Deployment of a software infrastructure for Ecommerce and business analytics in a small business

Author:

Martín de la Riva

Tutor:

Germán Gutiérrez Sánchez

Director:

José Antonio Díaz Nicolás

Abstract

This project will be a simulation of a real business project that can be found in the enterprise world. We will create an Ecommerce framework for a fictional medium-sized company and we will connect it to a Business Intelligence & Data Visualization software in order to let the enterprise perform business analytics, see their current performance, and provide enough data to final users (e.g. CEOs) to let them to make better decisions.

In order to do this project, we will first review the current possibilities to accomplish this type of project. Therefore, we will perform an analysis of the software being used in the market to decide the better one for a medium-sized business, accomplishing all the basic requirements it would need. So, we will do this analysis for the Ecommerce and Data Visualization software. And finally, we will study different solutions to post the shop on the web.

For this last goal of posting the web online, we will use a server service that provides a set of instances that can be powerful enough to support an Ecommerce platform. The server service finally selected is Amazon Web Services.

As Ecommerce platform, we will use Magento, an open source software, which has been leading the market of Ecommerce platforms for several years [1] [2]. Magento can be modified to create nearly any required feature (e.g. adding a contact section), and also, there is a huge community where other developers post their questions and/or solutions, solving problems, and of course implementing new functionalities.

For the Data Visualization part, we will use one of the main leaders on this section: Tableau. It is a powerful tool mainly centralized in performing Visualization procedures. Although it is not open source, and therefore it is closed to its current features, it provides everything that is needed for this project. We will use their Student full version to simulate what would the purchased version provide.

Then we will unify all these software elements so we can make business analysis with Tableau, using the information from Magento Ecommerce. We have investigated how to connect these two software frameworks, and also how to get the appropriate information to show in the Data Visualization app.

Finally, with the information obtained from Magento, we will use Tableau and create example charts as an introduction to the potential reports, etc. we can create with a tool like Tableau.

We will define the tasks and estimate a longitude of the project. Taking into account the tasks and the nature of the project, we will use as work methodology the Waterfall approach, performing only one delivery, as the requirements will not change during the course of the project, and therefore agile methods are not applicable.

We will end with the conclusions extracted from the project and the future work that can be added to have a complete and more powerful solution.

Acknowledgements

I would like to thank my family, who has always supported me and helped me achieve my goals.

I would also like to thank my teachers at University Carlos III, and at my Erasmus at Warsaw and Vienna Universities of Technology; my tutor Germán and every member of Sugerendo Sistemas, that helped me during the course of this project, and showed me a first glance at the business world.

I would also like to thank the people I have met during this last four years, that has helped me evolve not only academically but as a person.

"The purpose of computing is insight, not numbers." Richard Hamming

Index of Content

Index of Figures

Index of Tables

1 Definition of the Project

1.1 Introduction

In this project we are willing to simulate a real project in which a medium-size company (xx employees) wants: 1) to sell products via online in two different countries; 2) get visualization reports about their sales, in order to explore solutions and help in business decisions.

In order to accomplish this goal, we are going to implement a powerful Ecommerce software that can have the required features installed, and having it connected to a Business Intelligence or Data Visualization software, where simple queries can be used to generate easy and fast graphics.

This project has been developed while I am involved (an internship) in a small-size company specialized in Ecommerce solutions[1]. Therefore, in this project this company will play the role of the final user, defining the requirements based on the Ecommerce projects they achieve in their day-to-day activities.

1.1.1 Motivation

In the last 10 years, the use of online shopping has increased overwhelmingly, and currently a large percentage of sales are done through the Internet. This has increased the need of software development for Ecommerce, not only as a web is needed, but all the functionalities on its Back-End, including databases with the sales, operations, products, clients, etc.

Meanwhile, the development of systems within Business Intelligence area has also exploded on nearly any size of business, being proved to be a successful tool to make the best decisions on a company. It is mainly used for selling companies, which make it perfect for an Ecommerce business. Although some Ecommerce software have possibilities to create simple graphics, none has the complete freedom a Data Visualization platform provides, where any type of charts (Bar Charts, Pie Charts, Histograms, Scatterplots, etc) can be created, using any type of information. Therefore, an implementation between the information of an Ecommerce and a Business Intelligence software can be really useful and wanted.

1.1.2 Goals

In this project I am willing to acquire a better knowledge about Ecommerce and Business Intelligence. I want to get a deeper knowledge about the online market and how it is done, and

[1] Sugerendo Sistemas. www.sugerendo.com

increase my knowledge on the aspects of Business Intelligence, especially in Data Visualization. Also, I will investigate about how these two technologies are approached, in order to get the current picture about the different softwares that are being currently used.

Now, in order to accomplish a successful project, these will be the milestones to achieve:

- Basic installation of an Ecommerce application. This software will be chosen doing a benchmark of the different solutions in the market.
- Installation of a Business Intelligence or Data Visualization software, which will also have to be chosen through a benchmark of the current solutions.
- Implementation between both applications, so the Data Visualization software is connected to the needed tables on the Ecommerce database, and can display useful analytics.
- Accomplish the basic requirements defined by the fictional company.
- Accomplish the advanced requirements defined.
- Write documentation about all the steps done during the project, in order to have proper information in case this or similar project are to be done in the future.

1.2 Socio-Economic Environment

The environment of this project is the Ecommerce market in Spain, by 2016. Therefore, we are going to analyze how this market is evolving.

Since it starts in Spain, the Ecommerce market has not stopped growing, as we can see in Figure 1 taken from a study done by *Observatorio Nacional de las Telecomunicaciones y de la SI* (*ONTSI*), a national organism whose main objective is to monitor analysis of the telecommunications sector and the Information Society. This study reflects how the B2C (Business to Customer) ECommerce market growth from 2007 to 2014.

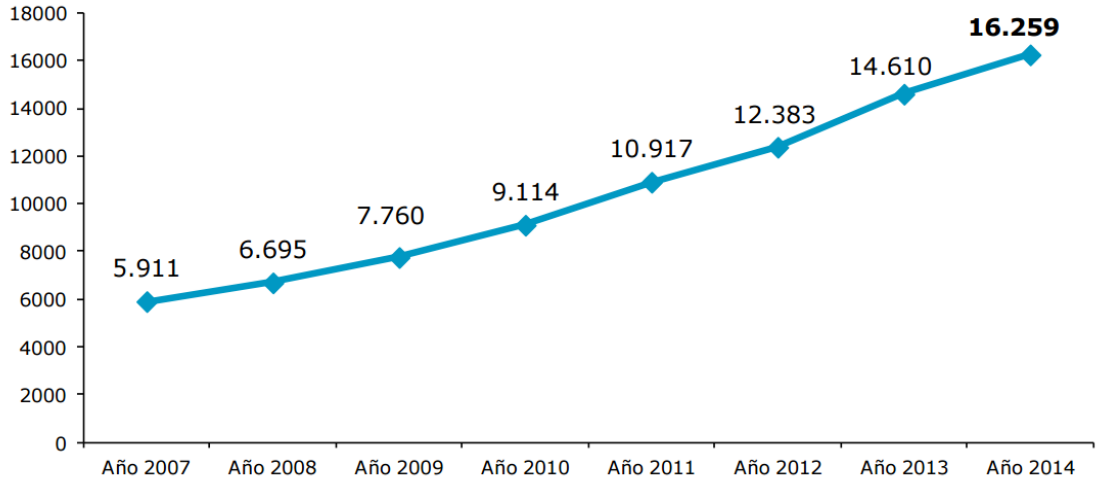


Figure 1: Evolution of Ecommerce in millions of euros in absolute terms [22]

This study shows that although the market is still in a growing phase, it is experiencing a deceleration. It has grown in absolute terms in 2014 a 11.3% in contrast to the growth it experimented in 2013 (18.0%) and 2012 (13.4%).

Also, as it can be seen in Figure # the spending per individual is also growing. This, joined with the fact that the number of buyers is also growing (Figure #), certifies that we are in a significantly growing market.

Año	Importe total
2007	739 €
2008	754 €
2009	749 €
2010	831 €
2011	828 €
2012	816 €
2013	848 €
2014	876 €

Figure 2: Evolution of annual average spending per individual online [22]

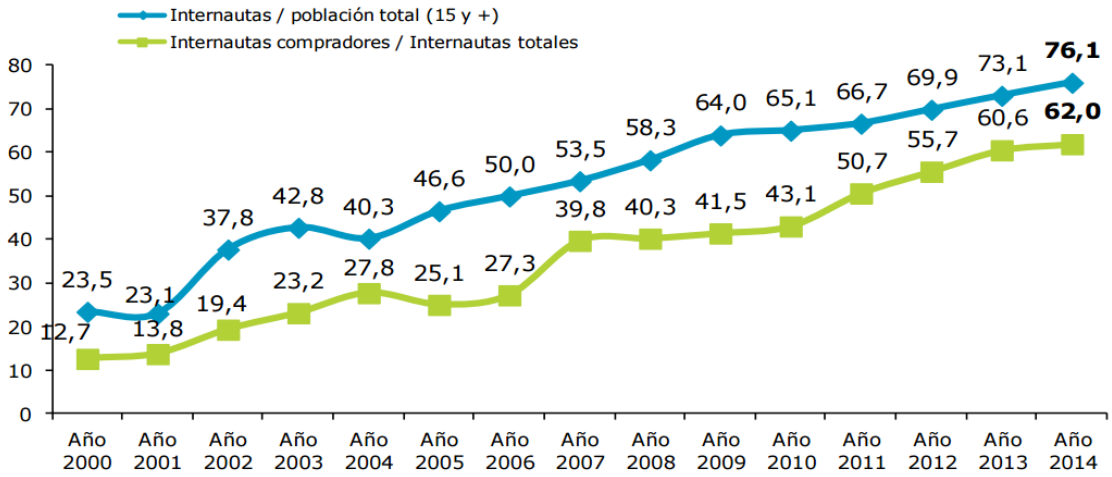


Figure 3: Evolution of internet usage (blue) and internet buyers (green) in Spain [22]

Right now, one of the reasons of this growth is the prices that are offered online (75.5% of the studied people), comfort (63.0%) and time saving (49.7%). Meanwhile, the main worries about Ecommerce are the shipping costs, the returning guaranty, the lack of customer support and providing personal information to perform the buy.

Also, the market is experiencing a growth in terms of mCommerce. In other words, commerce performed from smartphones. It has also been growing and it has now a 17.8% of the online buyers. This tendency keeps growing and shows the importance of giving a good support when entering the web through smartphones.

1.3 Legal Framework

We will now analyse the current legalization about Ecommerce to find what restrictions a project of this kind would encounter. The main restrictions we could find are in terms of privacy and protection of the customer.

1.3.1 Ley General para la Defensa de los Consumidores y Usuarios

This law provides information about the information that has to be provided before a buy and the restrictions within this buying procedure. The main indications to take into account for an Ecommerce are [23]:

- The information about price, delivery term, returning costs and conditions, before the buying procedure.
- Deadlines of delivery (**30 days** maximum).
- Customer's resignation right. It states mandatory that the online shop has to indicate clearly and comprehensive the customer's resignation right. The customer has to have a minimum 14 day term. If it is not properly indicated, the customer has 12 months to resign. Also, the seller has to refund the payment and the shipping costs.

1.3.2 Ley Orgánica de Protección de Datos de Carácter Personal

This law specifies the protection of privacy data from shop users. It states guidelines that the online shop must fulfil in order to store private information from a user. Although it shows other important aspects, these are the most relevant for the case of an Ecommerce [24]:

- The user has to give consent for its privacy data to be stored. This is done with terms and conditions that the user has to read and understand, to give permission to the commerce.
- If some privacy information is filtered or lost, the direct responsible will be the company that stored the data, including the possibility of applying a compensation.

1.3.3 Real Decreto-ley 13/2012, de 30 de marzo- Cookies

In terms of cookies, every web page has to properly inform users about the storage and usage of cookies in it.

A cookie is a program that is sent from the web page server to the visitor browser and is used to store information about the user (language, localization, etc.), and is normally used to perform analysis or provide a more complete experience.

In Spain, the web page has to properly inform the user about the cookies used in it and the cookie policy, giving him the chance to accept or decline the use of cookies [25].

They cannot be installed if the user does not give consent for it.

1.3.4 Price Policy

The only restriction in terms of prices is that it cannot be imposed below the commodity price [23]. This means that the practice commonly known as *dumping* [26] is forbidden. Besides this, there are no limitations in pricing, and it is allowed to sell at different prices in different shops (e.g. in physical and online shop).

1.4 Work Plan

In this section we will decide the work plan that we are using for this project. We will define the different tasks to complete and, taking them into account, decide what type of project life cycle fits best with it.

1.4.1 Tasks Breakdown

Taking the Goals defined in the Introduction, we can define the different phases we will have in this project, and with them the different tasks on each:

Table 1: Project phases and tasks definition

Project phases			
Name	**Description**	**Tasks**	**Duration**
Definition	Define the problem and the solution to implement, including the requirements it needs to comply.	1. Define the problem. 2. Specific Requirements definition. 3. Study current solutions.	3 weeks
Software Benchmark	Investigate the different software that exist in order to decide which one is the best for our project.	1. Ecommerce Benchmark. 2. Business Intelligence Benchmark. 3. Decide which software fits our project.	4 weeks
Implementation	Plan the design of the integration between both softwares.	1. Application Design 2. Implement Ecommerce Software.	8 weeks

	Implement and integrate the software and fulfill the specific requirements.	3. Program/install specific requirements. 4. Implement Business Intelligence Software. 5. Integrate both softwares 6. Create Data Visualization Charts.	
Documentation	Create a document describing the whole project and the final presentation.	1. Write Document. 2. Create Presentation.	16 weeks

1.4.2 Work Breakdown Structure

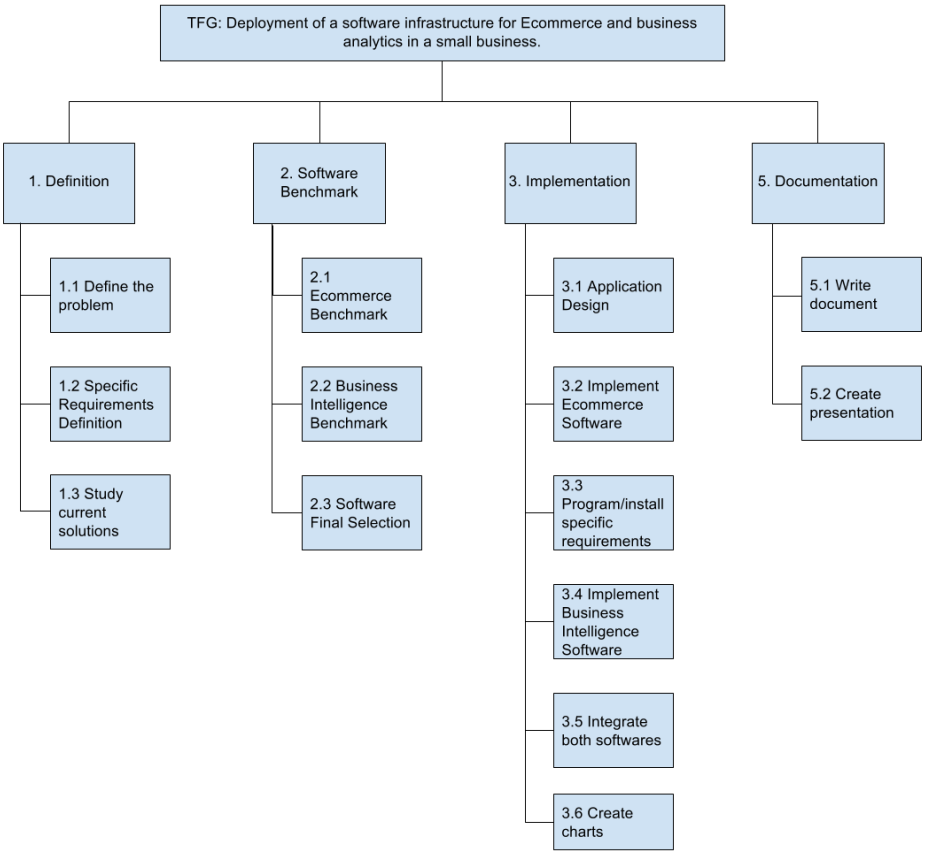


Figure 4 Project Work Breakdown Structure

1.4.3 Work Plan

Nowadays, there exist a lot of diverse life cycles approaches for projects. We can classify them in: i) classic methods and ii) agile methods. Agile methods are designed to have a more interactive communication between the user and the different members of the project, in order to minimize errors and implement the solution as close as possible to the needs of the user; and ensuring that all members of the project are on the same page and know what is everyone doing.

As the requirements are specified at the beginning and do not change along the process, agile methods are not adequate for the purpose of this project.

Then, we only have the classic approaches. Once again, we can choose between methods that emphasize interaction with the user (iterative-incremental delivery) or more 'rigid' approaches (incremental delivery and waterfall). Therefore, we will stick with the 'rigid' or user-less ones:

Incremental delivery is an approach that is based on dividing the project in different subprojects, delivering a fully functional product and improving it on each delivery. In the following image you can see a basic drawn explanation on how does it work:

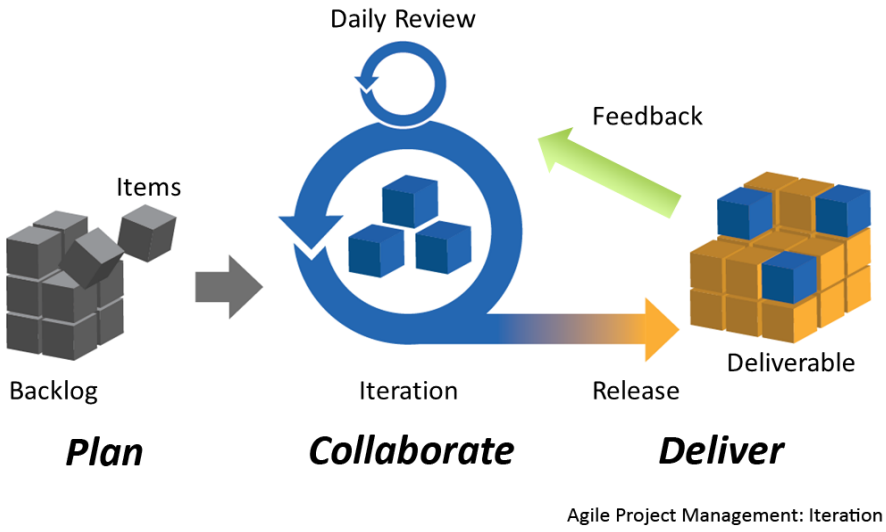


Figure 5: Incremental delivery sketch [3]

Meanwhile, waterfall is a classic method that consists on finishing the project on one single delivery, having sequential tasks. In other words, you do not start task #2 until you end task #1. This can be a risky approach, as if you find an error, you have to correct it on every task of the project, and it can make you waste a lot of time.

If we applied incremental delivery, we could divide the project in two subprojects (Ecommerce, and Data Visualization integration). Besides this, as we are doing the whole project in one delivery, we will use the Waterfall approach.

1.4.4 Time Plan (Gantt Chart)

With the work packages described in the previous section, we have estimated a project duration of around four months. The full deployment with the estimated duration per task can be seen in Figure 6. There we can see the estimated number of a total of 95 work days.

In Figure 7 you can see the Gantt Chart of the project tasks and the dependencies between them. As it can be seen, Documentation is done in parallel along the project is being done. Also, we have estimated a long Implementation phase (it takes up to 52% of the project) as we consider it is the most important task.

	Task Name	Duration	Start	Finish	Predecessors
0	**TFG: Deployment of a software infrastructure for Ecommerce and business analytics in a small business**	**95 days**	**Mon 08/02/16**	**Fri 17/06/16**	
1	**1 Definition**	**20 days**	**Mon 08/02/16**	**Fri 04/03/16**	
2	1.1 Define the problem	5 days	Mon 08/02/16	Fri 12/02/16	
3	1.2 Specific Requirements Definition	10 days	Mon 15/02/16	Fri 26/02/16	2
4	1.3 Study current solutions	5 days	Mon 29/02/16	Fri 04/03/16	3
5	**2 Software Benchmark**	**25 days**	**Mon 07/03/16**	**Fri 08/04/16**	
6	2.1 Ecommerce Benchmark	10 days	Mon 07/03/16	Fri 18/03/16	4
7	2.2 Business Intelligence Benchmark	10 days	Mon 21/03/16	Fri 01/04/16	6
8	2.3 Software Final Selection	5 days	Mon 04/04/16	Fri 08/04/16	7
9	**3 Implementation**	**50 days**	**Mon 11/04/16**	**Fri 17/06/16**	
10	3.1 Application Design	5 days	Mon 11/04/16	Fri 15/04/16	8
11	3.2 Implement Ecommerce Software	10 days	Mon 18/04/16	Fri 29/04/16	10
12	3.3 Program/install specific requirements	5 days	Mon 02/05/16	Fri 06/05/16	11
13	3.4 Implement Business Intelligence Software	10 days	Mon 09/05/16	Fri 20/05/16	12
14	3.5 Integrate both softwares	10 days	Mon 23/05/16	Fri 03/06/16	13
15	3.6 Create Data Visualization Charts	10 days	Mon 06/06/16	Fri 17/06/16	14
16	**4 Documentation**	**90 days**	**Mon 15/02/16**	**Fri 17/06/16**	
17	4.1 4	80 days	Mon 15/02/16	Fri 03/06/16	2
18	4.2 Create Presentation	10 days	Mon 06/06/16	Fri 17/06/16	17

Figure 6: Overall time estimation

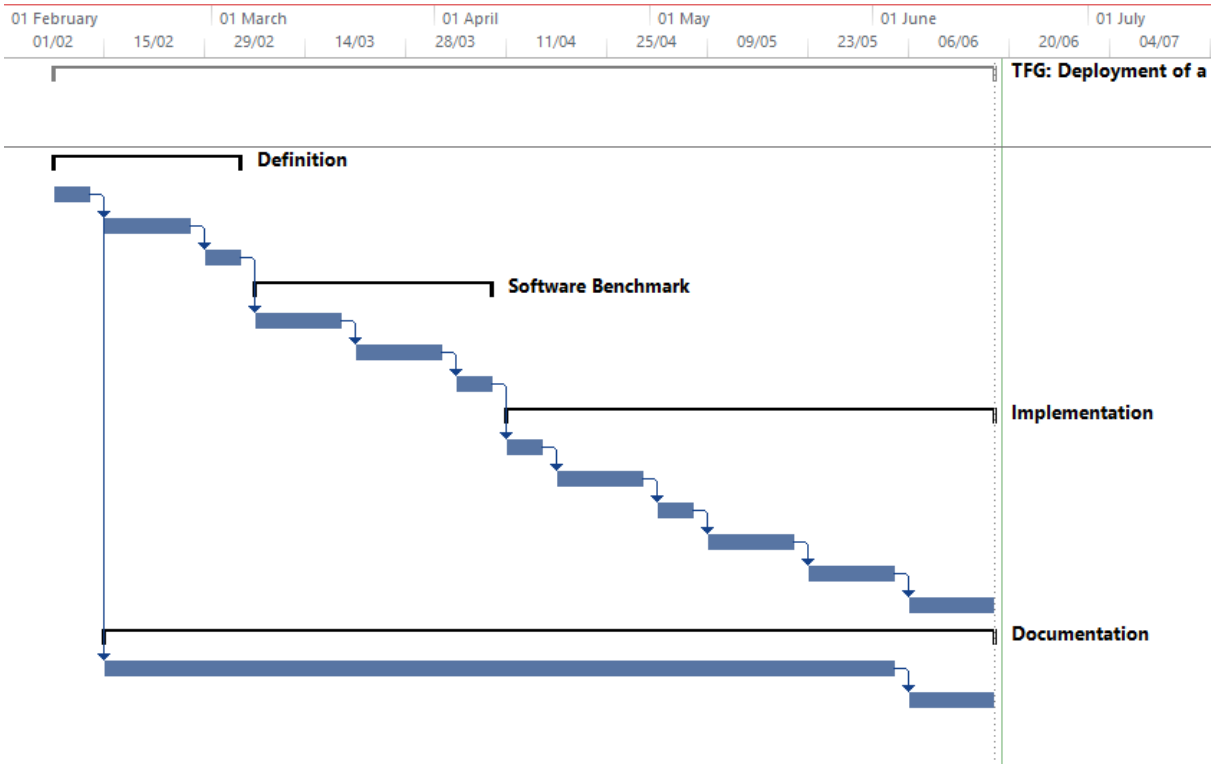


Figure 7: Gantt Chart

1.5 Resources & Budget

In this chapter we will create a list of the resources used for the completion of this project of investigation and installation of a business software. We will take into account the personnel resources, and the material for hardware and software.

