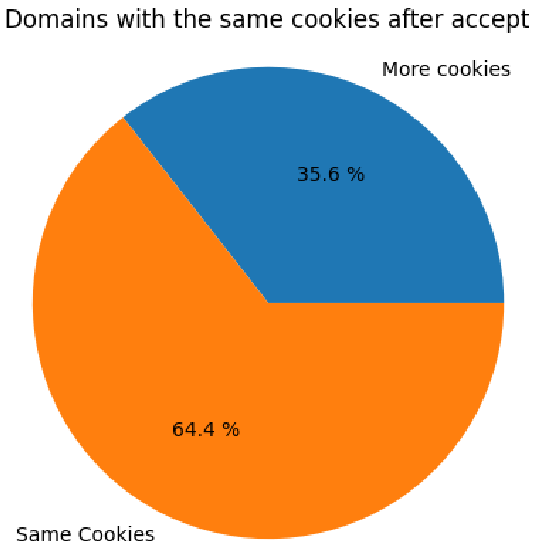


Figure 31: Domains with the same cookies after accepting 10000 domains.

On the other hand, in the case of modified ORM with Computer Vision, it is possible to see in Figure 32 that after clicking the accept cookies button, 73.4% of websites had added more cookies. It is important to remember that with this technique it was not possible to click the button in most websites, however, the ones that it was possible to click the button are introducing more cookies.

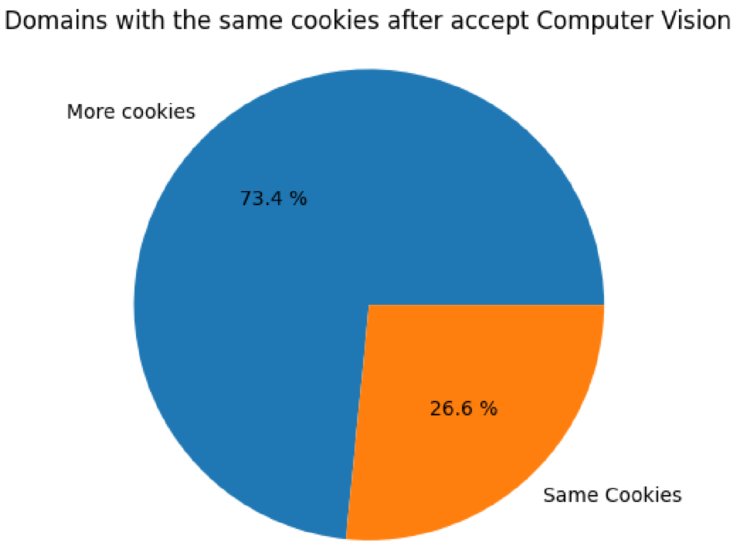


Figure 32: Domains with the same cookies after accepting CV.

The last Figure in this section is the case of modified ORM with Ninja Cookie plugin enabled. It is used to decline cookies on websites, so it should be fewer cookies. In Figure 33 it is possible to see that in 79.1% of websites that introduce cookies remain the same value. We can see that in 6.2% of the websites has reduced the number of cookies, and also we can see that in 14.8% of websites there are more cookies.

It is not normal that the Ninja Cookie plugin is adding cookies, so we did a manual inspection of the HTTP headers in that cases. We detect that the cookie is the same but it is repeated. So we can say that the Ninja Cookie plugin is working properly, nevertheless, in this case, it is blocking more than one time and the blocked cookies are repeated.

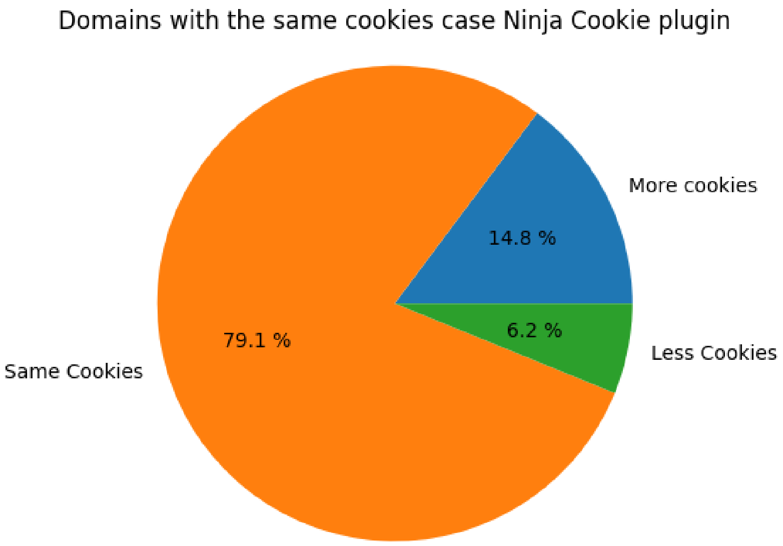


Figure 33: Domains with the same cookies with Ninja Cookie plugin.

We can conclude in this section that after interacting with websites, most of the cookies remain the same, the accept cookie button should add cookies, but a big number of web pages are adding cookies by default. In addition, the Ninja Cookie plugin is the same case, that should remove cookies, but the ones that are introduced are not possible to remove.

4.5 Country vs percentage NOT accepted

GDPR says that all websites, independently of the origin country, must ask consent from their EU customers to have their data processed. As we have seen in the previous results, it is not in that way, and more than 50% of websites are not asking for consent from their users.

In this section, we want to inspect the origin of the website countries that are not asking for consent. Furthermore, we divide the countries as first parties domains and third parties domains. Inspecting the database, we have seen that in every domain there was more than one country, this is because there are subdomains and we are collecting also the countries of the subdomains. So the first party is the country of the main domain, and the third party is the country of the subdomain.

In Figure 34 is possible to see the different countries that are found in the inspected domains. The blue lines are countries of the main domain, and the orange lines are countries of the subdomains. The relation is country vs the percentage of NOT accepted, so websites that was not possible to click on the accept cookies button.

Considering the origin of the country, for example, we can see Spain (ES) that is a common subdomain in lots of websites since this project is realized in Spain. It is possible to perceive that as a subdomain, it is present in lots of websites that do not ask for consent. On the other hand, if we inspect the case as a first-party domain, this value improves a lot, showing that they ask consent from their customers.

