

WORK MEMORY Q1 2021/2022

CRM MANAGEMENT WITH SALESFORCE

Author

DAVID AGUILERA IBÁÑEZ

david.aguilera.ibanez@estudiantat.upc.edu

Information Technologies Specialization

Bachelor Thesis

With the collaboration of INETUM ESPAÑA S.A

Thesis supervisor

RUBÉN TOUS LIESA

rtous@ac.upc.edu

Thesis director

JESUS RIVERA BONILLA

GEP professor

JOAN SARDÀ

joan.sarda@upc.edu



11th January, 2022



Abstract

The project consists in utilizing the CRM tool Salesforce to configure and build an application that must serve for the correct operation of the economic activities of a client of INETUM. Said client had, at the moment of the beginning of the project, an outdated system to carry out those activities.

During the construction process of the project, it has been possible to appreciate the development from scratch of a functional and useful application in the real world as well as the small situations and countermeasures that must be applied to a project of this magnitude.

On the other hand, I must emphasize the importance of using a good tool with the necessary functionalities to carry out this type of application. Salesforce incorporates many mechanisms that facilitate the development of projects and considerably reduce the investment to produce them, both economic investment and technical cost, since it does not require too advanced knowledge or too many personnel to build applications.

The result of the project is a useful and ergonomic application of an application that was out of date. The initial objectives and the requirements, both social, economic and technical, have been met.

Resumen

El proyecto consiste en utilizar la herramienta de CRM Salesforce para configurar y construir una aplicación que ha de servir para el correcto funcionamiento de la actividad económica de un cliente de INETUM. Este cliente tenía en el momento de iniciar el proyecto un sistema anticuado para llevar a cabo estas actividades.

Durante el proceso de construcción del proyecto se ha podido apreciar el desarrollo desde cero de una aplicación funcional y útil en el mundo real así como las pequeñas situaciones y contramedidas que se deben ir aplicando a un proyecto de esta envergadura.

Por otro lado debo hacer hincapié en la importancia de utilizar una buena herramienta con las funcionalidades necesarias para llevar a cabo este tipo de aplicaciones. Salesforce incorpora muchos mecanismos que facilitan el desarrollo de proyectos y reducen considerablemente la inversión para producirlos, tanto inversión económica como el coste técnico, puesto que no se requiere de conocimientos demasiado avanzados ni de mucho personal para construir aplicaciones.

El resultado del proyecto es una aplicación útil y ergonómica de una aplicación que había quedado desfasada. Se han cumplido los objetivos iniciales y los requisitos planteados, tanto sociales como económicos y técnicos.

Resum

El projecte consisteix a utilitzar l'eina de CRM Salesforce per configurar i construir una aplicació que ha de servir per al funcionament correcte de l'activitat econòmica d'un client d'INETUM. Aquest client tenia a l'hora d'iniciar el projecte un sistema antiquat per dur a terme aquestes activitats.

Durant el procés de construcció del projecte s'ha pogut apreciar el desenvolupament des de zero d'una aplicació funcional i útil al món real, així com les petites situacions i contramesures que cal anar aplicant a un projecte d'aquesta envergadura.

D'altra banda, he de posar èmfasi en la importància d'utilitzar una bona eina amb les funcionalitats necessàries per dur a terme aquest tipus d'aplicacions. Salesforce incorpora molts mecanismes que faciliten el desenvolupament de projectes i redueixen considerablement la inversió per produir-los, tant inversió econòmica com el cost tècnic, ja que no es requereix coneixements massa avançats ni de molt personal per construir aplicacions.

El resultat del projecte és una aplicació útil i ergonòmica d'una aplicació que havia quedat desfasada. S'han complert els objectius inicials i els requisits plantejats tant socials com econòmics i tècnics.

TABLE OF CONTENTS

1. Context
 - 1.1. Introduction
 - 1.2. Project Framework
 - 1.3. Concepts
 - 1.3.1. CRM:
 - 1.3.2. Salesforce:
 - 1.3.2.1. Salesforce data center locations
 - 1.4. Stakeholders
2. Justification
3. Scope
 - 3.1. Functional Requirements
 - 3.2. Potential obstacles and risks
4. Methodology and rigour
 - 4.1. Agile and tasks
 - 4.2. Version control.
5. Project Planning
 - 5.1. Description of the tasks
 - 5.1.1. Project planning
 - 5.1.2. Design of the project and its architecture
 - 5.1.3. Project implementation
 - 5.1.3.1. Private management
 - 5.1.3.1. Provider portal for quotes
 - 5.1.4. Testing and debugging
 - 5.1.5. Insertion of data
 - 5.1.2. Thesis preparation
 - 5.1.2.1. Thesis redaction
 - 5.1.2.1. Thesis defense preparation
6. Estimates and Gantt diagram
 - 6.1. Workload distribution
 - 6.1. Human resources
 - 6.3. Gantt Diagram
7. Risk Management
 - 7.1. Difficulty in the approval of the functional design.
 - 7.2. Have the required data on the dates agreed in the planning
 - 7.3. Apply a correct change management and acceptance by the organization of the new services.
 - 7.4. Technological difficulties with the client's systems.



7.5. My own understanding of Salesforce due to my lack of experience.

8. Budget

8.1. Costs per Personnel Activity

8.2. Costs per Activity Production

8.3. Other Costs

8.3. Total Cost

8.4. Management Control

9. Sustainability

9.1. Self-assessment

9.2. Economic Dimension

9.2. Environmental Dimension

9.2. Social Dimension

10. Development Introduction

11. Project Specification and Design

11.1. The Actors

11.2. Definition of requirements

11.3. Use cases

11.4. Class diagram

12. Implementing in Salesforce, Apex and SOQL

13. Private Management Application

13.1. Configuring the sales process

13.1.1. Profiles and data security.

13.1.2. Data inputs

13.1.3. Data relationships

13.1.4. Data Reliability

13.1.5. Mailing system.

13.2. Modelling in Salesforce for provider quotes

13.3. Using the Application for Management and Marketing purposes

14. Provider portal for quotes

15. Testing and Debugging

16. Data Insertion

16.1. Data Loader

17. Conclusion

18. Bibliography



Project Outline

1. Context

With the purpose of being as clear and organized as possible, I will divide this section into different subsections where I will be explaining the principal causes for the existence of this project as well as the main concepts and terms that must be taken into consideration before going any further.

1.1. Introduction

This is a Bachelor Thesis for an informatics degree (speciality in Information Technologies) made in the Barcelona School of Informatics with the collaboration and cooperation of Inetum. The purpose of a thesis is to demonstrate the ability of a student to build or participate in real-life projects using tools and techniques learned through the degree under the supervision of a tutor.

Inetum is an international consulting firm that provides digital services and solutions and has more than 27000 employees located in more than 26 countries around the world. The company made a re-branding in 2020, changing 10 years of being GFI and fusing with many other companies in order to give a new and fresh look to the company. Inetum does take part in numerous initiatives in Spain and works in multiple projects around the country.

In this particular case, I joined Inetum as a trainee and my responsible, Jesús Rivera, provided me with a project that needed to be developed and engineered. He gave me all the information and requirements that were necessary to start the project.

At the same time, Rubén Tous was my supervisor in representation of the university, and he helped me with different doubts and activities, in this case those activities were more related to the formalization and construction of the thesis itself.

1.2. Project Framework

The main objective of the project is to provide a third party, a new and polished portal to manage their relationships with their customers and potential customers using a Customer Relationship Management (CRM) system. For starters, a CRM is a technology for managing a company's relationships and interactions with customers and potential customers. These concepts will be explained in more detail in the following sections.

Our client is one of the greatest worldwide distributors of office material and work solutions. Commercialises with all kinds of objects; pens, paper, office chairs,

desktops, batteries and everything we can imagine in a work environment. It also provides IPEs (Individual Protection Equipment). In essence, our client is an intermediary; Makes possible for the products made by different manufacturers to arrive at the final clients. It operates in more than 20 countries and does business with many companies in different continents.

At the moment of the start of the project, our client had a supplier consultation website that the commercial department was using to quote special products made by their suppliers. In the context of Salesforce, a quote is a formal offer for products proposed at specific prices and related payment terms. Those are products that some clients may want in large quantities. For instance, picture a company with hundreds of employees. That company may want thousands of notebooks and perhaps it would be more professional if those notebooks had the company's logo and name on it. That is the added value of our client and why it secured its place among its market. We can imagine that it is important for them to have a good mechanism to control all their quotes.

The old consultation website had several problems that had to be solved in order to improve the user experience and guarantee the security of the consultations with their providers.

There are also some sub-objectives that are worth mentioning;

- Reaching quality standards to match the current philosophy and the status of the firm (Inetum).
- Developing interpersonal values such as companionship and to help prolongate the current state between colleagues in Inetum.
- Expand my knowledge in software development and in information technologies using cloud solutions to take part in different projects with the goal of improving the overall IT infrastructure and security.

1.3. Concepts

There are a couple of key concepts and tools that are required to be explored in order to fully understand the thesis.

1.3.1. CRM:

According to Christopher Sirk (2020) in [crm.org](https://www.crm.org), "A CRM gathers customer interactions across all channels in one place. Managing centralized data helps businesses improve customer experience, satisfaction, retention and service". CRMs are mainly designed for businesses that have some kind of activity related to marketing, service or sales.

Having all data centralized is very useful for retention of customers because they will not feel like they are repeating the same story again to your company. This is accomplished when all your agents check their information from the same sources. All data from social media, emails, phone calls or other sources will end in the same place, making everything more organized for all parties involved. Furthermore, depending on the tool used, some tasks can be automated leading to an increase in efficiency.

Moreover, having everything in one place helps when it comes to analyzing projects. Strengths and weaknesses are easily identified and that accelerates the time it takes for a company to react to certain decisions or adopt new directives.

As companies are becoming more remote distributed, teams are becoming more flexible. It makes sense to invest in a tool that puts all your data in one secure and manageable place.

1.3.2. Salesforce:

Salesforce is a multi-solution platform that has the goal of helping businesses with their processes. As the official [Salesforce website](#) puts it, Salesforce works to “bring companies and customers together.” Salesforce is a CRM tool that has been designed to bring to the table everything we have just discussed about CRM and its uses.

On the [same web page](#), the company claims that “Over 150,000 companies, both big and small, are growing their business with Salesforce.” in 2021.

Gartner, a global research and advisory firm known for their technology solutions' comparisons and analysis, published a “[magic quadrant](#)” in June 2021 where the principal competitors for the CRM suppliers market were analysed. Those firms were compared side by side and analysed depending on their strengths and weaknesses according to their company decisions, market share and potential. Salesforce was the preferred tool, situating itself on top. It is the leader of the CRM market in both the ability to execute their ideas and advances as well as the leader in the “vision” for its tool.

Salesforce has many data centers around the world, and they also have data on how and what processes are normally executed. With that data they can balance the loads and maximise the productivity of their servers and the use of data centers. It is cheaper for them to host a number of servers than it would be for a single company to host their required servers, and that is how they make their business.

It is a company dedicated to selling “Software as a service”. That is no more than a way of delivering centrally hosted applications over the net as a product itself. The product is a tool dedicated to building other applications. Software as a service is also known as On-demand software and “hosted” software. The use of these products is interesting for the customers because they are freed from a number of responsibilities, such as hosting servers and the need of using high-end computers.

1.3.2.1. Salesforce data center locations

To serve its massive traffic demand, Salesforce has placed its data center in many cities in five different countries. These countries include France, Germany, UK, Japan and the USA. With that selection they cover 3 continents, North America, Europe and Asia. Every client that subscribes to Salesforce, can create an instance to place their organization's CRM. That instance identifies the servers where a specific client is being hosted.

The servers can be consulted from the Salesforce web page and the identifier is simple. The first two letters determine the continent and the number is only the number of the server in that continent. If the access is from Europe, the initials would be “EU”.

1.4. Stakeholders

Many parties are involved in the project of the thesis, and most of them interact or benefit from this in a different way.

The client:

As the main stakeholder of this project, the client is highly interested in the development of the final product, as its usage will largely influence the management of the data in its infrastructure and will have a large impact on the functioning of the company.

Client employees:

They will be the ones that will be using the solution in their everyday work-life. All the automations and flows that involve anything related to CRM will be managed from the tool.



Inetum:

The consulting firm has the mission to help its partners to improve all their processes that have anything to do with technology. As Inetum was contacted to start with this project, it is of great interest for it to be as useful as possible.

Inetum employees:

Some employees at inetum will have to keep in touch with the client to maintain and keep the tool updated. Some of the necessities of the processes may vary, or some new processes may be added to the portal in the future.

The client's providers:

Some people external to the company access a portal to perform some tasks that are related to their relation and commercial takes part in. They will benefit from the new portal as it will have a better user interface and help with a better user experience.

The director and the tutor of this project:

On the one hand, the director of the project also works for Inetum, so he has a similar relation with the project as any other Inetum employee that may have been designed to collaborate with the continuous development of the project.

On the other hand, the tutor of the project is more interested in the correct execution of the project, as a manager of the thesis.

Myself, the author:

I am also a stakeholder, as I am very interested in the well -being of the project and its correct execution, as my degree depends on it.

2. Justification

Our client had an existing portal and some internal processes that were being used before the initialization of the project. There are some reasons as to why it was decided to use an updated CRM tool and start from scratch, with all the advantages and disadvantages that this brings with itself.

To start with this explanation, we must be on the same page. What was the real state of that existing infrastructure?

That existing infrastructure was being used for many years since it was first created, in a moment where CRM tools were almost non-existent or at least did not adapt as well to the company necessities as they are capable today. Many processes and internal policies have changed since then for the company, and all their infrastructure has been adapted, suffering many patches and hotfixes during the years.

There was a decision to be made, to keep the existing system or to build a new one. Every option had its pros and cons. They will be presented on the following page;

Maintaining a legacy application is a challenge in many ways, for instance:

- It becomes more difficult over time to find developers that are able to work with older software tools and APIs. Many external services may no longer be supported and that brings many security risks and concerns.
- Legacy apps may not be compatible with newer systems, and some people may prefer to use up-to-date systems, for example mobile phones or tablet applications.
- Updating an old system to adjust to the new workflows can be just plain impossible due compatibility issues.

But there are also some problems with starting with a new project, such as:

- The deployment must be orchestrated in such a way that it does affect as little as possible to the production environment of the company. Some companies cannot afford to fully stop their processes.
- Migrations are not easy. All the data and information that the company had previously to the update must be stored and accessed later. There cannot be any service gaps.

Salesforce has some advantages that must also be considered:

- There are thousands of salesforce experts around the world, so in the case that new hires are needed, it should be easier to find them. Furthermore, they

will have some experience with very similar projects and their adaptation time will be shorter.

- As many functions and options are pre-build in Salesforce, the setting up and building time of an application is boosted. It only takes to know how to adapt Salesforce to your organization.
- According to [Amazon](#), GE Oil & Gas achieved a total cost of ownership (TCO) savings of 52% by moving its legacy applications to the cloud. [David Linthicum](#) reports an average TCO saving of about 40% for industries such as financial services and communications, and the savings are usually higher in retail and other sectors that experience large seasonal or cyclical changes.

Because of the balancing of the points above, and the clear tendency and pressure for organizations in the market to invest in this kind of services, we decided to use a CRM tool. Moreover, as we saw in the previous section “concepts”, Salesforce is the leader in CRM, and Inetum has a lot of experience with it, making it the most cost-effective tool for project development and the preferred tool for this kind of projects.

3. Scope

The main objective of this thesis is to build from scratch an infrastructure using Salesforce for the client following their requirements. Apart from a centralized application where all data will be stored and managed, it is necessary to include an internal provider portal to which some actors participate as providers in a business to business relationship (b2b).

To conclude, it is also necessary to add mechanisms and automatizations to perform different processes more efficiently, and incorporate ways to gather information from some inputs on their webpage.

3.1. Functional Requirements

After specifying the objectives and sub-objectives of the project, we need a brief description of the requirements. With the purpose of readability and clarity, those requirements will be further elaborated later in this same document, as they will become more relevant in their respective sections.

