BACHELOR'S DEGREE THESIS

Degree in Data Science and Engineering

Title:Development of value metrics, dashboards and analysis for Real Estate portfolio management context.

Author: Pol Ruiz Farré

Advisor: Ramon Bragos Bardia

Department: Departament d'Enginyeria Electrònica

Month and year: juny 2022

Facultat d'Informàtica de Barcelona
Facultat de Matemàtiques i Estadística
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

Contents

TFG: Data analytics software for property managers

Abstract - English

Property management firms are traditional companies. In recent years, these companies have seen their profitability decrease due to inefficiencies. This has created an opportunity for those tools and software that allow them to optimize processes and be more competitive.
Business data analysis and visualization can be a pillar for companies that need to make decisions based on data to optimize their business.
This project is a feasibility study and development of a minimum viable data analysis software product for property management companies.

Abstract - Spanish

Las empresas de gestión de inmuebles son empresas tradicionales. En los últimos años estas empresas han visto sus rentabilidades disminuir a causa de ineficiencias y esto ha generado una oportunidad para aquellas herramientas y software que les permitan optimizar los procesos y ser más competitivos.
El análisis de datos de negocio y su visualización pueden ser un pilar para cualquiera de estas empresas que necesiten tomar decisiones en base a datos para optimizar su negocio.
Este proyecto es un estudio de viabilidad y desarrollo de un mínimo producto viable de un software de análisis de datos para las empresas de gestión de inmuebles.

Abstract - Catalan

Les empreses de gestió d'immobles són empreses tradicionals. En els últims anys aquestes empreses han vist les seves rendibilitats disminuir a causa de ineficiències i això ha generat una oportunitat per aquelles eines i softwares que els permetin optimitzar processos i ser més competitius.
L'anàlisis de dades de negoci i la seva visualització pot ser un pilar per qualsevol d'aquestes empreses que necessitin prendre decisions en base a dades per tal d'optimitzar el seu negoci.
Aquest projecte és un estudi de viabilitat i desenvolupament d'un mínim producte viable d'un software d'anàlisis de dades per les empreses de gestió d'immobles.

1. Introduction

The project idea is the outcome of a pivoted project started one year ago with my cofounder Guillem Rovira, another UPC student. The initial project aimed to help property managers communicate with tenants through an app as we realized that there was a clear pain in the communication between the two stakeholders and where there is a pain, there is an opportunity to solve it.

My co-founder and I understood that the real value of the project was on collecting the right data to optimize, simplify and automate most of the processes and avoid synchronous communication.

We realized that the go-to-market of an app that involved two types of users (tenants and property managers) was hard and we could simply provide value to the manager by analyzing his own ERP data without the need of the tenant and the complexity that this adds.

As I was in charge of the technical aspect of the project, I defined the type of solution we could get by analyzing only the data that the property managers had in their ERP and we both agreed to pivot the project into a “business to business (B2B)” data analytics software for property management firms.

Without the complexity of building two different go-to-markets and the retention of the most valuable part of the project (the real estate data), we laid the foundation for a much more solid project with an international aim.

This TFG is a market research and the MVP development of this new version of the project. I am in charge of doing this market research and the technical development. My co-founder (as we are both parts of Urbix SL, the proprietary society of the project) is in charge of talking to potential investors, building relations with different stakeholders, executing marketing and sales strategies, and searching for funding and public subsidies. We also had the support and collaboration of Empren UPC during the development of this project not only with knowledge support but also AWS 5000$ credit that we had used to test and deploy all the software and infrastructure of Urbix.

2. Objectives

The main objective is to study the viability of the project in an entrepreneurial way and build an MVP as a proof of concept to test with real-world users.

2.1. Business plan goals

1. Talk to potential users to get the requirements and value opportunities that the software could meet.
2. Define the functionalities that can fulfill the minimum viable product.
3. Validate the minimum viable product by getting feedback from real users.
4. Iterate the MVP adding (or modifying) the functionalities based on user feedback.
5. Make market research to evaluate the opportunities and potential clients with an international view. (TAM, type of clients, KPIs in different countries)
6. Develop the business model (monetization system, recurrence, target, pricing plans, premium features, pricing model)
7. Develop the go-to-market strategy
8. Explore the potential for growth of the project to study the viability and needs of a funding round.
9. Develop an inversion plan for growing the team, product, and going to market.

2.2. Technical goals

1. Development of frontend dashboards that satisfy business analytics requirements creating value for the potential users.
2. Development of integrations with some ERPs of the market to automate the data pipeline between the users and the platform.
3. Development of backend involving automated data retrieval and ETL
4. Development of backend user authentication system
5. Development of database architecture
6. Development of backend DB data request and data processing for fitting the dashboard graphs.

3. Business plan

3.1. Problem identification

The tools for data visualization and Business Intelligence (BI) don't reach all the potential the data hides. This type of tool tries to solve data visualization regardless of the business industry of the user and that's why they build their software that tries to adapt visualizations and charts to different types of data, at expense of complexity, personalization, and depth of data analysis.
In the real estate sector, the data has an essential importance for business development. Not even the big real estate agencies can afford to have a team of software developers dedicated to building software for analyzing this data deeply. In the majority of cases, the companies hire software consulting agencies to build simple BI dashboards without being able to exploit the data at the top level with deep data analysis due to a lack of know-how, resources, and data volume.

3.2. Solution

Unlike other tools in the market, we aim to build a solution for data exploration and analysis for the real estate industry. We offer a tool with different dashboards, KPIs, and alerts that covers completely all the needs of data analysis for the property management business.
Breaking the trend of the software tools for data visualization in the market, we base our tool on the real estate industry, at the expense of adaptability (to other types of data and industries) in return for a tool with much deeper data analysis and easier to use.
We implement dashboards with KPIs and charts from the executive team to the managers or their clients.
We want our software to substitute the other generic BI software in the market (power BI, google data studio, Tableau) that are used by the real estate property managers without technical expertise, hard integrations, or long onboarding processes.