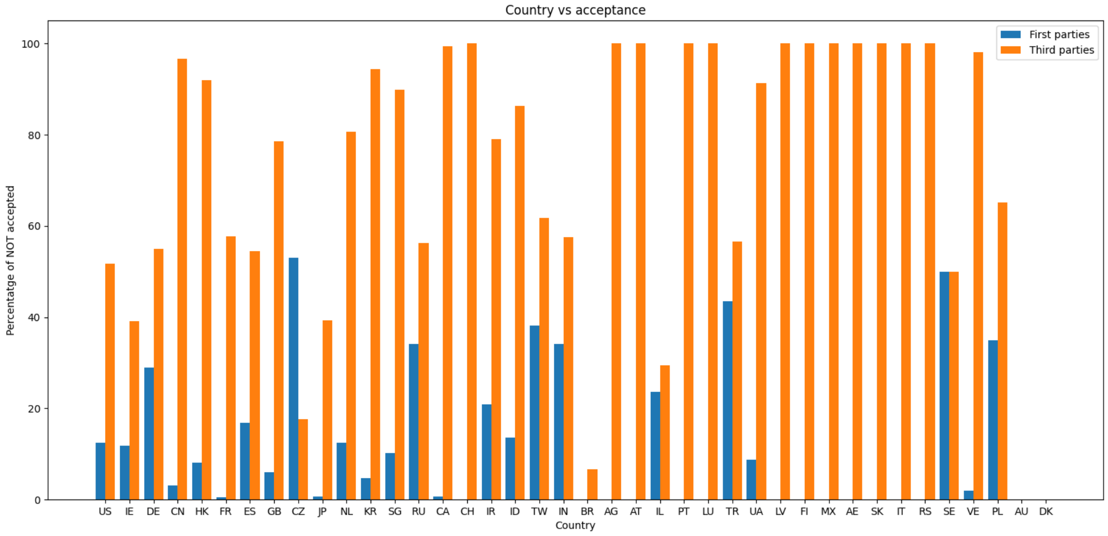


Figure 34: Country vs percentage NOT accepted.

Also, we want to distinguish between European countries and the ones that are not European. As we said, all countries must respect the law, however for example in Spain, it is less common to consult a Chinese website than a Spanish one.

In Figure 35 it is possible to see a map of Europe representing the values of the first parties domains. In general, these values as first parties are low, being the worst cases Sweden and the Czech Republic.

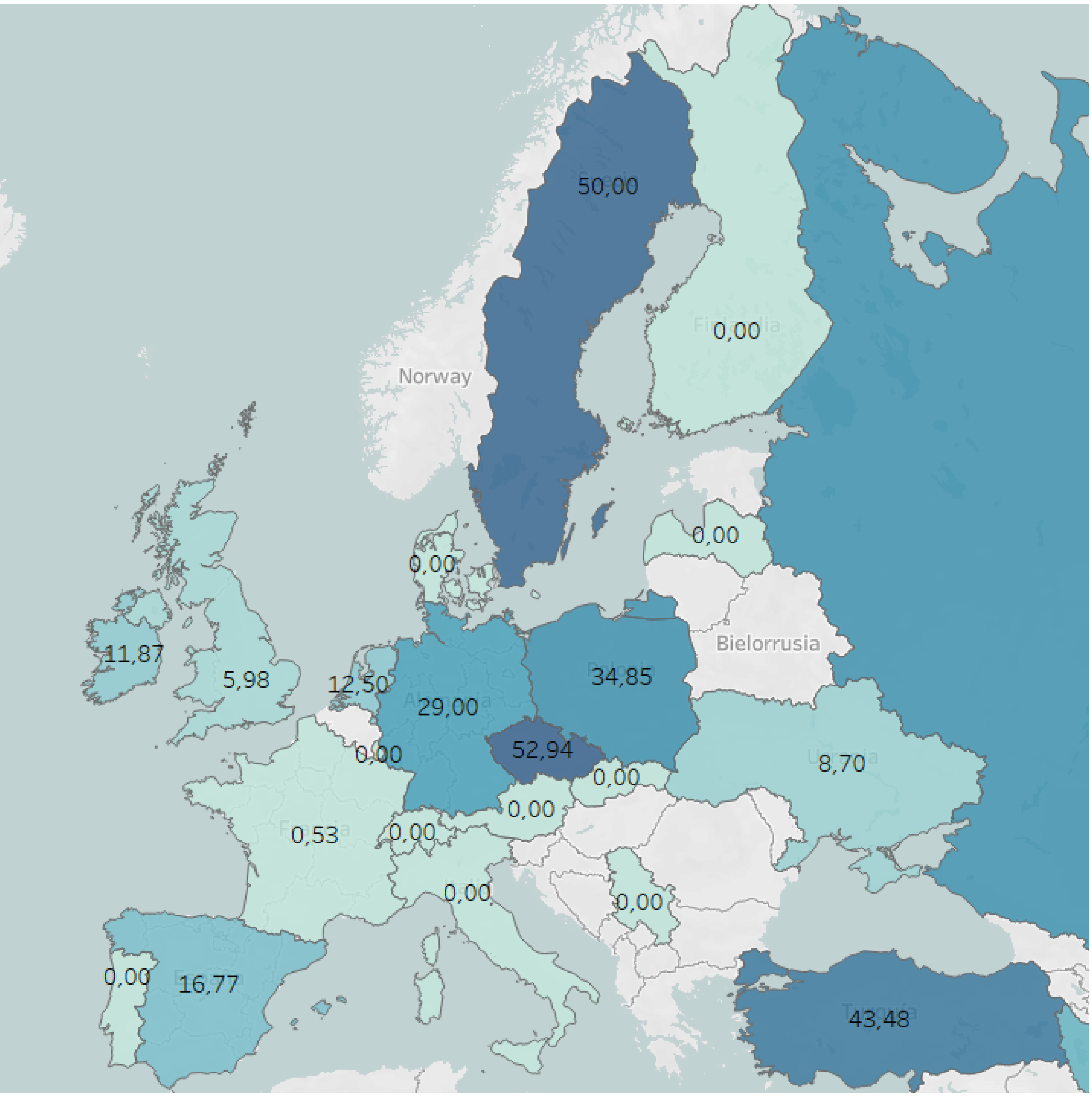


Figure 35: Map of Europe first parties.

On the other hand, in Figure 36 we can see the values of the orange columns represented. The results change a lot, being present as subdomains in domains that do not ask for consent from their users.

Countries such as Italy, Switzerland, Austria, have gone from having 0% of not asking for consent, meaning that consent was requested in all cases, to 100% of cases that do not ask for consent.