Some of these requirements are pretty self-explanatory and simple, as this kind of technology becomes more and more used the requirements seem obvious or basic, but they will be included nevertheless.

- The users should be able to interact with their application, adding, deleting or editing all the records that are relevant for them.
- Every user has a role in the company, and it will be reflected in the application. Some users have permissions and access to some objects that others may not see or utilize.
- Third parties (providers), will be able to log into their site, where they can interact with their relevant records.
- The flow or process of sales must be determined, from contact with a client to the completion of the order. With all the steps that are necessary.
 - Those steps will have some automatizations, including workflows to help the user to complete the processes benefiting from statistics provided by previous communications with the same or similar providers and clients.
- Users will create Dashboards to present the information in a more readable way, and they represent data and reports previously generated.

3.2. Potential obstacles and risks

There are some risks associated with working with different parties on a project. In this case I have identified four risks that may affect to the outcome of the thesis:

1. Difficulty in the approval of the functional design, involvement of key users in the taking of requirements, tests and approvals.
2. Have the required data on the dates agreed in the planning (Integrations with our client's products, access to environments, data to migrate, etc.)
3. Apply a correct change management and acceptance by the organization of the new services.
4. Technological difficulties with the client systems.

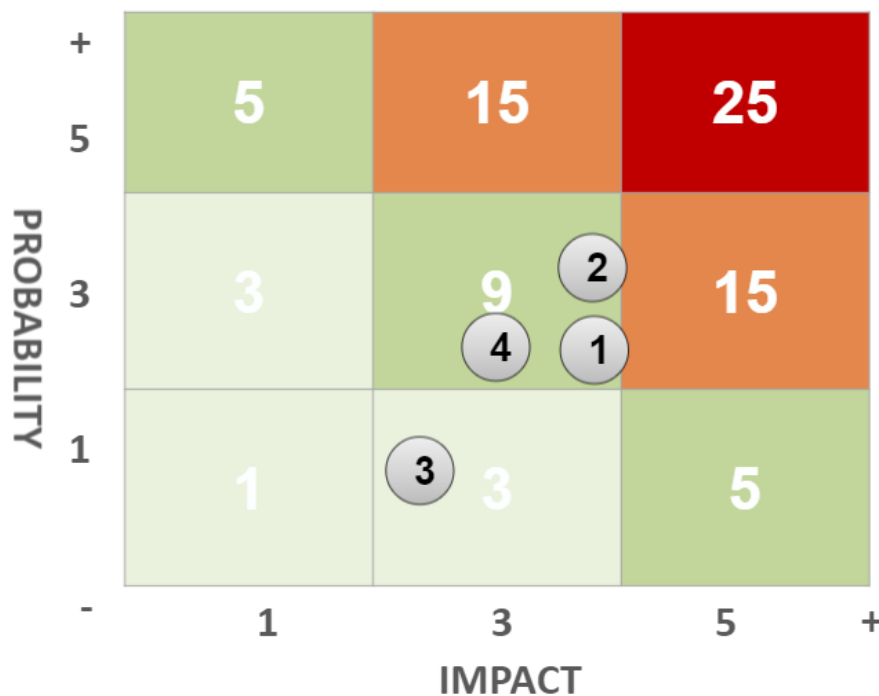


Figure 1: Visual representation of the risks. Source: made in Inetum.

4. Methodology and rigour

Defining a working methodology and sticking to it is of great importance for a satisfactory thesis development. The thesis has two reasons for being completed before December; a) the final client wants to reach at least an MVP before the end of the year, and b) the thesis must be presented this academic semester.

I will be following an Agile methodology, and therefore the work must be divided into smaller tasks. To keep under control the status of the project and to mitigate possible misconceptions due to communication issues, we will be holding daily meetings using the tool Microsoft Teams, where the project Manager will keep short but concise conversations with the development team to give feedback about the advances of the day before.

4.1. Agile and tasks

The tasks will be divided into four categories:

- To Do, Tasks that are still pending or, to be realized in the future.
- Current development; for tasks that are being realized at the time.
- Review; for tasks that are apparently completed, but are lacking part of the testing or the supervision of the project manager.
- Completed; for tasks that have been reviewed with positive feedback.

If a task is reviewed and has some improvement to do, it will return to a “Current development” state.

The division of the tasks will be explained deeply in future sections, but here is a brief view of what the task separation will look like.

Tasks related to the internal data management:

- A functional analysis of the application must be done.
- Modeling the queries of the providers.
- Implementation of a process for synchronizing products.
- Implementation of a workflow to manage the queries.
- Development of the flow of actions of the principal workflow.

Tasks related to the Provider portal:

- A functional analysis of the provider portal.
- Define the usability of each screen.
- Implementation of the screens of the queries by the providers and the process of those queries.

- Implementation of a private area of management, new queries and dashboards personalized for each user.

4.2. Version control.

Version control is one of the most commonly used tools for development where teams can leverage on their path to effective release management. Using a version control system has many advantages, such as enabling parallel development streams, helping to audit and keep trail for each stage of the development cycle and facilitating the coordination and storage of different versions of a project among others.

At the beginning, we will be developing our applications and basics using a sandbox project in Salesforce where things can be tested and executed freely, since it does not affect any external activity. Once we are set with all the infrastructure, the project will need a version control system for the parts of the application that have to be done outside the development playground, meaning all the classes in Apex code developed to further adapt Salesforce to our client will require to be treated apart, in accordance with Inetum's best practices.

5. Project Planning

I will dedicate about 400h of work in Inetum to this particular project and I expect to dedicate about 10 hours per week on average of personal work. The total amount of hours that will be committed to the thesis will round the 500 hours.

The project starts on June 22 and it is expected to end in mid September. This means that I will have around 70 working days to complete the project. Those hours must be distributed between different and smaller tasks to have a better control over the overall schedule and the planning.

The following pages will be dedicated to translating into words the initial idea for the planning and although it is expected to follow it tightly, it is possible that some modifications could be done to adapt the schedule in the future.

5.1. Description of the tasks

This section is dedicated to further elaborate the different tasks that have been briefly mentioned in the section "methodology and rigour". These tasks are performed in a certain order because of dependencies and organization-wide standards. Those dependencies imply that some tasks depend on others to be finished or at least partially completed to proceed.

Later, a graphical representation of these relations will be shown in the Gantt diagram and the specific time bounds will be set.

5.1.1. Project planning

Contextualization and Scope. A Project cannot be started without this. An objective must be set along with deadlines and the methodology. Furthermore, a temporal planning must also be set, to keep track of the progress of the project and to be able to react to possible delays. At this point the budget and the sustainability aspects will be detailed.

Analysis of the social and economical factors. As the concern for the environment grows and companies become more involved with sustainability, all projects must go through an extensive analysis to be sure it passes some standards. An economic analysis is also required to plan the required resources to complete the project and make an estimation.

5.1.2. Design of the project and its architecture

Definition of requirements. To define these requirements, some Inetum experts will hold some calls with the clients, where they will further elaborate in detail what they want and the first outline of the project will be discussed.

Standard for object management. After the definition of the requirements and the initial project planning, a modeling language must be used to express all the information in a structure that is defined by a universal set of rules. Anyone with a minimal knowledge of the modeling language will be able to read it and know how the project will be implemented and all the requirements it has.

5.1.3. Project implementation

As we have already said, this project contains two main parts; An internal application where the client can manage their data and a portal dedicated to the communication with its providers. These two tasks are not entirely sequential, so once the setting up of the private management is completed the portal can start to be worked on. Every day, a meeting will take place to talk about the advances of the projects, possible problems and think about possible solutions.

5.1.3.1. Private management

Functional analysis of provider quotes. Most of the analysis will be already done at this moment, but some minor details need to be discussed, so the project

advances in the right direction. The roles of the users and the profiles will be also defined.

Modelling in Salesforce for provider quotes. Salesforce has some pre-build standards that could be partially used for this project. Most of the job here is configuration and adaptation. Further work could be necessary in that regard.

Implementation of a process for product synchronization. In the case that Salesforce does not provide a standard model that would suit the necessities perfectly, some modifications need to be done to it. Even if it mostly fits, there is some setting to do about the products.

Developing the flow of actions of the Salesforce Workflow. One of the points of using Salesforce is to use its perks. Workflows help users to perform different tasks in a more efficient and guided way. Those Workflows must be configured.

5.1.3.1. Provider portal for quotes

Functional analysis of the provider portal. The uses for this application will be set here. At this point the general idea will be clear, but there could be some details that must be consulted with the client. The roles of the users and the profiles will be also defined.

Define the usability for each screen of the application. To define usability we must listen to the clients and follow the best practices in this regard. The portal must be a helping tool for their transactions and as such, it must be clear and easy to use.

Implementation of screens for provider management. All the actions a provider can do with the portal have to be implemented following the defined usability.

Implementation of the private area. The provider that accesses the portal will have available different information about the relation with our client. For instance, an overview of previous transactions and active quotes among other options.

5.1.4. Testing and debugging

The testing part of the project greatly depends on how many things outside of the salesforce standards the project has had to incorporate. We do not expect to build from scratch many things, so the testing will be mostly on the front-end.

5.1.5. Insertion of data

The previous data that the client wants to incorporate into the project should be included in their Salesforce space database.

5.1.2. Thesis preparation

The final degree Thesis has some tasks related to the university that add some workload to the extension of the project. Those tasks will be carried out outside of the work in Inetum.

5.1.2.1. Thesis redaction

In order to deliver and defend the thesis, it has to fulfill some requirements that include certain bureaucracy and the definition of different social and economic aspect among other things. Furthermore, all of the project itself must be written and the different decisions taken during the development should be explained.

5.1.2.1. Thesis defense preparation

As a last step, the thesis must be defended in a tribunal, where I will explain the project and the principal concepts and decisions made as well as show the final result.

6. Estimates and Gantt diagram

6.1. Workload distribution

Task name	Task Code	Dependencies	Duration (h)
Project planning	T1	-	40
Design of the project	T2	T1	40
Project Implementation	T3	T2	250
Private Management App	T3.1	T2	150
Functional analysis for provider quotes	T3.1.1	T2	25
Modelling in Salesforce for provider quotes	T3.1.2	T3.1.1	50
Implementation of a process for product synchronization	T3.1.3	T3.1.2	35
Developing the flow of actions of the Salesforce Workflow	T3.1.4	T3.1.3	40
Provider portal for quotes	T3.2	T2	100
Functional analysis of the provider portal	T3.2.1	T2	25
Define the usability for each screen of the application	T3.2.2	T3.2.1	30
Implementation of screens for provider management	T3.2.3	T3.2.2	25
Implementation of the private area	T3.2.4	T3.2.3	20
Testing and debugging	T4	T3.1 / T3.2	40
Insertion of data	T5	T3.1,T4	45
Thesis Preparation	T6	T5	90
Thesis Redaction	T6.1	T5	70
Thesis defense preparation	T6.2	T6.1	20
Total	505 h.		

Figure 2: Estimated hours per task. Source: Own elaboration.

6.2. Resources

6.1. Human resources

There are a few actors implicated in the development of the project, but the workloads are not equally distributed. I will be the one designating time to most of the tasks, but both my thesis tutor and my thesis director are important human *resources because they influence the development of the project and in the case of the director he will be implicated in the meetings with the client and in the daily calls that will be held during the process. There is also another important figure related to the thesis preparation and that is my university GEP responsible. He will be directing the first steps taken in the project where the project management and scope will be discussed.

6.2. Other resources

This thesis does not require any equipment that Inetum or the university did not previously have, but we must take at least the amortization into consideration.

These resources can be divided depending on its nature.

Hardware:

- Computer Lenovo laptop computer: 8GB of RAM and Intel(R) Core(TM) i7-8550U. CPU @ 1.80GHz.
- Router for internet connection: Mitrastar HGU GPT-2541GNAC.

Software:

- Windows Licenses. O.S, Teams, etc.
- Salesforce Licenses.

7. Risk Management

Risk management is a task of utmost importance when planning for a project. There are multiple variables that could affect the outcome and some of them cannot be fully controlled but can be mitigated if their effects are not desirable.

Some of the risks include:

1. Difficulty in the approval of the functional design, involvement of key users in the taking of requirements, tests and approvals.
2. Have the required data on the dates agreed in the planning (Integrations with the client products, access to environments, data to migrate, etc.)
3. Apply a correct change management and acceptance by the organization of the new services.
4. Technological difficulties with the client's systems.
5. My own understanding of Salesforce due to my lack of experience.

Some of these risks are more probable than others and they do not have an equal impact when emerging, therefore we have to prioritize some of them and put more effort to avoid their occurrence, while others can be easily mitigated beforehand.

We cannot count all the hours that depend on those risks, but just to be sure and proceed safely it is advised to count some of them in a controlled way. As a result of that, I have decided to count all the proposed hours and divide them by 5, as most of the risks have a probability of less than 10% of occurring. Therefore, some tasks have been inflated to handle these possibilities.

7.1. Difficulty in the approval of the functional design.

The approval of the functional design is discussed pretty early in the development of the project and it could add an iteration to that part. To redo the parts that are to be improved is not as time consuming as rewriting a whole new project. That could add around 10h to that part.

7.2. Have the required data on the dates agreed in the planning

There is a possibility that some of the data is not available when it should or is not presented as anticipated. The chances of this occurring are low, and the data affect once the project arrives at its end.

- If the data arrives later, we cannot do much but to wait, and all the Tasks that derive from this task (T5), can be completed at least partially.

- If the data is organized differently or contains different types of data. This case is very unlikely, but could happen. Depending on the case this could be easily fixed. But it could also be very complicated so it would be necessary to add 30h to the resolution of the project just in case.

7.3. Apply a correct change management and acceptance by the organization of the new services.

If the meetings and the communication with the client goes as planned, the probability of this risk should be really low. Those meetings are principally held to avoid this risk.

7.4. Technological difficulties with the client's systems.

Salesforce does not require a lot of resources at the client-end. It can be launched from an internet navigator as any other web and the application runs from both android and IOs systems.

Once the hardware is solved, we have to pay attention to the software. It could happen that some of the APIs that the client would add to the system are not compatible with our solution. In this case we should program some new APIs to at least fix those issues. It could add 40h to the project.

7.5. My own understanding of Salesforce due to my lack of experience.

In the case that we decide to implement something in such a way I am not familiar with, we should adapt the schedule as some formation would be required. Inetum provided me with an exhaustive formation and tools to handle the project anyway, so the probability of this risk is negligible.

8. Budget

During this section all the costs derived from the activity and development of this project will be explored. Moreover, this section will also contain the definition of diverse control mechanisms used for the management of the budget. With those mechanisms we expect to provide information to evaluate the relevancy of each deviance caused by the different analysed risks.

As the budget is a far-reaching concept, we will divide it into different sections with the intention of keeping things more clear and structured.

8.1. Costs per Personnel Activity

We have mentioned briefly the activities that the different actors will execute during the project before in the section named “human resources”. The cost for each activity or “task” is calculated by multiplying the hourly rate of the different actors that participate in said task per the amount of hours dedicated to it. Different personnel have different costs, so we must first classify these actors into different sets.

Since it is complicated to find the exact price for each actor, we will find the average hourly rates of a Salesforce Team Manager, Salesforce Developer and the ones from a Researcher and University professors. Those estimated rates can be found in [Glassdoor](https://www.glassdoor.com) in the area of Barcelona.

Project Role	Price per hour (€)	Roles played by Project Actor
Project Manager	29,25	GEP Tutor, Director, Supervisor, Intern
Salesforce Developer	19,59	Intern
Salesforce Sales Expert	19,31	Intern
Technical writer	19,78	Intern

Figure 5: Personnel hourly rates. Own elaboration.

To calculate the number of hours dedicated by each actor, we can use the following table, that has been made in accordance with the Gantt chart and the “Workload distribution table” seen in the section 6: Estimates and Gantt Chart.