3.3. Business model

Our business model is a software as a service (Saas) B2B model. This implies that we are going to charge a subscription fee to the companies, as our software is meant to be used by companies, the subscription will be an annual fee.
As bigger companies with a larger number of assets take more value from our software, we want to correlate the value we bring and use of our software with the amount we charge to our customers by creating a subscription model based on the number of assets our software will analyze. This type of pricing model also helps us automatically upsell the revenue of each user. If we expect our users to better manage their companies and grow their business, we are going to charge them more year over year if they keep growing.
Some Saas B2B products are used to charge based on the number of users (of the same companies) that use the product. We think that this type of billing will discourage the adoption of our software in all the departments of the companies (not only the executive team) being contrary to our objective of making a product necessary through all the company employees.
We want to provide a free unlimited version with some basic features to achieve some objectives:

- As our customer profile is defined by traditional companies with a lack of digitalization, we expect them to have no knowledge and experience in data analysis and visualization. This makes them reluctant to trust that with data analysis and some dashboards, they can be more efficient, make better decisions and impact directly on the revenue they get from every client. With a basic free version of our software, they can check easily that they can get value from data analysis from the minute one without paying for it. Once they get value, we expect them to migrate to the premium version.
- We can not lose the data. If we are not able to convert a potential client to a paying customer, either because our software is expensive or because we don’t communicate well the value we can offer, we still want their data. Our product and whole project have an intrinsic value thanks to the data we have, having a free version will allow us to get all the data from these potential clients that don’t convert to paying customers, and without this free version we would lose.
- Low the customer acquisition cost. As the users can try a free version before paying, they are more likely to try our product and get the value before paying and converting more potential customers to paying users.
- The potential customers that are already using a data analytics/visualization tool can try it for free without paying and then decide to change their main data analysis tool. A free versión will easily convince this type of user to try our software although they already are paying for another because they could try it for free.

- A free version of our software that already provides a lot of value can deter other competitors and could raise the market entry barrier for new competitors as we will be offering for free what they planned to offer in a paid version.

Apart from the main Saas model, a company that manages (stores and analyzes) that much data from the real estate industry has intrinsic value on its own. Real-time data like geolocalized unpaid rents or rotation of rent assets is a valued data not only by property managers but also by investment funds, property owners, banks, etc.
We plan to grow the business model to another type of customer with the same value proposition of providing value by exploiting our data.

When talking about the pricing, as said before, we expect to charge our customers by the number of assets they manage and we expect to be a high-ticket software as our tools impact directly on the revenue and growth of the companies that use our product and also knowing that the revenue for each asset yearly is between 1000€ and 5000€. Following this premise and after thinking and getting feedback from key stakeholders of the market, we have stipulated the following pricing model:

Number of assets	**Price per asset per year (€/asset)**
0-499	15
500-999	12
1000-1499	9
1500-1999	8
2000-2499	7
2500-3000	6
+3000	Contact sales for a personalized offer

3.4. Market research

Transactions in the Spanish market (2019):

- 800.000 purchases and rental operations on the main dwelling (570.000 purchases, 230.000 rentals)
- 37 corporate operations worth +1500M€
- +25% Spanish population lives on a rental asset
- +12500M€ on investment volume
- Office building operations worth +4600M€
- +2000M€ on investments transactions for assets intended for rent
- Real estate commercial market value: +500B€

The real estate companies in Spain:

- There are more than 55000 real estate companies
- The companies in this industry got +29.000M€ of revenue

“Socimis” and other big residential asset owners in Spain:
Even though there are big players in the real estate market, they only represent 4.2% of the rental market in Spain. Other real estate companies like Merlin Properties or Colonial are listed on the stock market with a market cap of +4180M and +4300M€.

Servicers and property management firms in Spain:
The assets from the big real estate funds and “socimis” are managed by six big players that manage real estate assets with a value of +220.000M€. Nevertheless, this represents less than 4.2% of the market share:

- Altimira: Santander. +315M€ revenue.
- Anticipa Real Estate: Blackstone.
- Servihabitat: Lone Star. Previously Caixabank.

Franchise, real estate agencies, and digital companies in Spain:
The market is highly fragmented, there are SMEs with less than 30M€ of annual revenue but the market is dominated by franchises:

- Tecnocasa: +470 offices.
- Engels & Völkers: +70 offices.
- DonPiso: +120 offices.
- Best House: +153 offices.

Real estate SMEs in Spain:
The real estate Spanish market is fragmented. More than 95% of the revenue generated in this market comes from SMEs.
These companies are usually familiar companies with annual revenue between 500K€ and 10M€.

In general, they are late adopters of technology and digitalization. While the companies from other sectors started using management software and ERPs more than 15-20 years ago, the real estate companies that use an ERP or management software have started using it in the last 5-10 years. Even some of them still are using Excel or handwritten books to manage their business.
SMEs are not used to using data insights to run their business. As it is a highly operational business, not using data to optimize the operations, some inefficiencies are not only appreciable but solvable using data analytics.
All this being said, most of them are conscious of their lack of digitalization and think that they have to evolve to not die vs other competitors or market changes. This opens an opportunity for technology and digitization software that helps them become more competitive in the actual market.
Nowadays some growing startups are breaking into the market using online strategy and getting astonishing results not only on revenue but also on growing ratio year over year. Some examples of these startups are Housfy with +15M€ revenue in 2021 and willing to end 2022 with +43M€ of revenue. Another example is Clickalia, the company has closed 450M€ of funding to keep growing on the international market.

United Kingdom real estate market 2019:

- 1180M transactions
- 34.8% population lives on rental assets
- +63B€ investment volume
- 1270B€ commercial assets market value

France real estate market 2019:

- 1050M residential real estate transactions
- 38.8% population lives on rental assets
- +42B€ investment volume
- 971B€ commercial assets market value

Italian real estate market 2019:

- 604M residential real estate transactions
- 27.6% population lives on rental assets
- +12B€ investment volume
- 729B€ commercial assets market value

German real estate market 2019:

- 17,5B€ residential real estate transaction volume
- 48.9% population lives on rental assets
- 1371B€ commercial assets market value
- 158,4B€ revenue from the companies in the real estate industry

3.5. Go-to-market

To get visibility of the product and attract qualified leads we will follow a communication and marketing strategy to evangelize about the potential of business improvements that a tool like ours could bring to them: As the market, in general, is not used to analyze their data and using these type of tools, we consider that we find ourselves in a “Blue ocean” market. Where our main goal is not to explain which one of our features is better than the competitor's but to teach the customers and the whole market about this new type of tools (data analytics) that can help them in their business and how easily they can use these tools without hard, long and expensive integrations. The main objective of our communication and content strategy will be to create demand in the market for our product.

For achieving this objective we will plan different action fronts:

- Partnerships with hubs: We will partner with some key hubs in the real estate and proptech-startup sector. For example, we are part of Innomads an innovation hub for real estate startups funded by a collective of property managers from Barcelona. Another example could be the CSIM real estate cluster founded and partnered by some of the bigger players in the property management industry.
- Collaborations with API Schools: In Spain, there are important institutions of groupings of APIs (property real estate agents) that promote tools, courses, and lectures useful for them. We plan to partner with them to help us promote our product and its benefits through lectures, courses, and sponsorship of some of their events.
- Participation in Proptech events: Proptech startups have captured the highest volume of venture capital investments in Spain 2021, 637.7M. As the proptech industry grows and attracts funding, more and more events take place in Spain with the participation of big players and attracting a lot of potential investors and customers. Being in this type of event as a visitor, expositor, sponsor or lecturer is a big opportunity to talk to potential customers about the problems we solve.
- Partner with ERPs: The property managers usually use software to better manage their companies. These software companies not only are the ones we have to integrate with to import the data recurrently but are also a good communication channel with our potential customers. The software is open to partnerships with revenue share for each customer they bring to our business and also are likely to recommend our software as it helps their customers.

