**TECHNICAL REPORT OF THE INITIAL SOFTWARE**

**REQUIREMENTS SPECIFICATION OF A SOFTWARE PROJECT**

1st of march

version 1.0



Pings design and development team

Juan Fernández del Pozo Romero

Guillermo Martin-Coello Juarez

Rodrigo García Redero

Daniel Varela Sánchez

**ID: EcoUAM-ShareIt**

**Change log table**

| **Version** | **Date** | **Contents** | **Purpose** |
| --- | --- | --- | --- |
| 0.1 | 23/02/2022 | Creation of the document | Create the document |
| 0.2 | 25/02/2022 | Creation of the introduction | Enter what the project will be about |
| 0.3 | 25/02/2022 | Writing the system interaction | Adding the mockups and explaining them |
| 0.4 | 28/02/2022 | Defining the project goals and scope | Structure the fundamental ideas of the project |
| 0.5 | 28/02/2022 | Detailing the Project Subsystems | Provide a brief inside look of the systems that will make up the project |
| 0.6 | 01/03/2022 | Front page creation | Create front page |
| 0.7 | 01/03/2022 | Appendix creation | Addition of the extra files |
| 0.8 | 01/03/2022 | Creation of the summary | Summary of the full document |
| 0.9 | 01/03/2022 | References and styling | Provide final details to the document |
| 1.0 | 01/03/2022 | No new contents | Revision of the full document |

**Abstract**

The eleventh edition of the EcoUAM competition for IT Innovation Projects has been published by the Autonomous University of Madrid (UAM). The project is linked to increased mobility in private transportation among UAM members.

The document outlines the software project that would be presented to meet the main requirements for a software with these capabilities. There would be four primary segments in the document.

The user management subsystem will offer methods for authenticating users using their UAM email and password. Users will be able to fill out their personal information, as well as view and reserve rides that have been listed. The Publication of Rides subsystem will include mechanisms for publishing new rides along with their details. The ride seat booking management subsystem will include techniques for reserving various rides. Methods for performing payments using VPA, which stands for 1€, will be available in the payment management subsystem. The implementation would be kept as simple as possible.

Using Google Maps or Apple Maps, the system would be able to provide location assistance. Also included is payment support via a payment gateway that will be used.

Finally, some functionality, such as in-app calls or customer service, would be impossible to implement. Sharing locations or communicating with drivers are examples of future possibilities. Some of the objectives include encouraging UAM employees to join and share rides, as well as lowering carbon emissions.

Index

[1. Introduction](#_19ag8en1ici2) **5**

[1.1 Description of the problem. Delimitation of the problem](#_q5f8f7quied6) 5

[1.2 Definition of the computable need of the group/s of people affected by the project (context, importance)](#_4hmhmrddvc52) 5

[1.3 Document goals, recipients](#_4a92dnax6q2c) 5

[1.4 Project goals, project beneficiaries, system goals, functionalities/services of the software system, users](#_5x4bgbbeynmm) 6

[1.5 Methods used/procedure. Relevant antecedents for the report](#_tni0nhyzchof) 6