Here I must make a clarification; for the counting of the hours in section 6 I only took into consideration the ones I would be doing as an Intern, working on the

project. The project itself requires some more hours that will be done by different actors, as it will be shown ahead. There are tasks that are done by two actors at the same time, namely the management and organization tasks, where for instance the developer talks with its manager to clarify or specify some things. In table number 6, all those hours are counted, because they affect the budget.

There is a total of 80 hours related to the work of a Project manager, obtained by summing the durations of the tasks T1 and T2.

A “Salesforce Sales expert” should have around 50 hours of work corresponding to the tasks T3.1.1 and T3.2.1.

The scope of the “Salesforce Developer” encompasses 285 hours of work on its own plus the work related to the support for the “Salesforce Sales expert” and the “Project manager”. This corresponds to the tasks T1, T2, T3, T4 and T5. Which sums a total of 415 hours of work.

The technical writer would be responsible for task 6, where 90 hours of work were counted.

In the following table we have information on the total cost for each Role:

Project Role	Price per hour (€)	Total Hours (h)	Total Price per Role (€)
Project Manager	29,25	80	2340
Salesforce Developer	19,59	415	8.129,85
Salesforce Sales Expert	19,31	50	965,5
Technical writer	19,78	90	1.780,2
Total Price per Roles			13.215,55

Figure 6: Price per Role. Own elaboration.

By summing all the prices per role we will obtain the total price for the personnel activities related to the thesis. We must note that although some of the work realised by the Technical writer focuses on the standards of the thesis and by extension the university, most of it is done anyway when documenting a project.

$$Total\ Personnel\ Cost = 2340 + 8129,85 + 965,5 + 1780,2 = 13.215,55\ €.$$

Personnel costs do have an added social charge that rounds about the 30% extra on the total cost. This percentage is an approximation.

Tax	Percentage (%)	Cost Before Taxes (€)	Taxes Cost (€)
Social Charges	30	13.215,55	3964,66
Total with taxes	17.180,21		

Figure 7: Taxes. Own elaboration.

8.2. Costs per Activity Production

This project requires some basic hardware and software to be produced. Due to the choice of using Salesforce, most of the hardware costs that would be related to the hosting of servers, and renting spaces are almost non-existent. By paying a license to Salesforce a client can have its final product functioning fully and remotely. Only a device with internet access is needed by both the developers and the clients.

It is estimated that all the work will be carried out using the same computer, and in the following table we can see the value and amortization.

Hardware	Units (u)	Price (€)	Average Usage (h)	Amortization (€)
Lenovo laptop computer	2	1100	300	171,87
Mitrastar HGU-GPT-2541GNAC	2	109	300	17.03
Total Hardware Amortization				188.9

Figure 8: Hardware amortization. Own elaboration.

To calculate the amortization we use the following formula:

$$Amortization (\text{€}) = Price \times \left(\frac{1}{Life\ expectancy}\right) \times \left(\frac{1}{Days\ of\ use}\right) \times \left(\frac{1}{hours\ of\ daily\ use}\right) \times h.\ used$$

Where:

Life expectancy = 4 years

Days of use for the project = 100

H. Used = Average Usage

This calculation becomes a bit more complicated when speaking about the software. Inetum uses Windows 10 operating systems alongside Microsoft Teams

for internal meetings and communications, and it is the tool we will be using when communicating and performing different management tasks.

As for the Salesforce licenses, this is where things are tricky. For the making of the project, meaning the design and implementation of the different functionalities, and even for testing them there is no need for a paid license. Only when we want to insert the data and start using it on a daily basis is when we must acquire some kind of license. Salesforce has a wide licensing system to choose from, and the manager of the project analysed which one suited best for the usage that is required.

In our case a Professional edition would suffice from the making of the Task 5 to the end. We will use a few hours to pass everything from our “playground” (how salesforce references to a free administrator environment) to our real organization. Therefore, we need salesforce licenses for 2 weeks before the deployment of the project. Those licenses are paid yearly and would be used even after the ending of the project for maintenance purposes. The cost is the full price of the licenses for the time used, as they are untransferable and every organization needs their own licenses.

The Microsoft licensing does have a cost as is being used by Inetum. Its usage must be contemplated in the budget. The prices for their licenses can be found on their [webpage](#).

Software	Quantity (u.)	Price (€)	Average Usage (h)	Costs (€)
Microsoft Windows 10 Pro	2	199,99	300	21,58
Microsoft Teams	2	12,5/month	60	8,47
Salesforce Professional	2	100/month	Not applicable	200
Total Software costs				230,05

Figure 9: Software amortization and Price. Own elaboration.

There are other costs related to the production and development of the project. Those are the costs associated with the place where the project is being made.

In this case, there are no travel or commuting costs due our actual full-remote work decision.

I live near Barcelona, in an area where the approximate price of the square meter is about 10 euros per month. Considering that my workplace is around 8 square meters, we could say that having the ability to work there costs me 80€ per month. As for the electricity cost, according to the page [tarifaluzhora](#) electric power costs 0.248€/Kwh during most of my working hours. My monthly bill is about 60 € but the part corresponding to the work is almost negligible because the personal computer adds about 100 Watts a day and my led illumination sums 60W. That would correspond to 5 cents per day. The project is done during the summer months, so there is no need for artificial acclimatation.

The internet is a fixed consumption for 29.99€ which gives me enough to navigate and develop without inconveniences.

Other items	Price	Daily usage (h)	Amortization
Work space	240	8	22,7
Electric cost	4.5	8	0,41
Internet cost	29,99	8	2,87
Total other costs			28,89

Figure 10: Software amortization and Price. Own elaboration.

The total cost derived from the activity production is: **444,93 €**.

8.3. Other Costs

Contingencies: Part of the budget must be reserved for possible unforeseen events or "contingencies". For this reason, we have to decide a contingency margin to cover partly the risks mentioned in their section above. A plus 10% of the total cost of the project should be enough.

Contingency cost: $18.590,93 + 10\% = 20.450,02 - 18.590,93 = \mathbf{1.859,09€}$

Incidental costs: There are also costs related to the adaptation of the alternative plans described that arise from risks becoming real problems. Here we will only take those plans and add value to the risks and the cost of them being produced. Moreover, we should add to the budget part of the estimated total cost for a contingency.

Most tasks have already slightly superior hour counting for the purpose of mitigating possible delays, so the budget addition for these incidents will be on the lower side.

Name	Estimated total cost (€)	Risk (%)	Budget addition (€)
Difficulty in the approval of the functional design	3500	11	329
Have the required data on the dates agreed in the planning	800	9	76,12
Technical difficulties with the client Systems	5000	9	458,78
Lack of experience	1050	20	101,89
Total risks costs			965,79

Figure 11: Incident costs. Own elaboration.

The total cost for the incidental costs adds **965,79 €** to the final cost of the project.

8.3. Total Cost

The total cost of the project is calculated by the sum of all the previous sections of the budget, meaning personnel costs, activity production costs and other costs.

Item Name	Cost
Costs per personnel activity	13.215,55
Taxes per personnel activity	3964.66
Costs per activity production	444,93
Incidental costs	965,79
Contingencies budget	1.859,09
Total Cost	20.450,02 €

Figure 11: Costs summatory. Own elaboration.

8.4. Management Control

In some projects it is possible for the budget and time estimates not being as accurate as desired. For that reason, the project must incorporate a mechanism to control different unforeseen circumstances. In this project we will add an estimated

deviation to each cost, so that in the case that some uncontrolled event occurs, the project has the ability of analysing it for future cases.

This deviation is calculated from the difference between expected resources used versus the real usage of resources. Every cost can be calculated following the same principle. The quality of the dimensioning of the project can be calculated after the project has finished, which may not help with this project but will certainly help in future cases.

The formula is as follows:

$$\sigma X = [(Estimated\ X\ value/h) - (Real\ X\ value/h)] \times Total\ h.\ consumed\ by\ X.$$

σ = *deviation.*

X = *Element being calculated.*

$h.$ = *Hours.*

When each element has been calculated, we will know the deviations on the initial budget. If the number is greater than 0, it means that the budget was oversized. If the number is less than 0, it means that the budget was too short. It is better for it to be a number superior than 0, with an approximate margin of +5% of the total cost of the project.

9. Sustainability

In this section the social and economical aspect of the project will be discussed. There are numerous aspects to take into consideration when speaking about the sustainability of an Information Technologies project, since they are becoming the central axis of many everyday interactions.

To start with that, we must first ask ourselves how much of an understanding we have about sustainability.

9.1. Self-assessment

Making the effort of analysing and discovering the current situation allows the students to gain perspective and to expand their knowledge in the matter in many cases. With this little exercise, a lot of projects acquire a renovated dimension, where all this matters gain in importance and visibility. In some cases it may occur that some things are designed differently in contrast to what the initial idea was because of different social, economic or environmental questions.

We are living in difficult times. There are many factors that contribute to a social and environmental outcome that is not desirable. Many of these factors can be grouped in two principal categories. The ones that derive from pollution and contamination, and the ones that derive from the current wealth distribution. We are all on this planet together so it is our duty to intervene and do as much as we can improve the situation or at least not make it worse. The project viability also depends on those matters. For a project to be successful, it is compulsory for it to be respectable with the environment and also to be adjusted to the current social situation.

During the making of this section I have had to face several questions that although I had some general idea regarding them, I did not have the required knowledge to answer them and adjust the project to those answers. Now, I feel more comfortable talking about these topics and have learned some key concepts that will certainly help me in the future.

In the following sections I will develop in more detail how the project affects in three different dimensions: firstly the economic dimension followed by the environmental dimension and last but not least the social dimension.

9.2. Economic Dimension

The economic dimension is where this project is conceived to live. As a last instance, it is a tool to facilitate the interaction between companies to perform their transactions. Those interactions had already taken place in the real world so this project focuses mainly on making them efficient. It has already been described in previous chapters of this thesis how it would affect those interactions. Making the process efficient means that with less resources and time, the outcome will be better than before, minimizing the energy consumption of both parties (our client and its providers).

For the cost of the project, we have chosen an environment that is both efficient for its purpose and scalable. Those perks are very important because they make the project viable economically in the long run.

9.2. Environmental Dimension

The project is also environment-friendly, as it benefits from high end technologies that when used well, improve the ecological footprint. For instance, the sole use of Salesforce (a cloud service company), means that our client won't have the necessity of holding servers on its own, and that Salesforce will hold all the data themselves alongside with the data of hundreds of other clients. This gives them the capability to reduce the global energy cost as they have the ability of increasing the performance of their machines using different techniques that can only be done when having a lot of economic power and hundreds of thousands of machines. According to their [climate action plan](#), salesforce has achieved in 2021 the goal of using renewable energies only for their economic activity.

9.2. Social Dimension

Obviously this project will benefit directly the people that have to use the applications in one way or another, but all the people somewhat related to the interactions generated by the use of the application will also receive part of the impact. For instance, employees that use the app will have an easier time performing their tasks, in a more user friendly environment. This will increase their productivity but will also improve the quality of their services, making the relations easier with other employees. Furthermore, it's an easy-to-learn environment, which will increase the chances of new incorporations participating in those activities, since the cost of instructing them would decrease.



Project development

10. Development Introduction

From this point onwards, this paper serves as documentation for a hands-on project construction, and in this part all the developing decisions and eventualities will be explained in detail. The project has already been designed and carefully planned in the first part of the thesis “project definition and management” and in this second and final part of the thesis the development will be explained in chronological order, following the gantt diagram made in section 6.3.

Alongside with the explanation of the implementation of the project, some concepts and concrete Salesforce methodologies and standards will also be explained, so that some of the decisions will make more sense to a person less experienced with Salesforce Development.

11. Project Specification and Design

11.1. The Actors

We have mentioned the actors before as stakeholders of the project so now we will only mention them and explain their roles when using the applications. Some of the stakeholders previously mentioned, do not take part in the utilization of the project in any way once it is deployed and published, so we will not count them as “actors”.

The client’s employees, as described before, are the one that will work for the company owning the final product or the application. There are also the client’s providers, the ones that will be put in contact with the client and will be “invited” into the provider portal to negotiate the different rates for their products.

Both of them will be renamed as “**users**”, and there are three kinds of users; **Sales Users**, **Marketing Users** and **Providers**. The first two are “the client’s employers”, the latter is, as the name suggests, “the client’s providers” or suppliers.

The Users are registered individuals, and as such they will have permissions to see different parts of the application and they will have different levels of access.

Finally, we have the **clients**. They represent “the client’s clients”. They are the principal reason for the project to exist. The clients will not have access to any of the parts within the application, but depending on the “case of use” they may get some data in form of mail or document. They also may initiate some of the processes that will be explained in the “use cases” section.

11.2. Definition of requirements

As we have previously mentioned, the project is divided into two main parts; the private management application and the provider portal for quotes. Even with this distinction, we must take into account that these parts are not independent and many processes occur between them. More specifically, the provider portal “drinks” from the private management app. It utilizes some of the data on it and that's one of the principal reasons as to why we started developing the management application first.

The idea is to build a space where all the data generated from the interaction between marketing, sales, the providers and the clients will be stored and managed. With said data we want to accomplish the following things:

- Have a visual representation of the whole **sales process**, from the contact with a client to closing. In this case the whole business idea revolves around “queries” or opportunities. A query represents a potential sale for a given account that you want to work or track. These queries have different stages, depending on the state in which we define the status of said query. They start on “New open” and go through “Pending”, “Proposal”, “Awaiting Approval Provider Final”, “Negotiation/ Review” and finally closed, that may be “Closed Won” if the query was finally “sold” or “Closed Lost” if for some reason we lost the client interest in said query.
- All the **data generated** in the previously described process must be saved and treated. With the accumulated data from different queries we will be able to make predictions or to evaluate the different performances of the actors or items involved.
- Different **application forms** that will allow further modifications. These forms should include information related to the object in scope. In the case a new user demonstrates an interest in a product or contacts the company, all the data has to be collected and stored for that user. If a sales user is in contact with a potential client, the sales user has to be able to open a new account for the user if it is the first time that he contacts the company or to open the existing account if the user already contacted the company in the past. If the user contacts via the web form, all the data that is not empty should be collected and stored in a new “lead”. A lead represents people that may be interested in the product or company. When a new lead is generated, it is of

great interest to try to convert that lead to an account. Once it is an account, we may initiate the “sales process”, creating a new query related to a lead.

- For a query to be valid, it has to have some **products linked** to it, as it is not possible for a client to be interested in zero products and have an open query.
- For those forms, different **data types** will be created. Each one of the data types has to allow the different things:
 - Edit the labels related to the field.
 - Make the field “required” for the form or page.
 - Arrange it in a different way.
 - Make it invisible in different parts of the application.
 - If the field has different options, it has to allow new options or manipulate the existing ones.
- Generate different **templates** for **auto email responses**. When a client is contacted and a new query is generated with the different products of interest, both the client and the providers have to receive a mail with the following-up of the query. In different stages of the query, they should be updated with different emails. Once the product is ready, the client has to be contacted with the details.
- **Approval workflows**. Some queries will exceed a price threshold, a supervisor should be informed and kept updated. For an important query to advance stages it is necessary to receive permission from the supervisor. The approval has to be shown via email.
- In some cases, **Alerts** have to be generated. When a new lead is generated, it is assigned to a sales user automatically. The process has to generate an alert to inform said user. Other processes also generate alerts; changes in the “sales process” related to the user, status changes in an approval process or changes in the status of a product made by a provider.
- The information collected in the application must be used to generate **automatic reports**. Said reports have to be modifiable and the visibility has to be personalized.
- **Dashboards**. With the data collected and organized in reports, visual representation of the different cooperative metrics must be shown to the

users. Every user has a personal “homepage” where important information can be found. This information allows to rapidly get in touch with its current situation (his open queries, monthly reports, etc.).