We will also calculate the costs of the whole project, including every resource used in it. We will simulate the costs of the personnel resources, creating the personnel roles that could participate in this project.

1.5.1 Personnel Resources

The workforce dedicated to this project consists of:

- **Martín de la Riva:** Student of the Bachelor's Degree in Computer Science and Engineering in Universidad Carlos III de Madrid. Responsible of the planification, investigation and development of the project as well as the execution of testing. At last he will have to document the project.
- **Germán Gutiérrez Sánchez:** Teacher at Universidad Carlos III de Madrid. Tutor of the project. In charge of helping the student on the realization and documentation of the project.
- **José Antonio Díaz Nicolás:** Project Manager at Sugerendo Sistemas SL. Director of the project. In charge of guiding the student on the orientation of the project.

Costs

Table 2: Personnel Costs

Category	Working hours	Cost per hour (€)	Costs
Project Manager	50	25	1.250,00 €
Analyst	150	20	3.000,00 €
Designer	50	20	1.000,00 €
Programmer	350	15	5.250,00 €
		Total	10.500,00 €

1.5.2 Hardware Resources

Table 3: Hardware Resources Costs

Description	Costs (€)	Dedicated time in months	Depreciation time in months	Costs
Desktop Pc	1.500	4	60	100,00 €
Laptop	700	4	60	46,67 €
Amazon Web Services Server	0	4	-	0,00 €
Total			**Total**	146,67 €

1.5.3 Software Resources

Table 4: Software Resources Costs

Description	Costs (€)	Dedicated time in months	Depreciation time in months	Costs
Google Drive	0	-	-	0
Google Documents	0	-	-	0
Microsoft Word 2016	135	4	30	18,00 €
Microsoft Project 2016	769	4	30	102,53 €
Magento CE	0	-	-	0
Tableau for Students	0	-	-	0
Putty	0	-	-	0
FileZilla	0	-	-	0
Apache	0	-	-	0
MySQL	0	-	-	0
phpMyAdmin	0	-	-	0
Sublime Text 2	0	-	-	0
Ubuntu 14.04	0	-	-	0
Windows 10	0	-	-	0
Total		4	**Total**	120,53 €

1.5.4 Total Costs

In order to calculate the total costs, we will add up the personnel, hardware and software costs. We will add the indirect costs. These include the costs that are derived from the project (e.g. internet, light, etc.). We will estimate a 10% of the total project for them.

Finally, we have to add the taxes. As the project is performed in Spain, we will apply the regular tax, which is 21% (IVA).

Table 5: Total Costs

Type	Costs
Personnel Costs	10.500,00 €
Hardware	146,67 €
Software	120,53 €
Total without indirect costs	10.767,20 €
Indirect costs (10%)	1.076,72 €
Total withour taxes	11.843,92 €
Taxes (21 %)	2.487,22 €
Total	14.331,14 €

1.6 State of the Art

On this chapter, we will analyze which options already exist relationed with the matter of study of this project (i.e. Ecommerce and Business Intelligence), and seek the different possibilities that offer similar characteristics, as well as other technologies that may be useful to the scope of the study. This includes investigation about Ecommerce, Data Analysis, and the available options to publish an Ecommerce shop online.

We will study which solutions exist in order to create and manage an online shop, what is business intelligence and what can be used to accomplish it and finnaly which options we have in order to set all up.

Also, this information will be extended in the following chapters, where I will carry out different software benchmarks.

1.6.1 Ecommerce

Online shopping emerged in the mid-90s, especially with the launch of eBay and Amazon. Traditionally, in order to create an ecommerce, A project would have to be carried out in which the whole shop would have to be programmed. Therefore, technology such as HTML, CSS and JavaScript, as well as a Database system would have to be used for this matter.

This option is messy and takes a lot of time to accomplish. As nearly everything in informatics, this can be 'automatized', and therefore, with time, that automatization started and software platforms for ecommerce emerged, changing completely the market. Now, you can create a basic online shop with just a few clicks. Although manual programming is still being done, now most of the online shopping web pages have a platform behind.

These platforms come with a Front and Back end. While the Front end is what the online visitors will see, in the Back end the Ecommerce owner or the developers are able to manage the online shop: edit the features it contains, manage the products, clients, and everything related with the shop and the web.

Of course, for an online shop, there is not a limited number of features that could be applied. Every online shop is different and requires distinct characteristics. For example, an Ecommerce focused on clothing would require to have a wishlist, while a supermarket that has an online shop may not require it.

This raises the question: What if our platform does not have all the features we need?

Leaving this question in the air for the moment, we will classify the Ecommerce platforms into two types: open source software and proprietary software.

Proprietary Ecommerce software, such as Shopify, will provide a fast and easy installation, but will limit the possibilities of the shop to the functionalities and customization that are already developed in the program. This gives no solution to the previous question. Besides this, Shopify is widely used, as some studies position it as the third Ecommerce used platform [2].

Meanwhile, open source solutions offer a different approach. Using an open source Ecommerce platform, having a notion on programming and web design, new functionalities could be created by any user. These solutions are currently leading the market, having Magento and WooCommerce as their main competitors [3].

Obviously, not every user has the knowledge to program a functionality for the Ecommerce software, and therefore these platforms usually have communities in which users share their solutions. Sometimes, there even exist companies specialized in developing these functionalities in order to sell them online to the community.

To summarize, now we know the difference between programming an Ecommerce manually and using an 'automatized' platform, that provides an already usable online shop. Also, we have identified the differences between using open source and proprietary software, showing the advantages of being able to program functionalities. This being said, we can conclude that the best option is to use an open source Ecommerce platform, in order to be able to provide the needs that the online shop may require.

As there are many options to choose from, in the following chapter we will perform a benchmark to decide which platform fits best our project.

1.6.2 Business Intelligence

Business Intelligence is a set of techniques that transform data into a business analysis that can result in important business decisions. This discipline is divided in several activities in order to accomplish this result. The first section of this activities would be to take all the raw data from the enterprise and transform it in order to unify it and have a central overall database. This can be defined as the Data Integration part.

The second section consist on the business analysis. Once all this data is stored in a solely place, the next step would be to visually analyze it. In order to do this there exist several software applications that, given a database, can produce graphics and dynamic visualizations. With this

visualizations, conclusions can be extracted from the company information, and therefore make decisions from them. This would be the Data Analysis and Visualization part.

In this project, we would only take into consideration the data from the online shop, ignoring other information from the company, as it is fictional and thus, it would be above the scope of this project. Therefore, the Data Integration part is to be omitted on this project, and we will focus on the Data Visualization section.

In the case of Ecommerce, there exist some platforms like the ones we mentioned in the previous sections that offer some basic graphics that can help out with taking business decisions. This is a good although limited approach, as those graphics are predetermined and cannot be edited. Therefore, in order to accomplish a correct solution, we need the freedom that can be accomplished by a Business Intelligence platform, which allows you to make any type of graphic wanted, and explore within the data of the online shop.

As there are plenty of software to choose from, we will do a benchmark to decide which Business Analysis software fits our project the best, as the one we will do in order to select an Ecommerce platform.

1.6.3 Server

In order to publish an online web, we need a server. This server will store the webpage with its files and database. This has to be a secure server as we will have stored all the data from the online shop, and if some hacker can enter into it, the business would be in danger. Also, it has to be a trustable server, in order to minimize the possibility of the server going down if any problem happens or there is a high traffic on the web.

There are several solutions to have a personal server online. First of all, there exist the possibility to create your own server. This can actually be done with a simple computer that is connected to the net and acts as a server. The main problem with this approach is that you will have to buy a domain. As this is very costly and it will be very complicated to acquire the domain name wanted, this approach is out of the scope of this project.

As another possibility, there exist a lot of services that offer servers online. This means that the owner offering the service has a repository of physical servers, and lets you use one via online. This is a much cheaper solution, as it can even be free if you do not have many requirements for your server quality, and it gets costlier as you upgrade those qualities.

Therefore, our best solution is to find a solution like this last one. As I previously pointed, this project is being done while I am involved in a small company as an internship. There, they have encouraged me to use Amazon Web Services, as it is one of the top cloud services for servers, and it has a free instance for a simple server, enough to carry a simple online web. Therefore, this is a suitable solution for the scope of this project.

2 Specific Requirements & Benchmark

In this chapter we will define the specific requirements that a basic Ecommerce should have, as well as the requirements that a fictional company could ask for. The representation for this task will be functional **requirements definition** and **use cases**.

After the specification of the requirements, we will perform a benchmark among the different software solutions for Ecommerce and Business Intelligence. A benchmark is the process of comparing more than one practices to measure its performance and decide the best solutions. In this case, we will perform a benchmark to decide which softwares are currently controlling the market and evaluate them in order to find out which fits this project the best.

2.1 Specific Requirements

As it was said before, the requirements will be specified by the company where I am currently doing an internship, based on their day-to-day work, creating a fictional company with general and common needs for an Ecommerce. This fictional company desires to sell both in Spain and USA, and therefore he will have special needs such as tax management or different currency management.

2.1.1 Functional Requirements

Functional requirements specify what the system must be able to do. In order to enumerate we will write FR and the number of the requirement. In this project, these are the functional requirements specified by our fictional company:

- **FR 1:** The system has to be able to upload and manage products.
- **FR 2:** The system has to be able to manage the shipping methods and its pricing.
- **FR 3:** The system has to be able to accept PayPal and Credit Card as payment methods.
- **FR 4:** The system has to be able to customize client transactional emails.
- **FR 5:** The system has to be able to manage at least two currencies.
- **FR 6:** The system has to be able to manage different taxes policies.
- **FR 7:** The system has to be able to display different languages.
- **FR 8:** The system has to be able to provide a secure connection using SSL internet protocol.
- **FR 9:** The system has to be able to connect with Nacex transport company.
- **FR 10:** The system has to be able to monitor web analytics.

- **FR 11:** The system has to be able to customize invoices dynamically in order to provide information about the company.
- **FR 12:** The system has to be able to have access to a Data Visualization software in order to visualize graphics information in order to perform Business Intelligence.
- **FR 13:** The system has to be able to allow users to register.

Use cases are graphical representation of how our system should work. It represents how the different actors from the system interact between them, defining its behaviour.

With them, we can define the requirements, having a clearer view of how the system is going to work, and helping in finding other requirements that must be implemented in order to accomplish them. Therefore, we will use them to specifically define every function our system will have. We will use the following template to define them:

Table 6: Use Case template

Use Case	Use case name
Description	Brief description of use case's purpose
Actors	Involved actors
Preconditions	Use case preconditions
System flow	1. Action one 2. Action two 3. ...

In this system we will have two types of actors:

- **Administrator:** This actor will be the user that has logged in through the control panel and therefore has administrator permission.
- **Final User:** This actor will be the user that is using the online web.

Now, we will structure our use cases using subsystems in order to clarify the reading. Each subsystem will define a section of our system, which will be defined.

2.1.2 Use Cases

Subsystem Product Management

This subsystem will contain the use cases where any product is handled by the Administrator. The system has to be able to create, edit and delete a product.

Use Case 1: Create a product

The Administrator will be able to create a product in order to add it to the online shop.

Table 7: Use Case 1: Create a product

Use Case	Create a product.
Description	Product creation procedure.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the product section.
System flow	1. The Administrator will select *Create new product*. 2. The Administrator will enter all the general information required for the product: name, categories, description and price. 3. The Administrator will upload the images that illustrate the product. 4. The Administrator will set the product visibility (e.g. if the product will be uploaded to the online shop). 5. The system will save all the information and, if it is visible, it will include the product into the online shop.

Use Case 2: Edit a product

The Administrator will be able to edit product details in order to modify its properties.

Table 8: Use Case2: Edit a product

Use Case	Edit a product.
Description	Product edition procedure.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the product section. At least one product has to be already created.
System flow	1. The Administrator will select a created product. 2. The Administrator will be able to modify the general information, the images and the product visibility. 3. The system will save all the new information and will react to the set visibility if it is changed.

Use Case 3: Delete a product

The Administrator will be able to edit product details in order to modify its properties.

Table 9: Use case 3: Delete a product

Use Case	Delete a product.
Description	Product elimination procedure.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the product section. At least one product has to be already created.
System flow	1. The Administrator will select a created product. 2. The Administrator will select the delete product option. 3. The system will double check if the administrator wants to delete the product. 4. The Administrator will again select the delete product option. 5. The system will delete all the saved information about the product and will delete it from the online shop if it was visible.

Subsystem Shipping Method Management

This subsystem will contain the use cases where the shipping methods are handled by the Administrator. The system has to be able to create, edit and delete a shipping method.

Use Case 4: Create a shipping method

The Administrator will be able to create a product in order to add it to the online shop.

Table 10: Use case 4: Create a shipping method

Use Case	Create a shipping method.
Description	Shipping method creation procedure.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the shipping methods section.
System flow	1. The Administrator will select *Create new shipping method*. 2. The Administrator will enter all the information required about the shipping method: name, pricing options, telephone number, etc. 3. The system will save all the information and will include it into the shipping options.

Use Case 5: Edit a shipping method

The Administrator will be able to edit product details in order to modify its properties.

Table 11: Use case 5: Edit a shipping method.

Use Case	Edit a shipping method.
Description	Shipping method edition procedure.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the shipping method section. At least one shipping method has to be already created.
System flow	1. The Administrator will select a created shipping method. 2. The Administrator will be able to modify all the information about the selected shipping method. 3. The system will save all the new information.

Use Case 6: Delete a shipping method

The Administrator will be able to edit product details in order to modify its properties.

Table 12: Use case 6: Delete a shipping method.

Use Case	Delete a shipping method.
Description	Shipping method elimination procedure.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the shipping method section. At least one shipping method has to be already created.
System flow	1. The Administrator will select a created shipping method. 2. The Administrator will select the delete shipping method option. 3. The system will double check if the administrator wants to delete the shipping method. 4. The Administrator will again select the delete shipping method option. 5. The system will delete all the saved information about the shipping method.

Subsystem Payment Transactions

This subsystem will contain the use cases where a Final User buys a product and is on its way to pay. The system has to be able to provide two basic payment methods: PayPal and Credit Card.

Use Case 7: PayPal payment

The Administrator will be able to edit product details in order to modify its properties.

Table 13: Use case 7: PayPal payment

Use Case	PayPal payment.
Description	The Final User is going to buy a product and decides to pay with PayPal.
Actors	Final User.
Preconditions	The Final User has to have a product on its cart and be on the cart section.
System flow	1. The Final User will select *Pay* option. 2. The System will redirect him to a checkout page. 3. The Final User will enter the required details: billing information, shipping information, shipping method, user details if the user is not registered, etc. 4. As payment method he will choose PayPal. 5. The system will redirect him to the PayPal payment service page and will wait for confirmation from PayPal. 6. The system will proceed with the procedure when it receives the confirmation. If after a few minutes he does not receive confirmation, it will stop the procedure.

Use Case 8: Credit Card payment

The Administrator will be able to edit product details in order to modify its properties.

Table 14: Use case 8: Credit Card payment

Use Case	Credit Card payment.
Description	The Final User is going to buy a product and decides to pay with Credit Card.
Actors	Final User.
Preconditions	The Final User has to have a product on its cart and be on the cart section.
System flow	1. The Final User will select *Pay* option. 2. The System will redirect him to a checkout page. 3. The Final User will enter the required details: billing information, shipping information, shipping method, user details if the user is not registered, etc. 4. As payment method he will choose Credit Card and will fill the Credit Card information. 5. The system will redirect him to the payment service page and will wait for confirmation. 6. The system will proceed with the procedure when it receives the confirmation. If after a few minutes he does not receive confirmation, it will stop the procedure.

Subsystem Currency Management

In this subsystem will appear all the use cases related to managing the currencies. Adding and eliminating currencies is crucial to our business, in case we want to sell in more than one country.

Use case 9: Currency addition

The Administrator will be able to add a new currency to the Ecommerce shop.

Table 15: Use case 9: Currency addition

Use Case	Currency addition.
Description	Adding one or more currency to the Ecommerce shop.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the currency management section.
System flow	1. The Administrator will select *Add new currency*. 2. The Administrator will select one or more currency from a list of allowed currencies. 3. The system will import the rate values from an outside expert web. 4. The system will save the information and apply the changes to the online shop, adding the option to change to the accepted currencies.

Use case 10: Currency removal

The Administrator will be able to remove a currency from the Ecommerce shop.

Table 16: Use case 10: Currency removal

Use Case	Currency removal.
Description	Removing one or more currency from the Ecommerce shop.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the currency management section.
System flow	1. The Administrator will select *Remove currency*. 2. The Administrator will select one or more currencies from a list of added currencies. 3. The system will save the information and apply the changes to the online shop, deleting the option to change to the removed currencies.

Subsystem Tax Management

This subsystem will contain the use cases where the Administrator manages the different taxes from each country and product. This is crucial to sell any product, but also to sell in more than one country. The system has to be able to add taxes rates per country, and apply them to a product depending on its type (e.g.: food usually has a lower tax rate).

Use case 11: Create a Tax rate

The Administrator will be able to create a new Tax rate.

Table 17: Use case 11: Tax rate creation

Use Case	Create a tax rate.
Description	Procedure of creating a tax rate.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the tax management section.
System flow	1. The Administrator will select *Add new Tax rate*. 2. The Administrator will fill the required information about the Tax rate: name, rate, country, and region. 3. The system will save the information.

Use case 12: Tax rate implementation

The Administrator will be able to implement a Tax rate to a set of products.

Table 18: Use case 12: Tax rate implementation

Use Case	Tax rate implementation.
Description	Implement a Tax rate to a set of products.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the tax management section. At least one Tax rate and one product has to be created.
System flow	1. The Administrator will select *Apply a Tax rate*. 2. The Administrator will select a Tax rate from the list of created Tax rates. 3. The Administrator will select a product or a set of products. 4. The system will apply the Tax rule to the selected products.

Use case 13: Delete a Tax rate

The Administrator will be able to implement a Tax rate to a set of products.

Table 19: Use case 13: Delete a Tax rate

Use Case	Delete a Tax rate.
Description	Procedure of deleting a tax rate.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the tax management section. At least one Tax rate has to be created.
System flow	1. The Administrator will select *Delete a Tax rate*. 2. The Administrator will select a Tax rate from the list of created Tax rates. 3. The system will double check if the Administrator wants to delete the tax rate. 4. The Administrator will again select the delete tax rate option. 5. The system will cease the Tax rule to the products it affects, if any, and will delete the information stored about the Tax rate.

Subsystem Language Management

This subsystem will contain the use cases where the Administrator manages the different languages the system can use for the online shop.

Use case 14: Language addition

The Administrator will be able to add a new language to the Ecommerce shop.

Table 20: Use case 14: Language addition

Use Case	Language addition.
Description	Adding one or more languages to the Ecommerce shop.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the language management section.
System flow	1. The Administrator will select *Add new language*. 2. The Administrator will select one or more languages from a list of allowed languages. 3. The system will save the information and apply the changes to the online shop, adding the option to change to the accepted languages.

Use case 15: Language removal

The Administrator will be able to remove a currency from the Ecommerce shop.

Table 21: Use case 15: Language removal

Use Case	Language removal.
Description	Removing one or more languages from the Ecommerce shop.
Actors	Administrator.
Preconditions	The Administrator has to be logged in. The Administrator has to be in the control panel, in the language management section.
System flow	1. The Administrator will select *Remove language*. 2. The Administrator will select one or more language from a list of added languages. 3. The system will save the information and apply the changes to the online shop, deleting the option to change to the removed languages.

Use case 16: Transactional emails management

The Administrator edits the transactional emails the online shop will send to its clients, so it is consistent with the enterprise appearance.

Table 22: Use Case 16: Transactional emails management

Use Case	Transactional emails management.
Description	The Administrator is willing to edit the transactional emails to the enterprise needs.
Actors	Administrator.
Preconditions	The Administrator has to be logged in and in the control panel.
System flow	1. The Administrator selects the *Transactional email* option. 2. The System will send him to a page where he will have all the options to edit the transactional emails template. 3. The Administrator will modify and/or the willing components. 4. The System will save the new transactional emails options.