[1.6 Document explanation](#_94aaznie3jn7) 6

[1.7 Explanation of the logical structure of the document](#_jpm2f0udz05b) 6

[2. Project Definition](#_j7vwzuhv68qt) **7**

[2.1. Goals and Functionality](#_ahhzhxw8uune) 7

[2.2. Initial Catalog Requirements](#_n1o1lulekrdw) 8

[2.2.1. Functional Requirements - FR](#_rjy9eeihwaxh) 8

[2.2.1.1. User Management - UM](#_8f5b7kchgwsb) 8

[2.2.1.2. Publication of Rides - PR](#_9jrv1oo98mmv) 9

[2.2.1.3. Ride Seat Booking Management - RSB](#_j3psn445iaji) 11

[2.2.1.4. Payment Management - PM](#_m1te4kvqsl4w) 14

[2.2.2. Non-Functional Requirements - NFR](#_7mfmrm1b4pgx) 15

[2.2.2.1. Operational requirements](#_e34n90xiib2u) 15

[2.2.2.2. Security requirements](#_o3ys2kg96tl0) 15

[2.2.2.3. Portability requirements](#_ironxzpqkijx) 15

[2.2.2.4. Verification/Identification requirements](#_56cqs0u2ks3y) 15

[3. System Interaction. Conceptual and Visual Design](#_s84jdk2urnrj) **16**

[3.1 Login & User Identification](#_g3zeowplvans) 16

[3.2 Main Page](#_4esfd9c7ixzm) 17

[3.3 New Ride](#_qjem2tw1l9ww) 19

[3.4 Rides](#_p5rh7x9cd2le) 19

[3.5 My Rides](#_kdwbo5mkwnc1) 22

[3.6 User Profile](#_4ep6h5w2hoqm) 26

[3.7 Balance](#_4h5osb8wkpmh) 28

[3.8 Messages](#_7uj7kv19duyh) 30

[4. Conclusions](#_g3i3rqfi8smo) **32**

[**References**](#_qr1l9f1gcvfd) **34**

[Services and applications used](#_793fslsps30t) 34

[Documents used](#_mnu5cb9kq7kx) 34

[Appendices](#_ebe8r75dn8ql) **35**

[Appendix 1: Brainstorming](#_7dfzzna9q4ku) 35

[Appendix 2:Competitive analysis](#_kmmxh42f6wby) 38

[Appendix 3: Meeting minutes](#_tjw7572rj5oj) 45

[Appendix 4: Old mock-ups](#_1jlem28byqg3) 59

# **1. Introduction**

The current level of global connectivity is a significant leap forward in human history. Humanity has never had a connection like this before, thanks to the Internet and other technologies. Resources can be found in every corner of our world.

Every day, every second, netizens from all over the world utilize mobile applications or web applications. Applications for ordering food, watching or downloading films or movies, reading, playing games, and, in our situation, care sharing have all been developed.

Car sharing is a terrific method to develop healthier habits for our environment and save money, among other things. These are some of the reasons why we wish to improve UAM mobility and showcase our car-sharing system (Share it!).

## **1.1 Description of the problem. Delimitation of the problem**

The eleventh edition of the contests for IT Innovation Projects has been launched by the Autonomous University of Madrid (UAM). The EcoUAM project competition is linked to increased mobility in private transportation at UAM. Using the Google Maps or Apple Maps Tool, it is necessary to integrate private transportation in an interactive system.

## **1.2 Definition of the computable need of the group/s of people affected by the project (context, importance)**

The system will notify and assist UAM users about private transportation options available for use as a form of mobility when arriving or departing from the UAM. This will be accomplished through the creation of an application design that allows users to easily access this information. The app will be a mobile app, as we believe this is the most commonly utilized method among UAM members.

## **1.3 Document goals, recipients**

The project's main goal is to create and offer to the EcoUAM competition a set of specifications for software that includes these characteristics. The material would be given to the EcoUAM service in such a way that it would be simple to understand.

## **1.4 Project goals, project beneficiaries, system goals, functionalities/services of the software system, users**

The purpose of the project is to create software that will allow UAM users to manage private transportation at the university. To do this, our team will develop a mobile application that will make managing personal vehicles easier via a car sharing app. Among other things, UAM members would be able to book and post rides.

## **1.5 Methods used/procedure. Relevant antecedents for the report**

Approaches such as brainstorming ideas among team members and categorizing them as original ideas are among the techniques employed.

Another process employed was competitive analysis, which required each team member to conduct an internet search and examine the features of other systems, categorizing them as advantages or negatives and extracting ideas that could be useful for our system.

Finally, we created mock-ups to gain a visual representation of how the program would look.

## **1.6 Document explanation**

The goal of the document is to clearly identify the problem that has to be solved. Identifying software needs for application implementation. Include any project-related advantages. Provide mock-ups that depict the app's design.

## **1.7 Explanation of the logical structure of the document**

The project definition would be the next section of the document, and it would be used to describe requirements for each subsystem. The mock-ups created to show the app's visual content would be the third section. Finally, there will be a conclusion where some key future decisions and conclusions will be reached.