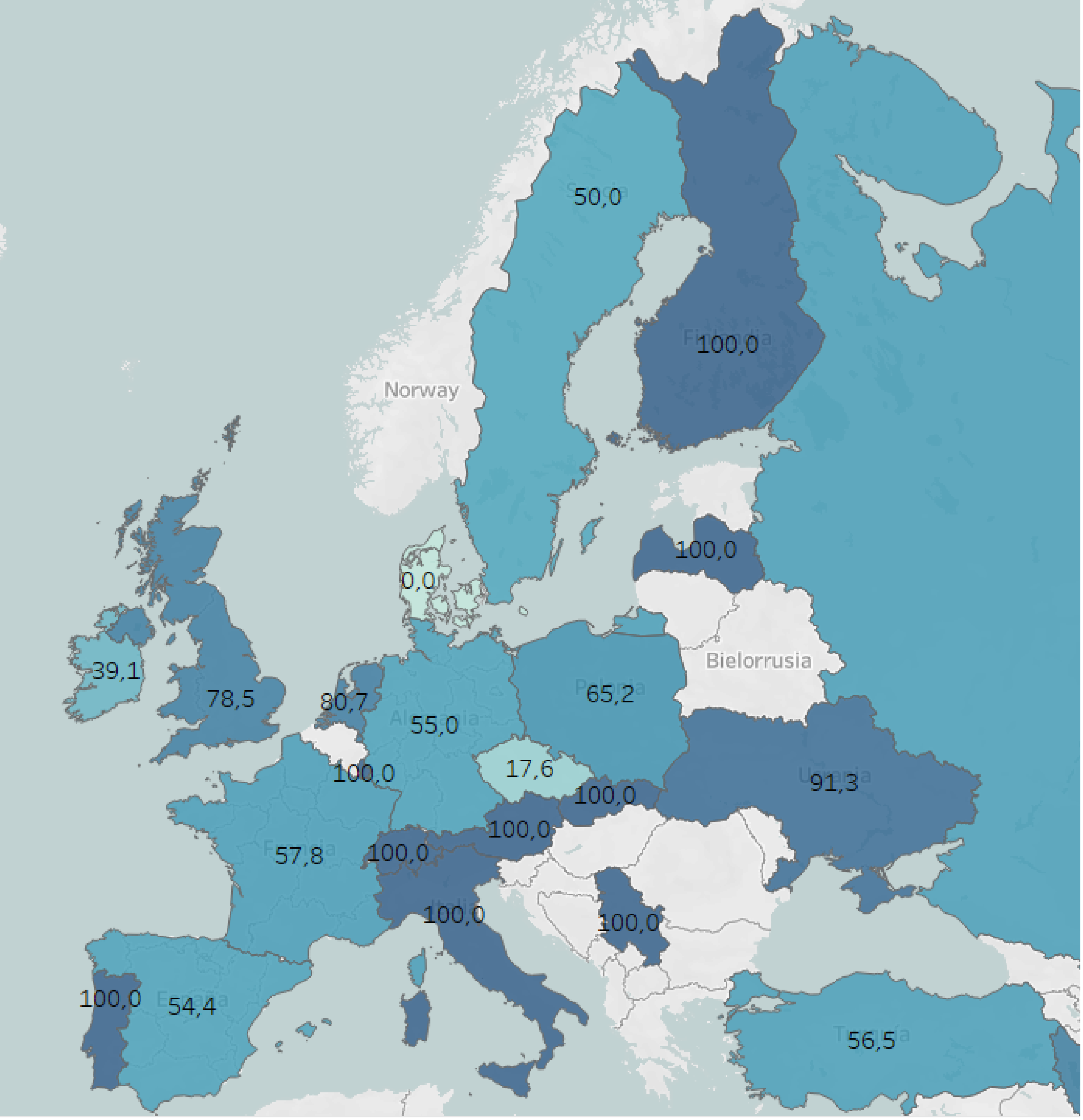


Figure 36: Map of Europe third parties.

Finally the last two figures of this section, Figure 37 and Figure 38 contains the same information as Figure 35 and Figure 36 but in world map. Figure 37 contains the values considered first parties domains, and Figure 38 contains the values being as subdomians.

It is possible to see in Figure 37 that the values are lower than the values of Figure 38.

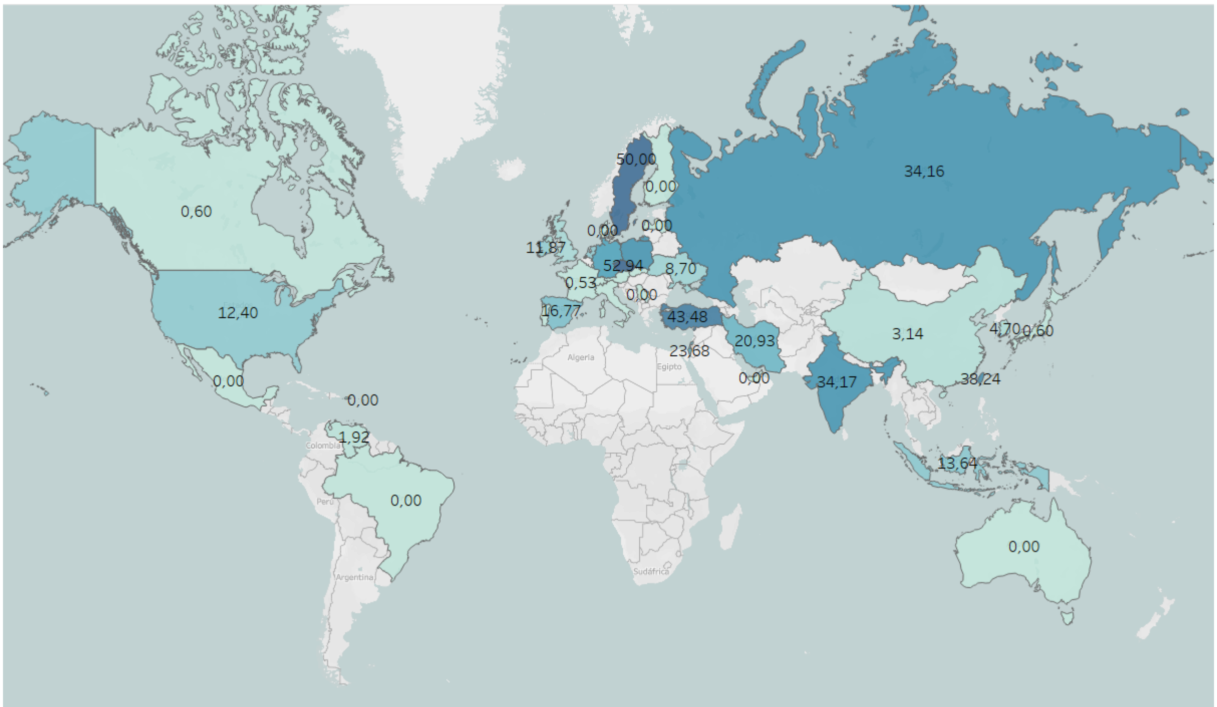


Figure 37: World map first parties.