2.2 Ecommerce Benchmark

2.2.1 First Analysis

Firstly, investigating in the web and using some benchmark documents, we are going to look for the most common Ecommerce platforms in the market, to then compare them taking their specifications and functionalities into account. Mainly, we are going to investigate it using public research benchmarks as well as analysis done by relevant Ecommerce themed blogs.

In the following graphic taken from the NBS System Benchmark [4], done in 2013, we can observe that the mainly used platforms for big companies are Oracle Commerce and IBM WebSphere. As we are willing to create our Ecommerce for a small-medium business, we will take into account the most popular solutions in smaller companies: Magento, PrestaShop, VirtueMart and openCart.

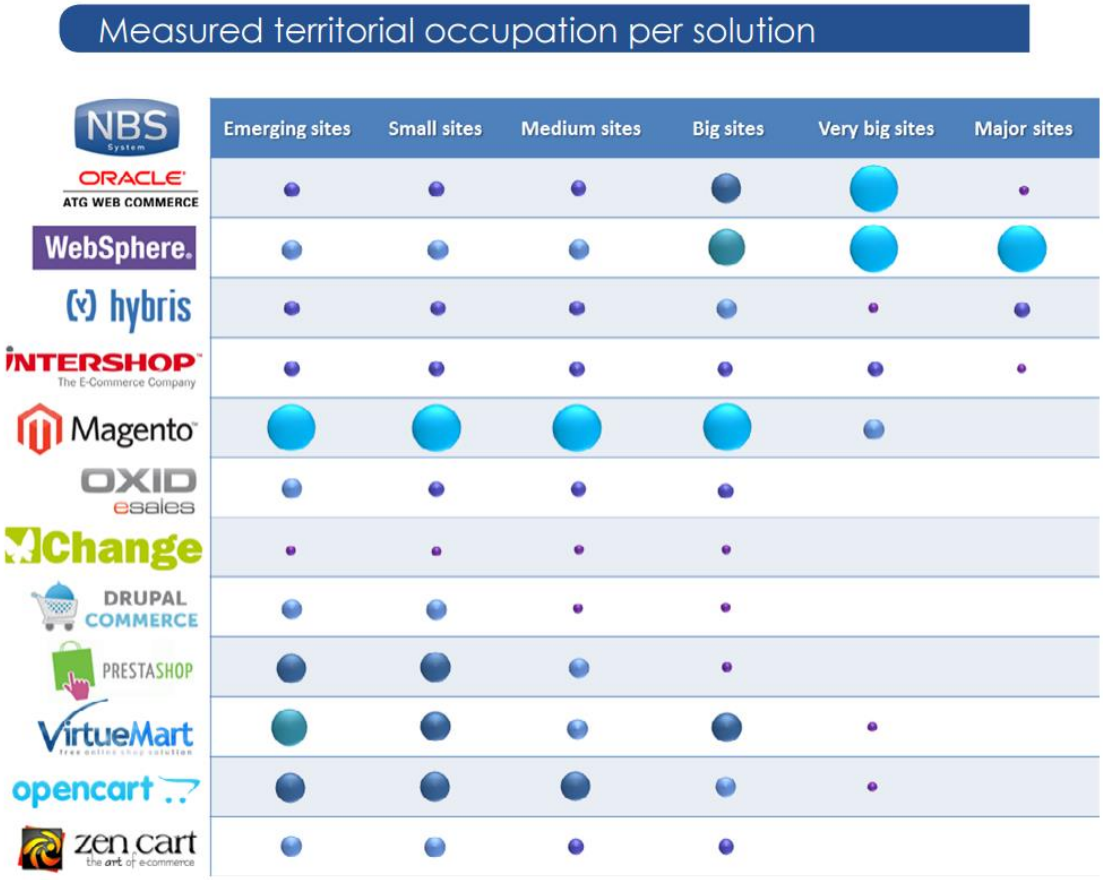


Figure 8: Measured occupation per platform. From: NBS System Benchmark 2013

As the previous benchmark was realized in 2013, we have decided to investigate using Google Trends the current popularity of this solutions, in order to appreciate the main changes that may have happened in the Ecommerce market. Coincidences on popularity can be found with the

analysis done above - Magento, PrestaShop and openCart are again shown to be popular -, although VirtueMart seems to have lost popularity during the last years.

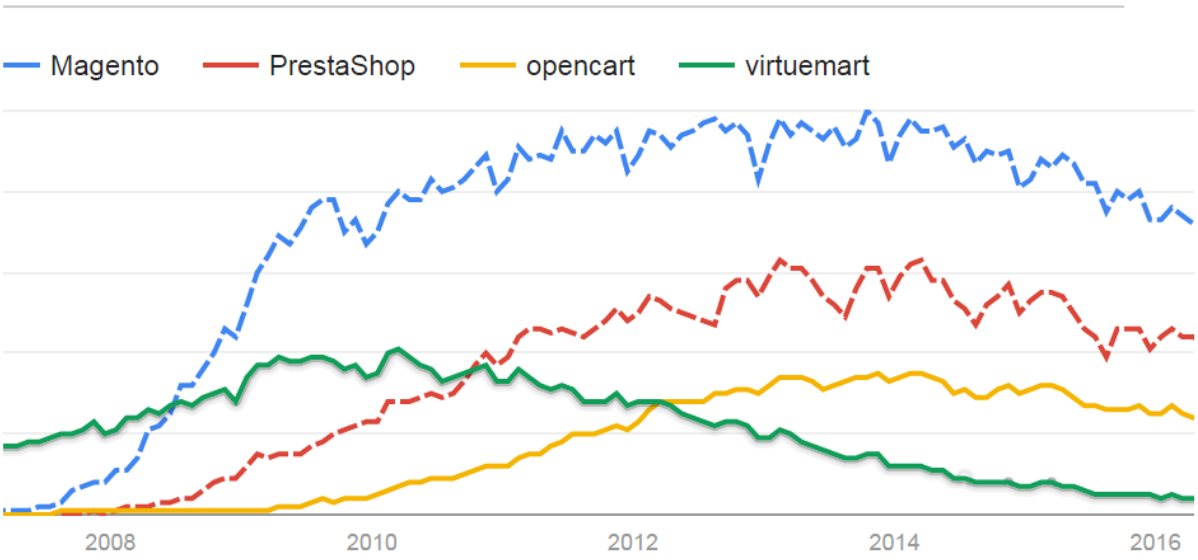


Figure 9: Interest shown towards the main Ecommerce platforms. From: Google Trends

We will also use other analyses that estimate the usage of Ecommerce platforms along the web. We are going to use the analysis done by two sources:

- AheadWorks[2], an Ecommerce blog mainly focused in Magento, which realizes an study of the usage of this platforms every 6 months.
- BuiltWith[3], an online tool that is used to explore the software used on a given webpage. It stores the information about the usage of any software, including Ecommerce platforms, and displays it on a pie chart online [3].

These two sources analyze the Top webpages of the Internet. In order to get these Top ranking, AheadWorks uses Ranking Alexa[4] and selects the Top 1 million. This ranking is a widely-used tool that estimates the overall ranking of a webpage along the entire Internet, in terms of visits. Meanwhile, Builtwith uses its own ranking.

These are the most recent analysis done by them:

[2] http://blog.aheadworks.com/
[3] http://builtwith.com/
[4] http://www.alexa.com/

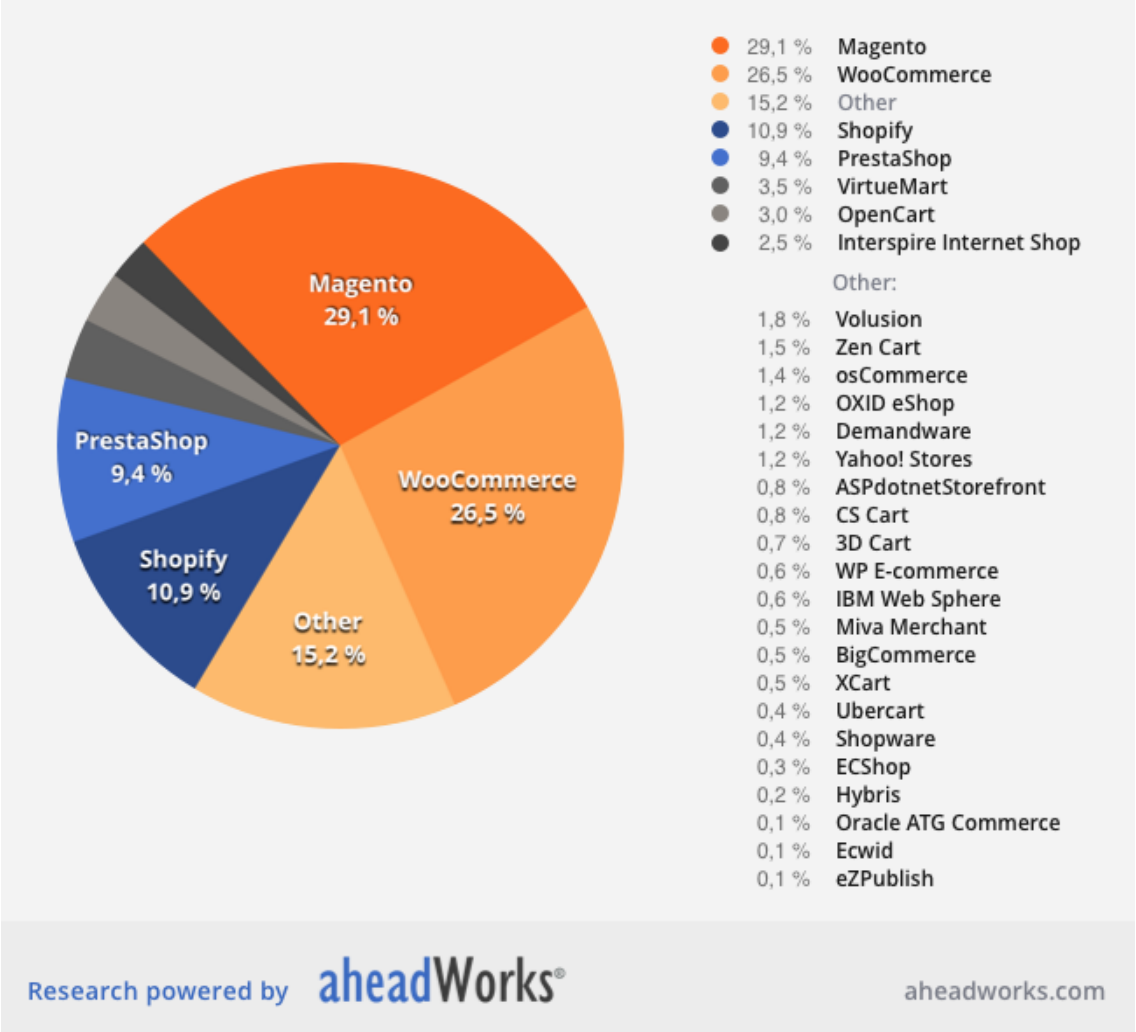


Figure 10: Research about the most used Ecommerce platforms on the Top 1 million ranking by Alexa. Carried out by aheadWorks in March 2016 [2]

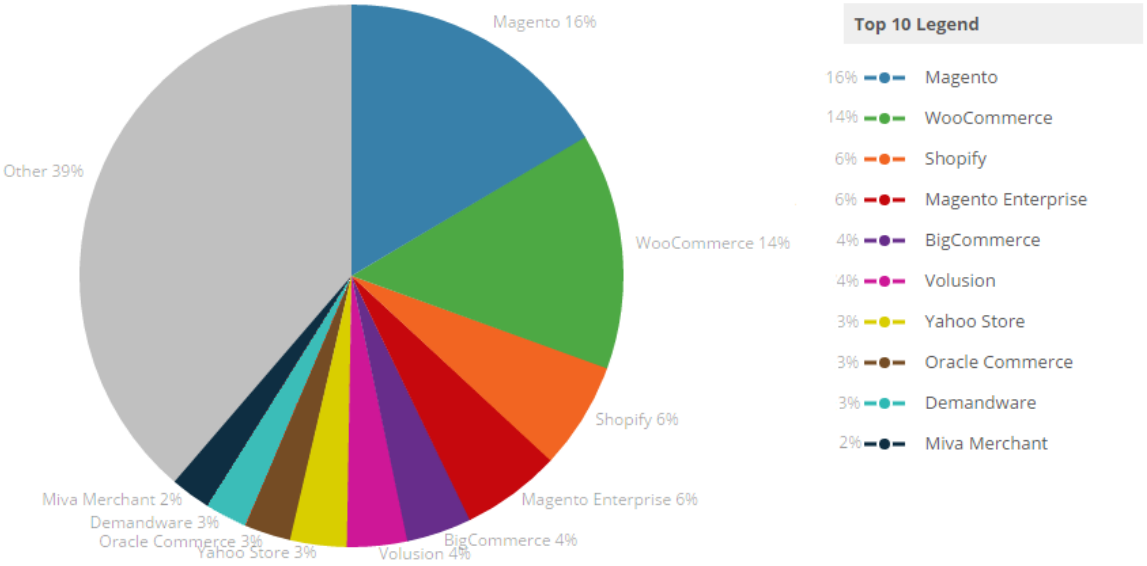


Figure 11: Analysis of the most used Ecommerce platforms on the Top 100k. Carried out by BuiltWith by May 2016 [3]

From these analyses we can verify the downfall of VirtueMart, and see something Google Trends did not tell us, the current low usage of openCart, being given even a lower value than VirtueMart (3,5% vs 3%) in the aheadWorks study. Also, we can advert the existence of two other platforms that we previously did not know, Shopify and WooCommerce.

VirtueMart interest have been dramatically descending during the last 5 years, as the Google Trends analysis told us. Watching other ecommerce performing reports also done by aheadWorks (July 2013 [1], April 2014 [5], October 2014 [6], May 2015 [7]) we can see that not only the evolution in interest tends to decrease, but so does its usage, while only 3 years ago it was the second most used platform and had over 11% of the market [1]. This downfall shows that the software must have gotten behind in characteristics or performance. Therefore, it does not seem a good idea to consider it, and we should discard it.

The case of **openCart** is similar, as its usage has also been decreasing over time. although this platform has never been above 5% of the share. This contrasts with the interest Google Trends displays on it, but this low usage shows low confidence in the platform, and using it could result in a big mistake for our business, due to a possible lack of functionalities or performance.

Shopify has a striking big share of the market (between 6% and 11%) being the third most used platform in both studies; and its popularity has been growing exceptionally since its launch, as it can be seen in Figure 10. Besides this, it is a closed source software. This is a big disadvantage in this type of software.

Although with a closed source software you already have everything done, and you get a fast and easy installation of the online shop, which is desirable, you are tied to the functionalities that already come with it. Meanwhile, with an open source platform, we are able to program our own functionalities, which allows a total customization of the Ecommerce. Also, this facilitates the existence of an online community, where users can share their own solutions, expediting and facilitating the process of installation of functionalities. Therefore, you have two advantages: The software is free, and you have a lot of online help, both in functionalities and blogs.

Then, although we are impressed by the market share performance of Shopify, we would rather go for an open source software, as we may have to implement solutions to comply the specific requirements defined by the user (i.e. the small fictional business).

WooCommerce is a free WordPress plugin that enables you to activate Ecommerce functions in your web page. It is a relatively new Ecommerce software that has irrupted

astonishingly in the market, equalizing and overpassing Magento in terms of usage [8], which prior to this dominated the market for years.

It comes with the basic functionalities needed by an Ecommerce, although it is limited in comparison with other platforms [9] (e.g. it lacks a lot of payment services from start), and requires to install extensions to catch up. In terms of costs it is a good solution for a low budget enterprise or even an individual. Besides this, we do not know yet the scope this solution could reach, due to its newness and its low success in high-value web pages [9]. Also, the online community (blogs and extensions) has not got the opportunity to grow as big as the other leading open source platforms yet, and is still a really young software that still finds some functionality issues, as we previously mentioned.

Therefore, in order to minimize the possibility of finding any errors, we will discard this possibility, also considering that our fictional small business could be able to have a budget big enough for a more complex software. Besides this, it seems to be an interesting software that points to be a potential leader in the future.

Figure 12: Interest shown towards the most used platforms. From: Google Trends

Now, we only have two platforms left, Magento and PrestaShop. They are both open source softwares with a macerated and big community, with a great amount of extensions done by online developers. In order to take the final decision, we are going to analyze them more deeply, investigating their functionalities, advantages and disadvantages. In the following table, we will show the main comparisons between them.

2.2.2 Magento vs PrestaShop

General information	Prestashop	Magento
Turnover to consider the solution	From 0+	> $ 500 K
Current market settlement	Tier 4	Tier 2 - 3 - 4
Editor direction	Tier 3	Tier 2
Profile (B:Brand, R:Retailer, P:Pure player, D:Department store, Pv:Private sales, B2B)	B,P	B,S,R,P
Opensource	Yes	CE Yes / EE No
Editor emphasis	Simplicity	Versatility
Editor emphasis	Cost	Community
Editor emphasis	Community	Costs
We are impressed by	Time to market	Ecosystem
We are sceptical about	Soaring someday ?	Cross Commerce
Min budget to create site (developping from scratch / customizing an example shop)	**10 K€**	**25 K€**
General KPI	**Prestashop**	**Magento**
Cost of the solution (license, integration, hosting)	★★★★★	★★★☆
Flexibility of licensing costs (peak, pay as you grow, on demand, on premise ...)	n/a	★★☆
Time to market (developping from scratch / customizing an existing example store)	★★★★★	★★★
Variety/Quality of demo shops to adapt to a specific use (apart from std demostore)	n/a	n/a
Marketing capabilities (promotion engine, coupon, gift card, etc.)	★★★	★★★★
Level of navigation & catalog presentation (faceted/multi-stores/multilingual/etc.)	★★★☆	★★★★☆
Eco-system (community, partners, integrators, forums, etc.)	★★★★	★★★★★
Backoffice friendliness & ease of use	★★★☆	★★☆
Technical KPI	**Prestashop**	**Magento (CE/EE)**
Development technical complexity (more stars => less complicated)	★★★★★	★★★
Number of third party softwares / extensions / services available	★★★★	★★★★★
Number of complementary product / services from the Editor	★★☆	★★★
Speed of the Front Office (customer web page rendering)	★★☆	★★/★★★☆
Front Office scalability (capacity & cost to serve more customers with less servers)	★★★	★★☆/★★★☆
Speed of the Back Office & scalability (number of simultaneous users)	★★★	★☆
Native CMS capabilities	★★☆	★★
Native Webservice capabilities (Interfacing with third systems, e.g. ERP or Logistic)	★★★★	★☆
B2C Feature list (default, without add-ons or side programs)	★★★★	★★★/★★★★
Mobile (Native App, Hybrid App, Responsive design)	★★★☆	★★★☆
Advanced features KPI	**Prestashop**	**Magento**
Advanced searchandising & user personnalization capabilities	★	★★
Multi / cross / Omni channel capabilities	★★	★
Advanced catalog management (PIM, multi catalog, attributes, bundles, etc.)	★★☆	★★
Integrated or natively interfaced OMS (order management system)	★	★
Advanced marketing tools (adaptive marketing, dynamic navigation tunnels, etc.)	★★	★★☆
Native B2B capabilities	★	★★

Figure 13: Magento vs PrestaShop. From: NBS System Benchmark 2013 [4]

Table 23: Analysis table of Magento and PrestaShop compared. From: Magento and PrestaShop web pages

		Magento	**PrestaShop**
Installation & System Requirements	System Requirements	PHP 5.4+	PHP 5.2+
		MySQL 5.6	MySQL 5.0+
		Linux x86-64	Windows / Mac / Linux
	Web Server	**Web Server** - Apache 2.x - Nginx 1.7.x	**Web Server** - Apache 1.3 - Apache 2.x - Nginx - Microsoft IIS
	Versions	Free Version (CE)	Free Version
		Paid Version (EE)	Paid Version
Market Targets	Enterprises Size	Small Enterprises	Small Enterprises
		Medium Enterprises	Medium Enterprises
		Big Enterprises	Big Enterprises
		Major Enterprises	Major Enterprises
Development	Installation difficulty	Medium	Easy

	Open Source Development	Open Source Development	Open Source Development
	Community Modules	Community Modules	Community Modules
	Module difficulty	Medium	Easy
Functionalities ***- Front End***	Products	Cart	Cart
		Wishlist	Wishlist
		Product search	Product search
		Product comparator	Product comparator
		Opinions	Opinions
		Rating	Rating
		Cross-Selling	Cross-Selling
		Up-Selling	Up-Selling
	Orders	Orders history	Orders history
	Social Networks & Content Management	Product sharing	Product sharing
		Newsletter	Newsletter
		RSS	RSS
	Appearance	Mobile version	Mobile version

		Responsive	Responsive
		Multi-language	Multi-language
Functionalities ***- Back end***	Products	Product Management	Product Mgmt
		Inventory Mgmt	Inventory Mgmt
		Product batches Mgmt	Product batches Mgmt
	Orders	Orders Mgmt	Orders Mgmt
	Logistics	Shipment Mgmt	Shipment Mgmt
	Prices	Price Mgmt	Price Mgmt
		Promotions	Promotions
		Discount coupons	Discount coupons
		Taxes	Taxes
	Statistics	Basic Statistics Package	Basic Statistics Package
	Maintenance	Back-up	Back-up
Training & Support		24/7	24/7
		Online	Online
		In Person	In Person

		Live Online	Live Online
		Online Videos	Online Videos
		Documentation	Documentation

Table colors meanings:

- **Green:** it has the described item.
- **Red:** it does not have the described item.
- **Blue:** defined specifications, descriptions.

In functionalities, both softwares provide the basics an Ecommerce would need; manage products, establish prices, create categories, etc. But if we want to go deeper into advanced functionalities, Magento is ahead. It already has product comparators or wishlists among other ones, while in PrestaShop we would have to find the appropriate extensions and install them.

From the table above we can conclude that, while PrestaShop is easier to use and customize, in contrast to Magento. Therefore, if we are looking for a fast and easy implementation for the Ecommerce, PrestaShop would be a good option, although a web based on Magento will be more complete and, eventually, will have more chance to produce more profit.

Both softwares have a great focus on their community, both being able to download templates and extensions (functionalities) done by other members of the community. This increases to an immense number of possibilities for the Ecommerce, and makes it easier to implement any new functionality. In this feature, Magento is better than PrestaShop, as their community is much bigger and is more compromised.

Besides this, in case we would want to create our own module, it would be much easier to implement it in PrestaShop rather than in Magento.