# **2. Project Definition**

In this section we will go over the different requirements that shall be met in our project, detailing the subsystems that will form the proposal. First we will introduce the project definition and comment on what our goals are, and the ideas that remain in or out of our scope. Secondly we will address in more detail the requirements and how they are classified.

## **2.1. Goals and Functionality**

When it comes to the goals of this project, the main functionality that our product must meet when it is delivered, we arrive at some well defined points. First, the product will provide a means of communication, management, and payment agreement between members of the UAM. The purpose is for the product to ease car sharing and help normalize this practice. The app will hold drivers, who will be able to post rides, for the riders to accept, and charge them for it. Riders may book, accept or cancel a seat in these rides, and pay a fare for it. The application must be accessible for all UAM members, and the payments will be virtual, through the app.

Regardingthe scope of the project, as aforementioned, the main goal is to deliver a practical way for UAM users to interact. This idea leaves out of the scope initiatives like private insurance or a 24h phone liveline for issues. We will also be hiring services of third party companies for the GPS infrastructure, so a personalized GPS navigator is out of the project. Since the project will include in-app chat, we are purposely leaving out an integration of any phone calling system. This would be a source of issues regarding economical maintenance, privacy and other policy infractions.

To provide a structured development of the project, we have divided it into four subsystems which integrate all the required functionality. The requirements have been grouped under the following: 1.The User Management Subsystem, which will is responsible for obtaining, validating and storing the user data; 2.The Publication of Rides Subsystem, which will enable publishing rides, configuring trip details, and deleting existing ones; 3.The Ride Seat Booking Management Subsystem, which will allow riders to search for rides, filter them and make a reservation. It will also check whether a reservation is possible and will take care of the credit traffic between users; 4.The Payment Management Subsystem, which is vital regarding the income and outcome of real money into the application. It will define the virtual currency and implement safe transaction protocols to enable the payment gateway.

## **2.2. Initial Catalog Requirements**

In this next part, we will define the project requirements needed to achieve the product objectives in a more specific way. First, we will present the functional requirements, grouped under subsystems. The non functional requirements will follow those and will be grouped into themes.

### **2.2.1. Functional Requirements - FR**

#### **2.2.1.1. User Management - UM**

* **FR-UM-1:** Only authenticated users will be able to access the application (except for the login page).

1. If the user is not logged in he will see a login page.
2. To access the page users have to write username and password in the login interface.
3. When the user writes username, password and other data fields correctly and clicks the login button the application functionality will become available.

* **FR-UM-2:** When users log in, the system will verify the access data (UAM email and

password) entered by them with the UAM authentication system.

1. Data written in the login interface (username and password) will be processed by the user authentication system.
2. If the authentication is accepted, the system will assign a session to the user

(authenticated user).

* 1. If the UAM authentication system returns a valid response a new user session will be created and the application will be shown to the user.

1. If the authentication system returns a not valid response the login interface will show again with a Invalid Username or Password message.

* **FR-UM-3:** When users log out, the system will delete their session (unauthenticated user).

1. To log out, the user must click on the settings button and then on the logout button. The user session will then end. The system will save all user data in case something has changed and then it will delete the current session.
2. The application will stop being available.
3. The login interface will show up for a new user to login.

* **FR-UM-4:** When users access the system for the first time, an “empty” profile will be created.

1. If the user hasn’t used the application before, a new user profile with the user data will be created when the login button is pressed.
2. A new virtual payment account will be created by the system with basic payment information of the user.

* **FR-UM-5:** Authenticated users will be able to see the rides published and reserved by them (historical) as well as the payments they have made for each ride.

1. Authenticated users can see a list with all published and reserved rides (by the current user) by pressing on the list button in the main page and then the my rides button. Rides will be organized on a historical order on the list. Payments of each ride will be shown next to each ride data.

* **FR-UM-6:** Authenticated users will be able to see the profile of the user offering a ride.

1. Authenticated users can select the user icon of a specific ride to see the user profile of the driver. This will display a profile page with all the information about the drive, including name, age, picture and reviews.