- Different roles have different **levels of access**. The providers should only be able to access the providers portal page, and the sales users only access to the queries, the leads and other information directly related to them. Marketing users must have visibility to the whole application except for some parts that the upper management may dictate. As such, the application must be moldable in that sense.
- **Security**. The management application should register every log in and its location. It has to allow only authorized logins from defined places or specific machines during the working hours. The provider portal on the other hand should be accessible for the providers, and the access has to be more lax. New users may be generated in the future, their activation and deactivation has to be contemplated.
- **Configure FAQs**. A link to the main web page must be available in the provider portal, and some information has to be present in said portal as a promotional carousel.

Furthermore, we have some requirements that are not functional. These requirements are related to the quality and usability of the final product. In this case, we do not specify what the application should do.

- The application must be **accessible from mobile devices or tablets** as well as computers. Some functionalities may be not available from the phone or tablets.
- **Usability**. Every menu and option must be easy to understand and to guess its functions and immediate implications.
- All the data inserted in the database must be **reliable**.
- **Security/ up-time**. The system must assure that data will not be lost and that it will be accessible when needed.
- The new tools and apps must be **compatible** with the processes that the client had and that are not going to be replaced by the system.

11.3. Use cases

A use case is a description of the steps or activities that have to be followed or executed to complete a process. In the context of the project, a case of use is a sequence of interactions that will be performed between a system and its different actors in response to an action that executed a primary actor on the system. With the diagrams of use cases, we can specify the different behaviours and communications that a system and its actors can have.

This section will be organized in the following way; each one of the use cases will be explained in more detail using a table-formatting to be more systematic and organized. We will start with the use cases related with the management app and later with the provider portal ones.

Explanation for each use case:

Use Case	Creating a Lead via management app
Description	Create a Lead and register it in the system.
Actors	Sales User, System.
Normal procedure	<ul style="list-style-type: none"> - Sales User clicks on the Lead tab - Proceeds to click on “new”, a form will open. - Then fill the fields. <u>Only Name</u>, <u>Last Name</u>, <u>Lead Status</u>, <u>Company</u> and <u>Product Category</u> are required fields. - Lastly, click on save.
Result	The Lead is created and is in the preferred stage. Modifications in the fields and further options are available.
Alternatives or possible errors	The process can be cancelled at any point. There are no errors possible at this point. Until the lead is not “converted” there are no “contacts” modified, so no check for duplicates is made at this point.

Use Case	Moving through different tabs
Description	Sales and Marketing users have the ability of moving to different tabs. There are fixed tabs and “temporal tabs”.
Actors	Sales User, Marketing User, System.
Normal procedure	<ul style="list-style-type: none"> - Click on the desired tab. - The system will redirect to it.

Result	The user will be in the desired tab and is able to see the information related to it.
Alternatives or possible errors	If the Tab is “temporal” the user has the ability of closing it. Tabs have an inbound desplegable menu that allows the user to perform some actions. To create an object related to the tab or to go directly to a record. No errors are foreseen here.

Use Case	Subscribe to Dashboards
Description	All users have the ability of subscribing to Dashboards. The dashboard must be accessible to the user to have the ability of subscribing to it.
Actors	Sales User, Marketing User, System.
Normal procedure	<ul style="list-style-type: none"> - Go to the desired dashboard (using a link or searching it in the menu). - Click on <u>subscribe</u>. - Configure the settings for subscription; Frequency, which days. At what time and the option of sending it to the mail address. - We can add more recipients to it.
Result	The user will have access to the dashboard and it will be scheduled as desired. If the option of emails is selected, it will be sent to its mail address.
Alternatives or possible errors	The process can be cancelled and no further actions will be performed. Recipients of the email need to have access to the dashboard or a blank page will be shown.

Use Case	Creating an account
Description	Accounts represent mostly companies in Salesforce. By creating a new account we can connect contacts to it.
Actors	Sales users, System.
Normal procedure	<ul style="list-style-type: none"> - First, we must click on the “Accounts” tab. Alternatively the desplegable will be selected. - Click on the “new” button. - Accounts only have one required field: “<u>Account Name</u>”. There are other optional fields that are used to store more information about that account. - Lastly, click on save, or “save and new” if more accounts are to be created.
Result	A new account will be created in our DataBase.

Alternatives or possible errors	The process can be cancelled and no further actions will be performed.
--	--

Use Case	Creating a Contact
Description	Contacts will be representatives of an account. Contacts that are not converted from leads are “providers”, not clients.
Actors	Sales users, System
Normal procedure	<ul style="list-style-type: none"> - First, we must click on the “Contacts” tab. Alternatively the desplegable will be selected. - Click on the “new” button. - Contacts have a few required fields; “<u>Name</u>”, “<u>Last name</u>” and “”. There are other optional fields that are used to store more information about that account. - Lastly, click on save, or “save and new” if more accounts are to be created.
Result	A new contact will be created. If we left the “account” field empty the contact will not be linked to an account yet.
Alternatives or possible errors	The process can be cancelled and no further actions will be performed.

Use Case	Converting a Lead
Description	When a new lead is registered in the system, either manually or via other means (website, emails, etc.), we have the option of converting it to a contact that will have a query or petition to our company.
Actors	Sales users, System.
Normal procedure	<ul style="list-style-type: none"> - Click on “Converted” and then click “select convert status”. - A new pop-up will be shown. We can select “Create New Account” or existing. The same with Contact. We can Create the “Query” or opportunity here too. With yet again the same options of creating new or choosing existing. - The Record Owner and Converted Status fields can be left as they are because it autofills even though they are required. - Click on convert.
Result	Depending on how the user chose in the “Convert Lead” menu, the result is a new account, contact and query. If “Choose

	Existing” was marked at any of those records, the information is added to the respective chosen record. For instance, if an existing account was picked, the contact and related Query will be related to that account.
Alternatives or possible errors	The process can be cancelled and no further actions will be performed.

Use Case	Deleting a record
Description	Users have the ability of deleting records that are owned by them or by anyone under their hierarchical scheme.
Actors	Sales users, Marketing users, System.
Normal procedure	<ul style="list-style-type: none"> - Every record shown in a list has a desplegable menu on its right. Click on it. - If the necessary conditions are met, three options are shown; <u>edit</u>, <u>delete</u>, <u>change owner</u>. - Click on delete. - A pop-up will show to make sure the correct option was selected. - Click on “delete”.
Result	The record will be erased from the database.
Alternatives or possible errors	<ul style="list-style-type: none"> - If a record with dependent records is marked to be deleted, we must first delete the related objects. - The process can also be canceled at any point. - There is also the option of deleting a record when looking at it. At the top right corner there is a desplegable menu with different options, one of them is to delete the object.

Use Case	Creating Queries
Description	A query represents a potential sale for a given account that you want to work or track. We can create them at different points of the process. As said before they can be generated when converting a lead or from other places.
Actors	Sales User, system.
Normal procedure	<ul style="list-style-type: none"> - First go to a contact that we want to relate the query to. The contact is the “client” in this case. - Second, click on the Related button and then new on queries.

	<ul style="list-style-type: none"> - A pop-up window will be shown. - All Required fields must be filled. These are "<u>opportunity name</u>", "<u>close date</u>", "<u>Stage</u>", "<u>Product Category</u>", "<u>Product Multiplier</u>" and "<u>Product discount</u>". - Click on save.
Result	A new Query will be created. We have yet to include the providers that will be chosen to deal with this specific query.
Alternatives or possible errors	The query can also be created from the queries tabs, and then we will have to select the contact to which the query is related in addition to the required fields.

Use Case	Add new Products
Description	Products represent the items that Lyreco sells to its clients. Many providers can manufacture the same products. When adding a product, we can start selling it and adding it to queries.
Actors	Sales users, system.
Normal procedure	<ul style="list-style-type: none"> - First, click on the products tabs. - Then, click on "new" on the top right corner of the screen. - A pop up will show. There are two required fields; <u>Product Name</u> and <u>Type</u>. - Click on "add product".
Result	A product is added.
Alternatives or possible errors	We can cancel the process at any time.

Use Case	Add quotes to a Query
Description	Queries need quotes for the product to be sold or "offered". Those quotes are going to be presented in the provider portal.
Actors	Sales users, system.
Normal procedure	<ul style="list-style-type: none"> - First, go to the query we want to add quotes to. - Later, on the related items part, click on add quote. - Quotes require a <u>Quote name</u> and a <u>provider</u>, that is an account, not a single contact. All contacts from that account will have access to that quote.
Result	A new quote is added to the query and it can be seen from the provider portal by the related providers.

Alternatives or possible errors	We could fail to select the account of the providers correctly. We can delete a quote at any time.
--	--

Use Case	Managing the products of a quote.
Description	A quote is always related to one query. When we create a quote, the products are added automatically, but we might desire to show only a few to the provider or maybe propose another for any reason.
Actors	Marketing users, Sales users, system.
Normal procedure	<ul style="list-style-type: none"> - First, we go to the quote that we want to manage. - We will see a table with information about the products. - From there we can click a desplegable button that has two options: edit or delete.
Result	The product will be able to be edited or to be deleted.
Alternatives or possible errors	The process can be cancelled and no further actions will be performed.

Use Case	Marking a contact as a provider
Description	When a contact is a provider, it is granted access to the provider site. An email is sent with its login information and from that point will have access to the quotes related to its account.
Actors	Marketing user, Sales user, system.
Normal procedure	<ul style="list-style-type: none"> - First, we have to go to the account that we know is our provider. - Second, we must mark that account as a partner account. - Then, we must go to the contacts related to that account that we want to give access to the provider portal. - Lastly click on "Enable partner user".
Result	The contact that we marked will have access to the provider portal.
Alternatives or possible errors	If the account is not marked as a partner account, all users that had access to the provider portal will lose it.

Use Case	Creating Dashboards
-----------------	---------------------

Description	Dashboards are useful visual representations. When creating a dashboard, we can select many options and configure it in many ways.
Actors	Marketing users, system.
Normal procedure	<ul style="list-style-type: none"> - First, go to dashboards. - Second, click on new and give it a name and add it to a folder. The folder determines which visibility it will have. - Third, click on components. Each component can represent different pieces of information. We can add 12 components and place them in different ways. - Lastly click on save.
Result	The dashboard will be configured as desired.
Alternatives or possible errors	We can cancel the process. We cannot save a dashboard to a folder that we do not have access to.

Use Case	Discovering a record/ using the search bar
Description	We can search for any record (data) at any point by clicking on the search bar.
Actors	Sales Users, Marketing Users, System.
Normal procedure	<ul style="list-style-type: none"> - First, go to the search bar on the top-mid of the page. - Second, type in the information that we know about a certain record. - All the results will be shown.
Result	The list of records that the user searched will be shown.
Alternatives or possible errors	Perhaps the record does not exist in the system or there was a typo in the search. In that case the system will return an empty list.

Use Case	Queries flow
Description	Queries have different stages. These stages represent different moments in the circle of life of the query.
Actors	Sales Users, System.
Normal procedure	<ul style="list-style-type: none"> - First, go to the query that we want to modify. - Second, click on the stage that we want the query to be into.

	- Click on the blue button “change stage”.
Result	The query will be now set to the new stage.
Alternatives or possible errors	Some stages require some actions to be done. We cannot mark a query as done if some of its quotes are still “pending”. Therefore we must first close a quote with the provider and then close the query.

Provider Portal use cases;

Use Case	Log in
Description	Initiate a session in the system to access the data.
Actors	Providers, Sales Users, Marketing Users, System.
Normal procedure	<ul style="list-style-type: none"> - First, go to the provided link in the welcome mail. - Second, enter your credentials - Lastly, click on log-in.
Result	The user will be logged-in to the system.
Alternatives or possible errors	If bad credentials are provided or the user is no longer admitted in the system, the log-in process will not be possible.

Use Case	Log out
Description	Disconnect or close the session a provider has open.
Actors	Providers, Sales Users, Marketing Users, System.
Normal procedure	<ul style="list-style-type: none"> - First, go to the user icon in the top right corner of the screen. - Click on log-out - The login page will be shown.
Result	The user is disconnected successfully from the system. The user must go through the log-in process again to access information.
Alternatives or possible errors	When inactivity is detected the account is also logged off. The process itself does not forecast any errors.

Use Case	Move between tabs
Description	The provider portal has a home page and a “quotes” tab. The

	home page shows general information whereas the quotes page shows information relevant to the quotes of the user.
Actors	Providers, Sales Users, Marketing Users, System.
Normal procedure	<ul style="list-style-type: none"> - First, click on the desired tab that you want to go to in the top menu. - An automatic redirect will be done.
Result	The user will see the information related with the desired tab.
Alternatives or possible errors	There are no possible errors at this point.

Use Case	Contact Support
Description	Sometimes a user might be lost with something or may require some guidance. A support option has some FAQs and a button to make contact with someone that may help.
Actors	Providers, Sales Users, Marketing Users, System.
Normal procedure	<ul style="list-style-type: none"> - First, go to the user icon in the top right of the page. - A desplegable will show. - Click on “contact support”.
Result	The user will be in a page where all information related with “contact support” is shown.
Alternatives or possible errors	There are no possible errors at this point.

Use Case	Find my quotes in the provider portal
Description	Providers must have a way to manage all their transactions with Lyreco. The “my quotes” page shows a list with all the quotes that are related to them. All include the ones that are already closed.
Actors	Providers, Sales Users, Marketing Users, System.
Normal procedure	<ul style="list-style-type: none"> - First, go to “my quotes” in tabs. - Then, all “my quotes” will be shown.
Result	The user will see all its quotes in the same page.
Alternatives or possible errors	It is possible that the table has not been accessed in a while and

possible errors	“the recent items” option is marked. In that case the table may appear empty. To change it just click on that option and select the one the user prefers.
------------------------	---

Use Case	Modify the Quotes
Description	The provider portal has been created for the providers to change the prices of the listed products or to select if they are interested in selling them or not.
Actors	Providers, System.
Normal procedure	<ul style="list-style-type: none"> - First, the provider must select the “my quotes” tab. - Second, click on the quote that we want to interact with. - Third, go to the related sub-tab. - There we will see the items that we can manage. We can edit the “sales price” and the status. With the status we determine if we want to sell that product or not. - Click on save.
Result	The quotes are changed as the user wants them.
Alternatives or possible errors	There are no possible errors, since the system will warn “on live” if any parameter has been introduced in a wrong or strange way.