- Blog contents: We plan to create different content related to the real estate industry and the use of data. From tips to better manage their company to strategies for getting new customers efficiently. All this content will be focused on providing value to our potential customers. We will also post use cases and success stories about our clients like the improvements of key metrics like revenue or margin improvement since they started using our software. This type of content will help potential customers to understand how we bring value directly.

- Social media and email marketing: We have identified that our potential customers are actively using social media like Linkedin to share content and keep updated with the news in the real estate industry. We plan to share our blog contents through Linkedin and our emailing lists and also to interact with different players through Linkedin to make us noticed inside the industry.
- Press: Our profile of customers still reads every day the typical newspapers like La Vanguardia, Expansion, or el Pais. They still interpret being in this type of press as an authority and professionality signal which conveys confidence to them. For that reason, we will feature some news and paid press to improve our authority in our industry.

Taking into account that our software will have an annual high ticket subscription for B2B (SMEs and corporates), the sales team will be the engine of the commercialization of our tool. In the first stages of our company, we don't expect our sales process to be touchless with the customers onboarding themselves into our platform. We expect the process to involve some type of demo meeting where someone from our team can explain to them the main functionalities and help them with the first contact with our product.
Although we want to provide value to all the company sizes (SMEs and corporates) we are conscious that SMEs products and sales processes are different from the corporates' ones. That's why in the first stage we want to focus on the SMEs. Focusing on small and medium companies, we will get financial sustainability as 100 paying customers implies more predictable revenues than one big client. SMEs also require a less mature product with fewer features and we can get more data once we start recurrently onboarding new customers.
The plan is to finance our product development with small and medium customers and once the product is mature enough to provide high value to corporates in a sustained way, we will start to extend our product to corporates by creating a dedicated sales teams with longer selling processes and even a software support team to onboard and assist these type of big players with the integration of our software to their companies.

3.6. Potential for growth

Taking into account the market size and the expected revenue per client we have approximate the total addressable market (TAM) in Europe.
In Spain, taking into account the number of operations, rental assets, and the number of companies and their revenues, we have come to a TAM of 500M€. If we extrapolate this number to the principal countries of the EU, we can expect a TAM of 1500M€ in the United Kingdom, 1500M€ in Germany, 1000M€ in France, 600M€ in Italy, and more than 2000M€ in the rest of the countries in Europe. All this brings this project with a total addressable market in Europe of more than 7000M€, more than sufficient to aim this project into the international market.
Although this software has a potential market of millions of dollars, we don't keep the vision of this project bound to a data analytics software for property managers. Since the beginning, we wanted to offer value by analyzing real estate data. We wanted the real estate data to be the foundations of our project and the different layers of data analysis and dashboards as the different floors of a much bigger building.
So, taking into account this vision, a data analytics software for property managers is only the tip of the iceberg. The data, they provide to us through the ERP is the most valuable data in the market. As the first-party data is coming in real-time just when it happens, we can know before than anyone when a new rent has been signed, which price, and when. We can know when someone is missing the payment of his rent and if this is an isolated case. These insights analyzed correctly and following anonymized practices to be compliant with GDPR can be useful to a much wider segment of companies implying a much much bigger TAM not only in Europe but worldwide.
We can provide value to professional service providers like consulting firms, Banks, real estate funds like Blackstone, and even governmental institutions. All this value is provided through different software solutions involving dashboards to complex machine learning for predictive analytics of the real estate market.

4. Minimum Viable Product

4.1. Product features

The product consists of three dashboards with different KPIs and charts from their ERP data analyzed.
The dashboards are divided by the type of data analyzed:

- Community.
- Rental
- Office (operations)

From the user profile (and the feedback received from the market) we found that our potential users are not familiar with data analytics, dashboards, and business intelligence. That's why we wanted to build an easy-to-use software with the minimum clicks (and filters) needed to look at the main metrics, KPIs and charts not only during the deploying stage but also in the further use of the product. Most of the dashboards are static with some replicated metrics to improve the user experience taking into account the profile of the user who is going to use our software.
The product should contain general features like authentication methods for login different users as well as profile data visualization and modification (password or email change).

4.1.1 Community Dashboard:

This dashboard consists of the data analyzed from the buildings and communities of the owner's assets that the user manages.
The aim of this dashboard is to be used not only by the owner or executive team but by the different administrators that manage the portfolio of communities.

KPIs:

- Number of communities
- Variation of the number of communities vs the year before
- Revenue this year
- Variation of the revenue vs the same period the year before
- Revenue from the recurrent fees
- Variation of the revenue from fees vs the same period the year before
- Revenue from industrial bonification
- Variation of the revenue from bonification vs the same period the year before
- Revenue this month
- Variation of the revenue vs the same period the month before

List:

- Community
- Administrator
- Current balance

Charts:

- Line chart with communities expenses per month
- Bar chart with the type of expenses vs the year before
- Pie chart with the type of expenses

Filter:

- Administrator
- Community name

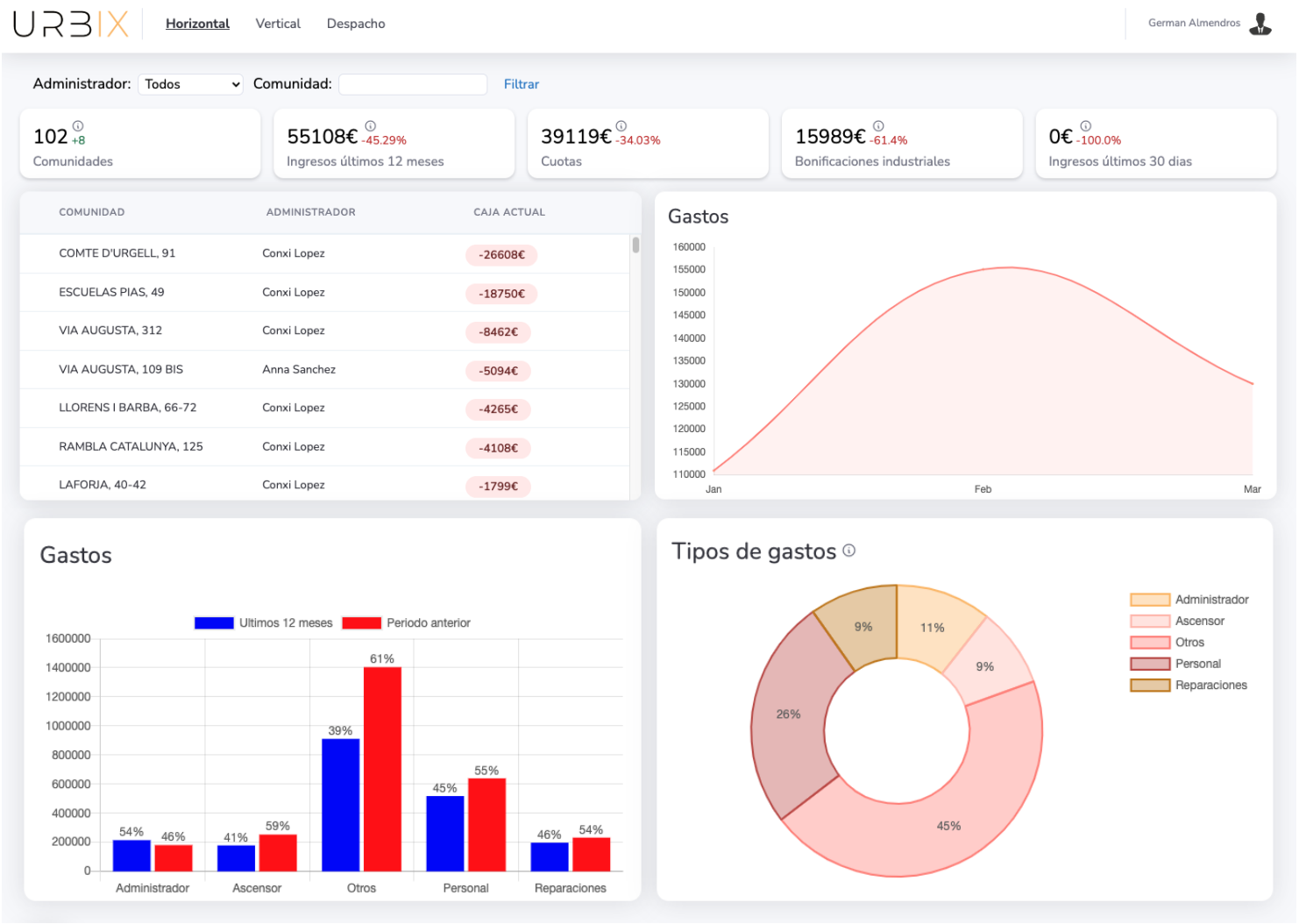

Fig 1: Community Dashboard

4.1.2 Rental Dashboard:

This dashboard consists of the data analyzed from the rental assets currently in the user's portfolio (including the ones on the market for rent and the ones currently rented).
The aim of this dashboard is to be used not only by the owner or executive team but by the different property managers that manage the portfolio of rental assets.

KPIs:

- Number of owners
- Variation of the number of owners vs the year before
- Number of assets under management
- Variation of the number of assets vs the year before
- Mean recurrent revenue per asset (from rent)
- Variation of mean recurrent revenue per asset
- Income this year
- Variation of income this year vs same period the year before
- Expenses this year
- Variation of expenses vs same period the year before

List:

- Owner
- Administrator
- Properties
- Communities/Buildings
- Balance

Charts:

- Bar chart with the number of rents by month (this year)
- Bar chart with the type of owner's expenses

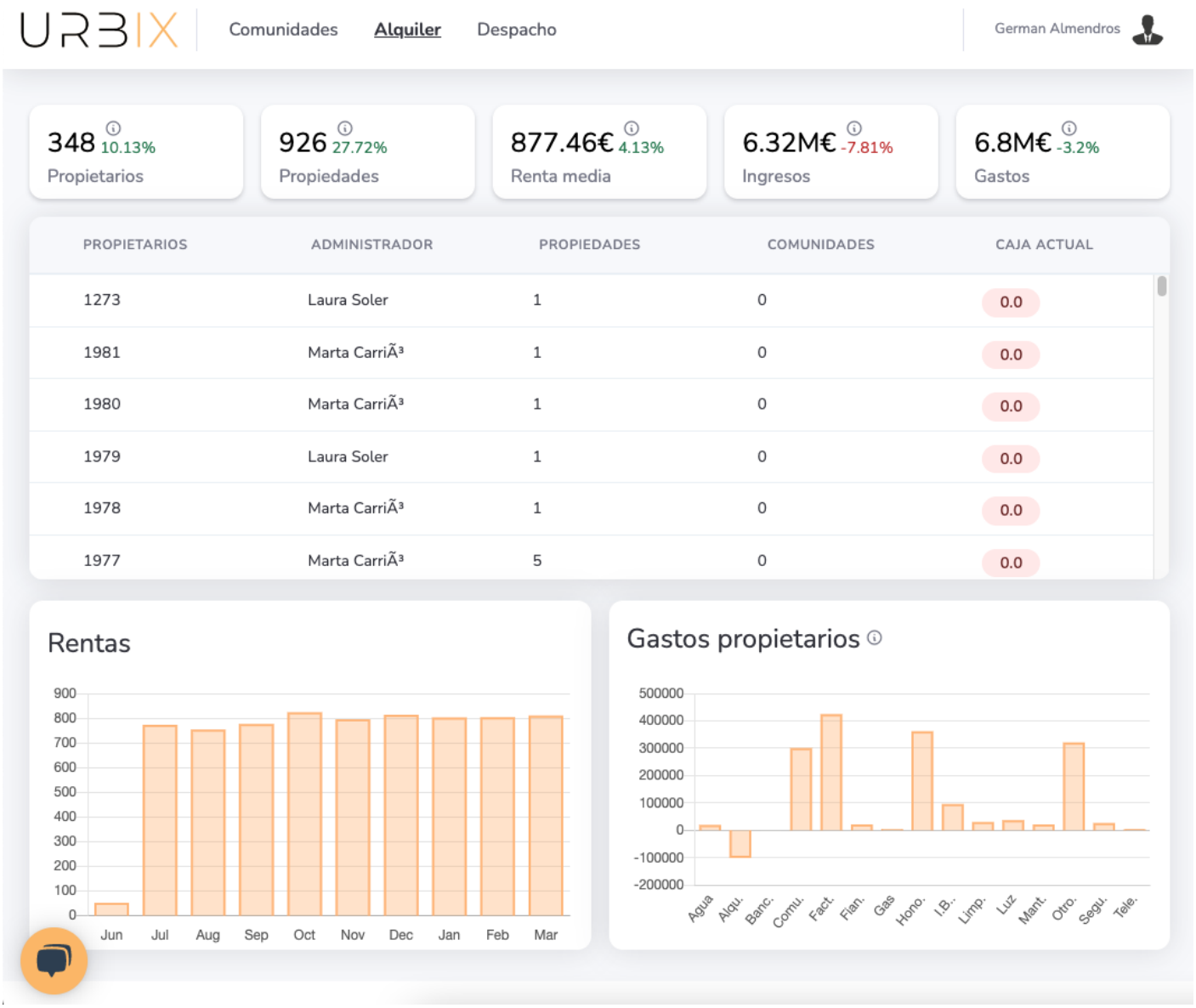

Fig 2: Rental Dashboard

4.1.3 Office-Operations Dashboard:

This dashboard consists of the data analyzed from the office and operations. From employee performance to revenue and incomes of each client.
The aim of this dashboard is to be used by the owners, executives, and/or managers of the company as they can look at employee performance metrics or the global income and revenue of the company.