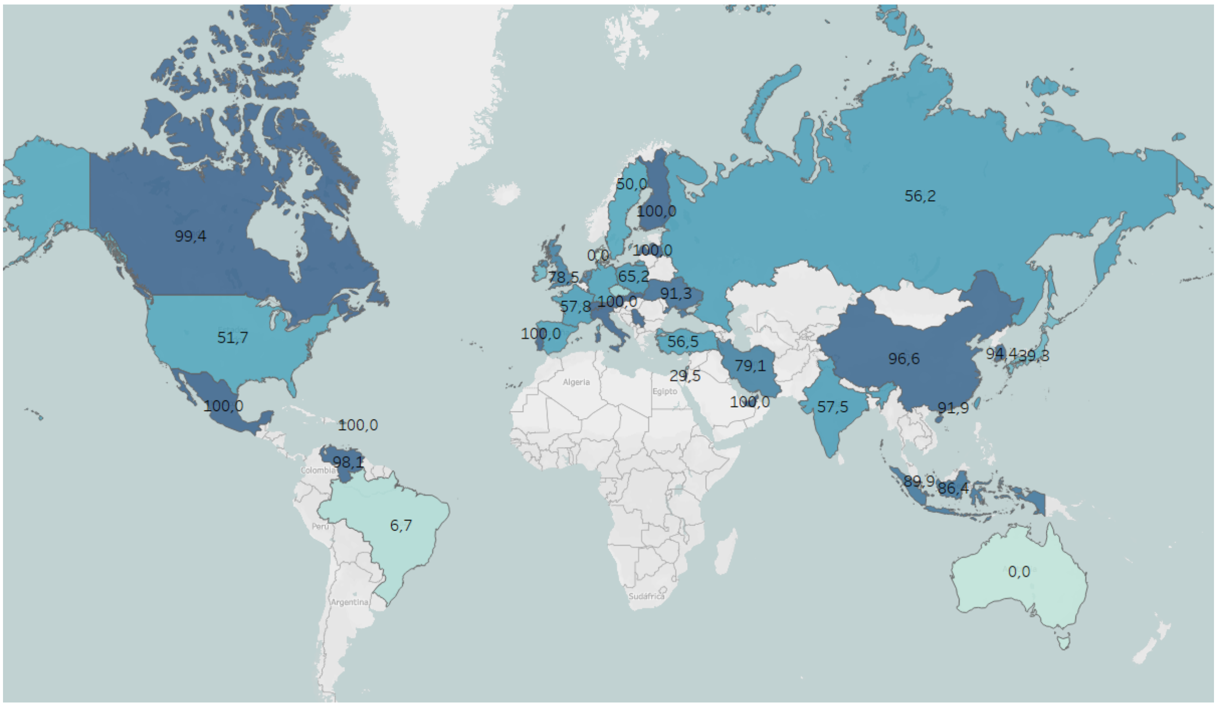


Figure 38: World map third parties.

To conclude this section, it is been observed that when a country corresponds to the main domain, usually is more respectfully with GDPR, asking for consent from their customers. However, when a country is a subdomain, the main domain usually does not ask for consent.

4.6 Cumulative Distribution Function (CDF)

This section, there are presented the results for different cases of the Cumulative Distribution Function (CDF). The CDF is an interesting graph because shows the distribution of the number of cookies.

In Figure 39 it is possible to see the CDF for the case of modified ORM with Selenium to accept cookies and the case of executing ORM by default, without any modification.

As it is possible to see, by default there are added cookies, but there is a higher number of fewer cookies. What this means is that for example, there are more websites that there are only introducing 1 cookie by default.

On the other hand, when we click the ”accept cookies” button, there are more websites with a higher number of cookies. In Figure 39 the blue line represents the case after accepting cookies, and we can see the dot blue point for the case of one cookie that is lower than the red one for the same case.

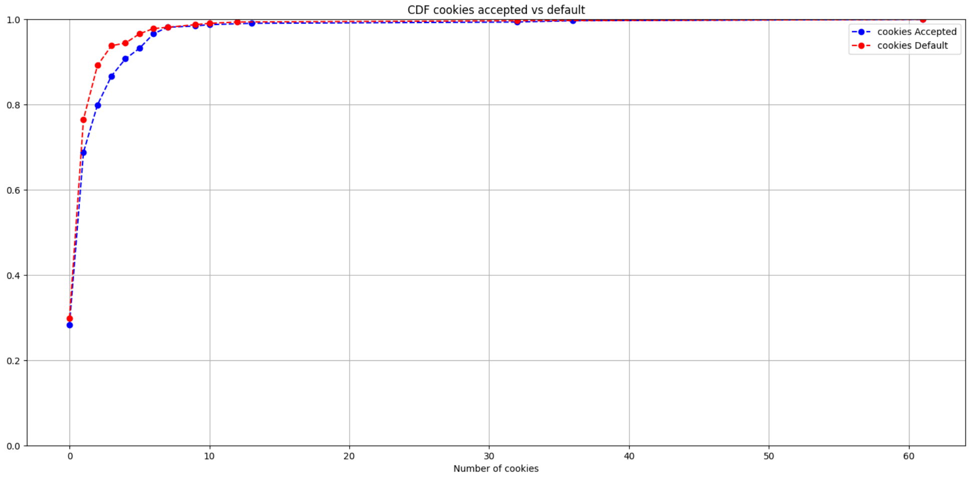


Figure 39: CDF cookies accepted vs default.

In Figure 40, after analyzing 10000 domains, in the case of added cookies by default and the added cookies after accepting, the number of cookies increases a lot. Although, there is a high number of websites with a few cookies.