11.4. Class diagram

Data Model used in Inetum for Salesforce:

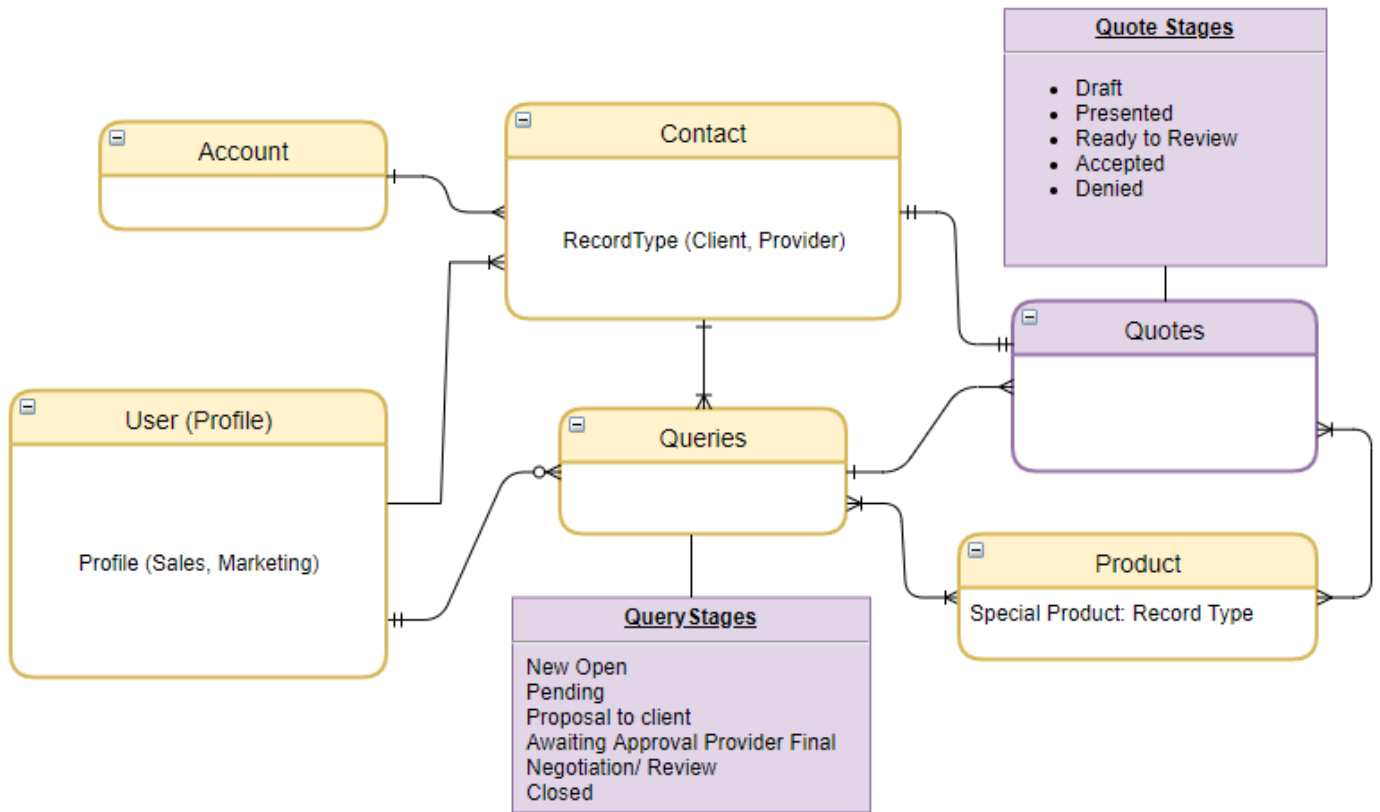


Figure 12: Data Model. Source: made in Inetum.

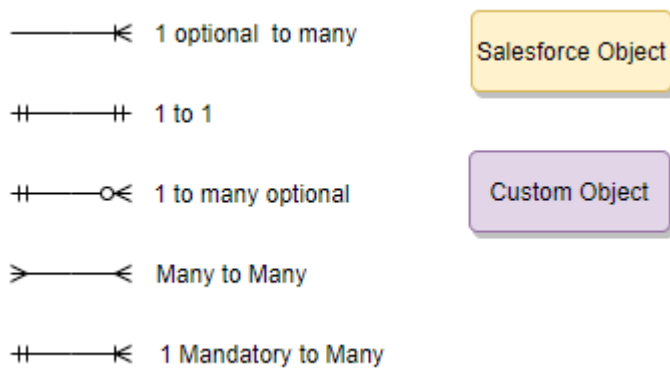


Figure 13: Object and relationships guide. Source: made in Inetum.

12. Implementing in Salesforce, Apex and SOQL

To develop in Salesforce, we have two main routes of action; Use the “administration mechanisms” that are offered in a point and click fashion- or we can code as in the old ways. Many things are achievable with the administration route, but there are some that have to be coded. There is only one way to know which ones are doable only by code, and that is knowing the environment and having experience with it.

To code for salesforce we will use the “Apex” language. Apex is an object-oriented and strongly typed programming language that enables developers to perform commands and transaction control statements on the servers in conjunction with calls to the Salesforce API. It utilizes a syntax that looks like the widely known Java, Apex allows developers to implement business logic to most system events, including related record updates, webpages and button clicks. To initialize a chunk of Apex code we can utilize triggers on objects and service requests.

Like java, Apex has expression syntax and loop syntax, as well as array and object notation. As a newcomer to Apex, with a light experience in Java you will not be lost. Furthermore apex is designed to access and utilize big databases. It can thread multiple queries and DML statements into units of work on the Salesforce Servers.

Along with Apex, we use SOQL (Salesforce Object Query Language) to interact with the organization’s database hosted in the Salesforce servers. It is really similar to the Structured Query Language (SQL) but integrated for Salesforce. Utilises the “SELECT” structure and works almost in the same way. There are some differences between them and we will explain why there are those differences in the following lines.

Because Apex and all its processes run in a multitenant environment, the Apex runtime engine strictly enforces a set of limits in order to ensure that some processes do not monopolize the shared resources. Elseway some tenants would suffer from the exhaustive use of other tenants. If a chunk of Apex code exceeds a defined limit, the associated governor issues a runtime exception that cannot be handled in any way.

There are different types of Governor Limits, and as a Developer they are one of the most important things to have in mind while designing and programming in Salesforce. Those types include:

- Size-Specific Apex Limits.
- Lightning Platform Apex Limits.
- Static Apex Limits.
- Miscellaneous Apex Limit.
- Per-Transaction Apex Limits.
- Per-Transaction Certified Managed Package Limits.

These limits have its reason in the shared use of resources, and a way of normalizing is to cut the number of operations that the code can do. To make it more understandable and visual, I will list some of these limits in a more tangible way;

- Total number of SOSL queries issued : 20.
- Total number of SOQL queries issued: 100.
- Total number of DML statements issued : 150.
- Total number of records processed as a result of DML statements : 10,000.
- Total number of records retrieved by SOQL queries : 50,000.

As we can see, from a Development perspective it is of utmost importance to ensure that our code is scalable and never hits governor limits, or it would be stopped and it would affect our business. There is a guide of best practices to avoid that and some of them are listed below;

- Bulkify your Code.
- Avoid SOQL Queries or DML statements inside FOR Loops.
- Using Collections, Streamlining Queries, and Efficient For Loops.
- Streamlining Multiple Triggers on the Same Object.
- Querying Large Data Sets.
- Use @future Appropriately.
- Use batch apex if you are working for more than 50000 records.
- Never make any SOQL, DML operation inside the loop.

13. Private Management Application

The private management application is the main part of the project, where all the sales processes of the client will be executed. The application will be accessible from a web navigator from any device. Furthermore is accessible from portable devices using the [salesforce application](#).

This application will have all the information relevant to the sales and marketing team in an organized way. To achieve this, we have conducted several meetings with the client to fully understand its company policies and internal processes. The main reason for the usage of Salesforce in this case was the almost immediate translation of the original company process to the one that salesforce has pre-build in all its deployments. The standard Sales process is highly intuitive and adaptable, meaning that most of the companies out there that dedicate their business model to sales can utilize most of the standard process.

The usage of Salesforce accelerates the process of building the application, since most of the job was tweaking and redefining the already built-in objects and fields. Thanks to this, we will have from day one a lot of functionalities working on our process. For instance, Salesforce has an object named “opportunities”, that is capable of tracking the status of a relationship with a client. From this, generating a report to keep track of the general scheme of things or to predict the viability of a particular opportunity using data science can be performed in a matter of minutes.

As we could see in our data model in section 11.5 most of the objects used for the development of this part of the project are Standard Salesforce objects. There are many particularities that had to be done using the different tools that salesforce offers, and we will explain some of them in the following pages.

13.1. Configuring the sales process

Once we have the data model and the data access clear, we must start using the app manager. The App Manager is the place in Salesforce where we are able to manage our apps for **Lightning Experience**. This tool allows us to make different apps connected to the same Salesforce Organization. Lightning experience is an up-to-date interface that performs really well and allows our teams to reach greater performance and productivity. The apps made in the app manager using lightning experience are called **Lightning Apps**.

Is at this moment when we have to decide what objects are going to be shown in the app. The different apps built have access to the same information, but they can

display it in many different ways. In our case, we will include the objects from our data model for salesforce from section 11.5. Those objects will be shown on the left top of our main app, just as shown in the figure below:

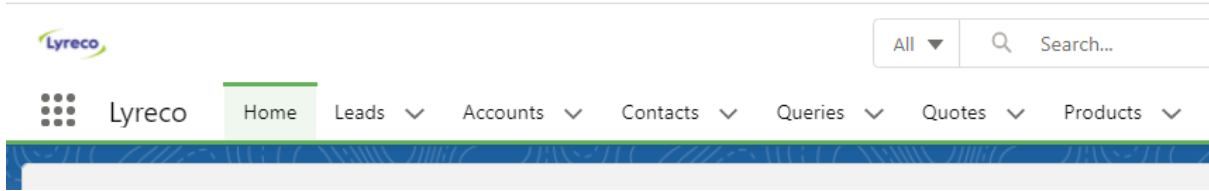


Figure 15: App Navigability. Source: Own creation.

We also added a “Home” tab, where general information will be displayed. The standard objects have a default name. We have previously mentioned that, and now is the moment to rename Opportunities to Queries, as our client wants it to be called.

Here we will have to pay attention to different areas in our application. There is a distinction between these areas but many are interrelated. It is presented in that way in favour of clarity.

13.1.1. Profiles and data security.

One of the first things that we must do is to configure the different profiles. A profile in salesforce defines how users access data and objects and what they can or cannot do within the different applications. Access to certain parts of the application can be granted or revoked with different tools and mechanisms.

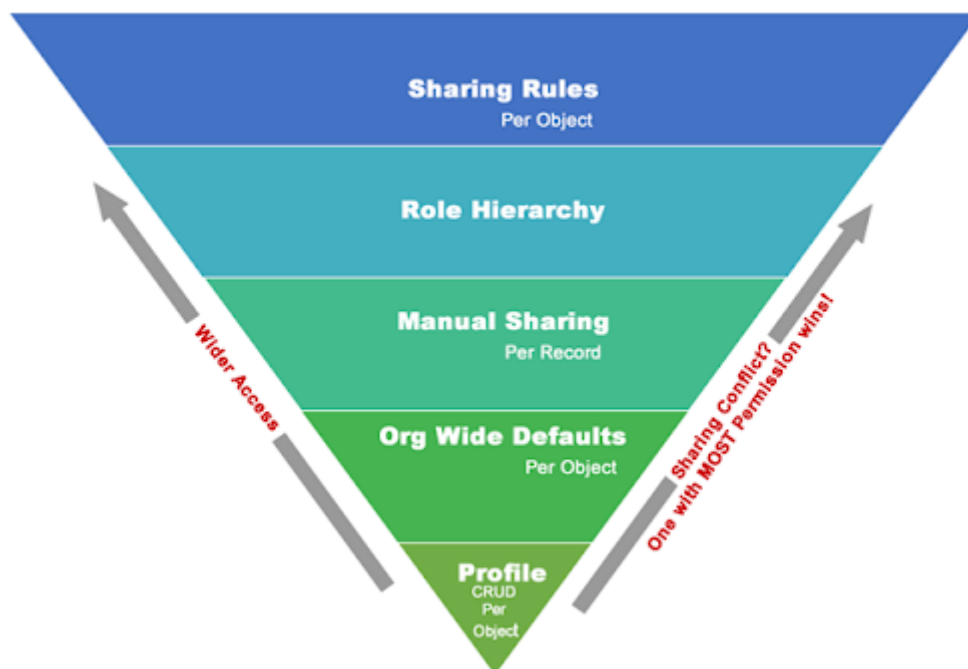


Figure 14: Access in salesforce. Source: Salesforce General.

As we can see in the figure 14, Profiles are the angular part to control this. It makes a lot of sense to define this part clearly before proceeding.

We start from the standard Marketing user, which we will clone to create the sales user. The sales user in our case has a lesser level of access, since they do not need to access the whole marketing package, therefore we only have to reject some permissions and we are done.

It is also possible to control login hours and valid IP ranges to connect into the application. These are some of the security measures that we will implement to improve the security of our systems by avoiding unauthorized accesses. Profiles can be used to make distinctions between users, we want the admins to be able to access at all times, and perhaps the managers too. This flexibility is very important when implementing these functionalities.

We can also enable a real-time event monitoring system which will give us a set of possibilities. For example, we will be able to know when the data has been accessed and by whom. Also shows the threats detected in our organization, such as anomalies in how users view or export reports, session hijacking attacks, or credential stuffing attacks.

This system does not only provide us with some logging measures for security, but also shows some metrics for usability and reliability. For instance we can know how long it takes a Lightning page to load.

13.1.2. Data inputs

Once we have our primary setup, we have to start planning ahead. We know, from the use cases section, that sometimes leads or prospects will be attracted through the client's webpage. Therefore, we have to prepare for the insertion of information via the web. Salesforce offers an option for this, and is called web-to-lead. Whenever a possible customer fills a form on our website and sends the information, it all is collected by salesforce and automatically a new Lead is generated. This lead can be assigned to whoever we want.

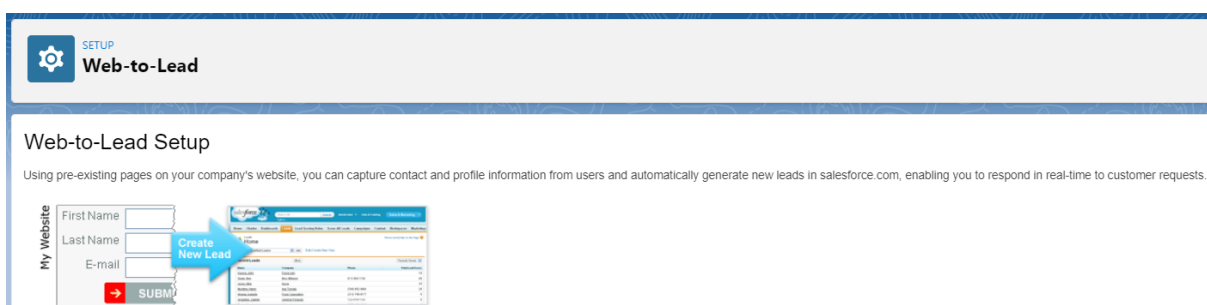


Figure 16: Web to lead Setup. Source: Own creation.

The process is simple, we must configure the html to include the redirection page, and link the fields of information within the form to fields in our database. These fields will be custom made and they will validate the data automatically. If an integer is expected and we receive a string, Salesforce will call an exception. The lead object will have some custom fields that are used to capture information. For the purposes of this project, we have made a fully operative concept for the web-to-lead, following existent forms found on the clients webpage.

In the following picture we can see how it would look on the spanish website of the client:



Figure 17: Web to lead example. Source: made in Inetum.

13.1.3. Data relationships

Following up with our setup process, we must link our objects the same way it is displayed on the data model. This is very simple; we only need to create custom fields within the different objects with the same data type, and then establish a relationship between them.

As explained in their user manual, an object relationship in Salesforce is a two-way connection between two different objects. These relationships are born by creating custom relationship fields on a given object. When the object is viewed, all linked objects are also seen. There are some relationships already defined in a basic environment of salesforce. Those relationships are called "standard relationships". An example of this are the relations between Accounts and Contacts. These relationships can work in one way only. In this case one account can have many

contacts related to it.

There are three types of relationships in Salesforce:

- Master-detail
- Lookup
- Hierarchical

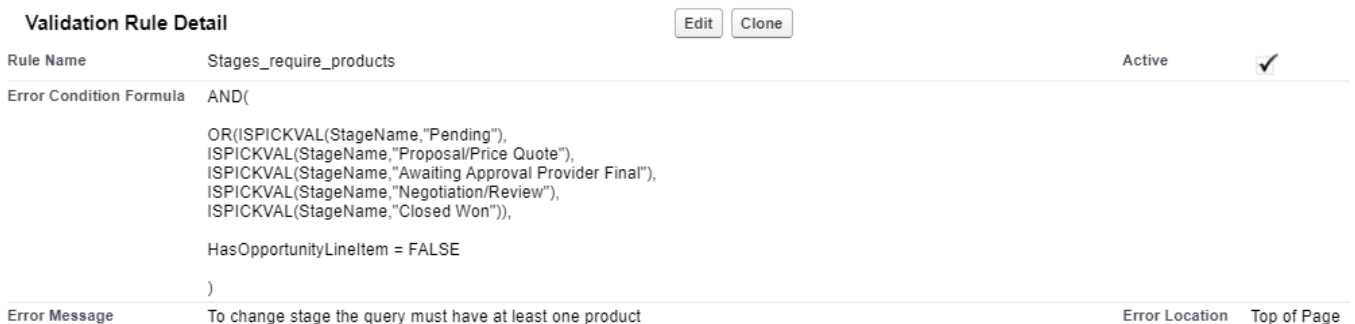
Some of the standard relationships are handy to us, since we can maintain most of the pre-build infrastructure.