* **FR-UM-7:** Authenticated users will be able to send and receive messages

1. The users will have a chat to interact between each other.
2. Users will see if the chat is for passenger or driver mode.
   1. Passengers will be able to speak with drivers of booked and confirmed trips.
   2. Drivers will see passenger chats for paid trips. Additionally they will also be able to chat with passengers that have only booked a seat.

#### **2.2.1.2. Publication of Rides - PR**

* **FR-PR-1:** Authenticated users will be able to publish rides for carpooling.

1. In the drivers main page there will be a button to create a ride once the button is pressed the ride creation page will be displayed.

* **FR-PR-1.1:** They must indicate the place of departure, date and time, number of persons admitted, whether smoking is allowed in the vehicle, destination, and price per seat. They can also include (optional) a textual description with pick-up points along the ride and/or type of vehicle.

1. The ride creation page will contain all the fields needed to create the ride.
2. Once the user fills in all the ride details, the publish ride button will activate and he will be able to publish his ride.
3. The ride then will get stored on the external rides database.

* **FR-PR-1.2:** Optionally, users can indicate whether it is an occasional or recurring ride.

1. For frequent rides drivers must indicate how often they take them and

a date when they estimate to quit taking this ride (maximum 2

months).

* 1. Once the ride is marked as recurring, a button for each day of the week will be shown and the user will be able to select the days of the week he is going to make this ride (we assume that the recurrence of the rides are weekly and that if there are some specific days the user will not make the ride he will cancel those days manually later).
  2. Also, when marked as recurring, a date will be asked to the user indicating when he will stop making the ride.

1. The system will register all the rides configured by the user (a single

ride for sporadic rides, and all possible rides for recurring rides).

* 1. The application will store in the database the ride and will later allow the user to see the rides that he has taken and published.
* **FR-PR-2:** The system will consider each repetition as an independent ride.

1. When a ride that is marked as frequent is published, the application will publish each of the rides one by one and treat them independently.

* **FR-PR-3:** Drivers will be able to change the number of available seats if it does not affect a reserved seat (that is, a seat that has been reserved cannot be removed).

1. In the driver main page he will have to press the list button, then the my rides button.
2. In the my rides page you can access the main details of each ride. When a ride is pressed the ride's complete details are shown, and if the ride is published by that user, he can change the details there and then press the button to save changes.
3. If the changes are valid, a page showing that the changes have been made shows, but if they are not, a n error page is shown.

* **FR-PR-4:** Drivers will be able to modify any information of the ride if no reservation has been made for it. When recurring rides are modified, drivers must indicate whether the modification affects just one specific ride or all rides from the given date on.

1. Following the steps on the previous description the driver can access the ride modification page where if no passenger has reserved his spot, the driver can make any change he wants.
2. Once the change is made, if the ride is recurring, a pop up will show that will ask the driver if the change should be applied to all the rides or just the single one edited.

* **FR-PR-5:** Drivers can cancel rides (they can be completely canceled or, in case they are recurring, just for one day).

1. The user would have to enter in the my rides page, then he would have to enter the details of a ride that he has published, from there a cancel ride button will show, and when it is pressed on a recurring ride, a message asking to apply the cancelation for one ride or for all will show.

* **FR-PR-5.1:** When a ride is canceled, the blocked credits of the passengers are refunded.

1. Once the ride is canceled by the driver, the system will automatically refund the passenger’s credits and the ride will appear on the system as canceled and never done.

* **FR-PR-6:** Drivers will be able to see the rides they have published (both those already made and those that are still pending)

1. In the driver main page he can press the list button that will take him to his rides page.
2. In that page he will find two tabs, past and future rides. By pressing on the one he wants to see he will be able to visualize the past or the future rides he has been on, will be on or published.

* **FR-PR-6.1:** Drivers will be able to enter the code given by passengers on the rides made that have pending payments. If the code entered is correct, the system confirms the journey made by the driver and the passenger (it initiates the transfer of passenger-driver credits, see Payment Management environment).