2.2.3 Conclusion

Finally, with this analysis we can conclude that although using PrestaShop would be easier, Magento is a much more complete and attractive software, having installed more of the required functionalities than PrestaShop, and having for the rest of them a larger community that can provide us with much more extensions in order to refine the system. Although it may

take more time to implement an extension in Magento than in PrestaShop, it will be easier to find the one that fits and works for our specifications.

Therefore, Magento Community Edition will be the chosen Ecommerce platform for this project.

2.3 Business Intelligence Software Benchmark

In order to carry out this benchmark, we will use analysis done by Gartner Inc. It is a company specialized in doing technology research and they are famous for their analysis about Business Intelligence software.

To have a base in which to study, we will start with the most recent Magic Quadrant [10]. The Magic Quadrant is a way of classifying software used by Gartner, which is divided in four sectors: Leaders, Visionaries, Niche Players and Challengers. Using their own explanation [11]:

- **Leaders** execute well against their current vision and are well positioned for tomorrow.
- **Visionaries** understand where the market is going or have a vision for changing market rules, but do not yet execute well.
- **Niche Players** focus successfully on a small segment, or are unfocused and do not out-innovate or outperform others.
- **Challengers** execute well today or may dominate a large segment, but do not demonstrate an understanding of market direction.

Taking this into account, we will not consider Niche Players, as they may not cover all our needs. In the following Magic Quadrant, that considers Business Intelligence platforms as of February 2016, we have platforms only as Leaders and Visionaries to consider.

Microsoft Power BI, Tableau and Qlik are the main leaders of this Quadrant, having a lot more to consider in Visionaries, where we can appreciate the inclusion of SAP, SAS, IBM, Pentaho or TIBCO Software.

Figure 14: Magic Quadrant for Business Intelligence and Analytics Platforms by Gartner as of February 2016

Now that we have chosen which Ecommerce platform to use, we are restricted to its characteristics. As we are using Magento, we have to consider that it stores all the information about the online shop in a MySQL database. Therefore, our Data Visualization app will have to be able to connect to a database of such kind.

Business Intelligence applications are very complex and costly to create, and therefore there exists very few cases where a free or open source version is available. In the following table we analyze this:

Table 24: Business Intelligence free software (as of 1Q 2016)

	Free / Open Source Version	Free Trial
Tableau	Yes	Yes
Qlik	Yes	Yes
Microsoft Power BI	No	Yes
Alteryx	No	Yes

SAS	No	Yes
SAP	No	Yes
MicroStrategy	No	Yes
IBM Watson Analytics	No	Yes
LogiAnalytics	No	Yes
ClearStory Data	No	Yes
Pentaho	Yes	Yes
TIBCO Software Jaspersoft	Yes	Yes
BeyondCore	No	Yes

Only five programs on the scope of our analysis are free or provide a free version. We will make two assumptions to continue with the analysis:

- Assume that the company wants to use a free version to minimize costs, assuming the risk of having less features.
- Assume that our company is willing to afford a priced version (at least for one user, which is usually around 1000$ and 2000$).

Therefore, we will continue the Benchmark distinguishing these two assumptions. We will look for a free software and a priced one that can accomplish our goals, and we will choose the best option for the project.

2.3.1 Free Software

In this section we will analyze the five softwares we have found that can be acquired for free: Tableau, QlikSense, Pentaho and TIBCO Software Jaspersoft.

From these softwares Jaspersoft and Pentaho are the least valued ones, and we may have to look into them more deeply. While Jaspersoft is a regular BI software, Pentaho provides the particularity that works on a Tomcat server.

This particularity distinguishes Pentaho from these types of softwares, as, if you install Pentaho on a server, you will be able to use it from any computer without the need of installing it on your own. Therefore, we will investigate the possibility of using this software, and try to applicate it on our project to provide business analysis.

Tableau has a free version that is quite powerful, having two important limitations: You can only save your work online, and you only have a few database connections. From the

available database connections, MySQL is not permitted. As we stated before, this requirement is crucial, and therefore we cannot consider Tableau free version.

SAS is usually a priced software [12], although it has a university free edition [13]. This software is more centered in use for statistics, more similar to R [14] than to a usual visual analytics software. In the Gartner analysis, the software considered classified in the Visionary sector is the priced one. Therefore, we will not consider SAS free version.

Taking this into account, we can conclude that the most attractive free version softwares for the scope of our project are QlikSense and Pentaho. QlikSense is a complete BI software that can be acquired for free (as a non-Enterprise version). Pentaho meanwhile, is interesting as we could install it in the same server where we have Magento to use it online.

2.3.2 Priced Sotware

In the case of the priced software we have eight platforms to choose from, but as we eliminated Tableau from the previous section due to its lack of MySQL connection, we will also consider its priced version.

As there are many softwares to consider, we will mainly focus on the Leaders Quartile (Tableau and Microsoft Power BI) as they are the ones that have more probabilities of complying our needs and be the most user friendly.

From this two softwares, Tableau is the most meaningful tool for our project, as it is mainly focused on the Visualization part of Business Intelligence, while other platforms (including Microsoft Power BI) spend a lot of resources in the Database part (which includes Data Warehouse, ETL, OLAP...). Although some data configurations can be fulfilled in Tableau using Calculated Fields, it is recommended to shape your data outside of Tableau [15]. But, as we do not want to perform any data configuration in this project, but only Data Visualization, this platform fits satisfactorily in our project.

Also, Tableau has a version for Students [16], which is a one-year full version. We will use it as a simulation on a bought full software.

2.3.3 Conclusion

To conclude, we should take into account three softwares for our final decision: QlikSense, Tableau and Pentaho.

Pentaho is valued really low in terms of usability. Being an open source software with not such a big community as Magento has, makes the installation, configuration and general usage more complicated. Also, it can overload the server we are using for this project, as it does not support a high capacity.

Comparing QlikSense and Tableau, the second one surpasses QlikSense in terms of Data Visualization. As pointed before, QlikSense spends resources in a part more centered in Database Warehouse, ETL or OLAP. Also, Tableau is valued better in terms of usability in the Magic Quadrant. Therefore, we will once again prioritize Data Visualization capabilities in this decision.

As we can simulate Tableau's full version using Tableau for Students, we will use this advantage

To conclude, we will test Tableau, as we can simulate Tableau's full version using Tableau for Students, we will use this advantage. Due to its great capabilities for Data Visualization, we will use this software, making the assumption that the fictional company is willing to afford a full version.

3 Implementation

3.1 Application Design

Now that we have decided which platforms we are going to use, we need them to work together. In order to do this, we have to analyze which possibilities do we have for them to interact together and decide which design fits better into our system.

3.1.1 Proposed Architecture

We have come to the conclusion that our applications can interact in four different ways, which are briefly explained on Figure 15. We will now analyze them one by one.

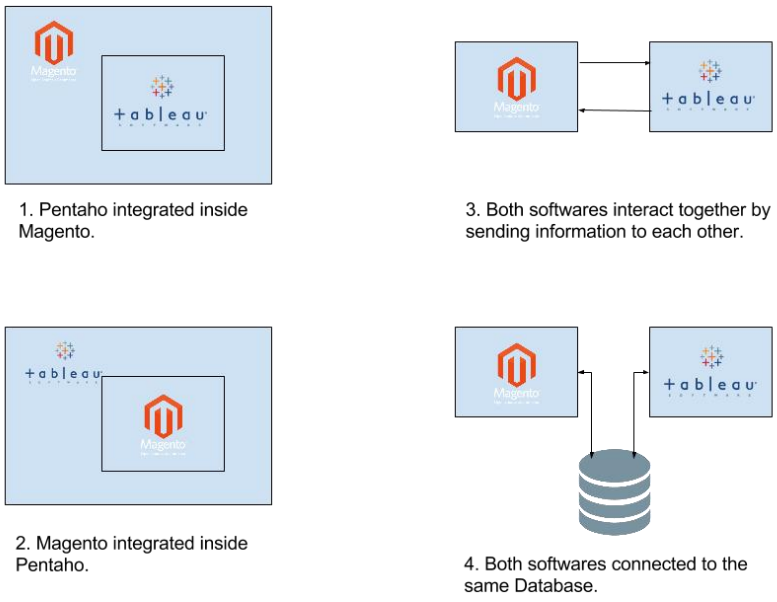


Figure 15: Possible architecture interactions for the system

As a first analysis, we will explore the possibility or improvement each of the architectures can provide us.

For the first option, we would have to implement a Magento module that creates a window connected to our Tableau. As Magento is an open source software, we could do this, and therefore we will consider this option for now.

For the second case, we would have to do the same as before but inversed, creating a Tableau module where you could connect to Magento. But, as Tableau is a closed environment software, we could not program over it. Also, as the Business Intelligence is intended to be an extension for the Ecommerce administrator to manage its business and act as a decision support

service, it would make more sense to implement it the other way around, as in the first option. Then, we are going to discard the second architecture.

In the third option, we would have to use web services in order to connect both platforms. Currently, Magento is able to use SOAP and REST [17], while Tableau does not support it. Therefore, this possibility cannot be implemented.

For the last case, both softwares should be able to connect to the same database. Magento stores all its information on a MySQL database. Therefore, for the connection to be possible we need Tableau to be able to use such a database. As we can see on the Tableau website [18], MySQL is one of the many database possibilities the software can deal with.

After this brief analysis, we can conclude that the second option is the most useless one. Therefore, we now have to analyze the advantages of the remaining integrations.

While the first option would be more time consuming, this fourth option seems to be the less time consuming. As in a project, we have to look for the easiest and fastest solutions, always considering that the needed requirements are accomplished. Of course, we cannot take only this into account. Therefore, we will also analyze which advantage can this architecture have against the first architecture.

In conclusion, with the last solution, we would have one database on our server, where both Magento and Tableau connects. As Tableau connects directly, it is constantly updated and, from the user point of view it can be more useful, as there is no need to enter into the Magento administrator panel to use it, but directly to the Tableau one.

3.1.2 Final Architecture

Now, having selected the architecture we are going to use, we can describe it in the following Figure 16.

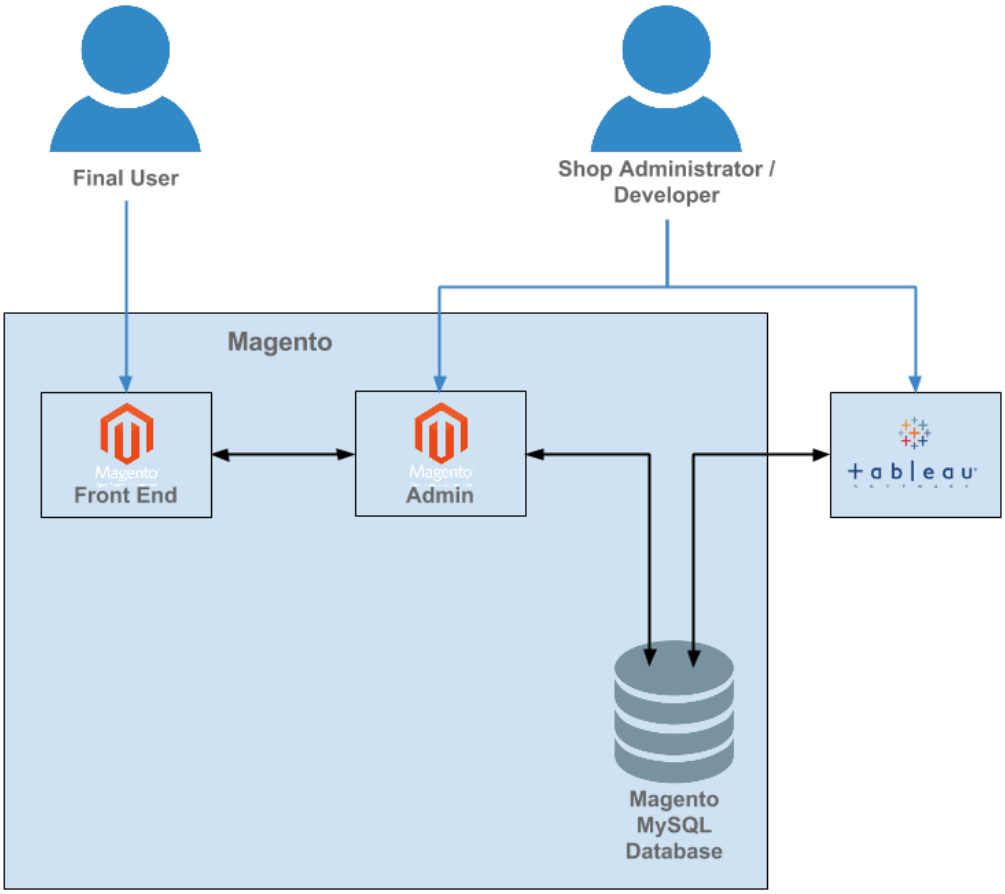


Figure 16: Final Architecture of the system

The shop administrator or a developer from the small business will have access to both Tableau and the Magento administrator panel, where it can make changes in the shop and create graphics in order to perform a business analysis. Tableau will be connected to the Magento database, so it is always up to date.

The final shop user will only see the front end, i.e. the online shop, where he can open an account and perform orders. Once these orders are performed, Magento will manage and store them in the database.

3.2 Ecommerce Implementation (Magento)

In this section we will explain how to implement the online shop using Magento Community Edition platform. For this goal it will be required to acquire an online server with LAMP architecture in which to host Magento, and install it. Also the needs by the small fictional company should be implemented, as well as the later maintenance of the software.

3.2.1 Server

The first thing we will need is a server that is able to host our online shop. As discussed in the State of the Art chapter, we will be using a hosting service for this goal; in this case, Amazon Web Services. We will open an account and create a Linux Amazon Instance. With this, Amazon Web Services provides us with a server on the web, which we can control from the AWS Console.

As Magento works in a LAMP server environment, we will then connect to it through SSH and FTP, and install it. For this purpose, we will use tools such as Putty (SSH) and FileZilla (FTP). With SSH we will be able to control our server using a virtual console from our personal computer. There we can install the required software to have a LAMP architecture server (Linux, Apache, MySQL and PHP). The processes of installation on the server (SSH, FTP and LAMP) are explained in the annexes 5 and 6.

The server requirements (i.e. Magento System Requirements) are Linux x86-64; Apache 2.x; PHP 5.4.x or 5.5.x; and MySQL 5.6 [19].

3.2.2 Magento

Now that we have our online server configured, we can proceed to install Magento. We will download the software from the Magento download page[5]. In this project we will use Magento version 1.9.2.4, from February 23, 2016.

Once downloaded we will transfer the file to the server storage (using FTP) to the server directory (in our Apache2 the directory is */var/www/html*). With this we can perform a basic installation (see Annex 2) and we will already have an online shop. Of course, this online shop has a low usage, as it lacks of:

- Products
- Payment methods

[5] https://www.magentocommerce.com/download

- Shipping methods
- Navigation tools
- Search engine
- Customized layout

And many other configurations that can be performed.

Every configuration of the Magento online shop is done through the administration panel, which is configured while the Magento Installation (see Annex 2). By default, it is in the path *http://pathtomagento/index.php/admin*.

3.2.3 Requirements Fulfilled

Magento functions can be divided in Core, Community and Local functions. The Core functions are already provided by the main Magento installation, while Community functions are acquired through their online community[6] (see Annex 3 to know how to install a community extension), and Local are functions manually programmed.

In this section we will investigate which requirements are fulfilled by the Core functions, and which could be fulfilled with Community or Locally programmed extensions.

Core

Of course, Magento comes with basic Ecommerces functionalities. Therefore, most of the functional requirements are fulfilled. Of course, it lets you manage products (FR1), shipping methods (FR2), paying methods (FR3), transactional emails (FR4), currency management (FR5), Tax policies (FR6), language configuration (FR7), SSL (FR8), invoices (FR11) and users (FR13).

For the remaining functional requirements (FR9, FR10, FR12) we will need to find other solutions.

Community

For functional requirements 9 and 10, we will use the following extensions:

- **FR 9:** The system has to be able to connect with Nacex transport company. As this is a usual feature, there already exist extensions in Magento Connect. The following can be used:
- https://www.magentocommerce.com/magento-connect/extension-de-nacex.html

[6] https://www.magentocommerce.com/magento-connect/

- **FR 10:** The system has to be able to monitor web analytics. There exist many web analytics services, although the most used one is Google Analytics, as it is very efficient and it is free. The following extension is the best valued in Magento Connect about Google Analytics.
- https://www.magentocommerce.com/magento-connect/google-analytics-by-fooman.html

Finally, to accomplish FR12, which is one of the main features of this project (*The system has to be able to have access to a Data Visualization software in order to visualize graphics information in order to perform Business Intelligence.*) we will explain its implementation in the following sections, using Tableau as we previously stated in the chapter of Specific Requirements and Software Benchmarks.

3.2.4 Maintenance

The last step would be to provide future maintenance of the online shop. This includes the following tasks:

- Upload and update catalog and products.
- Software updates. Every few months Magento delivers a new version, which is required in order to correct security failures and add new features.
- Backup: Perform backups in order to minimize the risk of losing the business information.
- Fault corrections from initial versions.
- Adaptations to changes in legislation, such as privacy laws.
- Resolutions of errors, for both the web or the server.
- Instruction on the platform use.
- Solve doubts about software handling and configuration.

3.3 Business Intelligence Implementation (Tableau)

In this section we will implement Tableau. We will explain how to install it and how to connect it to Magento applying the architecture chosen in the previous section Application Design. In that section we concluded that the best solution is to connect Tableau to Magento's database using a MySQL connection. Therefore, we will also explain how to configure correctly Tableau and the database to enable the connection between them.

To know how to install Tableau, see Annex 7.

Tableau is an interactive and user-friendly software to perform analysis. When opening Tableau, we will have a panel with all the possibilities to connect to our data. It can be used files such as MS Access, Excel, CSV etc. and many online connections, as it can be seen in the following picture.

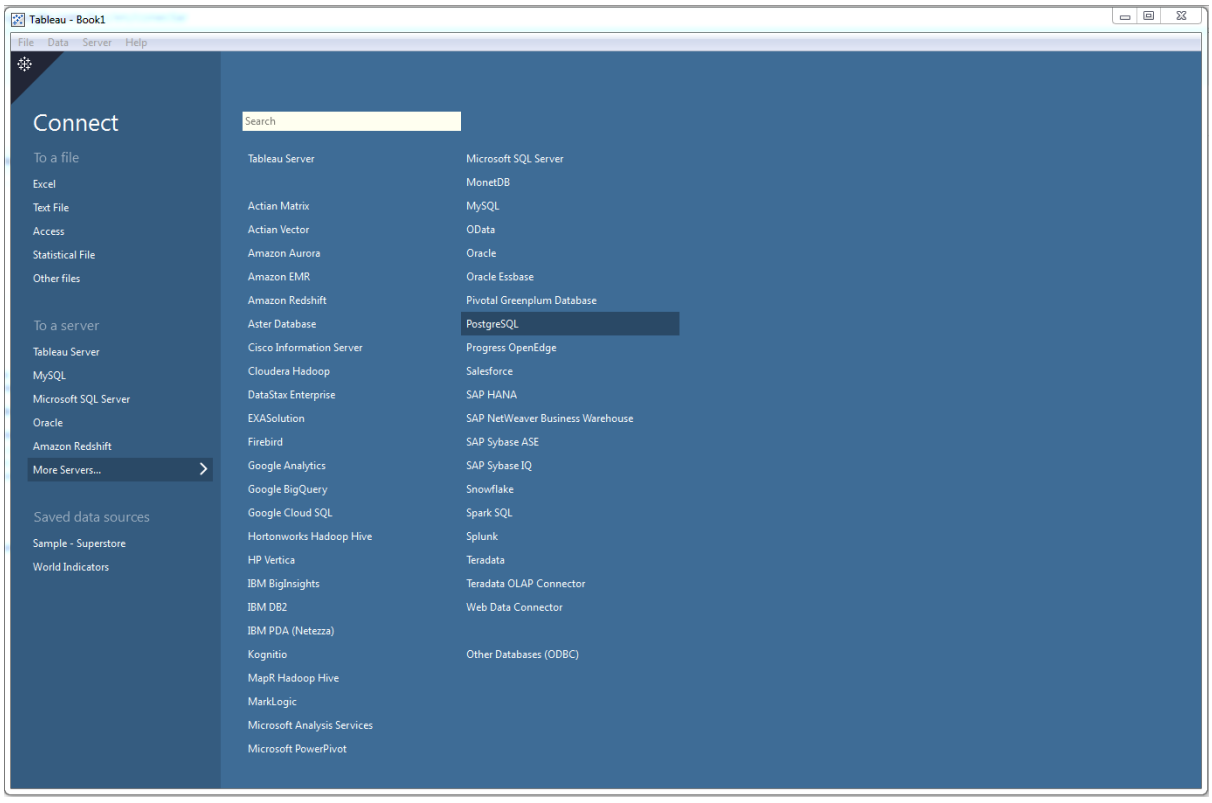


Figure 17: Tableau initial panel

We will probably have to previously install the latest MySQL ODBC connector[7] to use Tableau on our Database. Then, we will have to click on the MySQL and enter the required data (host IP, user and password).

[7] https://dev.mysql.com/downloads/connector/odbc/

By default, MySQL does not permit external connections. This means that, in order to connect to a MySQL database, we will have to be on the same host as that database. Therefore, as we will be connecting to a database that is installed on an online server, we will need to change this first.

In order to change this setting, we will open phpMyAdmin on our server and we will create a new user with usage privileges that can be connected from anywhere. As this decreases the security of the server and the online shop, we can configure this user to improve its security.

Of course, the user and password should be private and we could use a username that does not correspond to the default one (*root*). In this case we have called him *mage_tfg*. Also, when creating the user, we can configure it to have access only to the databases we want, in this case the Magento one.