Queries, contacts, users, products and accounts are the standard salesforce objects that will not require most attention since we are using them as they are. The relationships established in the data model are already built in and fully operational.

13.1.4. Data Reliability

Previously we have talked about data reliability, and this can be accomplished by different means in Salesforce. The most common way is the usage of “Validation Rules”. These rules are used to verify that the data that is entering the system in a record meets the standards of quality specified in the setup. A validation rule can contain an expression or a formula that evaluates the data that would be inserted. These expressions return a boolean value (True or False). If it evaluates to true, the validation rule will return a configurable text warning the user.

We need these rules to tweak our sales process more exhaustively. For instance, we know that a pending query or opportunity must have at least one product since there cannot be any empty request from a client. Therefore, we decided to make it mandatory to contain at least one product before we can change the stage of a given query to “pending”. This validation rule must be created in the query object, since it is the one that will be affected by it. It will look like this:



Validation Rule Detail		Edit	Clone	Active	✓
Rule Name	Stages_require_products			Active	✓
Error Condition Formula	AND(OR(ISPICKVAL(StageName, "Pending"), ISPICKVAL(StageName, "Proposal/Price Quote"), ISPICKVAL(StageName, "Awaiting Approval Provider Final"), ISPICKVAL(StageName, "Negotiation/Review"), ISPICKVAL(StageName, "Closed Won"),) HasOpportunityLineItem = FALSE))				
Error Message	To change stage the query must have at least one product			Error Location	Top of Page

Figure 18: Validation rule. Source: made in Inetum.

As we can see in figure 18, we can also choose where the error will be seen in case the expression evaluates to True.

We have another example below on how validation rules can be used; This time we can assure that no sales user can make a discount greater than a certain threshold. By creating a custom field and assigning it a value we can make the threshold editable. If the discount is greater than we defined, it will be impossible to present it

since a message will appear in that part (see error Location in figure 19) and the data will be rolled back.

Validation Rule Detail		Active <input checked="" type="checkbox"/>
Rule Name	Discount_is_lesser_than_Multiplier	
Error Condition Formula	Product_Multiplier__c <= Product_Discount__c	
Error Message	The discounted percentage must be lesser than the percentage multiplier.	Error Location Product Multiplier

Figure 19: Validation rule 2. Source: made in Inetum.

13.1.5. Mailing system.

When using Salesforce, we have the option of sending automatic mails if certain events happen. The mechanism is called “auto-response rule”. For our project, we will be setting multiple auto-response rules to fulfill our client’s requirements.

When we capture a new lead from the previously explained Web-to-Lead, we can choose to email the client with a greeting or simply letting him know that the inquiry has been processed.

This mechanism is also used when the sales user decides that it is time to send the offer to the client. The client is warned with an automated email containing the basic information of the quote. In the figure 20 below, we have an example of the setting up of an automated mail system.

Email Alert	
Email Alert on New Quote Presented	
E	
Email Alert Detail	
E	
Description	Email Alert on New Quote Presented
Unique Name	Email_Alert_on_New_Quote_Presented
From Email Address	Current User's email address
Recipients	Related Contact: Contact Name

Figure 20: Email Alerts. Source: made in Inetum.

Furthermore, the sales users and the marketing users will have the ability to see the email responses in the “Activity History” related list for a contact or lead, and it is also possible to connect their preferred email providers to Salesforce and choose to respond to the client from there.

That page has the following structure:

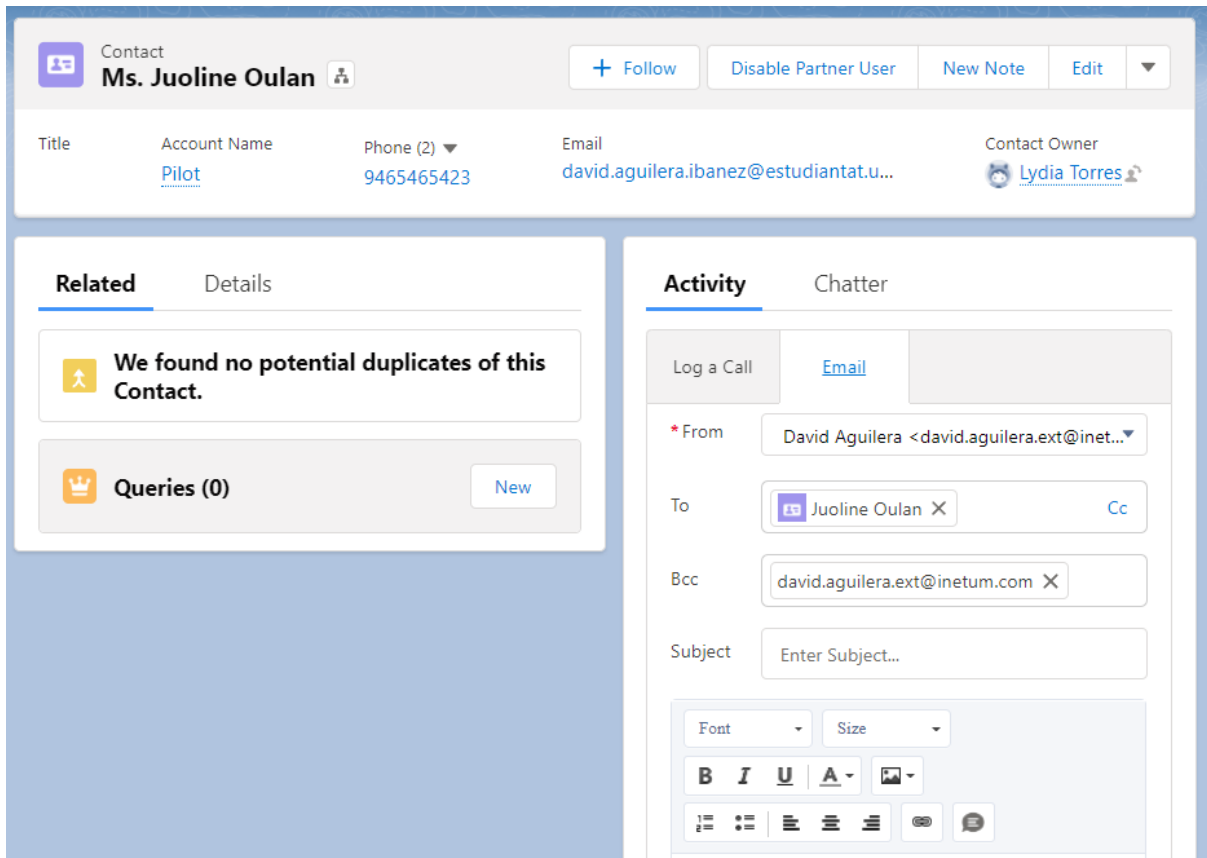


Figure 21: Email from Activity. Source: made in Inetum.

This panel is extremely useful for the day to day work of a sales representative. We avoid wasting a lot of time reducing context switching, we no longer need to go back and forth between different web navigators and we do not have to keep checking the emails of destiny, as it will be automatically filled with the relevant information.

13.2. Modelling in Salesforce for provider quotes

Quotes represent a proposed price for a product or service. An opportunity (Query in our case), that has products will need at least one quote. The product itself does not have a fixed price. Therefore, each Query can have multiple related quotes, and any of them can be synced to the Query. The problem with the standard behaviour is that when we sync a Quote, every product in the Query is updated with the values in that same Quote.

In our case the idea is that every Quote will be linked with a different provider, and every provider will have offerings for different products. Furthermore, it is possible to accept a product from a Quote and to refuse the other in favour of another provider.

For example, our client may want pens and notebooks. We, acting as a sales user, will create different Quotes, to link the products from the new Query to the different providers that we know that may be interested in selling said products. Once we have the information updated. Maybe it is in our best interest to select the pens from one Quote and the notebooks from the other. To be able to perform this action, a new behaviour is needed.

To implement that behaviour we have decided to make a new button which will call a “flow” responsible for this action. The button will be very visible, in the top right corner of the screen and will be very intuitive. There is another action required from us. We need to add a selectable field (or picklist in Salesforce), to mark a product in a quote as “awaiting”, “accepted” or “denied”. Depending on the status the product will be processed when pressing the button or not. This way we will be able to choose more swiftly which products we want. Moreover, that picklist must be included in the page, so that the sales users can see and edit with ease the different products.

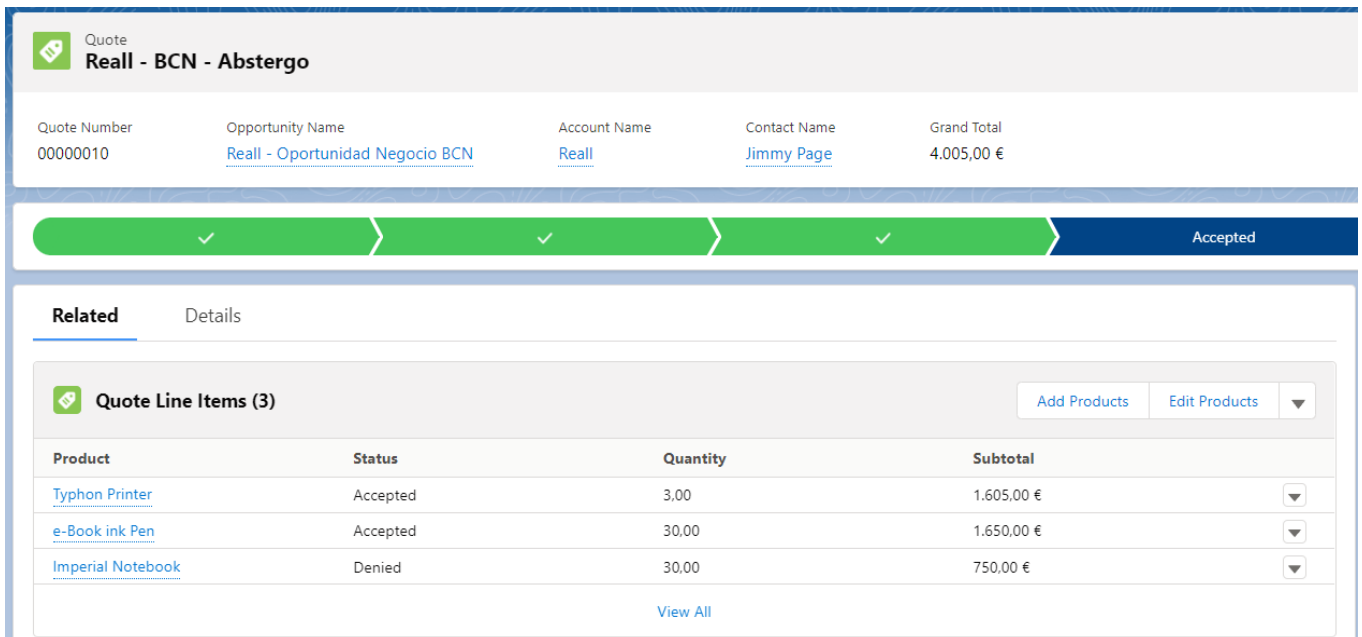
There is the possibility that a sales user accepts the same product from more than one Quote. Another possibility is that a product from a Query has no accepted Quotes. Those options will be discussed later, and we will explain how we took care of them.

From this point we have decided to include some visual aid for the reader, since using a text-only explanation may be too abstract.

A Quote in our application looks like the figure below; we have the Quote number, so we can quickly know which one we are looking at and we have the opportunity name (or quote name) that this quote is related to. Just below, we have an intractable menú that allows us to change the status of a Quote. Since a third party

can interact with some of the values in it (the price of the product will be inserted by a provider later on in the provider site), we need a mechanism to keep track of the stage. Once we present a Quote (second stage), a provider will receive an email saying that he can access the portal and interact with it. This Kanban helps the sales users to visualize the status, and it is very intuitive.

Below that, we have the related products with the Quote (known as Quote line items). Here we can see the status discussed before. Depending on the status this values will be updated to the ones in the Query.



Quote
Reall - BCN - Abstergo

Quote Number 00000010	Opportunity Name Reall - Oportunidad Negocio BCN	Account Name Reall	Contact Name Jimmy Page	Grand Total 4.005,00 €
--------------------------	---	---------------------------------------	--	---------------------------

Accepted

Related Details

Quote Line Items (3) [Add Products](#) [Edit Products](#)

Product	Status	Quantity	Subtotal
Typhon Printer	Accepted	3.00	1.605,00 €
e-Book ink Pen	Accepted	30.00	1.650,00 €
Imperial Notebook	Denied	30.00	750.00 €

[View All](#)

Figure 23: Quote page. Source: made in Inetum.

To implement the new action, we decided to use the incorporated “flows” in Salesforce. A flow is a tool that can create, update, look up and delete Salesforce records. It works like a piece of code, with different access to different records and is able to make many of the things we could archive with code. If there is anything that we cannot do with a flow, we can code it in Apex and then call it in the Flow.

This time the flow will be activated when a user clicks on the button. A flow looks like the following figure; If we know a bit of coding, we will notice that the structure is pretty similar. We have to access the records, select the ones that are of our interest in the action and then process them. Once we are done, we have to update them with the new information.

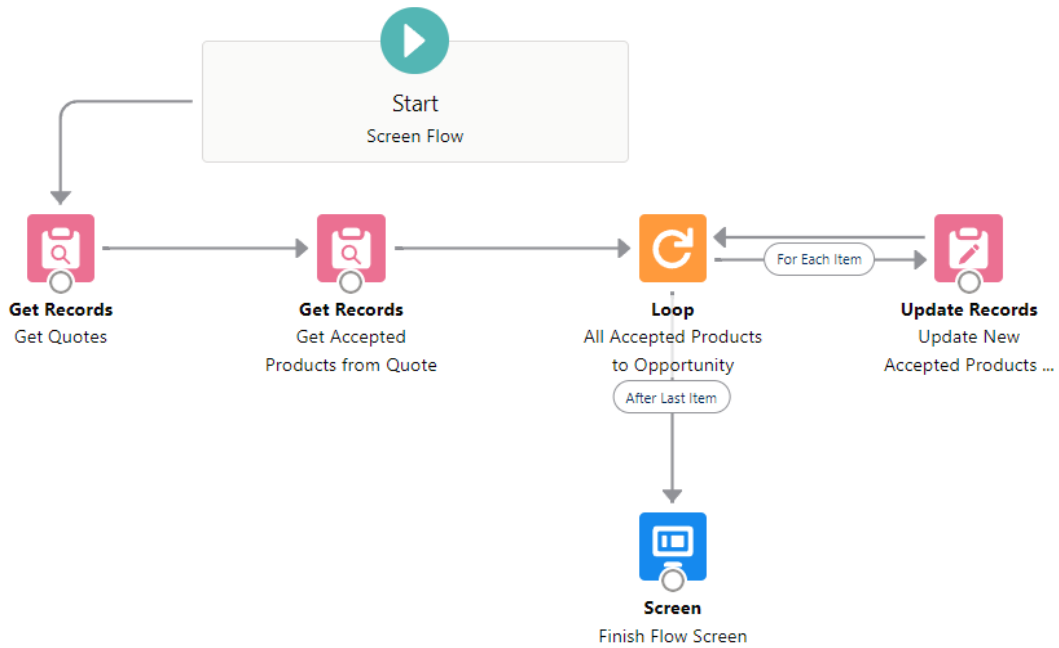


Figure 24: Flow example. Source: made in Inetum.

Everything is selectable. We do not need to know a programming language to get this far, as shown below, we only have to know the data model and how objects relate to each other to be able to “program” this action.