1. Once the ride time is past, the passengers will be able to see a pop up with a code once they access their current ride.
2. Once the time of the ride is past, the driver will have an option to input the passengers codes to confirm their ride. From the details of the current ride accessed in the rides page, he will find an option that says scan confirmation codes that will allow him to input the passengers codes in the text field and the system will confirm the payment from the passenger to the driver.

#### 

#### **2.2.1.3. Ride Seat Booking Management - RSB**

* **FR-RSB-1:** Users will be able to see the available rides.

1. The system will show a button for the user to interact and will display all available rides.
2. The available rides button will show on the main screen.
3. For users to see the rides they will need to be authenticated.

* **FR-RSB-1.1:** To see the rides passengers must indicate the pick-up point, date and time, and the arrival point.

1. In the rides display screen a filter interaction system to find rides, with a field to choose date, time origin and arrival.
2. To change the filters the user will need to press the filter screen.
3. Once these filters are used the system will only display rides that satisfy the search.

* **FR-RSB-1.2:** In addition, they will be able to define filters for ride details

1. The system will also show filters for details such as smoke and car type.
2. Users need to be logged in and have accessed the rides page.
3. User will use search filters to change pick-up point, date and time,

and the arrival point. At the same time the user can modify these details filters.

1. The application searches for the rides that satisfy the conditions.

* **FR-RSB-1.3:** When filters are applied the system will display all the results that match the specified criteria.

1. Users need to be authenticated.
2. Application search for rides that satisfy the search.
3. Users see the available rides displayed on the screen with the corresponding drivers dates and times.

* **FR-RSB-2:** Passengers will be able to see the ride details.

1. When the user is authenticated the system will display the option to see the ride detail of a specific ride the user selects on the rides menu.
2. The user needs to be authenticated.
3. The user has to enter the rides menu. The system will display the available rides.
4. The user must press a ride to open the rides detail menu.
5. The system will access the database and show all the ride information to the user.

* **FR-RSB-3:** Passengers can make seat bookings when they are seeing the ride details.

1. If a user is on a ride details screen a button can be pressed to book a seat on the desired car. The system will add the passenger to the respective ride.

* **FR-RSB-3.1:** For recurring journeys, the system will require passengers to confirm if the reservation is for a particular ride or it is intended to be a recurring booking (in this case the final date must be provided).

1. When the book seat button is pressed by the user a pop up message will be shown asking if the ride is recurrent. If so, the system will repeat the process for all the recurrent rides of that trip.

* **FR-RSB-3.2:** The system will calculate and show the user the total cost of the book.

1. A user that has been previously authenticated
2. The user is in the process of booking a ride.
3. The system will calculate the total cost of the booking and display it to the user.

* **FR-RSB-3.3:** The system will verify that the user has the required credit in his/her virtual payment account.

1. The system will access the database to find the user balance and check if the user booking has enough credits to fulfill the book cost.
2. If the user has enough credits to pay the book, the system will change the user balance account in the database, where the cost in credits will change from available credit to blocked credit.
3. After making a successful booking by a user, the system will show a reservation request on the request menu of the driver client of the application.
4. If a reservation request is done, it will show to the driver application.
   1. A request for a ride has been made by a user.
   2. The system shows the rider a new request popup on the request menu icon in the main page.
   3. The driver enters the request menu and the system shows all reservation requests pending for the driver.
   4. The driver accepts the request. The system adds the user to the corresponding rides in the database.
   5. The system generates a booking voucher with a unique code and sends it to the passenger.
5. If the driver does not accept, the passenger credit that has been blocked for that ride will be unblocked.
   1. If a reservation request is done, it will show to the driver application.
   2. A request for a ride has been made by a user.
   3. The system shows the rider a new request popup on the request menu icon in the main page.
   4. The driver enters the request menu and the system shows all reservation requests pending for the driver.
   5. The driver rejects the request. The system removes the request and unblocks the previously blocked money of the passenger in the database.
6. If there are not enough credits, the user cannot book the ride(s)
   1. The user is authenticated
   2. The user access a ride detail page
   3. The user tries to book a ride with the book ride button.
   4. The system checks in the database if there is enough credit.
   5. If there is not the system would not allow the booking. Feedback of the cancellation of the process is shown in the user interface.