KPIs:

- Number of owners
- Variation of owners vs month before
- Number of assets
- Variation of assets vs month before
- Number of communities
- Variation of communities vs month before
- Income this year
- Variation of income vs the same period the year before
- Income this month
- Variation of income vs the same month the year before

List 1:

- Community name
- Income from recurrent revenue
- Income from service bonifications
- Related income from rental assets in the community
- Manager/Administrator

List 2:

- Owner
- Number of assets
- Number of communities managed where there are rental assets
- Income from assets
- Manager/Administrator

Charts:

- Line chart with a total income of current year
- Pie chart representing the percentage of income by type of income

Cards 1:

- Community administrator's name
- Number communities under management
- New communities acquired current year
- Retention of communities from the year before
- Income

Cards 1:

- Rental asset manager's name
- Number assets under management
- New assets acquired current year
- Number of owners
- Retention of rental assets from the year before
- Income

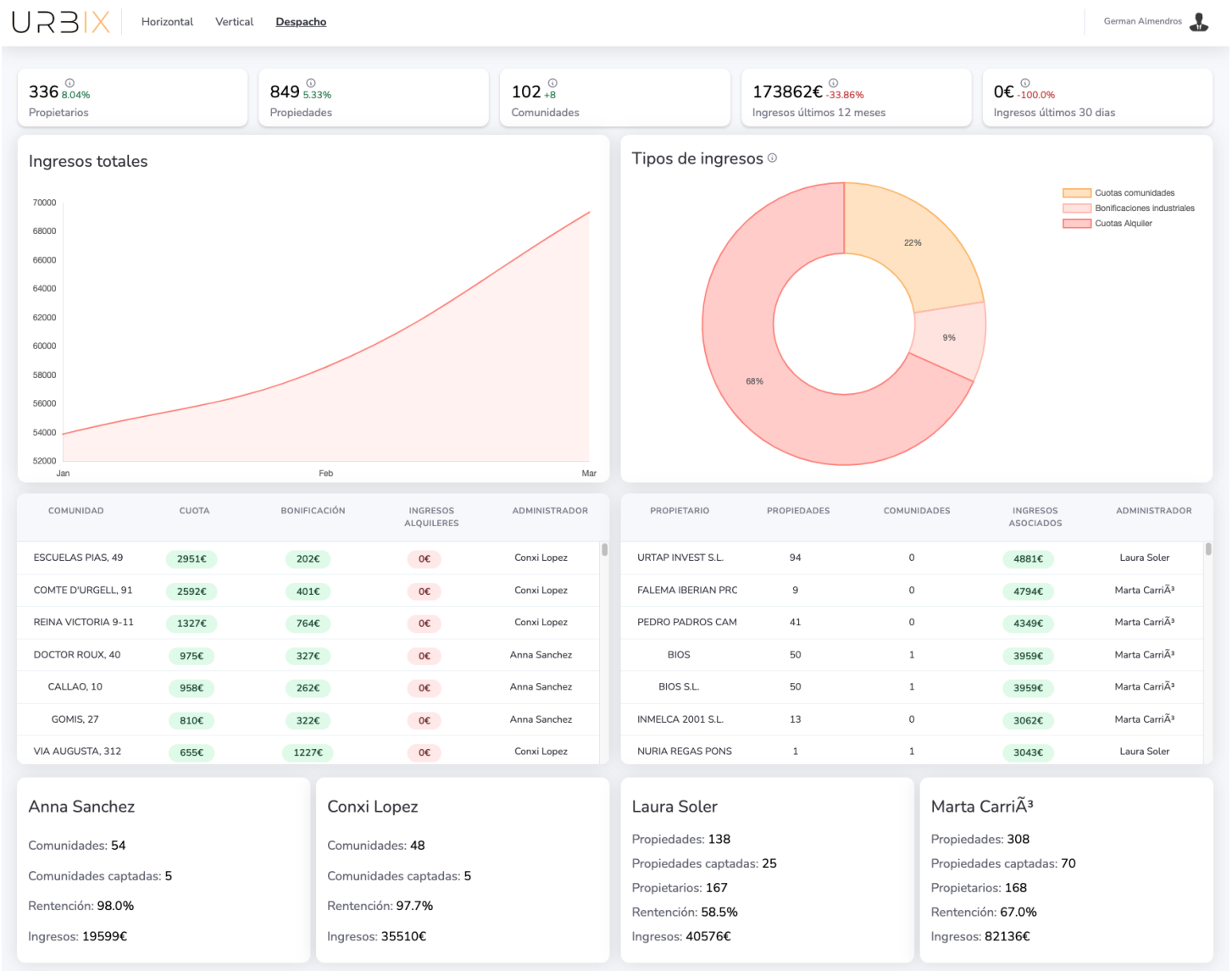

Fig 3: Operations Dashboard

4.2. Product technical needs

The MVP aims to help users take better and faster decisions by looking at the dashboards:

- ERP integration: As we need the dashboards to be updated with the last data, the product needs integration with the software the users use to manage their business. We also want to minimize the integration time for each new user and we want to be even self onboarding.
- Recurrent update: As the decisions can not be taken using outdated data, the MVP has to update the dashboards recurrently.
- Data standardization: As we aim to build a scalable product, we have to take into account in the ETL that different clients have different ERPs and data schemas and they have to be able to integrate with the MVP with minimum deployment time.
- Short response time: If we want the users to use the product on a day-to-day basis, we have to make sure that the response when loading the data to the dashboard doesn't take too long. A long wait will imply a bad user experience and a less "sticky" product.