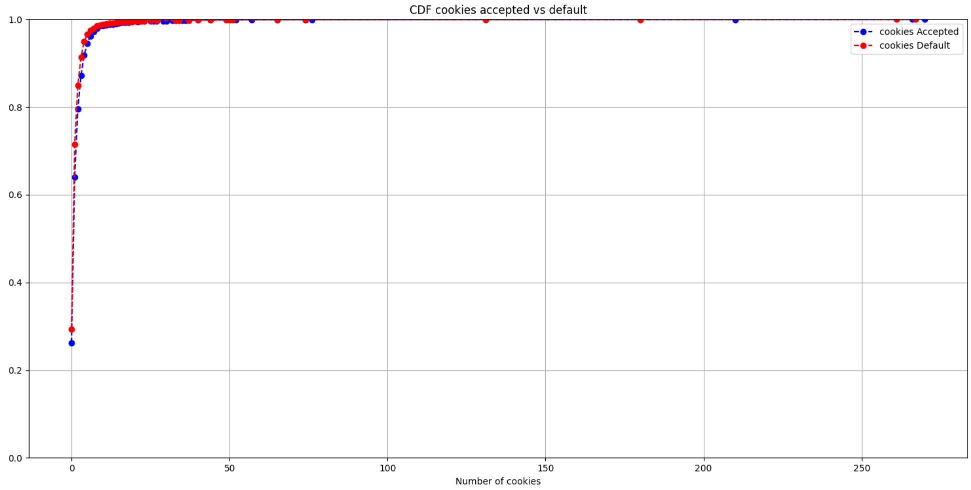


Figure 40: CDF cookies accepted vs default 10000 domains.

In Figure 41 it is possible to see the CDF for the case of the Ninja Cookie plugin enabled in ORM, and the case of executing ORM by default, without any modification. Theoretically, using the Ninja Cookie plugin should be fewer cookies, so the blue points should be higher for lower cookies. However, as mentioned in other sections, in some cases Ninja Cookies is blocking more than one time and is repeating cookies, so it counts as more cookies. Despite that, we can see that the blue line and the red line are practically equal.

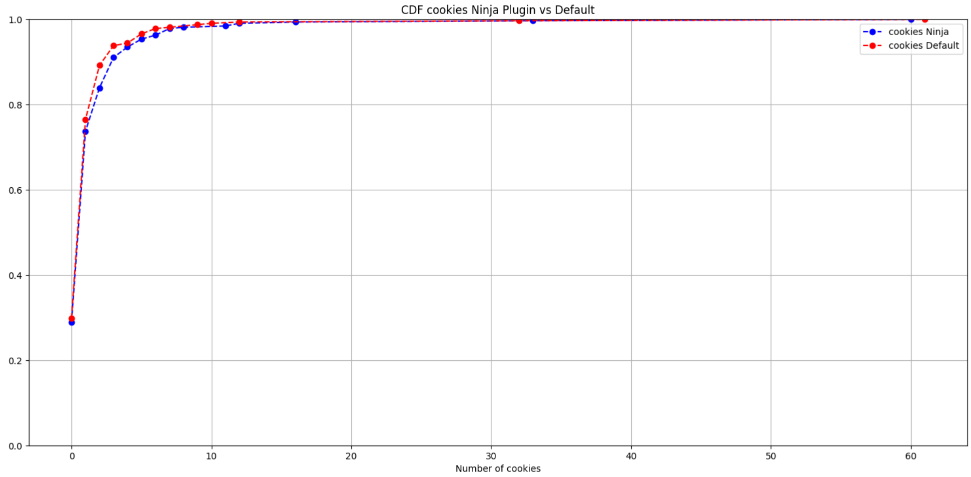


Figure 41: CDF Ninja Cookie plugin vs default.

In Figure 42 it is possible to see the CDF for the modified ORM with Computer Vision to accept cookies and the ORM by default. In this case, we can see that for lower values of the number of cookies added there are fewer websites, and the values increase more for a higher number of cookies.

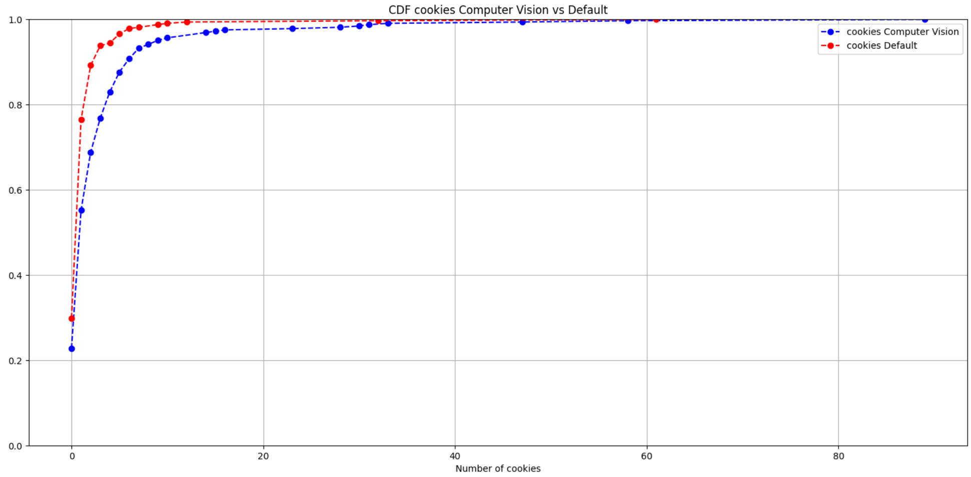


Figure 42: CDF cookies computer vision vs default.

Finally the last figure, the Figure 43 it is the representation of all CDF cases in one plot. We can see that the default ORM and Ninja Cookie plugin have similar values, and the methods to accept cookies increase the number of web pages with a higher number of cookies.