* **FR-RSB-3.4:** The system will indicate the occupation of the booked seat(s) in the ride.

1. When a user sees a ride detail screen of a specific ride, the occupation of the seats will be shown.
2. When a book is validated, the system will add the user to the ride.
3. The system will check for the number of users of a ride on the database and will display them showing some visual feedback on how many seats are left on the ride detail page.

* **FR-RSB-4:** Passengers may cancel any reservations made before the beginning of the ride. A passenger can press the cancel button on the details of the my rides page before the beginning of the ride.

1. The user accesses the my rides button in the main page.
2. The system shows all current rides booked or done by the user.
3. The user selects the ride desired to cancel. The system displays the ride details with a cancel button. The user can press the cancel button.
4. The system will access the database to change the user balance and the available seats.
5. If the cancellation has been made within 24-3 hours before the ride a 30% of the price would be not returned.
6. If the cancellation has been made within 3-0 hours before the ride there would not be a refund.

* **FR-RSB-5:** Passengers cannot reserve seats on different rides that overlap in time.

1. A passenger reserves a seat on one ride at a specific hour.
2. The passenger tries to reserve another seat on another ride in that same hour.
3. The system cancels the operation and shows to the user some feedback response indicating why the procedure could not be achieved.

* **FR-RSB-6:** Passengers will be able to check the rides they have reserved (both those from the past and the ones for the future)

1. User is authenticated.
2. The user presses the my rides button in the main page.
3. The system displays past history of rides. A button is also displayed to change to the future history rides.
4. The system will display an icon next to each future ride showing the state of the book (pending, accepted or rejected).

* **FR-RSB-7:** Riders will receive notifications related to bookings.

1. Drivers can change the details of the rides or cancel the ride. For changes that affect the ride such as changing the date the users will get a notification.
   1. If the ride is canceled by the driver, the system will automatically refund the passenger’s credits and emit a notification.
2. For recurring journeys, passengers will be notified before rides that a scheduled ride is approaching.
3. When the driver accepts the booking request.

* **FR-RSB-8:** Drivers will receive notifications related to bookings.

1. When a request for a ride has been made by a user.

#### **2.2.1.4. Payment Management - PM**

* **FR-PM-1:** Registered users will have a virtual payment account (VPA) associated with their identity.

1. Registered Users have a profile section with the VPA settings menu
2. The sub-section has a form for non configured accounts
   1. The form asks for the credit card number, expiration date, card pin
   2. The user has to validate its identity
   3. The card remains set for future payments in time
3. Accounts with already configured VPA will have an option to add more cards

* **FR-PM-2:** VPAs will initially have a balance of 0 credits. One VPA credit is equivalent to €1. Credits can be divided into cents.

1. The User has a balance section on his profile
2. The current credit is displayed in units and cents

* **FR-PM-3:** VPAs will manage available credits

1. Available credits are the amount of credits that the user can use at a given

moment.

1. Blocked credits are user’s credits reserved by the system as a guarantee of

payment for booked rides. Blocked credits cannot be used by the user unless

they are unlocked.

1. Total credits are the total amount of credits associated with the account at a

given moment (the sum of available + blocked credits).

* **FR-PM-4:** Authenticated users can transfer credits to their VPA from their bank account through an operation by means of virtual POS (online payment).

1. The user has a “Deposit” option on its balance section.
2. The user enters the desired amount to add to the account.
3. The system proceeds with the POS tool.
4. Transferred credits will be added to the available ones.
   1. The new credit is computed and displayed.

* **FR-PM-5:** When users book a ride, the system will block at the VPA the number of credits equivalent to the cost of the ride.

1. The user has enough credits to book a ride.
2. The system checks that the credits required are sufficient.
3. The credits are blocked at the user VPA.
4. The blocked credits are reflected in the user balance section.