4.3. Technologies used

Front-end:
- HTML
- CSS (Tailwind)
- JS
- ChartJS framework

Back-end:
- Python (Django)

Data Processing:
- Apache Spark (Pyspark)
- Aws lambdas
- Python (Pandas)

Infrastructure:
- Aws EMR/Glue (clusters for running spark scripts)
- Amazon lightsail (for the landing page)
- Aws S3 (data storage)
- Aws RDS Postgresql (database)
- Aws EC2 (backend and frontend servers)
- Aws Lambdas (functions for getting the ERP data, deleting files and inserting files to DB)
- Aws CloudWatch (monitor infrastructure)

DevOps:
- Github
- Docker & Docker-compose (container deploy for backend and frontend)
- Nginx (container for backend servers API calls in production)
- Github actions (automate CI/CD and production deployments)
- Aws EventBridge (trigger lambdas and spark scripts)

4.4 Tech infrastructure and processes

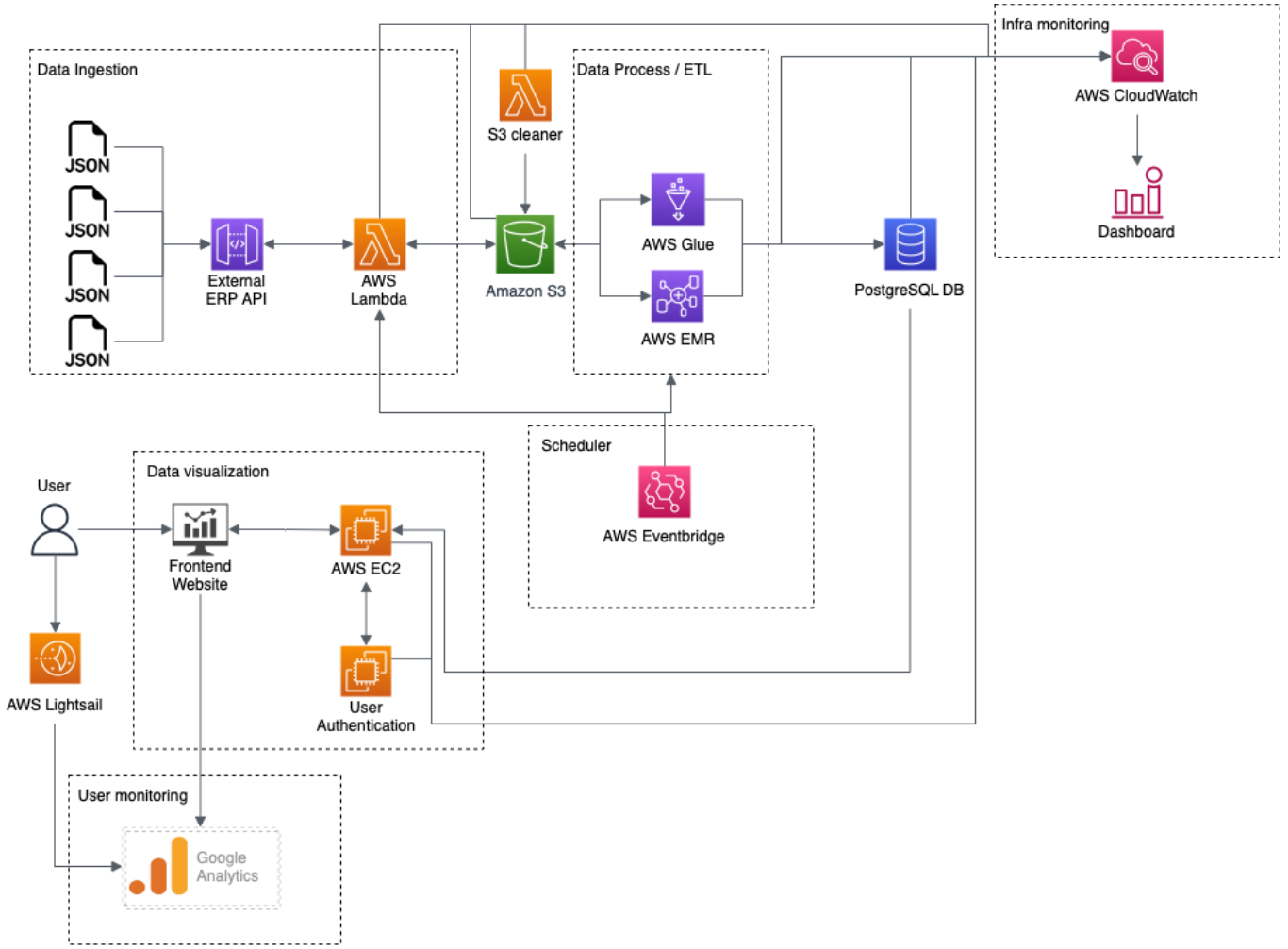

Fig 4: Design of the infrastructure of the project

4.4.1. Data Ingestion

The data ingestion is done through integration to the client ERP software via API. The ERP company provides us with an authentication key for each user.

These APIs have different endpoints to retrieve the different data. The API enables us to collect the data we want and retrieve it to us in JSON format. On the different endpoints, we can choose which data we want, when we want this data, and from which user.

For each ERP we have different endpoints and the data is saved and processed separately depending on the ERP of the user. This enables one pipeline for each ERP and any users that use an ERP that we are integrated with can use the platform without further development providing significant scalability.

Our different lambdas get triggered by AWS Eventrbridge schedulers at a given time every day or week depending on the type of data to retrieve. If the data is likely to be updated like asset incidences or expenses, the data is retrieved daily. Asset information is more static data that can be retrieved weekly to save processing computation (and money).

Once triggered, the Lambda access the user data (auth token, name, etc) needed to call the APIs endpoints and store the data retrieved and invoke asynchronously the other lambdas that call the different API endpoints and save the data into S3 in the corresponding path with the corresponding name.

4.4.2 Data Process

The data process is based on different PySpark scripts running on top of AWS Glue and AWS EMR (data processing cluster). Although AWS Glue provides a friendly interface for building ETL scripts, I decided to build raw pyspark scripts to not only better optimize and personalize the transformations process but to also be able to export them and run them into EMR if the scale requires to have full availability of data pipelines and real-time processing.
Every spark script is in charge of transforming an ERP data unit. We call a data unit the data retrieved from one endpoint of the ERP API. Some of these scripts, following the behavior of the data retrieval process, are triggered daily or weekly.
Once the script is triggered, it reads a config file from S3 with the data schema. Then it loads the data from the S3 paths specified and transforms it to fit the standard scheme of the Postgresql database (independently from the ERP source). Once the data is transformed, the data is saved to the database and also to the S3 to easily backup in case something happens to the DB.

4.4.3 Infrastructure monitoring

All the different components and infrastructure running on amazon web services are connected to AWS CloudWatch.
This amazon service receives the logs from all the services and analyzes them to monitor the key metrics.
From the AWS Lambdas, we analyze the number of calls to the different functions, the mean running time and the proportion of success and error for each call, and the concurrent number of calls.
From the S3 file system, we analyze the number of operations performed in the file system (PUT, GET, DELETE) and storage metrics like GB in use. These metrics not only help to know how much we access the file system but to relate these to the expected cost of using S3.
From AWS Glue, we analyze the number of runs, number of errors, number of cores used in the process, and execution type per script. These metrics help us monitor the performance of the ETL pipelines not only by knowing if the processes are running without errors but also to know which processes should be running with a bigger cluster to reduce the execution time.

From the Postgresql database, we analyze key metrics to ensure that the database is responding with the expected latency and without errors. We also monitor the different connections made to take into account the normal use of the database and potential optimizations to be done to improve the performance. These metrics are from CPU usage or database connections to reading and writing latency.
From EC2, we analyze the logs to study the performance of our servers (where the containers run) and make sure they are up and running without issues.