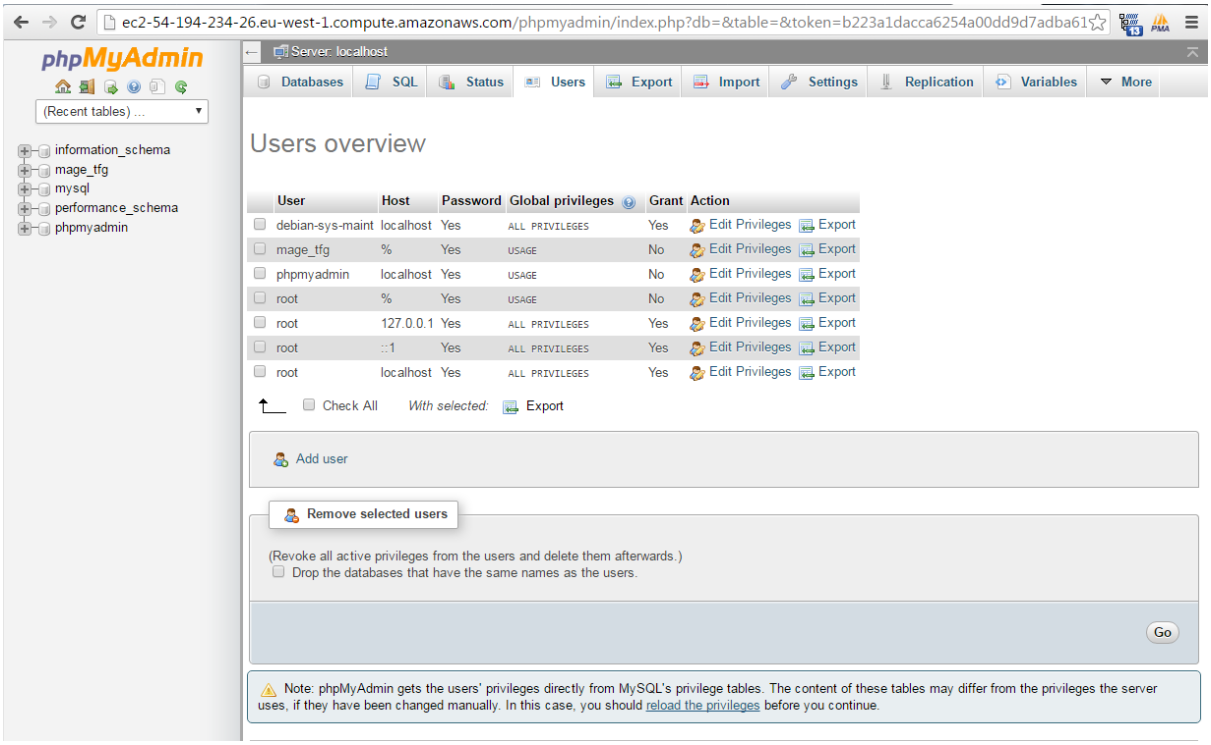


Figure 18: phpMyAdmin users section

With this step completed, we can connect to the database from Tableau. In the next section, we will have to select exactly which data are we going to use. For this purpose, Tableau lets us select the databases we have access to and the tables we want to use. With these tables, we can choose different ones selecting the joins that unify them. In Figure 19 we can see an example, where we join two tables through their joining foreign key. It can also be seen how the selection of databases and table is performed.

In the following section we will analyze Magento database and how to complete these connections and extract information from the database.

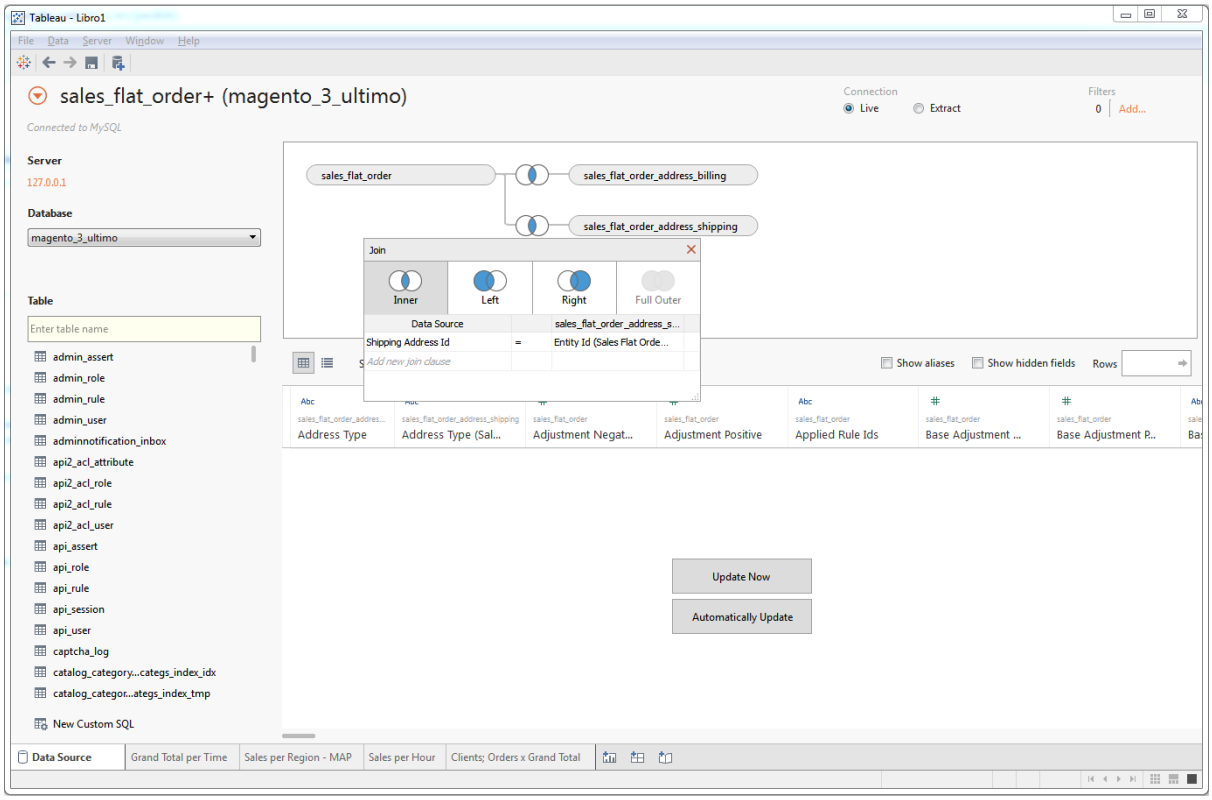


Figure 19: Tableau Data Source panel. Table join example

3.4 Ecommerce and Business Intelligence Integration

In this section we will explain how to connect both softwares using the architecture we chose in the previous section Application Design. Therefore, we will explain how to connect Tableau to the Magento database.

Also, we will explain how to connect to the different tables and how to extract information from them, showing then charts that can be created with the extracted data to analyze the business and how have they been created.

3.4.1 Connection

The Ecommerce database we want to connect to is on an online server. Therefore, we will have to perform an ODBC connection (Open DataBase Connectivity) in the case of Tableau. These are two access standard for database connections.

To connect we will have to fill the requirements:

- Hostname
- User
- Password

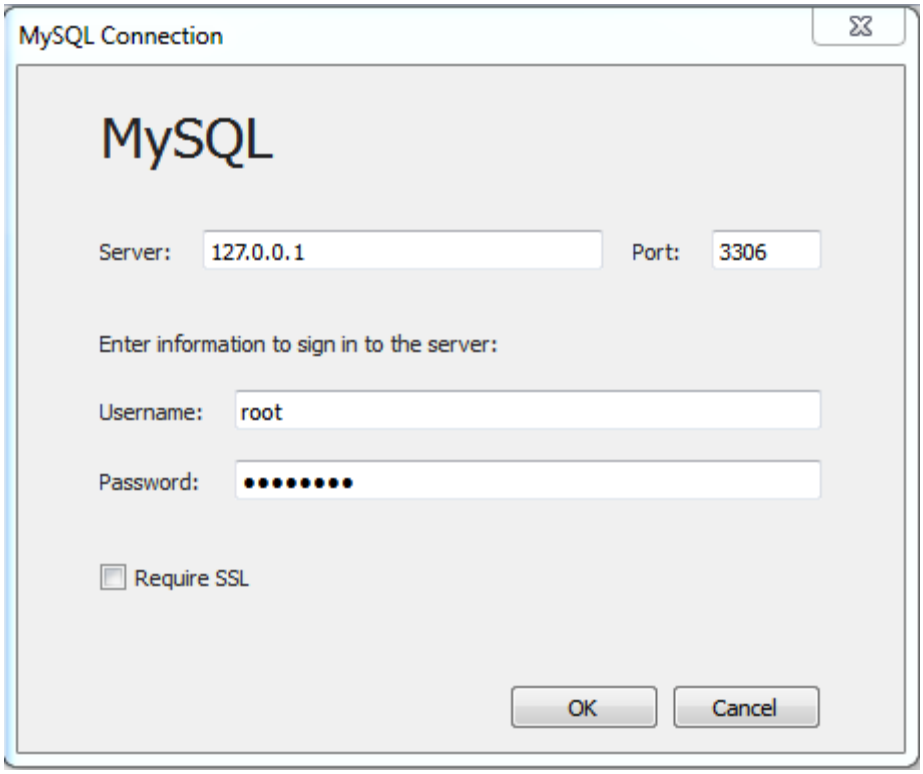


Figure 20: MySQL Connection example

And we will be connected to all the databases the user has permission to see. In our case, we will only need the connection to our Magento database.

Each software has its particularities to perform this connection, and they both need the installation of each connector. To see a deeper explanation on how to connect them, see previous section *Business Intelligence Implementation*.

3.4.2 Magento Database

Now, in order to create charts that analyze our business, we will have to explore how Magento stores the information and how its database works.

Magento has a huge over-structured database, with more than 200 tables The number of tables depends on the Magento version and the extensions installed, as they can modify the database. Thanks to Anna Voelkl, who did an analysis of the 1.9.2.2 version database, we can look into its structure and extract which parameters will we need [20].

Figure 21: Magento database structure. Version: 1.9.2.2

As we are using version 1.9.2.4, this database is good enough for our scope, as there are no changes in the main tables, as we have checked while creating graphics.

As it can be seen in the image above, Magento database is enormous and a complete chaos, and we will have to analyze it carefully. The most important information we will have to extract are data about Sales and Products. In this image, the tables about Sales are in the right part of the image (light blue square), and the tables about Products are in the square on the top left (light yellow), named Catalog.

Now, we will analyze the main tables of this section and how to join them. As some tables have over 50 or even 100 attributes, we will select the ones that are useful for our project. For each table we will add a description and a table with all the relevant attributes, describing them and indicating their possible values.

3.4.3 Magento Tables

Table 25: Magento tables and relevant attributes

Table	Relevant Attributes	Description	Possible Values
sales_flat_order Stores the information about orders processed. It saves one tuple per order.	Entity Id (PK)	Order identifier	Integer
	State	Order status	{cancelled, closed, complete, new, pending_payment, processing}
	Grand Total	Total paid	Integer
	Billing Address Id	Billing address identifier	Integer
	Shipping Address Id	Shipping address identifier	Integer
	Customer Id	Customer identifier. Null if the sale is done by Guest user	Integer
	Customer Email	Customer Email	String
	Created At	Order creation date	YYYY-MM-DD HH:MM:SS
sales_flat_order_address Stores information about shipping and billing addresses.	Entity Id (PK)	Address identifier	Integer
	Address Type	Address type	{billing, shipping}
	Country Id	Country identifier	ISO 2-char code

	Region	Region name	String
	City	City name	String
	Postcode	Postcode	Integer
sales_flat_order_item Stores information about products per order.	Entity Id (PK)	Order item identifier	Integer
	Order Id	Order Identifier	Integer
	Product Id	Product identifier	Integer
	Name	Product name	String
	Price Incl Tax	Product total price	Float
catalog_product_entity Stores information about products.	Entity Id (PK)	Product identifier	Integer
	Entity Type Id	Type of item (in this case, product, which is 4)	Integer
catalog_category_entity Stores information about product categories.	Entity Id (PK)	Category identifier	Integer
	Entity Type Id	Type of item (in this case, categories, which is 3)	Integer
catalog_category_product Stores connections between products and categories.	Category Id (PK)	Category identifier	Integer
	Product Id (PK)	Product identifier	Integer

Figure 22: Selected tables from Magento database

3.4.4 Magento Table Connections

In Tableau, we have to select the tables in the Data Source Panel and perform the required joins between them, in order to extract the needed data. In Business Intelligence, this selections are referred to as Data Marts. Now, we will explain which Data Marts have we selected and how have we made the connections.

Data Mart 1: Sales and addresses

In this Data Mart we have selected all the sales (*sales_flat_order*) and the addresses of each sale (*sales_flat_order_address*). With this information, a lot of analysis can be done, as we have information about sales, the customers that performed them and the addresses to be sent.

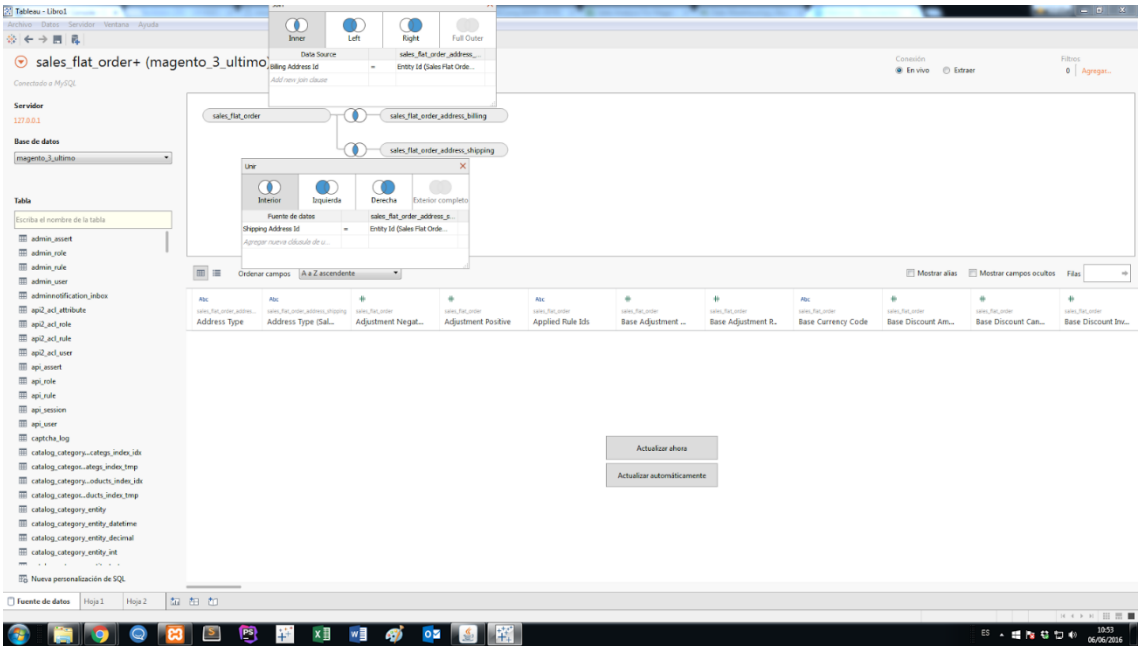


Figure 23: Data Mart 1 Sales and addresses

Table 26: Data Mart 1 Connections

	sales_flat_order	sales_flat_order_address
Connection 1: Shipping Address	Shipping Address Id	Entity Id (PK)
Connection 2: Billing Address	Billing Address Id	Entity Id (PK)

The table *sales_flat_order* stores as *Shipping Address Id* and *Billing Address Id* the identifiers of the addresses for shipping and billing. These are stored in *sales_flat_order_address*. Therefore, joining them through their identifier, we will have a Data Mart in which, per tuple, we will have all the information about the order, including the shipping and billing addresses.

Data Mart 2: Product and categories sales

In this Data Mart we have again selected the sales (*sales_flat_order*) but distinguishing the different products (*catalog_product_entity*) and the different categories on it (*catalog_category_entity*).

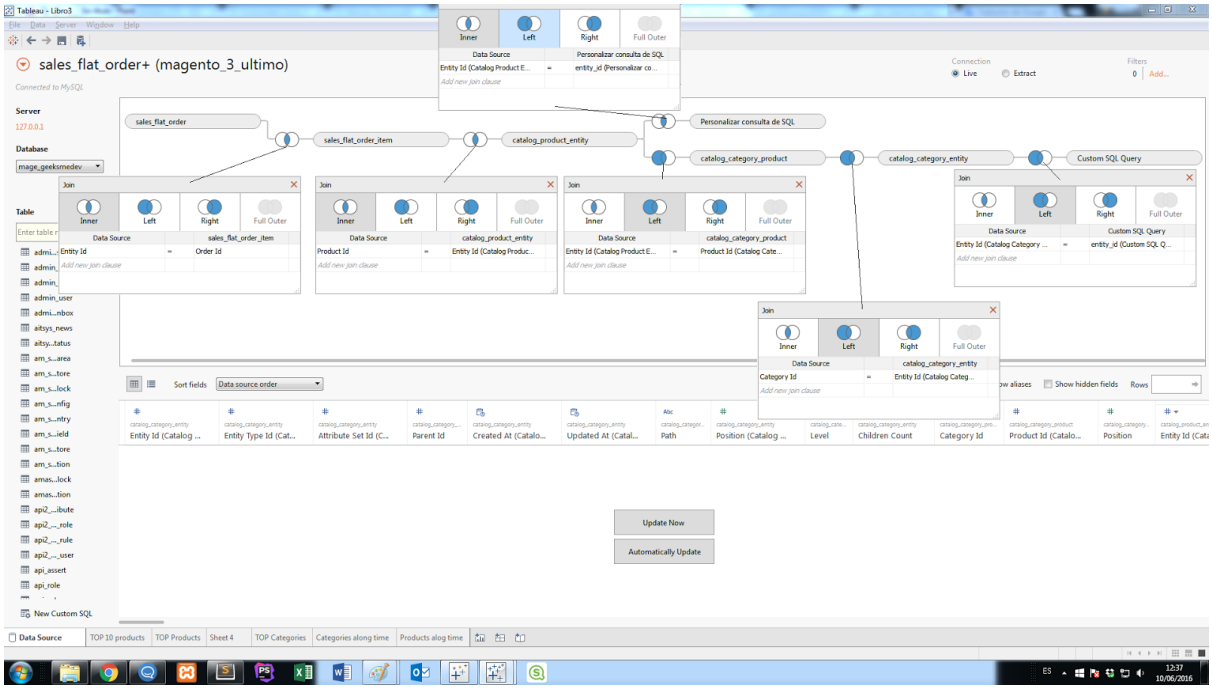


Figure 24: Data Mart 2 Product and category sales

Table 27: Data Mart 2 Connections

	sales_flat_order	**sales_flat_order_item**
Connection 1: Sales Items	Entity Id (PK)	Order Id
	sales_flat_order_item	**catalog_product_entity**
Connection 2: Products	Product Id	Entity Id (PK)
	catalog_product_entity	**Custom SQL Query 1**
Connection 3: Product Name	Entity Id (PK)	Entity Id
	catalog_product_entity	**catalog_category_product**
Connection 4: Product to Category	Entity Id (PK)	Product Id
	catalog_category_product	**catalog_category_entity**
Connection 5: Category	Category Id	Entity Id (PK)
	catalog_category_entity	**Custom SQL Query 2**
Connection 6: Category Name	Entity Id (PK)	Entity Id

In this case, to extract the product and categories name, we have to perform two different queries. This is because of the way Magento stores this information, which is in *catalog_product_entity_varchar* and *catalog_category_entity_varchar* respectively. In this tables Magento stores every data that involves a String about the Products/Categories, including its name. In this queries it can be clearly seen the over-structuration of Magento. To search the name we have to go to the EAV tables (Entity Attribute Value) and extract information such as type of attribute and entity type, to search the coincidence in *catalog_*_entity_varchar* and extract the wanted type of attribute. The SQL code for these queries are:

Custom SQL Query 1 (Product name):

```
SELECT `value` AS product_name, `entity_id`
FROM catalog_product_entity_varchar
WHERE entity_type_id = (SELECT entity_type_id FROM eav_entity_type WHERE
entity_type_code = 'catalog_product')
AND attribute_id = (SELECT attribute_id FROM eav_attribute WHERE
attribute_code = 'name' AND entity_type_id = (SELECT entity_type_id FROM
eav_entity_type WHERE entity_type_code = 'catalog_product'))
```

Custom SQL Query 2 (Category name):

```
SELECT `value` AS category_name, `entity_id`
FROM catalog_category_entity_varchar
WHERE entity_type_id = (SELECT entity_type_id FROM eav_entity_type WHERE
entity_type_code = 'catalog_category')
AND attribute_id = (SELECT attribute_id FROM eav_attribute WHERE
attribute_code = 'name' AND entity_type_id = (SELECT entity_type_id FROM
eav_entity_type WHERE entity_type_code = 'catalog_category'))
```

The rest of the attributes that form the joins are self-explanatory. As a side note, the joins for category have to be Left-joins instead of Intersectional-joins as there exist object that do not belong to a category, and an intersection would result in the elimination of the product rows that do not belong to a category. With the Left-join, these are not erased and therefore the problem is solved.

3.4.5 Graphics Implemented

In this section we will show and explain the different graphics that have been performed for a Magento Ecommerce using Tableau and the previously defined Data Marts.

For this purpose we have used sample data from a Magento Ecommerce that is currently shutted down.

As you will see, creating graphics with Tableau is a rather simple task, although you will need some practice to be fast at it. First, you have the *Columns* and *Rows* parameters, where you can put attributes and they will be shown graphically. Then, to ease the use we have the windows *Show me* where we can select custom charts, which will be represented if we have the required type of attributes on screen. Finally, probably the most customizable window is *Marks*, where you can alter the graphics, separating by colors, shape, size, or just to distinguish depending on certain attributes.

Also, Tableau lets you to filter using the window *Filters*, where you can move an attribute and select which parameters from that attribute are considered and not considered. Also, all these parameters are customizable (colors, size, label, etc), letting us built the charts as customized as we want.

One very important feature of Tableau is the ability to create *Calculated Fields*, where, using the attributes we already have, we can program a new one, using Tableau's programming code, which is simple and has a full explanation of all functions next to the programming box. To create the graphics we are about to show, we have created the following *Calculated Fields*.