* **FR-PM-6:** Users will be able to request the reimbursement of all or part of their "available" credits in their VPA to a bank account.

1. The user has a “Withdraw” option on its balance section.
2. The user enters the desired amount to subtract from the account.
3. The system proceeds with the reimbursement module.
4. In case of success, the system will deduct the reimbursed amount of available

VPA credits.

* 1. The new credit is computed and displayed.
* **FR-PM-7:** When a driver confirms that he/she has made a ride with a passenger, the transferred credits will be added as available credits of the driver and will be deducted from the blocked credits of the passenger.

1. The driver confirms a drive with a specific passenger is over.
2. Credits blocked from the passenger VPA are transferred to the drivers VPA.
3. New balances are computed in both accounts.

* **FR-PM-8:** When a ride is canceled, the corresponding blocked credits are refunded to its respective owners (unblocking them, subtracting them from the blocked credits and adding them to the available credits).

1. The driver cancels a ride
   1. Some passenger cancels a certain ride
   2. The passenger specifies the reasons to cancel the ride
2. The credits are unblocked and remain in the passengers VPA
3. The new balance is re-computed

### 

### **2.2.2. Non-Functional Requirements - NFR**

#### **2.2.2.1. Operational requirements**

* **NFR-1:** There will be only one application usage profile. This user can be at the same time both rider and driver, switching between them when required with such provided options.

#### **2.2.2.2. Security requirements**

* **NFR-2:** The payments will follow an external POS protocol and format

#### **2.2.2.3. Portability requirements**

* **NFR-3:** The project must be deployable mainly on mobile devices

#### **2.2.2.4. Verification/Identification requirements**

* **NFR-4:** Only members of the UAM will be able to sign up for this service.

1. The app will check the validity of a provided UAM-ID

# **3. System Interaction. Conceptual and Visual Design**

For the Conceptual and Visual design of the application different subcategories will be explained showing the different functionalities of the application. For each category functionality will be explained by text referencing visual representation by mockups.

## **3.1 Login & User Identification**

The first page seen by the user will be the login page (Image I) where the user will be able to enter his/her corresponding data to login to the application and access the other functionalities. When the user logins it will be moved to the main page (Image III).

If the login data is not correct a login page with some feedback will be shown (Image II).

|  |  |
| --- | --- |
| Image 1: Login Page | Image 2: Invalid Login |

## 

## **3.2 Main Page**

The main page of the application (Image 3) consists of a map (provided by an external partner) with a search bar to find specific locations. On the bottom part of the screen a slide button will be displayed to choose between a map with all rides shown and a map with only user booked rides shown (“my rides”). On the right side a “+” button is displayed to create a new ride (Image 9). The options button (third in the left) will show the options menu (Image V). The next button is the rides button (Image 4), where after pressing it the user will be able to choose between the “my rides” menu (Image 20) or the “all rides” menu (Image 11). Not to be mistaken with my rides and all rides options for the map. The next button for the requests menu (Image 46), where the user will be able to check the rides requests. The last button is for the messages page (Image 49).

|  |  |
| --- | --- |
| Image 3: Main Page | Image 4: Rides Button |

The options menu (Image 5) will show different options to choose from. The Balance option will redirect to the balance screen (Image 39). The policy button will show the company policy (Image 6). Help button will show the help menu (Image 7). A logout button will also be shown to the user. If pressed the user session will end and the user will be redirected to the login screen (Image 1).

|  |  |
| --- | --- |
| Image 5: Menu | Image 6: Policy |

The help menu (Image 7) will have a Frequently Asked Questions button to redirect the user to an external source provided by the host to show typical problems the user may encounter. A “delete my account “ button will also be shown to provide the user with the decision to clean all the collected user profile and data and be returned to the login page (Image 1). Before the account is deleted, a security message will be displayed to confirm the choice (Image 8). Some links to external social media to contact the host will be shown in this page to help the user communicate with the service provider.