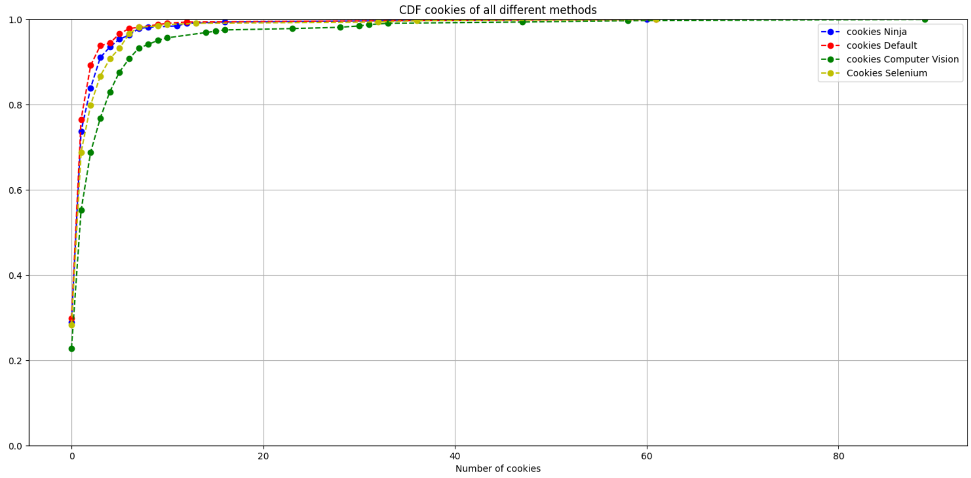


Figure 43: CDF all cases.

4.7 Famous domains vs number of cookies

In this section, we have chosen 10 websites that we consider famous or it can be more used in Europe. So we have considered that Chinese web pages are on top of the Alexa ranking, but with low probability will be visited in Europe.

Furthermore, we have chosen websites that contain cookies, to become aware that pages that are more probably to use in Europe are introducing cookies by default.

For example, google.com is a very famous and used domain. It is at the top of the Alexa ranking (number 1 in the list). As we can see in Figure 44 is introducing one cookie. Also, another example is amazon.com, which is introducing two cookies. In the following lines, we can see the cookie values of these domains, which we should inspect in more detail to determine if they are strictly necessary or are introducing something more.

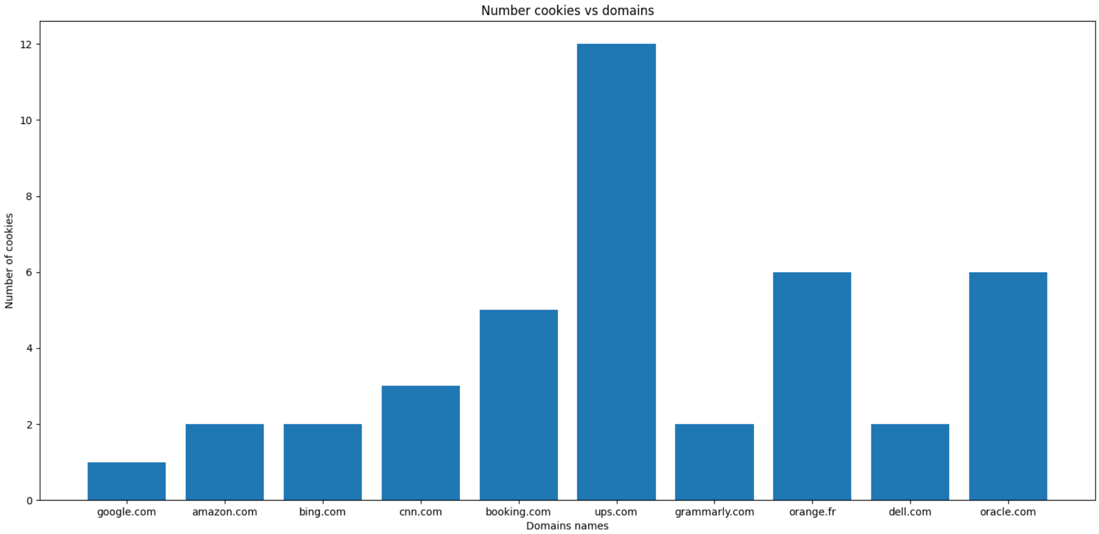


Figure 44: Famous domains vs number of cookies.

Value of google.com cookie header:

'set-cookie': 'CONSENT=PENDING+086; expires=Fri, 01-Jan-2038 00:00:00 GMT; path=/; domain=.google.com; Secure', 'status': '200', 'strict-transport-security': 'max-age=31536000', 'x-frame-options': 'SAMEORIGIN', 'x-xss-protection': '0'

Value of amazon.com cookie header 1:

'set-cookie': '$domain = .amazon.com$', '$status$': '200', '$strict - transport - security$': '$max-age = 47474747; includeSubDomains; preload$', '$vary$': '$Content-Type, Accept-Encoding, X-Amzn-CDN-Cache, X-Amzn-AX-Treatment, User-Agent$', '$via$': '$1.1bb2e82992d0b0058ac58c15961575f6c.cloudfront.net(CloudFront)$', '$x-amz-cf-id$': '$A6fB5G6cUAaAlsl-9Q8VXRttOSmSwnQjhW32_3r67q6oog2biLBccg ==$', '$x-amz-$