Country name. Checks the ISO codes in *Country Id* from table *sales_flat_order_address* and transforms it in countries names. We have only transformed the ones used in our sample data for this case:

```
CASE [Country Id]
WHEN "AU" THEN "Australia"
WHEN "BE" THEN "Belgium"
WHEN "CA" THEN "Canada"
WHEN "ES" THEN "Spain"
WHEN "FR" THEN "France"
WHEN "GB" THEN "Great Britain"
WHEN "HR" THEN "Australia"
WHEN "IT" THEN "Italy"
WHEN "LU" THEN "Australia"
WHEN "US" THEN "United States of America"
END
```

Customer Guest/Signed-in. Distinguish between Guest and Signed-in users. It separates users in those two options, checking the attribute Customer Id (if it is null, the user does not have an account):

```
IF ISNULL([Customer Id]) THEN "Guest"
ELSE "Signed-in" END
```

Data Mart 1 Graphics

Graphic 1: Aggregated Grand Total over time

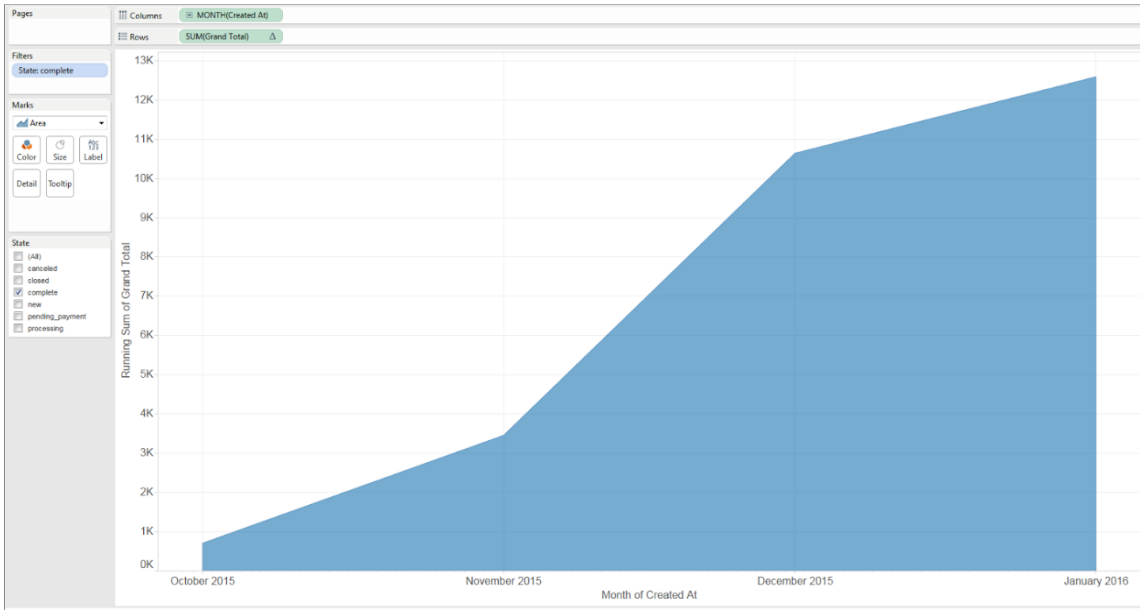


Figure 25: Aggregated Grand Total over time

In Figure 22 we can see how the money gained by sales increases over time, aggregating the total money.

We have included a filter to show only the orders that are classified as *complete*, from attribute *State*.

Used attributes:

- CreatedAt (Month and year) (table *sales_flat_order*)
- Grand Total (table *sales_flat_order*)
- Count of orders Ids in *sales_flat_order* (Total number of orders).
- State (Filter: completed) (table *sales_flat_order*)

Graphic 2: Grand Total and Sales over time (Months)

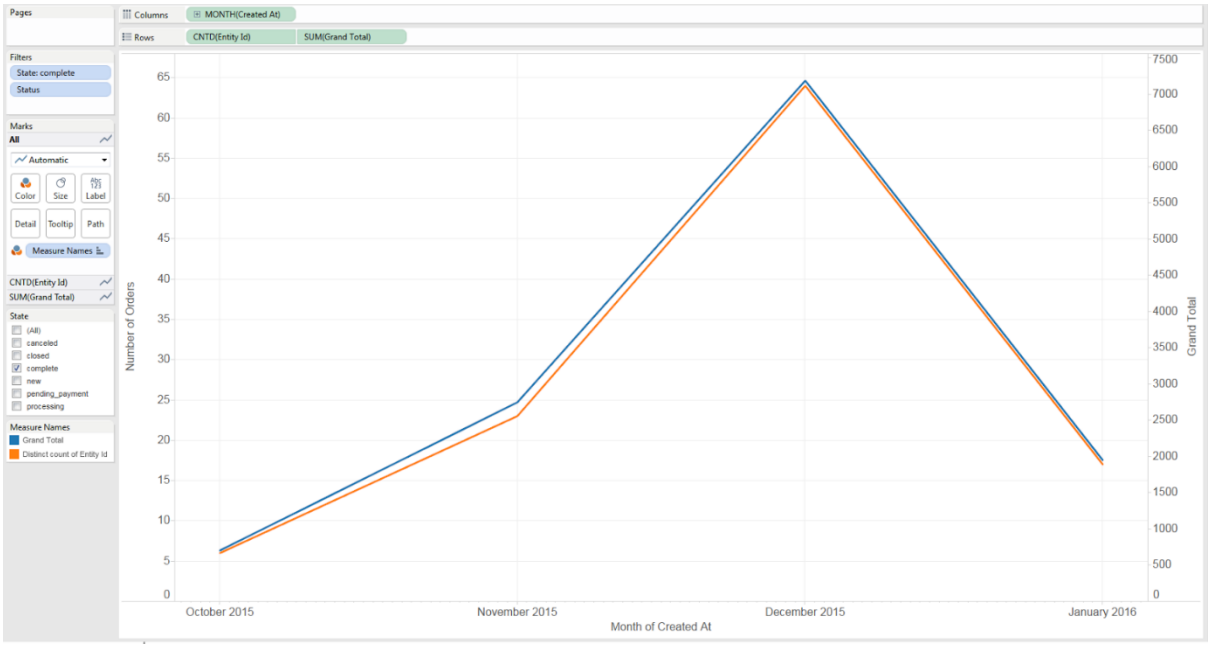


Figure 26: Grand Total and Sales over time

In Figure 26 we have the first graphic. It shows the total earned money and the total amount of orders performed per month. In this case, we only have data from October 2015 to January 2016.

We have included a filter to show only the orders that are classified as *complete*, from attribute *State*.

Used attributes:

- CreatedAt (Month and year) (table *sales_flat_order*)
- Grand Total (table *sales_flat_order*)
- Count of orders Ids in *sales_flat_order* (Total number of orders).
- State (Filter: completed) (table *sales_flat_order*)

Gaphic 3: Grand Total and Sales over time (Days)

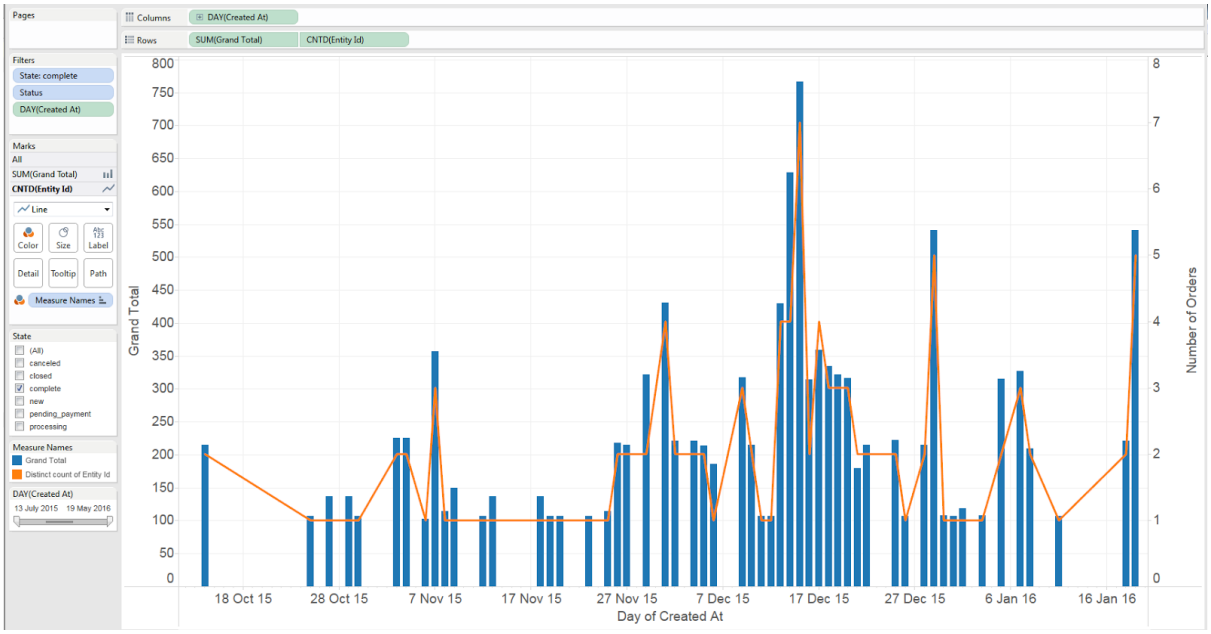


Figure 27: Grand Total and Sales by days

This graphic has the same idea as the last one, although this time we have divided it by days and used a bar chart. The blue bars indicate the money acquired that day (Grand Total), and the orange line indicates the number of orders. We have only considered completed orders.

Used attributes:

- CreatedAt (Day) (table *sales_flat_order*)
- Grand Total (table *sales_flat_order*)
- Count of orders Ids in *sales_flat_order* (Total number of orders).
- State (Filter: completed) (table *sales_flat_order*)

Graphic 4: Sales per Weekday

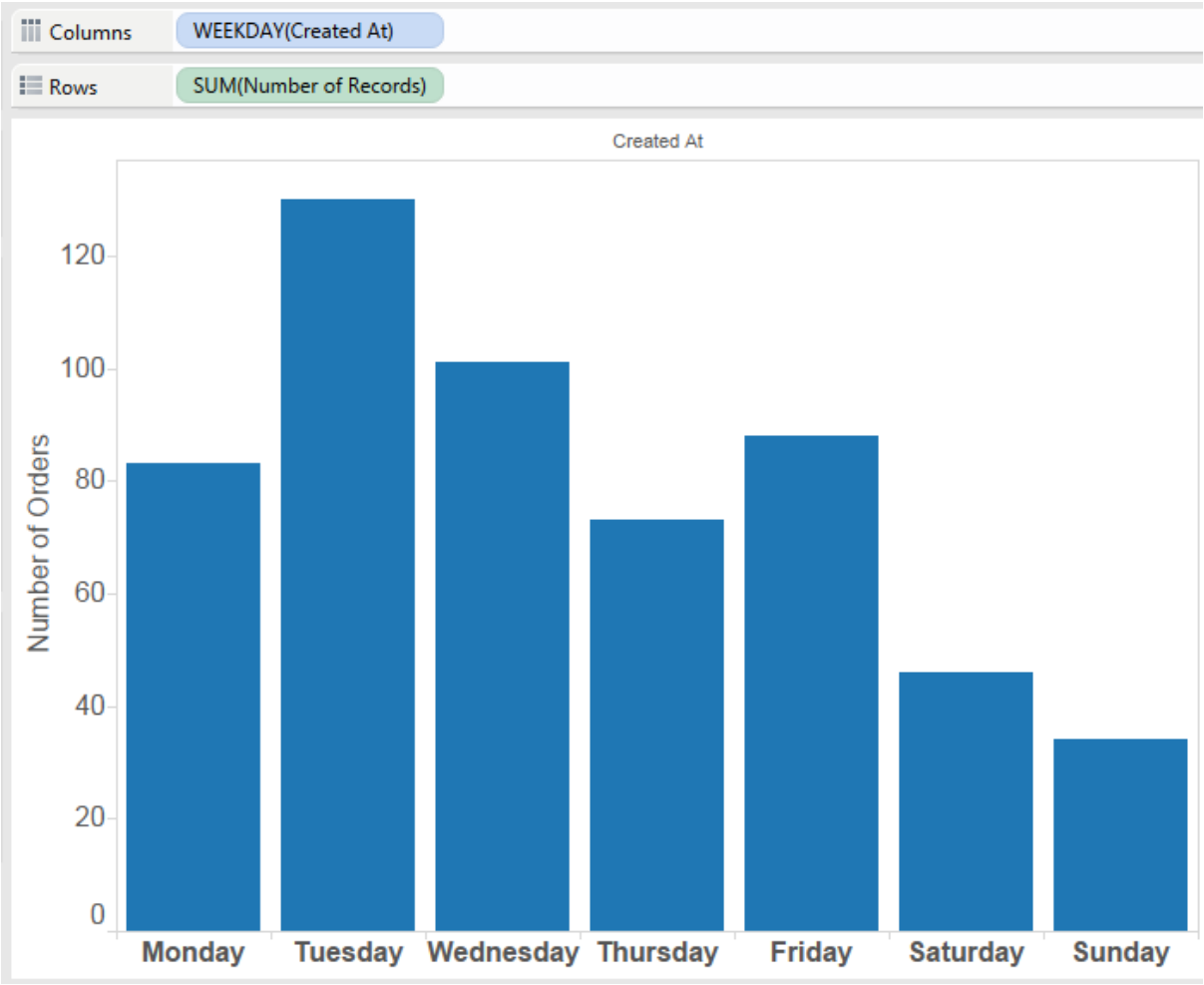


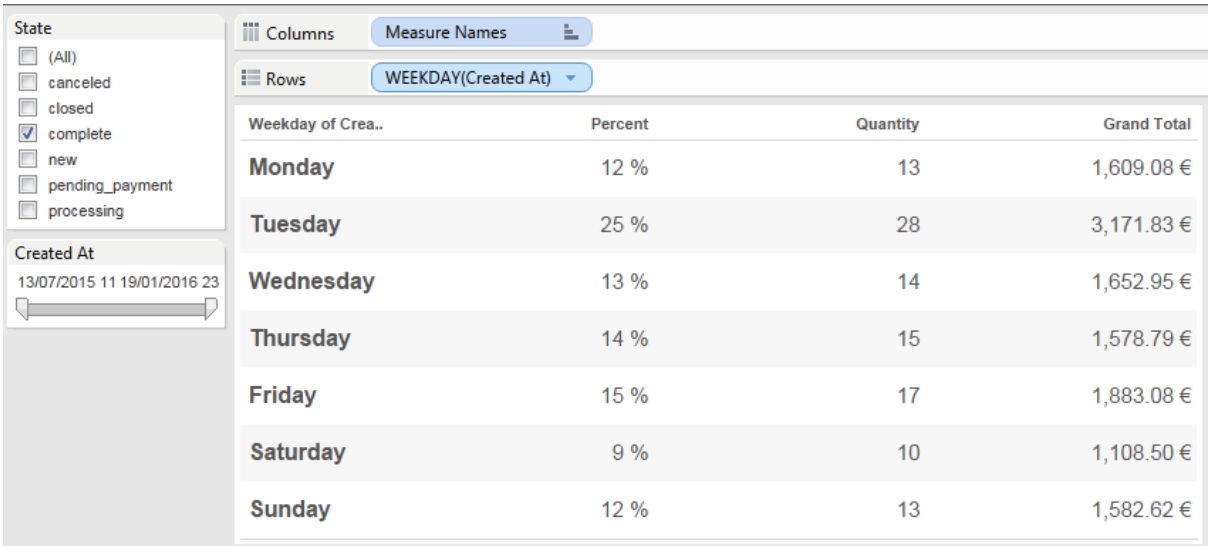
Figure 28: Sales per Weekday

Here we can see the trend for the most common days to perform orders in our Ecommerce. In this case we are considering every type of order as we want to analyze in which days people are connected to our Ecommerce to perform operations.

Used attributes:

- CreatedAt (Weekdays) (table *sales_flat_order*)
- Count of orders Ids in *sales_flat_order* (Total number of orders).

Graphic 5: Sales per Weekdays Analysis



Weekday of Crea..	Percent	Quantity	Grand Total
Monday	12 %	13	1,609.08 €
Tuesday	25 %	28	3,171.83 €
Wednesday	13 %	14	1,652.95 €
Thursday	14 %	15	1,578.79 €
Friday	15 %	17	1,883.08 €
Saturday	9 %	10	1,108.50 €
Sunday	12 %	13	1,582.62 €

Figure 29: Sales per Weekdays analysis

In this table we make a further analysis on the previous Weekdays sales bar chart, analyzing the percent of sales per weekday, the number of orders and the total money spent. We have filtered only the completed orders, as those are the one that have generated money, and added a filter to select the range of dates in which this analysis can be seen.

Used attributes:

- Created At (Weekdays) (table *sales_flat_order*).
- Count of orders Ids in *sales_flat_order* (Total number of orders).
- Calculated field that calculates the percentage of count of orders.
- Grand Total (table *sales_flat_order*).
- State (Filter: completed) (table *sales_flat_order*)
- Created At (Filter)

Graphic 6: Most frequent sales per hour

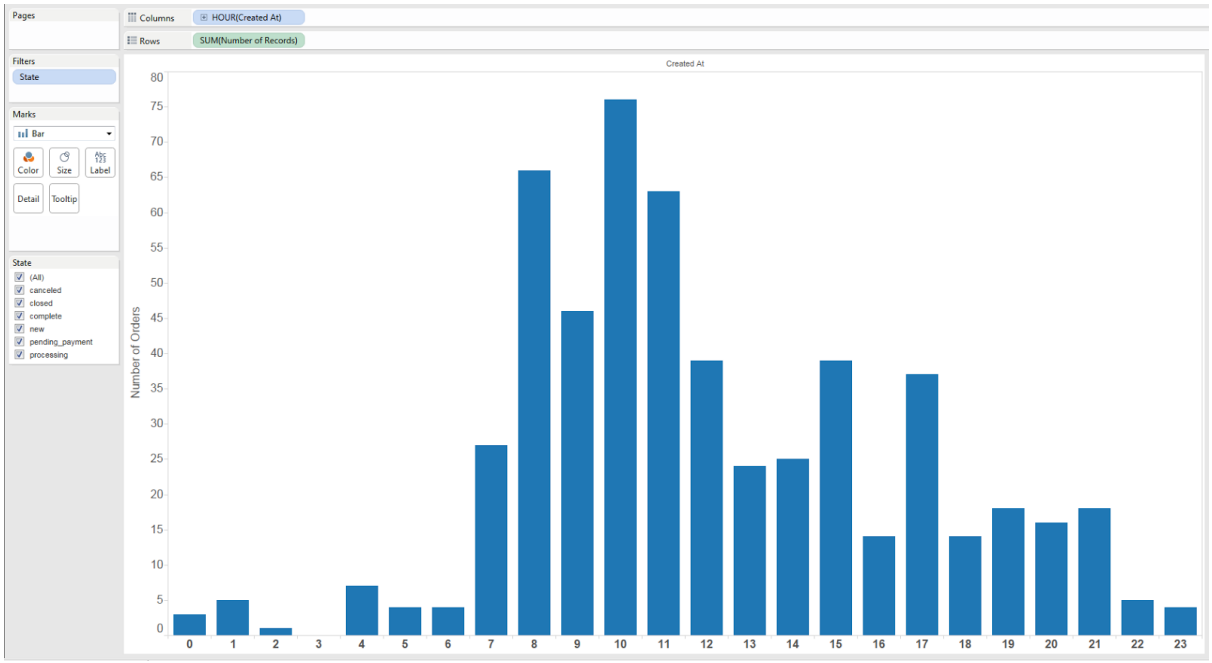


Figure 30: Most frequent sales per hour

In this graphic we can see which hours are the most frequent in which people perform orders. In this case we are considering every type of order as we want to analyze at which hours people are connected to our Ecommerce and perform operations.

Used attributes:

- Created At (Hour) (table *sales_flat_order*).
- Count of orders Ids in *sales_flat_order* (Total number of orders).

Graphics 7: Sales per City and Country

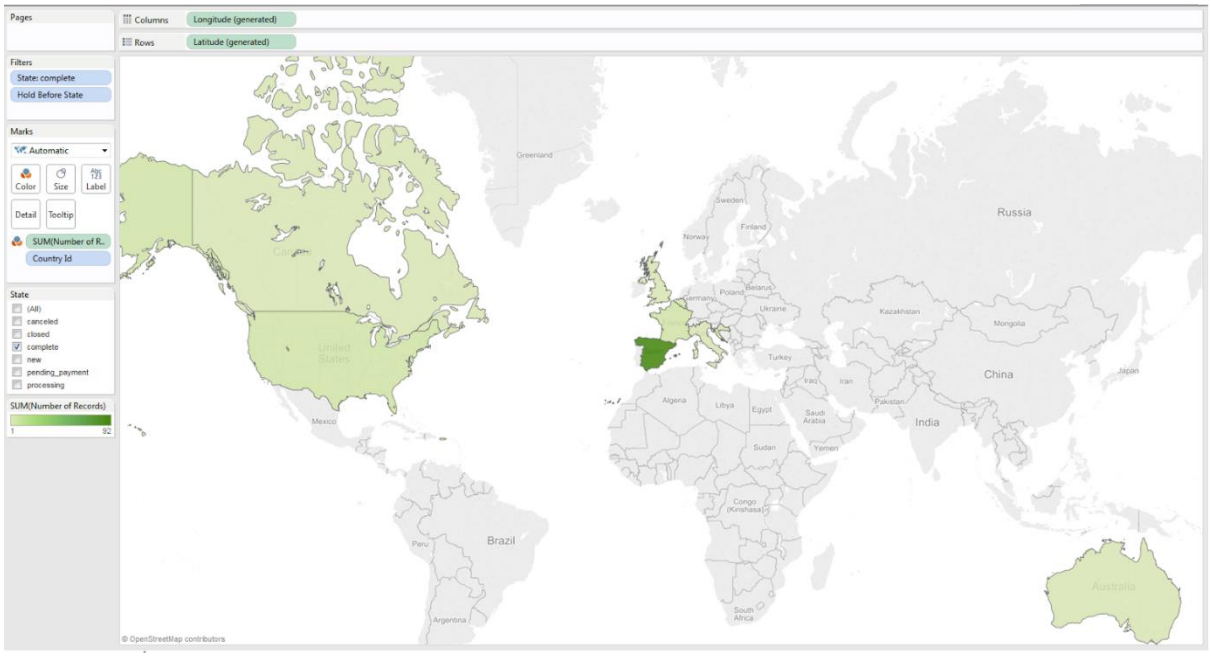


Figure 31: Sales per Country

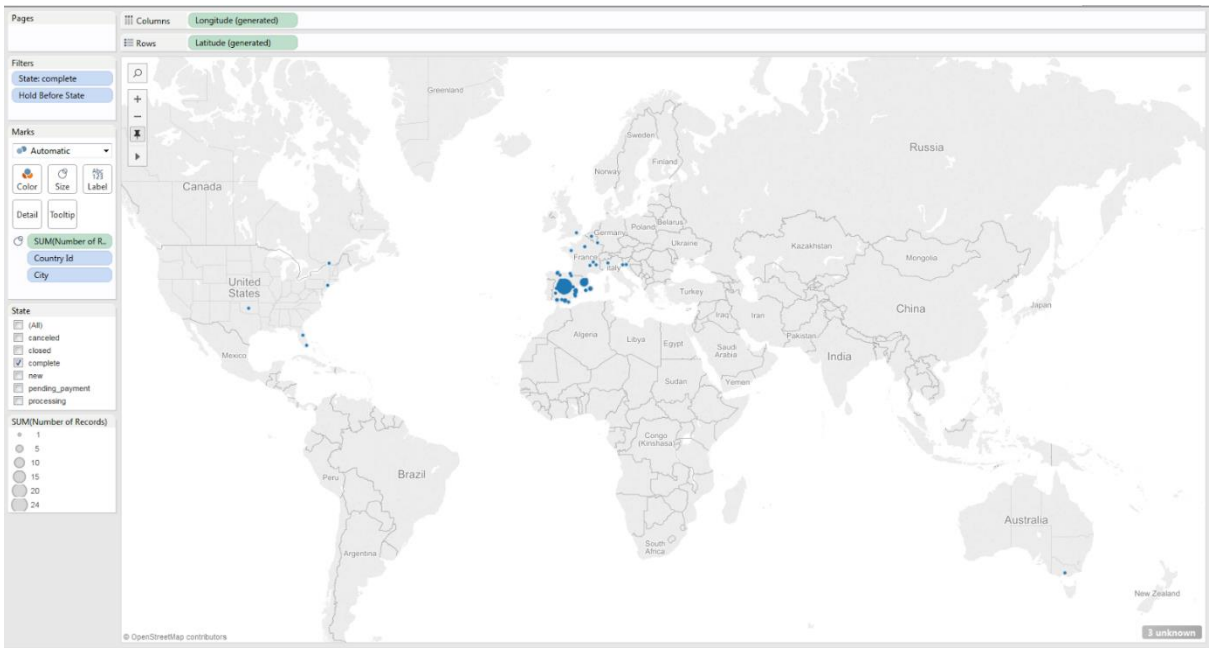


Figure 32: Sales per City

In Figure 31 and Figure 32 we have two map graphics. We have used the data from shipping address (*Country Id* and *City*) to analyze it. In the first one the size of the blue circles indicates the number of orders performed to ship to that city; in the second map the color intensity indicates the quantity.