4.4.4 Data visualization

The data visualization runs on the frontend and backend dockerized containers built on top of AWS EC2 Linux servers.
The HTTPS calls from the user in the different endpoints are handled with Django (python backend framework). Django not only handles the calls from the users but also makes sure each user is authenticated using the authentication system built on top of the Django framework.
Every user should have logged in before retrieving any data. If the user is authenticated, each endpoint retrieves data from the database and it is processed with pandas to fit the final charts. The data retrieval process is made using SQL queries optimized to make sure the time is as low as possible. Some data is not aggregated using SQL but pandas because the C embedding that the panda's library uses makes it more efficient to process the data in the backend rather than in the SQL query. Once the data is processed, it is sent to the frontend with additional information to display the HTML, CSS, and javascript needed to render the different dashboards.

4.4.5 Landing page

The landing page is the webpage used as the face of our software and company. This page runs on top of AWS Lightsail and is optimized for ultra-fast loading speed to make the user experience (UX/UI) the best for the leads and potential customers.
The page runs with the https://urbix.org/ URL while all the software runs in https://app.urbix.org/.
Having the landing page separated from the main software frontend helps us optimize the landing page to the extreme in terms of SEO, loading speed, user volume scalability, and optimize in other terms the software server like data processing speed, libraries needed, etc. Following a scalable coding architecture is also a good practice to split the different services (in this case dashboards and landing pages) for a better programming experience.

4.4.6 User monitoring

On the frontend side, we use google analytics to study the behavior of the users on our platform. We analyze the number of sessions each user does per day, the number of time spent per session, the dashboards they watch more and related metrics.
We also implemented Hotjar, a software that creates heat maps based on the usage of the users in our platform and also records the user sessions. This type of software helps us better understand the behavior of the users, how they use our product, what we should improve and where they spend their time.

4.4.7 Data storage

For data storage we use two services: AWS S3 and AWS RDS (PostgreSQL).
AWS S3 is a file system like HDFS and we are using it as our data lake. We store the different files following a structured path system. Config files are stored in the "/config" path. Raw files obtained from the ERPs are stored following the path "/raw/erp/type_data/user_datetime.json". This way we can split the raw data from the processed one and identify the data from the same ERP (following the same schema). We store the processed data in a parquet format optimized for reading with Spark (as a backup if something wrong happens to the database) in a path like "/processed/type_data/user_datetime.parquet".
AWS RDS is the database service from Amazon. We have built the RDS with Postgresql. This database is the database where user information and processed data are stored. We have a table for each type of data that we want to store and the corresponding primary keys and foreign keys. For example, we have a table for the rental assets information where we store information about each asset like which user is managing it, where is the asset, what type of asset is, and much more. Being the asset_id and the user_id the primary key of this table, we relate this with a foreign key to the table asset_expenses where we store information about the expenses of this asset.
The different tables are optimized using different indexes or materialized views to ensure a good response when retrieving the data to visualize it in the different dashboards.
Some data (raw and processed) is saved in the S3 and the database and even duplicated to make sure we can improve and modify the different features of our product. Having the processed data in the S3 and the Postgresql helps us make changes to Postgresql making sure we are not losing the data. Having the raw data saved in the S3 data lake makes it easier for us to modify core analytical aspects like adding more analysis making sure we can modify the Spark ETL scripts and re-run these scripts to fit the new metrics with the data that we already retrieved days ago. This helps us improve our product faster with minimum impact on the user experience during the implementations.

4.4.8 Scheduler

We have implemented different schedule rules using Aws eventbridge. With these rules, we can trigger at the same time (now implemented one time per day) services like the AWS lambdas that retrieve data from the ERPs of our users. This is what makes all the pipelines work automatically every X time without the intervention of humans.

One simple configuration change and we can go from daily pipeline execution (data retrieval, processing with ETL and saved to the Postgresql) to minute pipeline execution for having almost real-time support and keep the dashboards updated up to the granulometry we want to define.

4.6 Further improvements

After analyzing the use case of the whole software and the metrics associated with it, we have found some clear bottlenecks that should be improved to not affect the user experience.
The improvements are in the line of upgrading the MVP to a version 1.0 of the product fully prepared for the market.

4.6.1 User experience

The first thing we should improve is the loading time of the dashboards. Now every dashboard performs multiple SQL calls into the whole database to get the data and then processes it individually with pandas every time a dashboard frontend endpoint is called. All these different calls to the SQL and transformations to fit the front-end dashboards are done synchronously with every call. The first improvement we could apply is to rely on all these queries and transformations to different endpoints as a REST API and then call them asynchronously. This could speed the response of loading the dashboard by 5x taking into account the different queries every dashboard has.
Another improvement on the line of optimizing the end-user experience could be caching the different aggregated and transformed data that is displayed in the dashboards. We could implement a caching system with a Redis database where we cache all the different end transformations to fit the dashboards. This data could be updated every time new data is added to Postgresql and the different transformations could be done in “offline” mode. The queries made every time a dashboard endpoint is called should be significantly faster as no transformations will be needed and the queries will be done in a cache database.
For data reliability, more scalable infrastructure, and aiming at international use of our software, we could implement sharding to our main Postgresql database to assure minimum time response, no blocking operations, and more redundancy.

4.6.2 Authentication

This type of software, where sensitive data of our clients are managed and displayed, the authentication system and security are key. Now, the software authentication system is built on top of Django’s authentication framework on a different isolated server with a different database. Nevertheless, as it is critical to have security standards, we should rely on this authentication management system on third-party software like auth0 (https://auth0.com/). As the complexity of the user management grows for different clients involving different stakeholders with different levels of access to the different dashboards, third-party software can make it much easier to accomplish the expected security standards.

4.6.3 Code refactoring

The code of this project was built without taking into account a paradigm of development like SOLID and without taking into account practices like TDD. This came up with hard maintainable code with low testing. Refactoring the different endpoints following the improvements defined before is necessary for keeping the code clean and maintainable.
The refactoring should be made based on the testability of all the code with unit tests and following a clean architecture of DDD.

4.6.4 Security

In the actual MVP version, the data analytics software runs on HTTP, no HTTPS. We should migrate to HTTPS to assure end-to-end encryption and no data leaks. Also, using HTTP in some browsers implies not being able to access the page due to security concerns.
We should apply some security measures to the infrastructure, for example, we should cut the incoming connections to the database that are not coming from specified server IP or VPN (one of our servers). This avoids potential security issues of unwanted external connections that could imply critical data leaks.

4.6.5 Document the code for scalability

Following the refactoring improvement, we should document better the code and infrastructure to be able to onboard new hires to the development team. Adding documentation for the different API endpoints and the ETL scripts and lambdas would be an improvement in the time of explaining to new developers how our code is structured, and how our API and data pipelines work.