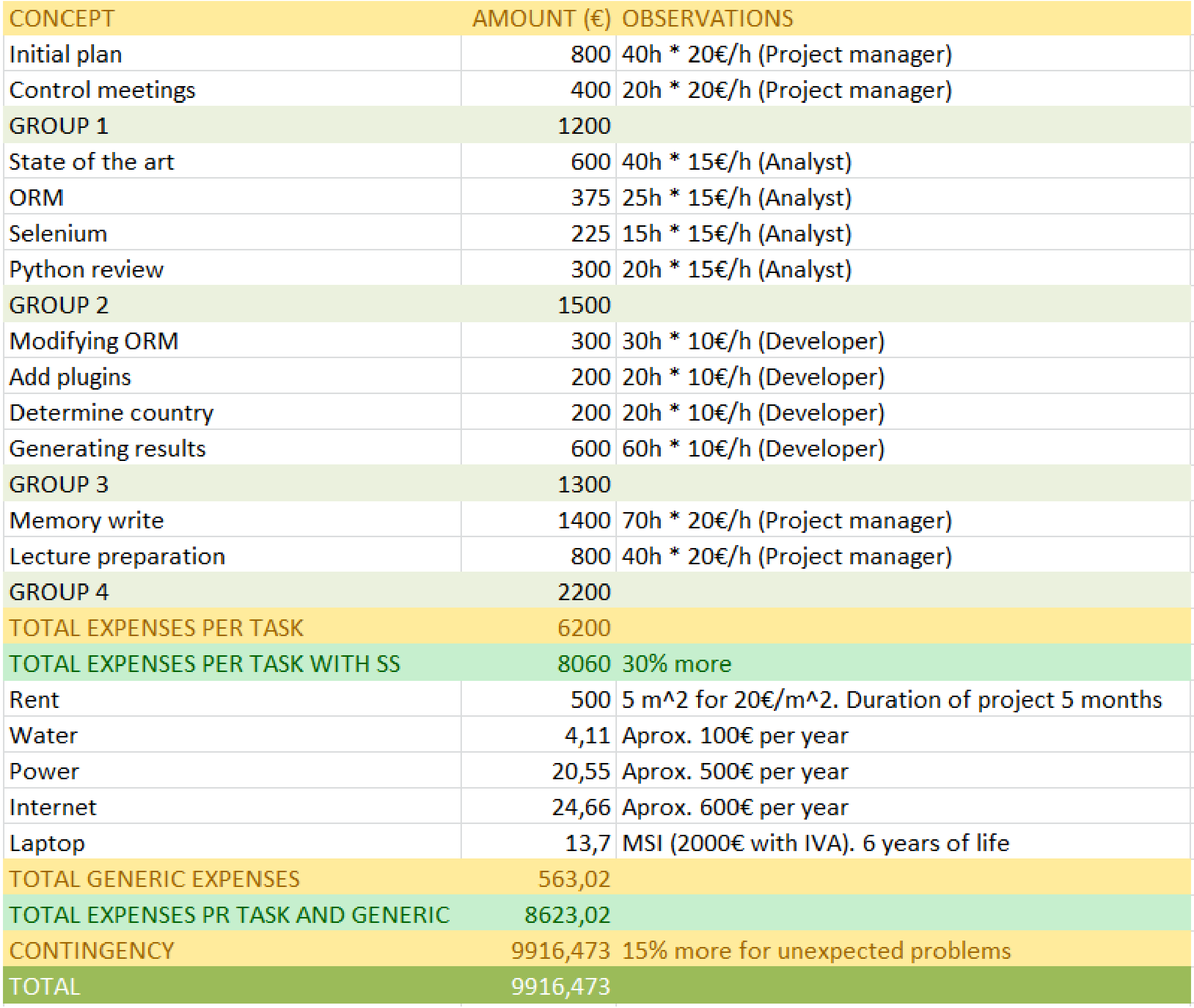
$cf-pop$’: ’$MAD51-C3$’, ’$x-amz-rid$’: ’$H3FTC6T7AVFSCMR3WHZS$’, ’$x-cache$’: ’$Missfromcloudfront$’, ’$x-content-type-options$’: ’$nosniff$’, ’$x-frame-options$’: ’$SAMEORIGIN$’, ’$x-ua-compatible$’: ’$IE=edge$’, ’$x-xss-protection$’: ’1;’

Value of amazon.com cookie header 2:

’Set-Cookie’: nad-privacy=0; Domain=.amazon-adsystem.com; Expires=Wed, 01-Jul-2026 15:59:34 GMT; Path=/; Secure; HttpOnly; SameSite=None’, ’Strict-Transport-Security’: ’max-age=47474747; includeSubDomains; preload’, ’Vary’: ’Content-Type,Accept-Encoding, X-Amzn-CDN-Cache, X-Amzn-AX-Treatment,User-Agent’, ’p3p’: ’policyref=”https://www.amazon.com/w3c/p3p.xml”, CP=”PSAo PSDo OUR SAM OTR DSP COR”’, ’x-amz-rid’: ’N4657QNKGMQA10C22NN1’

5 Budget

In this section, we detail the budget of the project. We consider that a developer will earn 10 euros per hour, an analyst will earn 15 euros per hour, and the project manager 20 euros per hour. Defined the salary and the task, we obtain the total expenses per task. We have to add the taxes that we consider that are 30%. Also, there are expenses related to renting, power, water, etc. So we have to add to the total expenses per task the generic expenses. Finally, we have to take into account that always can appear unexpected problems, so we add 15% for contingency. Finally, the total amount of money to develop this project is 9916,47 euros.



CONCEPT	AMOUNT (€)	OBSERVATIONS
Initial plan	800	40h * 20€/h (Project manager)
Control meetings	400	20h * 20€/h (Project manager)
GROUP 1	1200	
State of the art	600	40h * 15€/h (Analyst)
ORM	375	25h * 15€/h (Analyst)
Selenium	225	15h * 15€/h (Analyst)
Python review	300	20h * 15€/h (Analyst)
GROUP 2	1500	
Modifying ORM	300	30h * 10€/h (Developer)
Add plugins	200	20h * 10€/h (Developer)
Determine country	200	20h * 10€/h (Developer)
Generating results	600	60h * 10€/h (Developer)
GROUP 3	1300	
Memory write	1400	70h * 20€/h (Project manager)
Lecture preparation	800	40h * 20€/h (Project manager)
GROUP 4	2200	
TOTAL EXPENSES PER TASK	6200	
TOTAL EXPENSES PER TASK WITH SS	8060	30% more
Rent	500	5 m^2 for 20€/m^2. Duration of project 5 months
Water	4,11	Aprox. 100€ per year
Power	20,55	Aprox. 500€ per year
Internet	24,66	Aprox. 600€ per year
Laptop	13,7	MSI (2000€ with IVA). 6 years of life
TOTAL GENERIC EXPENSES	563,02	
TOTAL EXPENSES PR TASK AND GENERIC	8623,02	
CONTINGENCY	9916,473	15% more for unexpected problems
TOTAL	9916,473	

Figure 45: Budget table.

6 Conclusions and future development

This chapter will present the conclusions of this project after understanding what is GDPR, the corresponding rules, and the usage of the cookies. Also, the problems found during the development of the project and future study lines that this work can follow.

GDPR is the European regulation that has a specific point that says websites must ask their EU customers for consent to have their data processed.

Based on this, in this project several methods have been used to interact with web pages and analyze if they are compliant with GDPR, analyzing the usage of the cookies.

The first objective of the project was to find out what would happen if we ignored cookies and what was the behavior of the web only executing ORM. We have seen that in that case, a big number of websites are introducing cookies by default.

The second objective of the project was to find out what would happen if we accepted cookies and how this changed the operation of the web. To do so, we have followed two approaches, the first one is modifying ORM using Selenium and the second approach is the usage of Computer Vision to detect the button and click on it.

A comparison with these two methods has been done, showing that Computer Vision has more difficulties detecting the button. Although, both methods have shown that more than 50% of the websites visited do not ask for user consent.