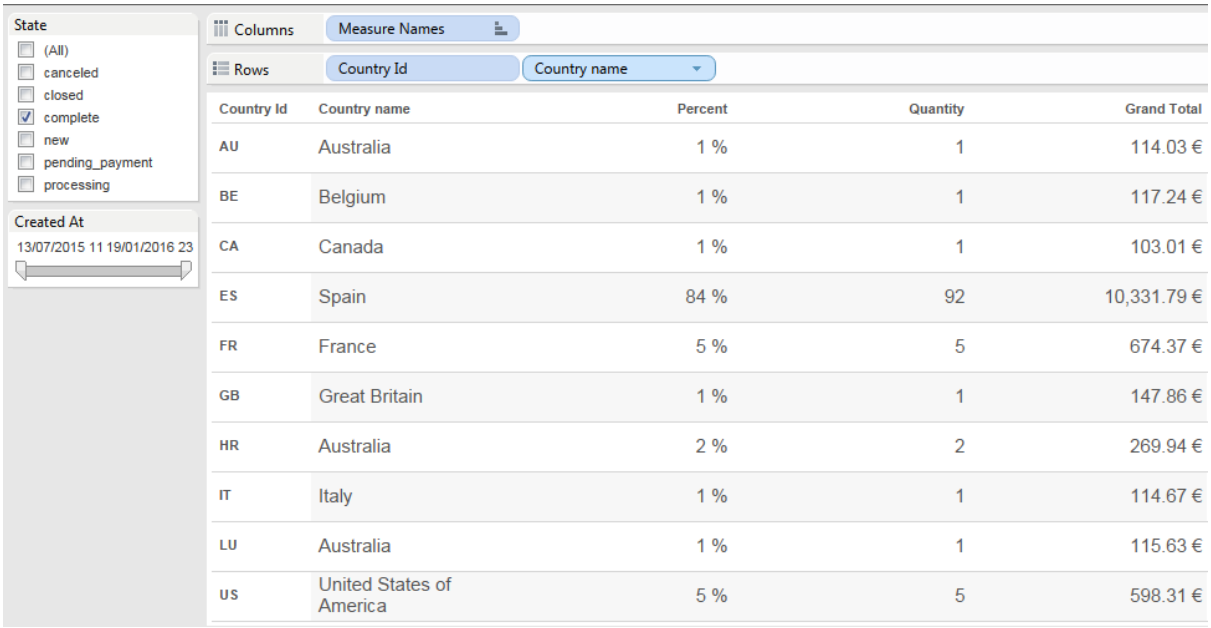
Again, we have only considered the completed orders.

We can see that this Ecommerce sells mainly in Spain, although it also has some market in occidental countries.

Used attributes:

- Count of orders Ids in *sales_flat_order* (Total number of orders).
- Latitud and Longitud, generated using the attributes *City* and *Country Id* from table *sales_flat_order_address*.
- State (Filter: completed) (table *sales_flat_order*)

Graphic 8: Sales per Country Analysis



Country Id	Country name	Percent	Quantity	Grand Total
AU	Australia	1 %	1	114.03 €
BE	Belgium	1 %	1	117.24 €
CA	Canada	1 %	1	103.01 €
ES	Spain	84 %	92	10,331.79 €
FR	France	5 %	5	674.37 €
GB	Great Britain	1 %	1	147.86 €
HR	Australia	2 %	2	269.94 €
IT	Italy	1 %	1	114.67 €
LU	Australia	1 %	1	115.63 €
US	United States of America	5 %	5	598.31 €

Figure 33: Sales by Country Analysis

In this table we make a further analysis on the previous Country sales map, analyzing the percent of sales per country, the number of orders and the total money spent. We have filtered only the completed orders, as those are the one that have generated money, and added a filter to select the range of dates in which this analysis can be seen.

Used attributes:

- Country Id (*sales_flat_order_address*).
- Country name: calculated field, transforms ISO code to country names.
- Count of orders Ids in *sales_flat_order* (Total number of orders).
- Calculated field that calculates the percentage of count of orders.
- Grand Total (*sales_flat_order*).
- State (Filter: completed) (table *sales_flat_order*)

Graphic 9: Sales per City Analysis

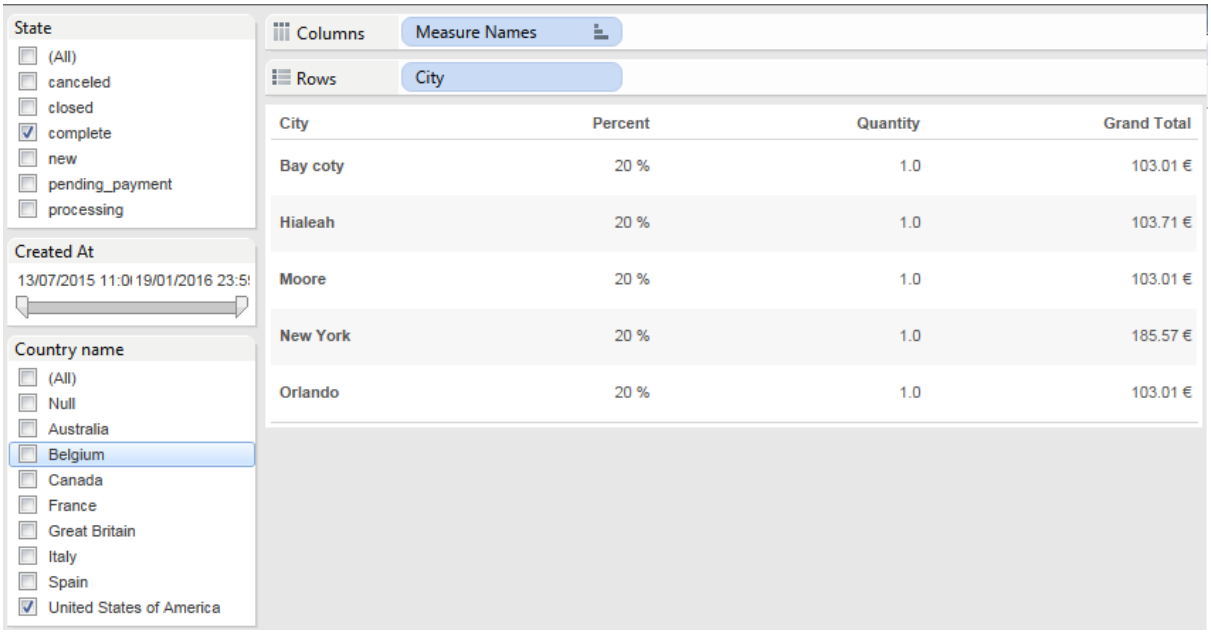


Figure 34: Sales by City Analysis

Here, we perform the same analysis as in the previous graphic, although this time we can filter through countries, in order to see the most profitable cities in each country.

Used attributes:

- City (*sales_flat_order_address*).
- Count of orders Ids in *sales_flat_order* (Total number of orders).
- Calculated field that calculates the percentage of count of orders.
- Grand Total (table *sales_flat_order*).
- State (Filter: completed) (table *sales_flat_order*)
- Country name (Filter)

Graphic 10: Client representation; Sales x Average Total

Figure 35: Clients representation, Sales x Average Total

Now, we represent clients in a graphic where the X axis represent the number of orders performed and the Y axis the average money spent by that person. Therefore, we can find the different type of clients and separate the ones that perform a lot of orders / spend the most money on it, to send them special offers.

We had more than one way to separate clients. Customer Name, Customer Surname, Customer Id and Customer Email. Name and surname do not identify users enough, as there could be more than one customers with the same name/surname, and the Id is only stored if the user has an existing account. As there are many cases of unsigned users operations, the best option is the email, which is also useful if we are planning on sending marketing through it.

Used attributes:

- Count of orders Ids in *sales_flat_order* (Total number of orders).
- Grand Total Average (table *sales_flat_order*).
- Customer Email (table *sales_flat_order*).
- State (Filter: completed) (table *sales_flat_order*).

Graphic 11: Countries representation; Sales x Average Total

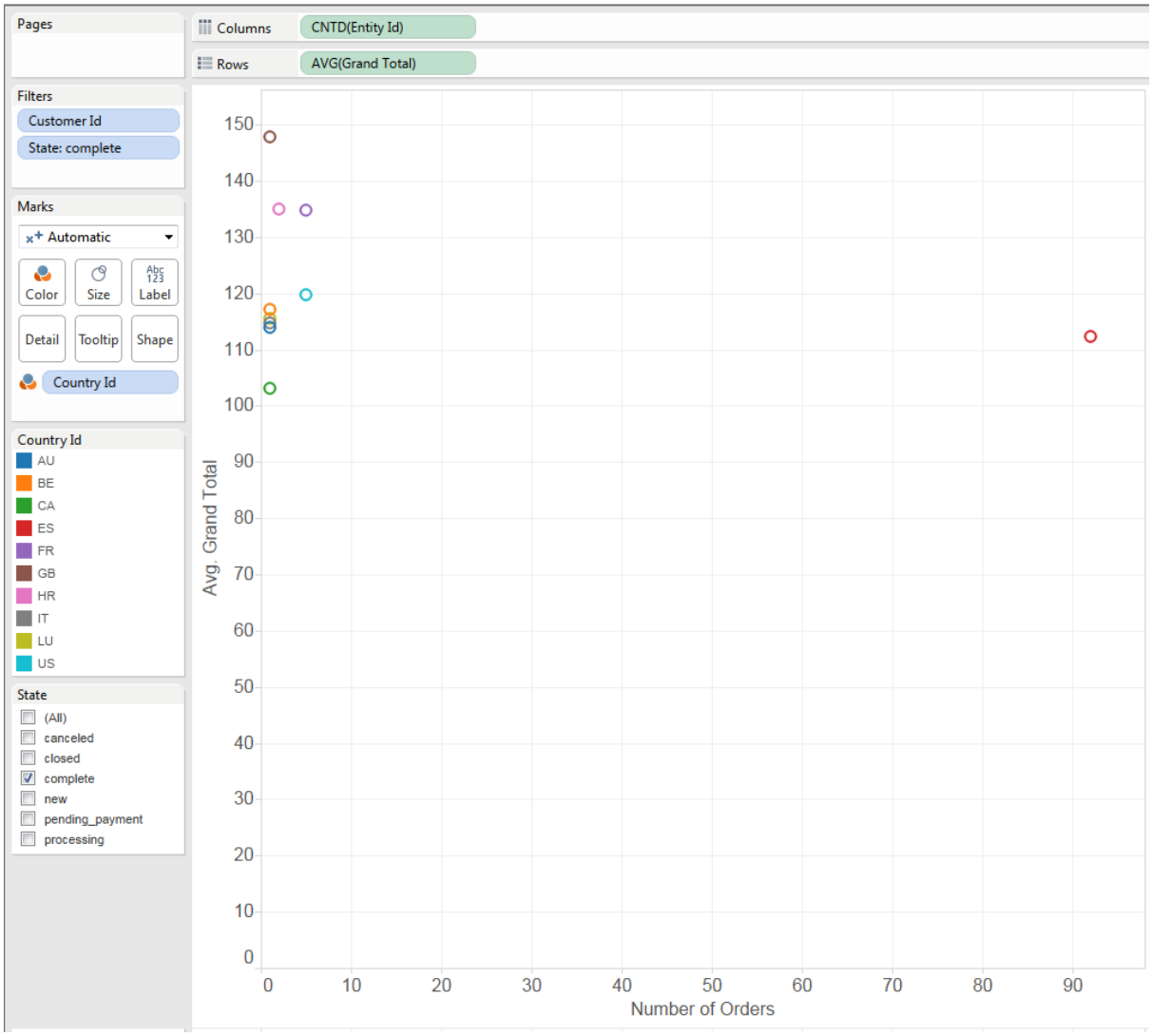


Figure 36: Countries representation, Sales x Average Total

Here we have done the same as in the previous graphic, but distinguishing the different countries, to find the higher performing countries. This can help to make decisions about which countries should the commerce focus in. It can be seen that Spain is the main leader in this Ecommerce.

Used attributes: The count of all the different Ids in *sales_flat_order,* Grand Total and Country Id.

Used attributes:

- Count of orders Ids in *sales_flat_order* (Total number of orders).
- Grand Total Average (*sales_flat_order*).
- Country Id (table *sales_flat_order*).

- State (Filter: completed) (table *sales_flat_order*).

Graphic 12: Guest vs Signed-in sales

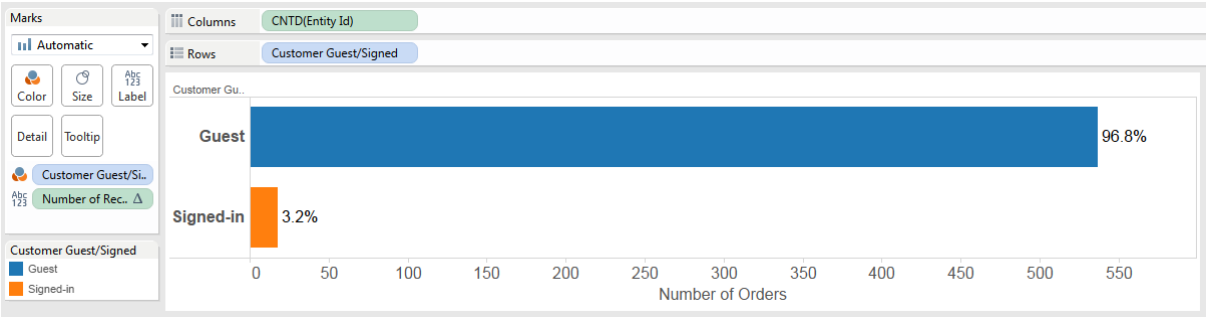


Figure 37: Guest vs Signed-in sales

In this graphic we are showing the comparison between operations done by users with a signed account and users without it. Clearly, not many people register themselves in this Ecommerce.

Used attributes:

- Count of orders Ids in *sales_flat_order* (Total number of orders).
- Customer Guest/Signed-in. Calculated Field to distinguish between Guest and Signed-in users.

Data Mart 2 Graphics

For the second Data Mart, as it was centered in products and categories, we do not have as many graphics as in the previous one, which was centered in sales, customers and billing/shipping addresses, where a lot of information can be taken.

Note that we have used a different Magento sample database for this Data Mart, as the one used in Data Mart 1, although it had more data, it did not have its products classified in categories, impeding a category name analysis.

Finally, it must be pointed out that beside the graphics implemented in this project, more could be created to perform a more extensive analysis. Of course this would require a deeper investigation into the database.

Graphic 13: Top X Product Bestsellers

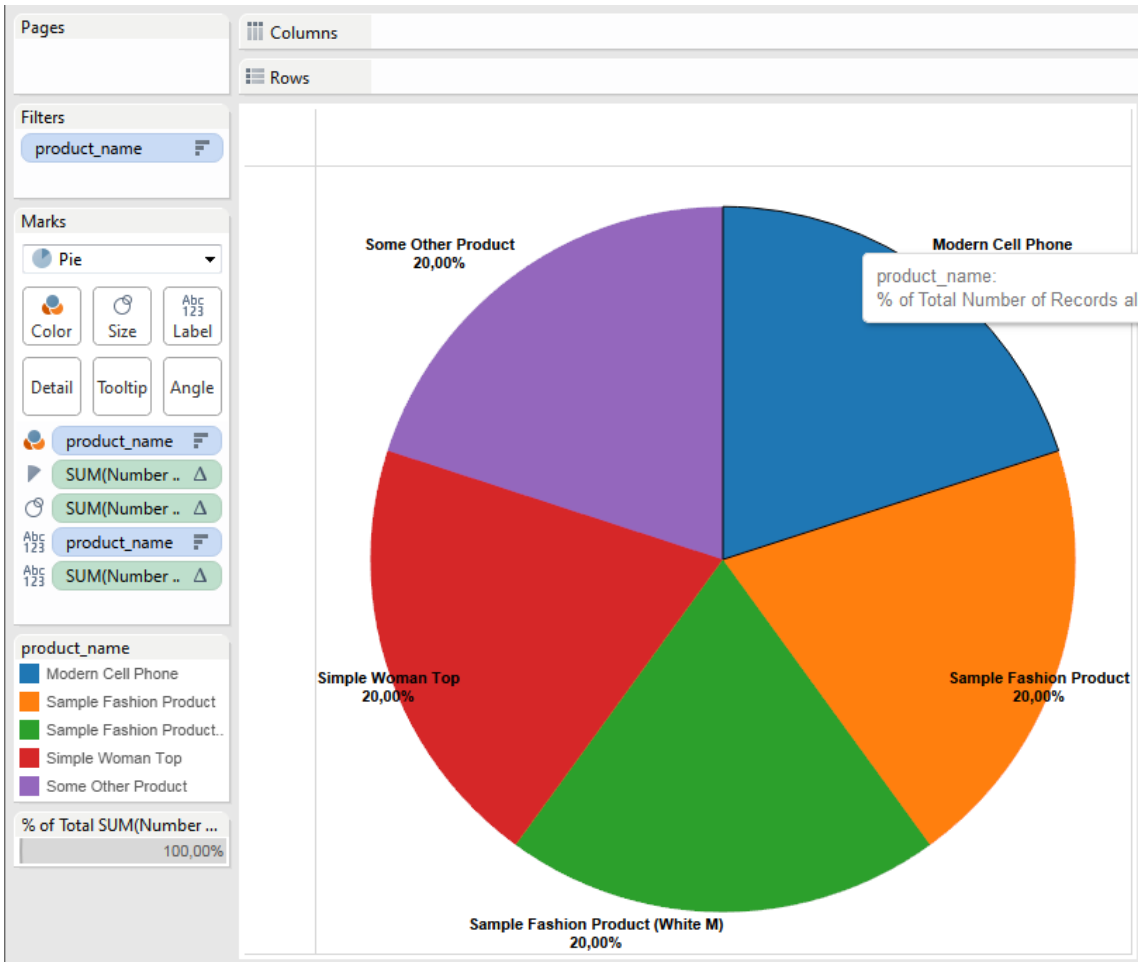


Figure 38: Top X Product Bestsellers

In this graphic we show a Pie chart with the Top sold products. Any number of products can be chosen, although we have chosen the Top 5.

Used attributes:

- product_name (taken from *Custom SQL Query 1*).
- Number of Records. Calculates the number of instances of the product (i.e. Quantity). Also used to calculate the percentage.

Graphic 14: Top X Product Bestsellers Analysis

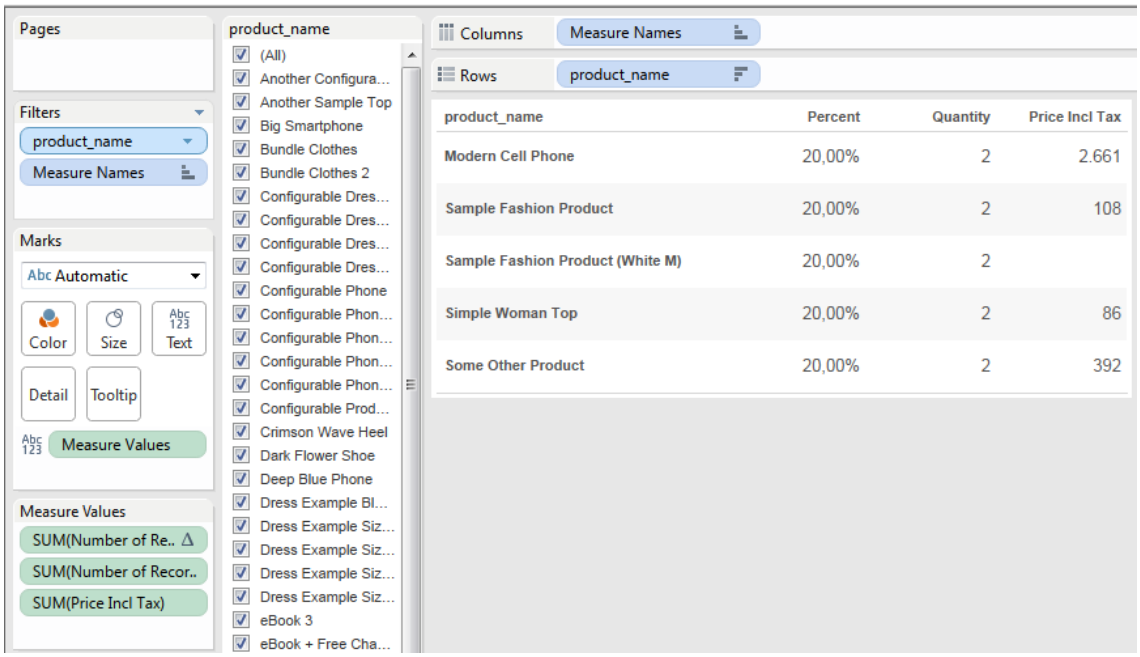


Figure 39: Top X Product Bestsellers Analysis

Now we are performing a deeper analysis on Graphic 13, calculating the percentage of each product over the Top products (to see if there are big differences between them), the number of sells, and the total amount of money acquired from that product.