Edit Update Records
Update New Accepted Products Stats (Update_New_Accepted_Products_Stats)

*How to Find Records to Update and Set Their Values
 Use the IDs and all field values from a record or record collection
 Specify conditions to identify records, and set fields individually

Update Records of This Object Type
 *Object
 Opportunity Product

Filter Opportunity Product Records
 Condition Requirements to Update Records
 All Conditions Are Met (AND)

Field	Operator	Value
OpportunityId	Equals	Current Item from Loop All_Accept...
AND Product2Id	Equals	Current Item from Loop All_Accept...

+ Add Condition

Set Field Values for the Opportunity Product Records

Field	Value
UnitPrice	Current Item from Loop All_Accepted_Products_to_...

Cancel Done

Figure 25: Setting a flow. Source: made in Inetum.

To add this action (Flow), to a button we only have to relate the button with said action, looking like the figure below;

Quote Action
Select to Query

Action Detail		Edit	Delete
Label	Select to Query		
Standard Label Type			
Name	Select_to_Query		
Description	This button adds the subtotal of the accepted products into the opportunity of this quote.		
Flow	Synchronize quote stats to Query		

Figure 26: Linking a flow to a button. Source: made in Inetum.

We will finally have our new working button in our application. As we said, it is displayed on the top-right corner of our screen and is blended in the page in an aesthetically pleasing way. Our sales users will easily use it.

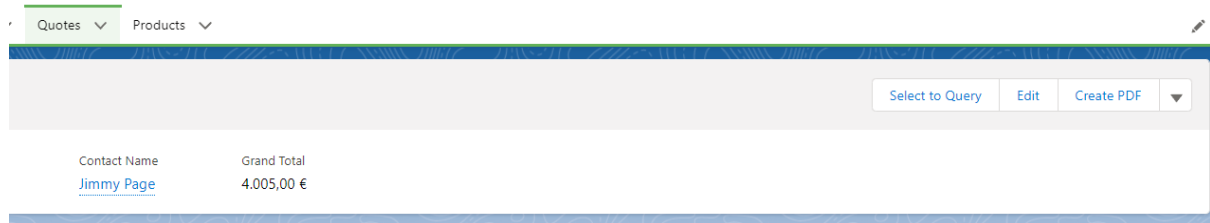


Figure 27: New “select to query” button. Source: made in Inetum.

As the last step, we have chosen to show a visual feedback for the user;

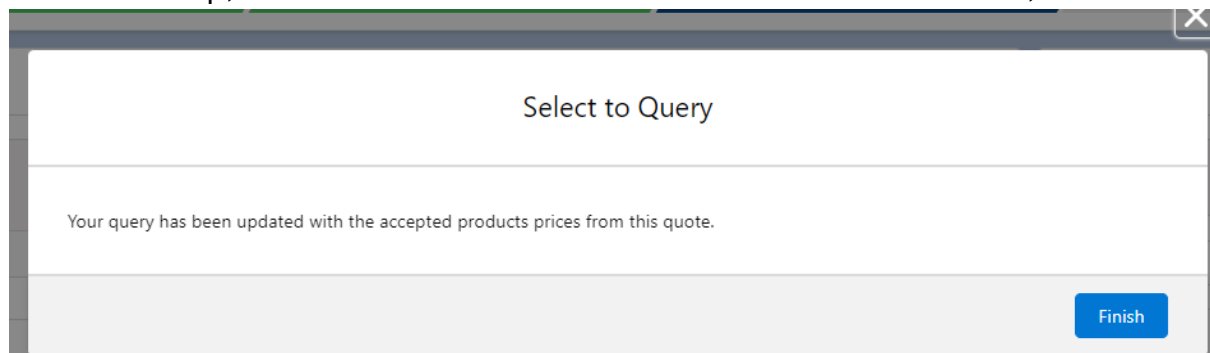


Figure 28: Feedback for “select to query” button. Source: made in Inetum.

That is all for the Quotes. We must now focus on the Queries. In a Query we can find different related objects, for instance, we will find Products and Quotes. They will be displayed like the ones in the following figures;

Figure 29: Query related objects. Source: made in Inetum.

Every one of these objects is linked to their corresponding information.

Figure 30: Query page. Source: made in Inetum.

Once we have managed all the Quotes and the sales process for a query is ending, we may choose the state to “closed”. “Closed win” if the client accepted the proposals or “closed lost” if didn’t. At this point we decided to include a Trigger made in Apex. The trigger function was to check whether the information in the query is correct or not. As we said before, a query may have the same product accepted in different quotes, and that should not be possible.

If there is any mistake, the user will be warned when clicking on the closed stage; for example, if the query has products that have not been accepted it will show the following message;

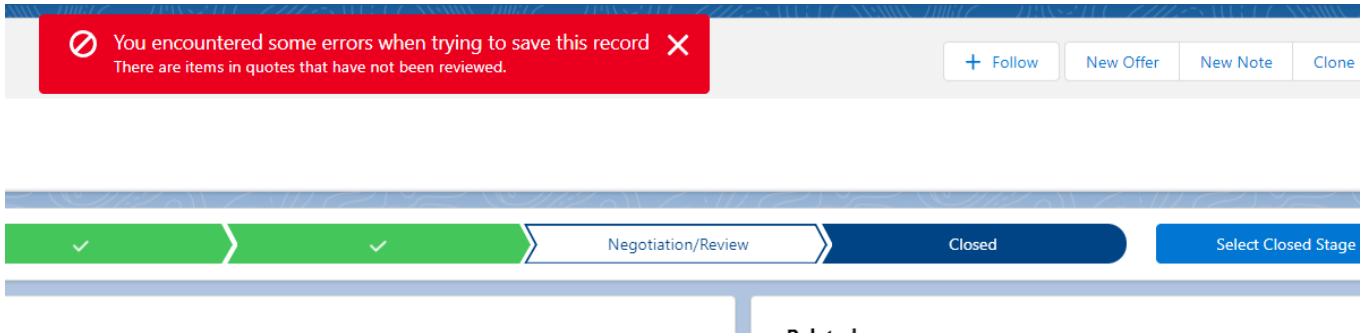


Figure 31: Trigger message. Source: made in Inetum.

The trigger is done in Apex, and it looks like the following Figure;

```
trigger CompletedOpportunity on Opportunity (after update) {
    Set<Id> oppSet = new Set<Id>();
    for(Opportunity opp: Trigger.New){
        if(opp.isWon) oppSet.add(opp.Id);
    }
    //Set for cheking duplicate accepted products.
    Set<Id> accProdSet = new Set<Id>();
    //Set for cheking accepted quotes.
    Integer hasQuotes = 0;
    //Get all quotelineitems in an specific Query/ opportunity
    for(Quote quo :[SELECT Name, Status, (SELECT Product2Id, Status__c, OpportunityLineItemId FROM QuoteLineItems) FROM Quote WHERE OpportunityId = :oppSet]) {

        for (QuoteLineItem quolineItem : quo.QuoteLineItems) {
            //all not-accepted products should be marked as Denied before wining an opportunity.
            if((quolineItem.Status__c != 'Accepted') && quolineItem.Status__c != 'Denied')
                opp.addError('There are items in quotes that have not been reviewed.');
```

Figure 32: Trigger building. Source: made in Inetum.

As we can see, Apex looks very similar to Java, a widely known programming language. There are some particularities when programming in a Salesforce Organization.

With the goal of maintaining an optimum performance among all their servers and increase availability to all customers, Salesforce made the choice of balancing transaction loads by imposing three kinds of limits;

- Concurrent API Request Limits
- API Timeout Limits
- Total API Request Allocations

When any of these limits in an organization is exceeded, an error is returned.

In our case this affects the way we make Queries. The system is called SOQL, and is very similar to SQL.

The key concept is that we must make the lesser quantity of queries that we can possibly accomplish. In the figure 32 we can see that we only made one Query and then accessed a data structure to process it. That way our organizational limits are kept to a minimum. For the making of this project these limits have not been any inconvenience. Perhaps in the future the project will grow and that's why we take them into consideration from the beginning.

13.3. Using the Application for Management and Marketing purposes

Salesforce incorporates different systems that allow management and marketing to perform accurate studies and predictions. In our case, we created Marketing users for this reason. Our client (Lyreco), did specify that in the future it would be interesting to make use of some of the tools that Salesforce has to offer for these purposes.

For Management, most of the work is made by Salesforce. We only have to keep working with our client and make their demands possible. For example if they want to create new dashboards to show specific items or relations, we have two routes of action;

Either we give some formation to someone in Lyreco to be able to make them themselves (the process is very intuitive and user friendly, so no advanced technical skills are required), or we create them on-demand.

Among the possibilities for management we can check the performance of different initiatives by comparing the results of different items across different timestamps or we can check whether some clients may be a “waste” of time and resources. We can also see which products are top sellers with a minimum effort on our side.

14. Provider portal for quotes

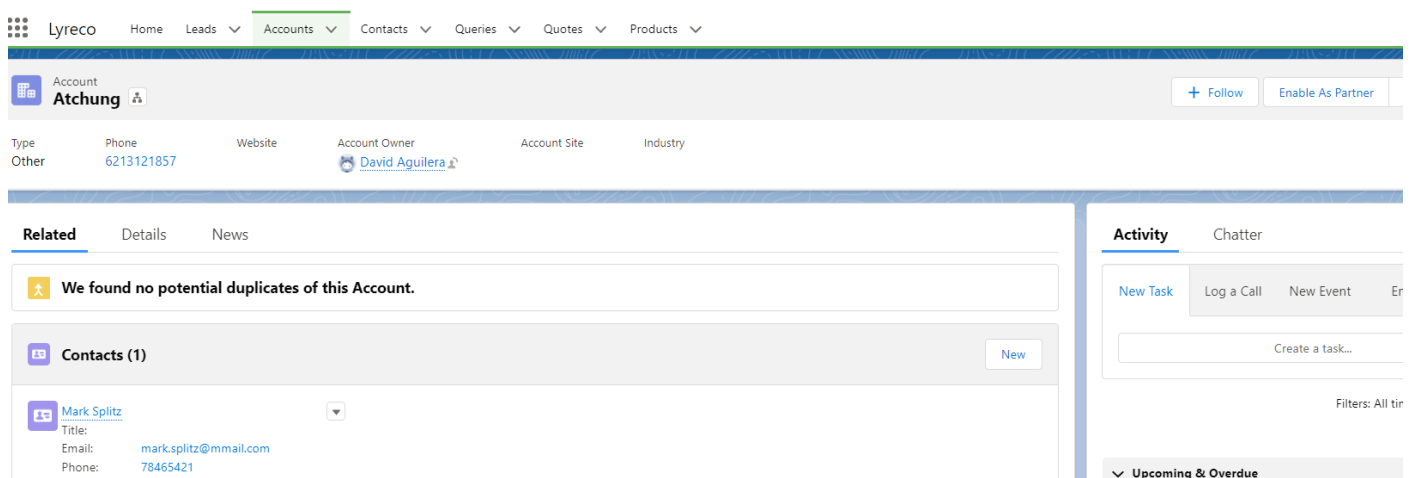
For the making of the portal for quotes that will be accessed mostly by our providers we can use Salesforce Sites. By using sites, we can create publicly accessible applications and websites that are directly connected with our Salesforce organization (our management app). These sites do not require a username or password to be accessed, but for the purposes of our application we will have to create a login page and make a mechanism to identify the users and show them personalized and relevant content.

All the data that is managed in the site, or the provider portal from now on, is integrated in Salesforce. That means that we won't have to worry about validating the data or making scheduled tasks to update the portal.

Is in this portal where the providers will be able to see their Quotes and where they will be able to change the prices of the individual items. We want the whole process to be as simple as possible, and we will explain now how we made it;

To begin with, a provider is stored as a Contact in Salesforce, meaning that in order for a provider to be able to login in the portal it will have to be registered in the Management Application as a Contact. Contacts are created by Sales Users. A contact is related to an Account, where in this case the account represents the company that will provide us with their products (the provider company). It makes sense that for a contact to be a provider and have access to the portal, he should be related with a company (Account), that we consider a provider company.

Once the contact and the account for a provider is created, we have to incorporate a mechanism for it to be able to connect to the portal.



The screenshot shows the Salesforce interface for an Account named 'Atchung'. The navigation bar includes 'Lyreco', 'Home', 'Leads', 'Accounts', 'Contacts', 'Queries', 'Quotes', and 'Products'. The account details section shows 'Type: Other', 'Phone: 6213121857', 'Account Owner: David Aguilera', and 'Industry: Other'. The 'Related' section displays a message: 'We found no potential duplicates of this Account.' Below this, there is a 'Contacts (1)' section with a 'New' button. A contact named 'Mark Splitz' is listed with details: Title, Email (mark.splitz@mmail.com), and Phone (78465421). The 'Activity' section on the right includes options for 'New Task', 'Log a Call', 'New Event', and 'Create a task...'. At the bottom, there is a section for 'Upcoming & Overdue'.

Figure 33: Enabling an Account as Provider. Source: made in Inetum.

First, we will have to enable the Account, by clicking on the button that we designed in the top-right corner of the screen in the Management App. Once we have enabled them as “partners”, we will have the same option in the user related to the Account that we want to have access to the portal.

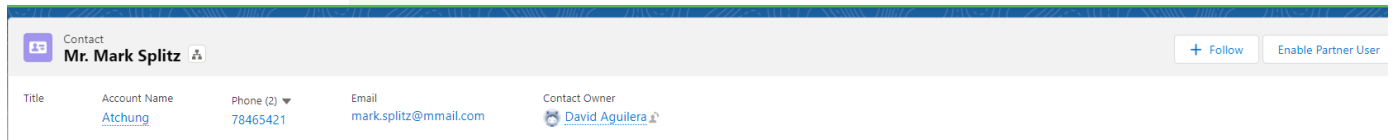


Figure 34: Enabling a User as a Provider. Source: made in Inetum.

Once we enable it as a “partner user”, an automatic email notification will be sent to him with a link to the provider portal.



Nombre de usuario

Contraseña

Iniciar sesión

Recordarme

[¿Olvidó la contraseña?](#)

[¿Es un empleado de Inetum? Iniciar sesión](#)

Figure 35: Portal Login Page. Source: made in Inetum.

The first time the user clicks in the portal via the link sent in the email, the password will be configured by the user. The username will be the mail of the contact and it

will be shown in the mail. In the case the password is lost, an email will be sent to the user to their email address to reset the password.

Once logged in the portal, the user will have access to a main page with information about Lyreco, special offers and other marketing items as well as to a page with all his quotes and another page with some commonly asked questions.

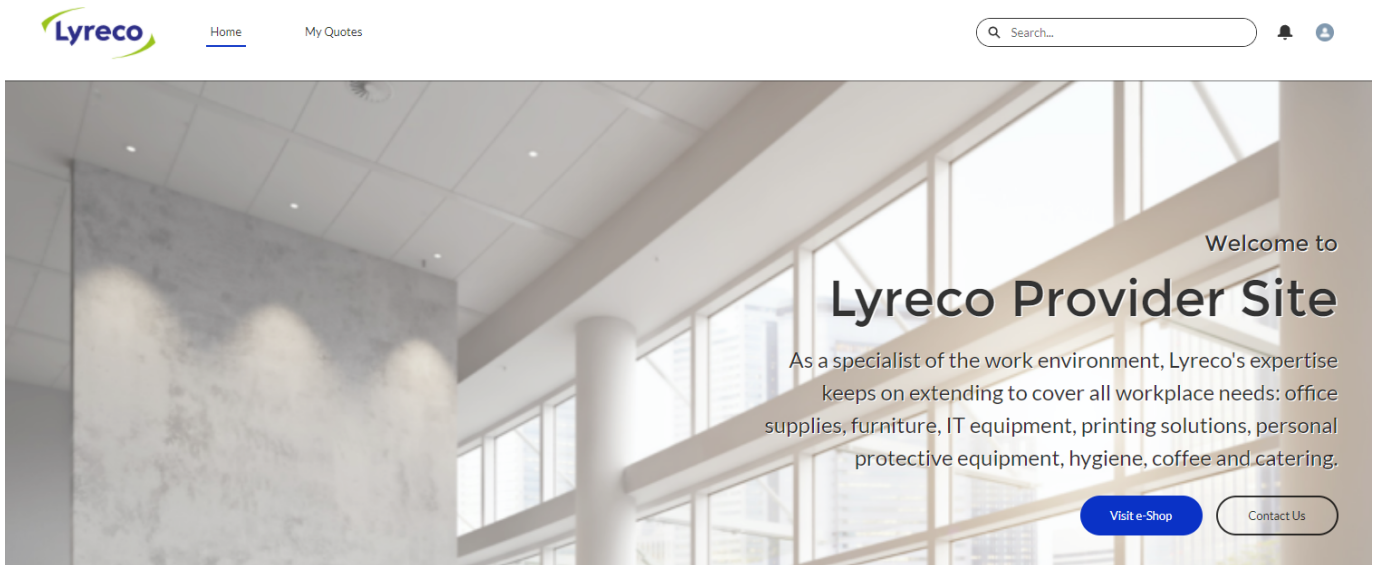


Figure 36: Portal Home Page. Source: made in Inetum.