In the case that it is possible to click the button and accept cookies, there is an important number of sites that do not add more cookies, leaving the same ones that already existed.

The third objective of the project was to find out what would happen if we blocked cookies and how this changed the operation of the web. We have seen that in that case the cookies that already are inserted by default will remain and very few cookies are blocked.

Although, it is analyzed the distribution of cookies in each case. As we mention, lots of web pages are introducing cookies by default, but usually are few cookies. If it is possible to accept cookies, the number of cookies introduced by websites increase. On the other hand, rejecting cookies, more or less the number of cookies in each domain will remain the same.

Finally, GDPR says that every website that wants to collect information of an EU user must ask for consent before introducing any cookie, regardless of the country of origin of the domain.

So the last results related to the country of origin, show that countries of the main domains tend to ask for user consent. On the other hand, countries that are subdomains, show that the percentage of domains that ask for user consent is lower.

Finally, all these methods to detect if websites are following the law and generate statistics, are integrated into the ePrivacy Observatory. In this observatory is possible to analyze different tracking methods and the level of tracking of each domain.

During the development of the project, we try to integrate all the different methods with the actual ORM. Due to compatibility reasons, we found that it was not possible to integrate it. The main problem was to modify ORM to integrate the Ninja Cookie plugin, it was required lots of modifications of the actual ORM version and a lot of time.

As a future line of development, it could be interesting to modify the ORM to integrate all the different methods to interact with websites.

Also, the results are generated with a small number of websites of the Alexa ranking due to different problems of space, and limitations of the computer. Another future line of development could be the generation of statistics with a large number of web pages of the Alexa ranking.

It is important to emphasize that with the integration of the modules with the newest version of ORM, automatically will be easier to generate the statistics with a large number of websites of the Alexa ranking.

References

[1] Théo and Guillaume. Ninja cookie, Apr 2021. Available online at: https://ninja-cookie.com/, last accessed on 09.05.2021.

[2] Ismael Castell-Uroz and Pere Barlet-Ros. Online resource mapper (orm), Apr 2021. Available online at: https://github.com/CBA-UPC/ORM/tree/master, last accessed on 11.05.2021.

[3] Daniel. I don't care about cookies, Apr 2021. Available online at: https://www.i-dont-care-about-cookies.eu/, last accessed on 11.05.2021.

[4] Selenium webdriver with python tutorial - javatpoint. Available online at: https://www.javatpoint.com/selenium-python, last accessed on 06.06.2021.

[5] What is gdpr, the eu's new data protection law? - gdpr.eu. Available online at: https://gdpr.eu/what-is-gdpr/, last accessed on 09.06.2021.

[6] Cookies, the gdpr, and the eprivacy directive - gdpr.eu. Available online at: https://gdpr.eu/cookies/, last accessed on 10.06.2021.

[7] Alexa - top sites. Available online at: https://www.alexa.com/topsites, last accessed on 10.06.2021.

[8] Google fined £91m over ad-tracking cookies - bbc news. Available online at: https://www.bbc.com/news/technology-55259602, last accessed on 11.06.2021.

[9] The spanish data protection authority fined the company vueling for the cookie policy used on its website with 30,000 euros — european data protection board. Available online at: https://edpb.europa.eu/news/national-news/2019/spanish-data-protection-authority-fined-company-vueling-cookie-policy-used_en, last accessed on 11.06.2021.

[10] 5 biggest gdpr fines so far [2021] – data privacy manager. Available online at: https://dataprivacymanager.net/5-biggest-gdpr-fines-so-far-2020/, last accessed on 11.06.2021.

[11] List of fines — gdpr fines - inplp. Available online at: https://gdpr-fines.inplp.com/list/, last accessed on 11.06.2021.

[12] eprivacy observatory. Available online at: http://tars.cba.upc.edu/index.php, last accessed on 11.06.2021.

[13] M. Veale et al. M. Nouwens, I. Liccardi. Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. (English). *Conference on Human Factors in Computing Systems - Proceedings*, 2020.

[14] C. Matte C. Santos, N. Bielova. Are cookie banners indeed compliant with the law ?. (English). *Plsc 2019*, pages 1–51, 2019.

[15] La matriz de confusión y sus métricas – inteligencia artificial –. Available online at: `https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/`, last accessed on 11.06.2021.

[16] A. Narayanan S. Englehardt. Online Tracking: A 1-million-site Measurement and Analysis. (English). *ACM CCS*, pages 1–20, 2016.

[17] P. Barlet-Ros I. Castell-Uroz, J. Solé-Pareta. TrackSign: Guided Web Tracking Discovery. (English). *Infocom 2021*, pages 1–10, 2021.

[18] Welcome to python.org. Available online at: `https://www.python.org/`, last accessed on 11.06.2021.

[19] Mysql :: Download mysql workbench. Available online at: `https://dev.mysql.com/downloads/workbench/`, last accessed on 11.06.2021.

[20] Chromium - the chromium projects. Available online at: `https://www.chromium.org/Home`, last accessed on 11.06.2021.

[21] Oracle vm virtualbox. Available online at: `https://www.virtualbox.org/`, last accessed on 11.06.2021.

[22] El motor de tu trabajo — slack. Available online at: `https://slack.com/intl/es-es/`, last accessed on 11.06.2021.

[23] Get ubuntu — download — ubuntu. Available online at: `https://ubuntu.com/download`, last accessed on 11.06.2021.

[24] Tus proyectos - overleaf, editor de latex online. Available online at: `https://es.overleaf.com/project`, last accessed on 11.06.2021.

[25] Github. Available online at: `https://github.com/`, last accessed on 11.06.2021.

[26] Visual studio code - code editing. redefined. Available online at: `https://code.visualstudio.com/`, last accessed on 11.06.2021.

[27] Google meet. Available online at: `https://meet.google.com/`, last accessed on 11.06.2021.

[28] Software de análisis e inteligencia de negocios. Available online at: `https://www.tableau.com/es-es`, last accessed on 11.06.2021.

[29] Teamviewer: soporte remoto, acceso remoto, asistencia técnica,colaboración en línea y reuniones. Available online at: `https://www.teamviewer.com/es/`, last accessed on 11.06.2021.

[30] mbasart/orm-analysis. Available online at: `https://github.com/mbasart/ORM-analysis`, last accessed on 11.06.2021.

[31] mbasart/orm-cookies. Available online at: `https://github.com/mbasart/ORM-cookies`, last accessed on 11.06.2021.

Appendices