4.6.7 Testing

For a marketable software product testing is key. In the actual versión of the MVP, there are only some unit tests on the API side of the software. We should add a significant amount of tests to cover all the code-critical aspects. We not only have to add more unit tests on the API side but also different integration tests and end-to-end testing of the whole backend.
We should also build tests for the different lambdas scripts and spark ETLs as it is a critical part of the project.

4.6.8 Optimize CI/CD

Adding a pipeline of continuous integration (CI) and continuous deployment (CD) is key in a software project that runs on production.
In the MVP version, there is only a tiny CD pipeline that deploys the last version of the main GitHub branch into the production servers.
To this pipeline, we should add testing before merging a different branch to the main branch and stop the branch from merging if all the tests are not passed. We should also add some commands to run into the production server for collecting the static files in case there are changes as now we have to do it manually.

4.6.9 Monitoring

In the MVP we had taken into consideration the monitoring of the infrastructure and user behavior but we should go further if we want a production-ready fully scalable product.
We should add logging in the different API endpoints of the server (and containers) to track the different requests handled by the different users. These different metrics should help us understand the workload of the containers, helping us to better identify the bottlenecks and improve the performance of the application. These should also help us identify the different interactions the user has with our software and where we should put the effort for improvements.
There are different software and open-source projects that let us monitor all this. One example of commercial software with a free version that could work is Datadog. Another example of an open-source project with a self-hosting versión (just pay the cost of hosting) is Grafana.

4.6.10 Build integrations API

As defined before, the integration of our software with the ERP of our clients is key to the success of this project.
In the first stages of our go-to-market, we know that we will have to push and integrate our software with the ERPs by ourselves. Nevertheless, when this type of project grows, they open an API to enable any software to integrate with their software.
That’s why we should start building the foundations of an open integration API to enable any ERP to build their integration with our software and open to new potential customers without the need of pushing ourselves for an integration with their ERP.

4.6.11 Build onboarding automation

If we want to be ready to grow and onboard new customers in a recurrent way, we have to automate the onboarding process.
In the MVP, every new user has to onboard manually although we are integrated with their ERP. We have to download and process the existing data in the ERP and add the customer to the automated data pipeline manually.
When talking about user onboarding, time and simplicity are key. We should be able to deliver the dashboards to use within less than 5-10 minutes from the registration of the user. If this process takes hours or days it may lead to the loss of interest of the potential customer apart from the obvious scalability issue of onboarding manually every customer that registers to our software.

5. Test with real users

5.1 Context

During the development of this project and MVP, we tested with real users two times:

- A proof of concept test with defined datasets of batches of 20 assets with three of the most important property management companies in Barcelona: Forcadell, Finques Feliu and Mas i Fill. This test was a proof of concept for many basic dashboards and user interfaces, without real-time data processing and with no integrations with the ERP.
- A full test with real-time automated data processing pipelines fully integrated with the ERP of the stakeholder. This was a test of the MVP functional and ready to be used on the day-to-day basis of a property management firm. The test was done with the company Finques Almendros, with more than 1100 assets under management.

5.2 Use cases

With the different tests we could identify some use cases of direct improvements in day-to-day operations thanks to our data analytics software.

First of all, as they had all the information of different assets in the same place analyzed, they could better tweak the different processes to achieve a better margin on returns.

With the economic KPIs, they can better manage their economies of scale and adjust the different budgets according to the expected cash flow. These types of decisions are the ones that, by what they have told us, they can’t do without our tools.

All the metrics of expenses of the different assets can be really useful in the meetings with clients and internally to identify key improvements and also to guide new clients with new assets about the expected expenses and income they can achieve thanks to the analysis of similar assets.

Having the different asset's cash in their coffer helps identify fast the ones that need a capital injection quickly to avoid the return of expense invoices of that asset. The real-time of this type of analysis is key to planning and preventing technical defaults of some assets that experience an increase in expenses during a short period.

5.3 Feedback and future improvements

The feedback from the proof of concept and the final MVP, was really good. In general, the different people from the companies we did the tests identified much more use cases than I identified at first. These use cases were not only improving the day-to-day tasks and decisions they have to make but also they were able to perform critical tasks that they couldn't do before.

Both agreed that this could be an essential tool for better managing their business. The ones that participated with the proof of concept urged us to integrate with their ERP (TAAF) to pay for the full version of our software.

German Almendros from Finques Almendros that was the one that could use the full MVP version of the software fully integrated with their ERP assured us that having this connectivity with the software was key for having real-time analysis and once we launch the full commercial version, he will be the first of recommending our software to the other companies with the same ERP.

With all this different feedback, the proof that the product works and solves critical problems from potential customers is solid. Nevertheless, we have to implement different functionalities to grow the value proposition with further analysis. For example, they mentioned that adding more filters could help them analyze by themselves some groups of assets sharing some specs. Also, adding a geolocalized analysis by zones could be useful to identify trends by localization (like increases in the rent or expenses).

6. Conclusions

The project has been a great opportunity to learn from the beginning to the end how to make a business idea come true. Starting from the idea, doing the market research, building the MVP, and validating the idea with real users. The market research was fundamental to understanding the business, the sector, the stakeholders, and the users and to identifying their needs and how our data analysis software could help them solve these issues. I can say that the goals defined in the objectives of this project have been fulfilled.

Defining the business model and go-to-market is key in this type of project to assure the viability of the project before devoting a long time to the development of the MVP.

The MVP development was the most important and challenging part of the project. I had to learn and apply many new tools and technologies that I had no previous experience with. All while establishing continuous feedback conversations with the stakeholders to make sure that the MVP feed their needs.

Testing the MVP with real users was the most gratifying part. It was great to see that the users not only found our software useful but also easy to use. All in all, it has been a great learning experience which I have enjoyed a lot and I am looking forward to continuing working on this project in the future.

Although we can take this MVP as a success, there is a lot of work to be done not only on the product and technical side but also in the go-to-market strategies implementation. There are some critical challenges to be solved like the scalability of the software that depends on ERP integrations and turning this MVP into a sticky product for our users.

8. Bibliografia

[1] *Housfy revenue plan for 2022 - https://elinmobiliariomesames.com/empresas/housfy-preve-alcanzar-los-43-millones-de-facturacion-en-2022/*
[2] *Proptech information and venture capital status in real estate Spanish market - https://twitter.com/carlosblanco/status/1518939101025689600/photo/1*
[3] *Authentication service - https://auth0.com/*
[4] *Funding round Clickalia - https://www.ejeprime.com/empresa/clickalia-recauda-450-millones-de-banco-santander-y-el-fondo-estadounidense-fith-wall.html*
[4] Django documentation - https://docs.djangoproject.com/en/4.0/
[5] Pyspark documentation - https://spark.apache.org/docs/latest/api/python/
[7] Pandas documentation - https://pandas.pydata.org/docs/