In the home page we will also find some dashboards with some basic information about the relationship between our companies and how our deals are going. As an example we have included the following figure;

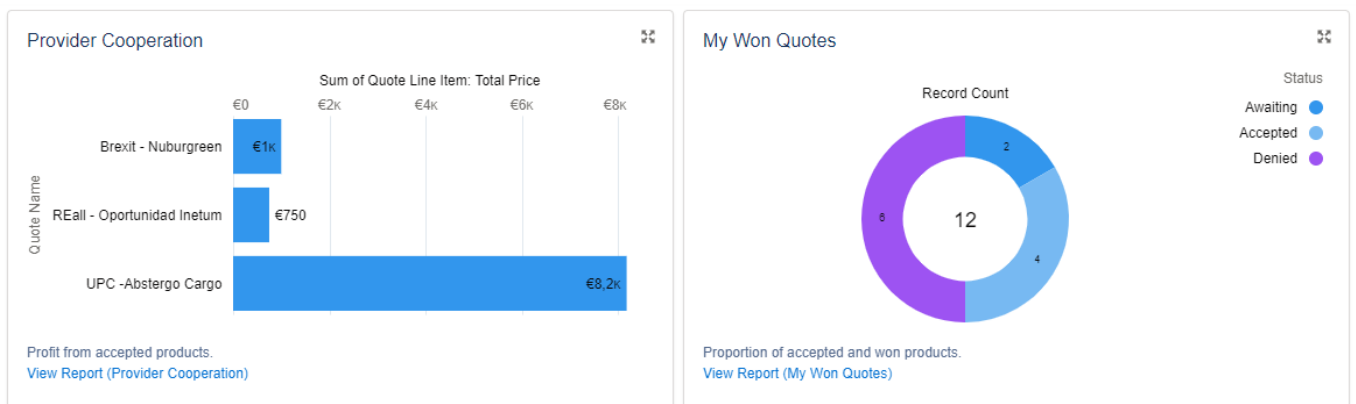


Figure 37: Portal Home Page Dashboards. Source: made in Inetum.

In the “my quotes” page, we will show information about all our quotes, with an intuitive U.I where the user will be able to navigate through all their quotes, active or not. This menu is simple, but it shows all the relevant information so that the providers does not feel that it is tedious or that he is losing his time.

<input type="checkbox"/>	Quote Name	Opportunity Name	Expiration Date	Subtotal	Total Price	Status
<input type="checkbox"/>	Brexit - Nuburgreen	Brexit.SA-		3.500,00 €	3.500,00 €	Accepted
<input type="checkbox"/>	example 1	Inetum- Aurelie 05082021		6.750,00 €	6.750,00 €	Denied
<input type="checkbox"/>	Inetum- Aurelie - Pilot	Inetum- Aurelie 05082021	19/8/2021	6.750,00 €	6.750,00 €	Accepted

Figure 38: Portal Quotes. Source: made in Inetum.

Every one of these quotes can be clicked on, and in the related items we will be able to modify the sales price, the status and the quantity of a product that we are willing to sell.

Quote
Inetum- Aurelie - Pilot

Quote Number	Opportunity Name	Account Name	Contact Name	Grand Total
00000007	<u>Inetum- Aurelie 05082021</u>	<u>Inetum</u>	<u>Juoline Oulan</u>	6.750,00 €

DETAILS **RELATED**

Quote Line Items (3)			
Product	Status	Quantity	Subtotal
e-Book Ink Pen	Denied	50,00	3.000,00 €
Imperial Notebook	Denied	50,00	1.250,00 €
Separator 3000	Denied	10,00	2.500,00 €

View All

Figure 39: Portal Quotes products. Source: made in Inetum.

List Price
60,00 €

* Sales Price

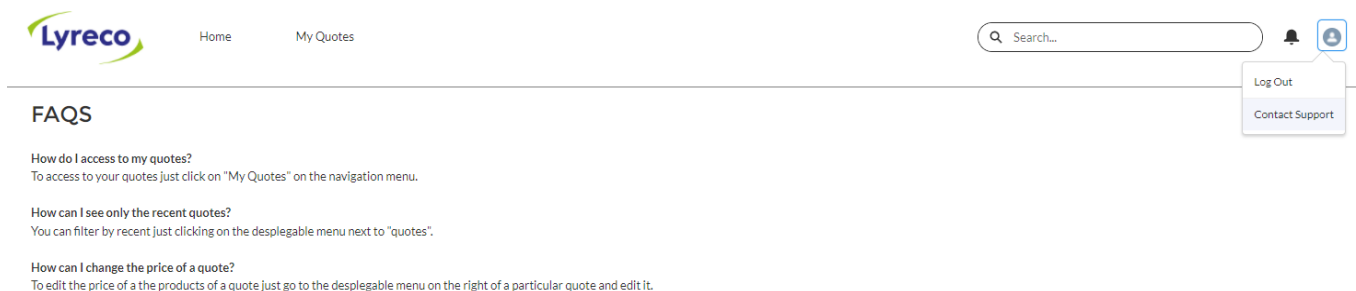
* Quantity

* Status

Figure 40: Modifying the products in the portal. Source: made in Inetum.

To end with this section, we left the FAQs for the end. A provider user can get support by multiple means, or by clicking the contact support button that appears on the user icon in the top right corner of the page or by contacting our sales users.

We added some possibly common asked questions to the portal so that we limit the time lost navigating the different menus.



Lyreco Home My Quotes Search... Log Out Contact Support

FAQS

How do I access to my quotes?
To access to your quotes just click on "My Quotes" on the navigation menu.

How can I see only the recent quotes?
You can filter by recent just clicking on the desplegable menu next to "quotes".

How can I change the price of a quote?
To edit the price of a the products of a quote just go to the desplegable menu on the right of a particular quote and edit it.

Figure 41: FAQs. Source: made in Inetum.

15. Testing and Debugging

Testing is a task of great importance when developing software since we must assure a degree of confidence in our application. There are different types of testing, and each one of them has a different purpose.

First off we have to do an “Acceptance Testing”, so we make sure that the whole system works as intended. As we are using Salesforce, we do not need to do a performance test or a stress test, since the system is already bound by some well-known limits that regulate those aspects. We must trust our environment and that Salesforce will carry on with their SLAs.

When creating functionalities using apex code, we must create Unit tests. A unit is the smallest testable component in an application. Each one of the classes will be covered with a unit test. The minimum coverage that we want from a class is 85%.

To test and debug we have been trying all the cases of action that are possible in our application and provider portal. We simulate business operations and all the possible decisions that can be made in our application. In the time that we were creating new actions and buttons, we have been testing its correct work and expected results. Luckily for us, most of the items used in the making of this project have been standard objects, so there was little to no need to test its reliability.

16. Data Insertion

Data insertion is the last step of our project. Once we have a fully operational application we would normally just start using it, but since our client already had a working system and therefore had important data generated from their previous business operations, we have to insert that data into our new system.

We must differentiate between different types of data. We have users, that are the employees of our client company (the ones we defined previously as sales users, marketing users or administrators). We have Contacts (either companies or individuals) and we have old transactions, that are the queries and quotes generated from them. To insert all this information we must first do a work of standardisation and follow an order. Some fields are required, and these required fields depend on the object. For example, Users require the following list of fields in order to be inserted into the Salesforce's Database:

- Alias
- Username
- Email
- First Name
- Last Name
- Locale (LOCALESIDKEY)
- Language (LANGUAGELOCALEKEY)
- Email Encoding (EMAILENCODINGKEY)
- Time Zone (TIMEZONESIDKEY)
- Currency (CURRENCYISOCODE)
- ProfileId (not profile name but rather the 15-character ID from the URL of the profile in the UI).

As we can already guess just by looking at some of the fields, probably the client's database does not contain some of them. Extra and unexpected old fields are not a big problem, since we can easily add them into our object as additional information.

Required fields are common in standard salesforce objects. We have to keep in mind that those objects have many pre-build in functionalities and the operability of most of them depends on the fields of an object.

Another important thing to keep in mind is load order. When we insert information we have to keep in mind our data model. For instance, a query requires an owner user, therefore we should first insert the users, the contacts and then the queries. Elseway, when we create a query using data loader an error will emerge, saying that the user does not exist and the query will not be added. Thankfully the tool that we are going to use has some features that will be very handy to our work.

16.1. Data Loader

Data loader is a tool used to import, export and delete data from an organization. The feature that will be of most use in our particular case is the intelligent data mapping. With said feature, we are able to match the old field names and types to the new field requirements, and therefore be able to use the existing database. Moreover it has an option to detect duplicate entries, removing all the manual and tedious work of having to check them one by one.

There is one catch, and that is that we can only insert 5 million records a day. We can use scheduled imports in that case, so there is no problem with that either.

As we can imagine, inserting data from all databases can be a big deal, but once we have standardised the data and tweaked the data loader tool, the process is somehow easy.

17. Conclusion

As a way to conclude, I will make a little overview and a little self-evaluation of the project once completed as well as the methodology used.

From the beginning we had clear objectives with a well planned route to follow, after all I had the opportunity to work in a company that specializes in this kind of operations, and they had the experience and resources to make that possible. I had the responsibility of making a portal for a client using Salesforce, which is the tool of choice in these cases because it really does fit like a glove and makes a process that in normal circumstances would take much longer to accomplish and almost impossible for a junior developer to make almost on its own feasible. In that regard, I must say that Salesforce did a really good job making this service and I expect for it to be more and more used in the near future.

Concerning the work methodology and philosophy of the company I have nothing to say but to praise the company. The planning does affect a lot in how a worker perceives the work and how to manage the stress, and using an agile approach helps to avoid being stuck in a certain phase or technical circumstance.

I think I did a good job in the making of the project. I participated as much as I could and provided as many ideas as my initial lack of experience allowed me. We must take into consideration that my work was very enclosed, meaning that although some things could have been made differently most of the project did not leave much room for imagination or improvisation. I understand this approach, since it makes all the projects very similar and easy to understand and maintain by different individuals, but I consider myself a highly creative person and therefore I do not see it in my future. Furthermore, in my opinion, the learning curve in Salesforce is not very pronounced. If you know how to program a bit beforehand that part is mostly done and you only have to understand how the environment works and to manage some tricky peculiar parts.



To end with the conclusion I would like to add that cloud tools are unquestionably dominating the market and they are steadily growing. Salesforce and cloud computing is an impressive market and a very engaging option for those interested in software development.

18. Bibliography

- Barcelona salaries and remuneration. (2021). Retrieved 10 October 2021, from <https://www.glassdoor.es/>
- Salesforce Climate Action Plan. (2021). Retrieved 10 October 2021, from https://www.salesforce.com/content/dam/web/en_us/www/assets/pdf/reports/salesforce-climate-action-plan-2021.pdf
- Inicio | Recursos de licencias de Microsoft. (2021). Retrieved 13 October 2021, from <https://www.microsoft.com/es-es/licensing/default>
- Positive digital flow – INETUM | Inetum Polska. (2021). Retrieved 25 September 2021, from <https://inetum.pl/en/2021/01/14/inetum-leader-in-the-field-of-digital-transformation/>
- What is CRM?. (2021). Retrieved 25 September 2021, from <https://www.salesforce.com/crm/what-is-crm/>
- What Is CRM? Customer Relationship Management System Meaning Defined. (2021). Retrieved 25 September 2021, from <https://crm.org/crmland/what-is-a-crm>
- What is Salesforce?. (2021). Retrieved 25 September 2021, from <https://www.salesforce.com/products/what-is-salesforce/>
- Salesforce - Wikipedia. (2021). Retrieved 25 September 2021, from <https://en.wikipedia.org/wiki/Salesforce>
- Center, 2. (2021). Gartner's 2021 Magic Quadrant for the CRM Customer Engagement Center. Retrieved 25 September 2021, from <https://www.zendesk.com/blog/gartner-magic-quadrant-crm/>
- Cloud-Native or Lift-and-Shift? | Amazon Web Services. (2021). Retrieved 25 September 2021, from <https://aws.amazon.com/es/blogs/enterprise-strategy/cloud-native-or-lift-and-shift/>

- Discover Salesforce Mobile & Close Deals Anywhere. (2021). Retrieved 26 November 2021, from <https://www.salesforce.com/eu/solutions/mobile/app/getting-started/?d=70130000000IxSo>
- Salesforce Sharing Model Visualized - Salesforce Dev/Admin Topics - Salesforce tutorials - Apex, Visualforce. (2021). Retrieved 26 November 2021, from <http://www.salesforcegeneral.com/salesforce-articles/salesforce-sharing-model-visualized.html>
- Help And Training Community. Quotes. (2021). Retrieved 26 November 2021, from https://help.salesforce.com/s/articleView?id=sf.quotes_overview.htm&type=5
- Salesforce Related Objects | Workato Docs. (2021). Retrieved 26 November 2021, from <https://docs.workato.com/connectors/salesforce/related-objects.html#related-objects-field>
- Help And Training Community. Auto-Response rules. (2021). Retrieved 26 November 2021, from https://help.salesforce.com/s/articleView?id=sf.creating_auto-response_rules.htm&type=5
- Salesforce Developers. (2021). Retrieved 22 December 2021, from https://developer.salesforce.com/docs/atlas.en-us.salesforce_app_limits_cheatsheet.meta/salesforce_app_limits_cheatsheet/salesforce_app_limits_platform_api.htm#
- Salesforce Flow. (2021). Retrieved 22 December 2021, from https://trailhead.salesforce.com/en/content/learn/modules/business_process_automation/flow
- Help And Training Community. (2021). Retrieved 23 December 2021, from https://help.salesforce.com/s/articleView?id=sf.sites_overview.htm&type=5
- Salesforce Developers. (2021). Retrieved 23 December 2021, from https://developer.salesforce.com/docs/atlas.en-us.dataLoader.meta/dataLoader/data_loader.htm



- Salesforce Developers. (2021). Retrieved 23 December 2021, from https://developer.salesforce.com/docs/atlas.en-us.dataLoader.meta/dataLoader/data_loader.htm
- Home. (2021). Retrieved 23 December 2021, from <https://dataloader.io/>
- Salesforce Data Center Locations - USA, UK, Germany, Japan - Europe, Asia, Australia, North America. (2021). Retrieved 27 December 2021, from <https://datacenterlocations.com/salesforce/>
- *What is SaaS (Software as a Service)?*. Salesforce.com. (2022). Retrieved 3 January 2022, from <https://www.salesforce.com/saas/>
- Dureja, A. (2022). *Salesforce Governor Limits - Salesforce Blog* salesforce. Salesforce Blog. Retrieved 3 January 2022, from <https://www.sfdcpoint.com/salesforce/salesforce-governor-limits/>.
- *Salesforce Developers*. Developer.salesforce.com. (2022). Retrieved 3 January 2022, from https://developer.salesforce.com/docs/atlas.en-us.soql_sosl.meta/soql_sosl/sforce_api_calls_soql.htm