Again, we can choose any Top we want. In this case, we chose Top 5.

Used attributes:

- product_name (taken from *Custom SQL Query 1*).
- Number of Records. Calculates the number of instances of the product (i.e. Quantity). Also used to calculate the percentage.
- Price Incl. Tax (table *sales_flat_order_item*). To calculate all the money acquired, we performed a Sum over all the instances of the product.

Graphic 15: Top X Categories

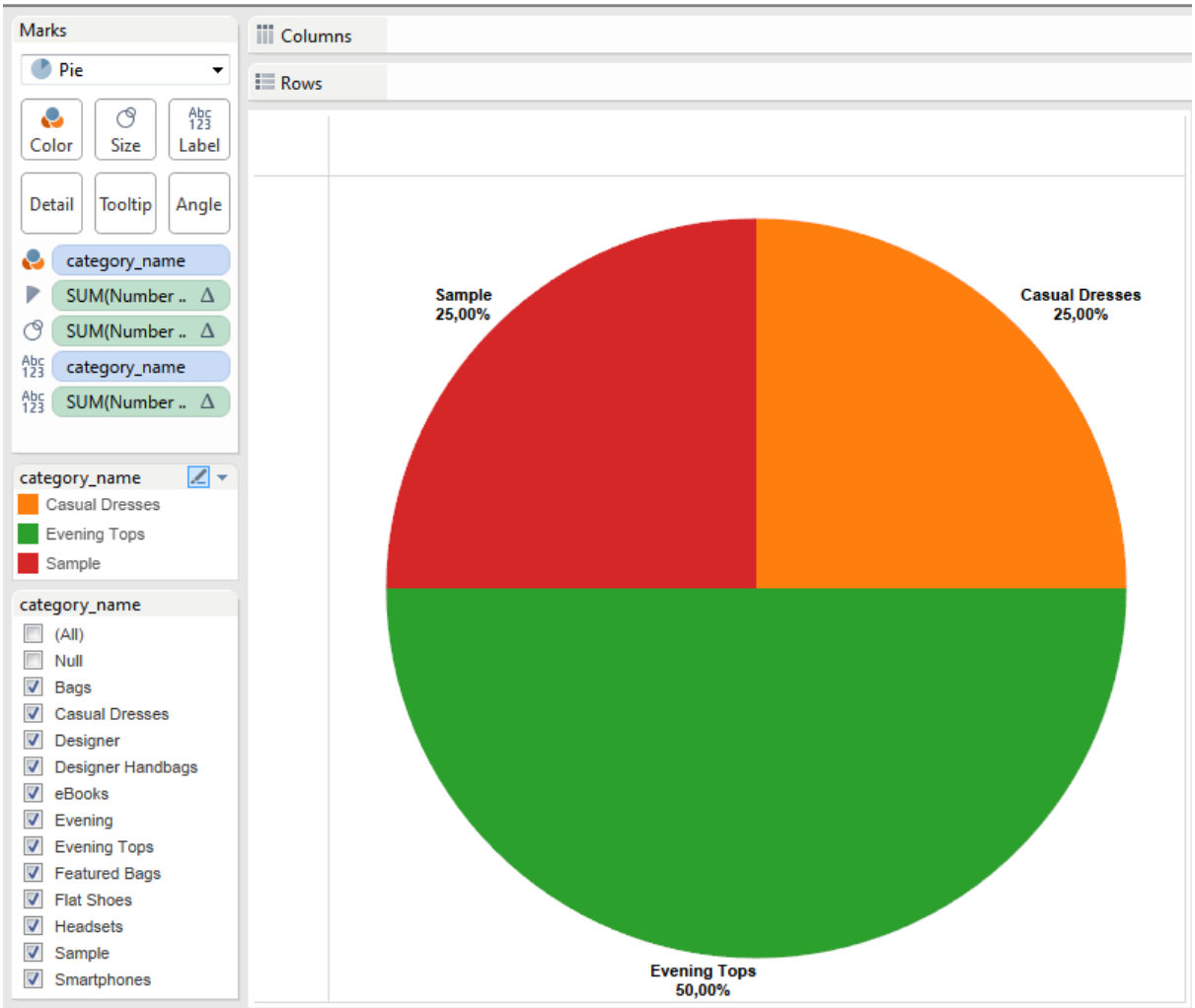


Figure 40: Top X Categories

As we did in graphic 13 we are showing a Pie chart with the Top sold categories. Any number of categories can be chosen, although we have chosen the Top 5. Some products do not have a category (it is not mandatory) and therefore we have excluded them from the sample, filtering them as it can be seen in the picture.

Used attributes:

- category_name (taken from *Custom SQL Query 2*).
- Number of Records. Calculates the number of instances of the category (i.e. Quantity). Also used to calculate the percentage.

Graphic 16: Top X Category Analysis

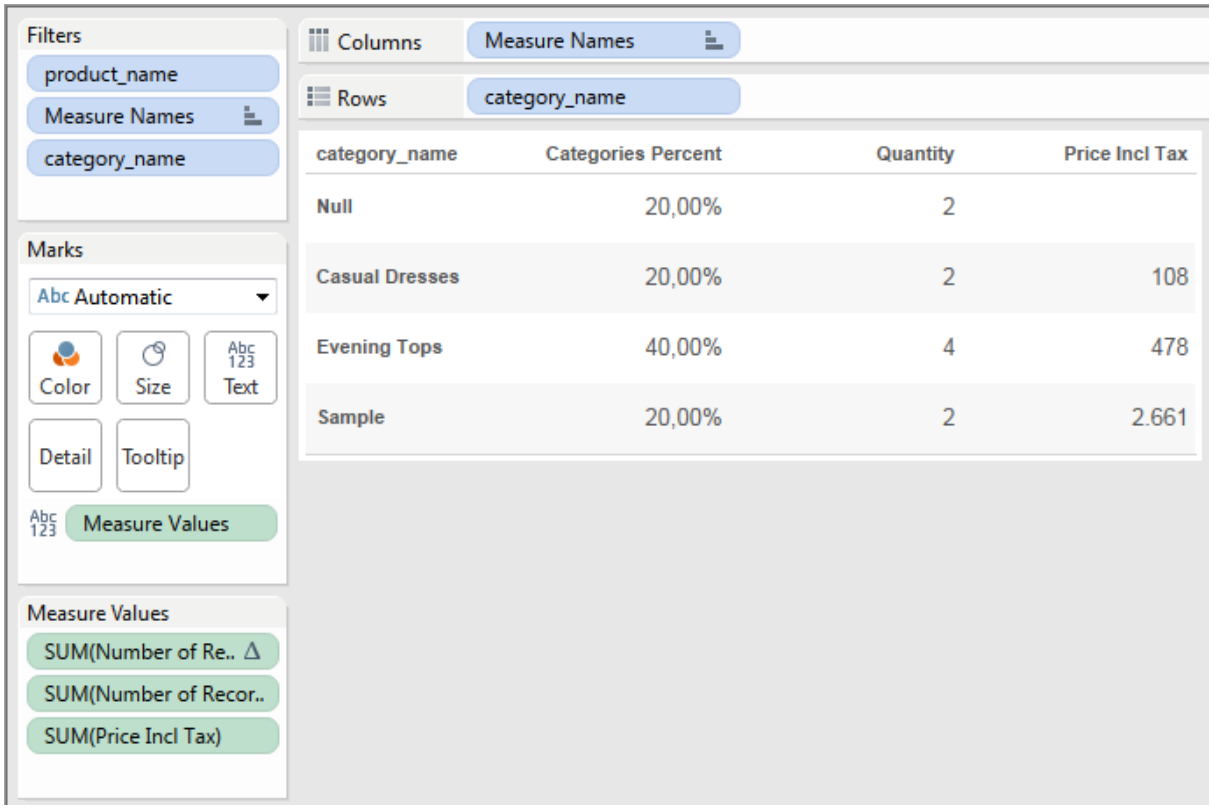


Figure 41: Top X Categories Analysis

Now we are performing a deeper analysis on Graphic 15, calculating the percentage of each category over the Top categories (to see if there are big differences between them), the number of sells, and the total amount of money acquired from that category.

Again, we can choose any Top we want. In this case, we chose Top 5.

Used attributes:

- category_name (taken from *Custom SQL Query 2*).
- Number of Records. Calculates the number of instances of the product (i.e. Quantity). Also used to calculate the percentage.
- Price Incl. Tax (table *sales_flat_order_item*). To calculate all the money acquired, we performed a Sum over all the instances of the product.

Graphic 17: Categories Performance over time

Figure 42: Categories performance over time

In this graphic we are showing the performance in terms of number of sales per different category of products. We are using the number of orders instead of the gained money as it would be biased to the more expensive categories. Only taken into account completed orders.

Used attributes:

- Created At (Month and Year) (table *sales_flat_order*)
- category_name (taken from *Custom SQL Query 2*).
- Number of Records. Calculates the number of instances of the product (i.e. Quantity).
- State (Filter: completed) (table *sales_flat_order*)

4 Conclusions & Future Work

In this chapter we will summarize the conclusions taken at the end of the current project.

4.1 Conclusions

The main goals indicated at the start of the project have been achieved, as I have increased my knowledge about the main platforms used nowadays in the Ecommerce and Business Intelligence area, and their current use and position in the market.

We have found that Magento, PrestaShop and WooCommerce are the current main leaders in the Ecommerce market. Although there are differences between them, we have concluded that Magento is a more complete software than PrestaShop and WooCommerce and a good choice for a company willing to grow.

In Business Intelligence, we have tried the most powerful platforms in the market, using for our project Tableau, which is mainly centered in Data Visualization and being therefore the most potent software to create visual graphics.

We have achieved to install Magento in an online server as our Ecommerce, performing the main configurations on it, adding the main characteristics such as products and categories.

Also, we have achieved to connect its database to Tableau, learning how to use it and creating graphics that may be relevant to help a business make decisions and see the performance of its online shop.

4.2 Future work

As to future work, the improvements that can be done in an Ecommerce are limitless. Improvement could be performed in terms of maintenance and customization. In this project we did not implement a theme for the Ecommerce nor any layout customization, although in a real project it is important to have a good looking and attractive web page.

In terms of maintenance, as well as the tasks we mentioned in the section about the Ecommerce implementation, we could include the addition of new functionalities and extensions to have a more complete experience for the user and the Ecommerce.

In terms of performance, a better instance in which the Magento Ecommerce is to be supported could be purchased, as when more visitors attend the web page it will require a server powerful enough to endure them.

In functionality and Ecommerce performance in the future other technologies beside business analytics could be implemented, such as Google Analytics to have analysis of the traffic on the web; an ERP (Enterprise Resource Planning) to manage the data of the business activities of the Ecommerce; and CRM (Customer Relationship Management) to have an improved interaction with the company's customers.

Also, in terms of security it could be improved in many forms. First, the access to Magento and phpMyAdmin administration panel could be installed in different locations, different from the default ones we used (*admin* and *phpmyadmin* respectively). Also, in the case of phpMyAdmin we could create different users that could only access to the databases we want.

As for this project, focusing in the Business Intelligence part, an OLAP cube could be performed in order to enhance the analysis part. An OLAP cube works with a main Fact table and different dimensions. A structure for this cube could be as follows, taking many of the tables and attributes mentioned in this project:

Table 28: Possible OLAP cube. For future works

Time	Customer	Product	Location
Date	Customer ID	Product ID	Address ID
Day	Name	Category	Country ID
Month	Email	Name	Region
Year		Price	City
			Postcode
FACTS: Sales Order ID, Customer ID, Product ID, Shipping Address ID, Billing Address ID, Order State, Order Date (CreatedAt).			

Of course, also a lot more graphics could be performed by a developer that investigated the Magento database or an expert on it, although that could be included in the maintenance of the Business Intelligence app.

Bibliography

[1] aheadWorks, "eCommerce Platforms Survey: Who Rules Olympus These Days and Who Will Gain Power in The Future?," July 2013. [Online]. Available: https://blog.aheadworks.com/2013/07/ecommerce-platforms-survey-who-rules-olympus-these-days-and-who-will-gain-power-in-the-future/. [Accessed March 2016].

[2] aheadWorks, "Magento 2 Contributes to the Global Ecommerce Platforms Market," February 2016. [Online]. Available: https://blog.aheadworks.com/2016/03/magento-2-contributes-to-the-global-ecommerce-platforms-market/. [Accessed March 2016].

[3] Wikipedia, "Iterative and incremental development," [Online]. Available: https://en.wikipedia.org/wiki/Iterative_and_incremental_development. [Accessed June 2016].

[4] "BuiltWith," [Online]. Available: http://trends.builtwith.com/shop. [Accessed June 2016].

[5] NBS System, "Benchmark of e-Commerce Solutions," 14 June 2013. [Online]. Available: https://www.nbs-system.com/wp-content/uploads/benchmark_short_english.pdf. [Accessed March 2016].

[6] aheadWorks, "E-commerce Platforms Comparative Study Based on Alexa1Million Top Sites," April 2014. [Online]. Available: http://blog.aheadworks.com/2014/04/e-commerce-platforms-comparative-study-based-on-alexa1million-top-sites/. [Accessed April 2016].

[7] aheadWorks, "E-commerce Platforms Popularity Study, October 2014," October 2014. [Online]. Available: http://blog.aheadworks.com/2014/10/e-commerce-platforms-share-investigation-october-2014/. [Accessed April 2016].

[8] aheadWorks, "Ecommerce Platforms Popularity, May 2015: Two Platforms Take Half," May 2015. [Online]. Available: https://blog.aheadworks.com/2015/05/ecommerce-platforms-popularity-may-2015-two-platforms-take-half/. [Accessed April 2016].

[9] aheadWorks, "Ecommerce Platforms Popularity, October 2015: Top Five Solutions Take Three Quarters of the Market," October 2015. [Online]. Available:

https://blog.aheadworks.com/2015/10/ecommerce-platforms-popularity-october-2015-top-five-solutions-take-three-quarters-of-the-market/. [Accessed April 2016].

[10] aheadWorks, "Magento vs. WooCommerce: Do They Have Any Reason for Competition," June 2015. [Online]. Available: http://blog.aheadworks.com/2015/06/magento-vs-woocommerce-do-they-have-any-reason-for-competition/. [Accessed April 2016].

[11] Gartner Inc., "Magic Quadrant for Business Intelligence and Analytics Platforms," February 2016. [Online]. Available: https://www.gartner.com/doc/reprints?id=1-2XXET8P&ct=160204. [Accessed April 2016].

[12] Gartner Inc., "Gartner Magic Quadrant," 2016. [Online]. Available: http://www.gartner.com/technology/research/methodologies/research_mq.jsp. [Accessed April 2016].

[13] SAS Institute Inc., "SAS products," [Online]. Available: http://www.sas.com/en_us/software/business-intelligence.html#view-all-products.

[14] SAS Institute Inc., "SAS University Edition," [Online]. Available: http://www.sas.com/en_us/software/university-edition.html . [Accessed May 2016].

[15] Wikipedia, "R Programming Language," [Online]. Available: https://en.wikipedia.org/wiki/R_(programming_language) . [Accessed May 2016].

[16] "Can we perform ETL in Tableau," July 2014. [Online]. Available: https://community.tableau.com/thread/145612. [Accessed May 2016].

[17] Tableau, "Tableau for Students," [Online]. Available: http://www.tableau.com/academic/students . [Accessed April 2016].

[18] Magento, "Getting Started with Magento Web APIs," 9 June 2016. [Online]. Available: http://devdocs.magento.com/guides/v2.0/get-started/bk-get-started-api.html. [Accessed June 2016].

[19] Tableau, "MySQL," [Online]. Available: https://onlinehelp.tableau.com/current/pro/online/mac/en-us/examples_mysql.html . [Accessed May 2016].

[20] Magento, "System Requirements for Magento Enterprise Edition and Community Edition (Current Shipping Versions)," [Online]. Available: http://devdocs.magento.com/guides/m1x/system-requirements.html. [Accessed April 2016].

[21] A. Voelkl, "Magento CE 1.9.2.2 database diagram," 17 November 2015. [Online]. Available: http://anna.voelkl.at/magento-ce-1-9-2-2-database-diagram/. [Accessed May 2016].

[22] Observatorio Nacional de las Telecomunicaciones y la SI, October 2015. [Online]. Available: http://www.ontsi.red.es/ontsi/sites/default/files/informe_b2c_2014_edicion_2015.pdf. [Accessed June 2016].

[23] Nominalia, “7 requisitos legales para montar una tienda online en España,” [Online]. Available: http://www.escueladeinternet.com/7-requisitos-legales-montar-una-tienda-online-espana. [Accessed June 2016].

[24] Boletín Oficial del Estado, "Ley Orgánica de Protección de Datos de Carácter Personal," March 2011. [Online]. Available: https://www.boe.es/buscar/act.php?id=BOE-A-1999-23750 . [Accessed June 2016].

[25] Boletín Oficial del Estado, “Sec. I. Pág. 26880,” 31 March 2012. [Online]. Available: https://www.boe.es/boe/dias/2012/03/31/pdfs/BOE-A-2012-4442.pdf. [Accessed June 2016].

[26] Wikipedia, “Dumping (pricing policy),” June 2016. [Online]. Available: https://en.wikipedia.org/wiki/Dumping_(pricing_policy). [Accessed June 2016].

Annexes

Annex 1: Access to Magento environment

The Magento shop can be accessed from anywhere as it is in an online server of Amazon Web Services. The main links are:

- This links to the front end of the implemented Magento shop: http://ec2-54-194-234-26.eu-west-1.compute.amazonaws.com/magento/
- This links to the back end (administrator panel) of the implemented Magento shop: http://ec2-54-194-234-26.eu-west-1.compute.amazonaws.com/magento/admin. To enter it will be required the user and password, which are *admin* and *sugerendo2016* respectively.

Annex 2: Install Magento

To install Magento we will first need to have downloaded the Magento version we want. From Magento download page. In this project we are using Magento v. 1.9.2.4. Then, we move it to the main directory of our server (in apache2 with ubuntu 14.04 it would be */var/www/html*), and we extract it there.

Now, we open the path *http://serverIP/magento* and, as it is the first time we enter, it will show an installation guide in the screen.

Annex 3: Install Magento Extension

As Magento is an open source software, there exist software modifications or improvements that are programmed by people from anywhere. These are called Magento Extensions, and can be found on the Magento Connect webpage, where there can be posted as in a *Google Play* or *Apple Store*.

Then, in order to install these extensions created by the community, you have to go to their respective page in Magento Connect. Once there, you have to click on the big blue *Install Now* button. Then, you have to read and accept the extension license agreement, and a link or *Extension key* will appear. This key has to be copied.

Then, we have to go to our administration panel. Access the *Magento Connect Manager* on the administration panel (In the *System* tab), enter your admin credentials, and a page will open where you can enter the key in the *Paste extension key to install* section. Then you click

Install, and when Magento has loaded the extension, click *Procced*. A terminal will show the process, and if there are no errors, a Package Installed message will appear on the screen.

With this, we now have the extension installed in our Magento.

Annex 4: Install Amazon Web Services Instance

To use an AWS instance, we will first have to open an account and enter the required information (username, password, credit card, address, etc.).

Once our account is activated, we will have to go to the AWS console and go to *Services/EC2/Launch Instance*. There, we select the instance we want to install (in our case we installed *Ubuntu Server 14.04*). In the next steps we have to select the different instance options (type of tier, instance details, storage capacity, instance tags, security groups). In out case we installed the free tier (*t2.micro*), selected a storage of 15 GB and added a security group that could enter via ports 80 (HTTP), 22 (SSH), 3306 (MySQL) and 8080 (for Pentaho).All this options can be configured at any time.

Annex 5: Access to server via SSH and FTP

We will explain how to access the server via SSH and FTP using Putty and FileZilla respectively.

When installing the server, we will receive an IP and a key in order to enter it. The key can be either a written password or a file. In case it is a file, we will need it in extension *.ppk* to use it in the mentioned programs. In our project case, Amazon Web Services delivers a *.pem* file. In order to transform it to *.ppk* we will need the program PuttyGen. In it, we will only have to load the key and click on *Save private key*. With this action our *.pem* key will now be a *.ppk*.

Now, we install Putty (SSH) and FileZilla (FTP). In Putty we enter as Hostname *user@IP*, and in the section Connection/SSH/Auth we load the generated private key (*.ppk*). Then, a terminal will open and we will be connected to our server via SSH.

In FileZilla we click on the *Site Manager* and add the server, entering the required information (server IP, user, keyfile) and the protocol, which will be *SFTP-SSH File Transfer Protocol*. When connected we will have a file manager opened of our server and we will be able to send files through it.

Annex 6: Install LAMP server

Once connected to the server via SSH (see Annex 5), we will have to enter the following code lines to install the basic software for a LAMP server architecture (Linux, Apache, MySQL and PHP):

- *sudo apt-get install apache2*
- *sudo apt-get install mysql-server*
- *sudo apt-get install php5 libapache*

In order to have a more interactive manage of the MySQL database, we can also install phpMyAdmin, which is a user interface that facilitates its usage:

- *sudo apt-get install phpmyadmin*

Annex 7: Install Tableau on a PC

In order to install Tableau (either Tableau Desktop or Tableau Public) we have to visit Tableau download page and select the version we want. Then, an installer will be downloaded. To complete the process, we just have to execute it, and read and accept its license. Tableau will be installed in a few minutes.

Also, to get a Student pass, you can certify your student status and Tableau will lend you a one-year full license.