|  |  |
| --- | --- |
| Image 7: Help | Image 8: Delete Account |

## **3.3 New Ride**

When the user chooses to create a new ride the create ride page will be shown (Image 9). The user will need to fill all the expected data to press the publish button. The user will need to specify a price, number of available seats, optional comments, a destination (if the origin is UAM) or an origin (if the destination is UAM) by the destiny confirmation menu (Image 26) accessed by pressing the right arrow. The user can also mark the ride as recurring and select the days it will be repeated. When the Publish button is pressed a confirmation will be shown (Image 10). The back arrow button will bring back the user to the main page (Image 3)

|  |  |
| --- | --- |
| Image 9: Create Ride | Image 10: Ride Published |

## **3.4 Rides**

The rides menu (Image 11) will display a list with all the rides available for the user. On the top of the page the user will be able to change between rides from the UAM or rides to the UAM. At the button right corner a filter option will be shown, and if pressed (Image 12) different filters will be shown like smoking, date, time or car type. If there are no rides, feedback will be displayed (Image 13). If a ride is pressed a detail menu will be shown (Image 14)

|  |  |  |
| --- | --- | --- |
| Image 11: Rides | Image 12: Ride Filters | Image 13: No Rides |

The detail page (Image 14) will display all the necessary details to book a ride. Origin/destination, driver, date, time, price, comments, filters, available seats and a sum up of the route. Current user credits will be shown on the top right corner. A reserve seat button will be shown only if there are still free seats. If pressed, it will show the user feedback of the reservation (Image 15). If the reserved ride is recurrent the application will ask the user if he/she wants a recurrent booking (Image 16). If so, a menu will be displayed to choose the recurrent booking options (Image 17). If the user already has a book on that date and time, the book will fail showing feedback (Image 18). If the user doesn’t have enough credits for the book, the reservation will fail with the corresponding feedback (Image 19). There is a button with a text bubble in the right part of the screen to access the new message menu (Image 51).

|  |  |  |
| --- | --- | --- |
| Image 14: Ride Details | Image 15: Ride Reservation | Image 16: Ride Reservation Recurrent |

|  |  |  |
| --- | --- | --- |
| Image 17: Ride Reservation Recurrent II | Image 18: Ride Reservation Concurrent | Image 19: Ride Reservation No Founds |

## **3.5 My Rides**

The my rides menu (Image 20) shows the rides the user has booked. A slider can be used on the top of the screen to change between past rides (Image 20) and future rides (Image 33). If there are no past or future rides, feedback will be shown (Images 21 & 22 respectively). Rides requested need to be approved by the driver. If the driver has denied the request, a cross will be displayed next to the ride. If it was approved a tic will show. If the request is pending approval, a yellow clock will show. If you click on an approved ride the detail menu will be shown (Image 27**).**  If a pendant of approval is clicked instead, a variation of the detail menu will be shown (Image 30). If the ride is completed it can be reviewed with a star system (1 is a terrible ride 5 a perfect ride).

|  |  |  |
| --- | --- | --- |
| Image 20: My Rides | Image 21: No Past Rides | Image 22: No Future Rides |

The edit ride menu (Image 23) will let the creator of a ride modify the ride characteristics (price, seats, smoking, comments, origin, destination, date, time, vehicle and recurrency). The option to cancel the ride will also be available. When changes are applied a pop up will ask if the changes are unique to the current ride or for all the rides (Image 24). Some modifications will not be possible (such as the change of the amount of seats) if seats have already been booked (Image 25).

|  |  |  |
| --- | --- | --- |
| Image 23: Edit Ride | Image 24: Edit Ride II | Image 25: Edit Ride III |

|  |
| --- |
| Image 26: Confirm Destiny |

The ride details menu (Image 27) will show the details of the booked ride. By pressing on the cancel button a pop up will be displayed to confirm the decision (Image 28) and if so, some feedback of the operation will be shown (Image 29). The page will show the canceled ride (Image 34).

|  |  |  |
| --- | --- | --- |
| Image 27: Cancel Ride | Image 28: Cancel Ride II | Image 29: Cancel Ride III |

The ride details for a non confirmed request (Image 30) will have a cancel request button instead of a cancel ride button. The cancellation will be automatic with no need for confirmation. The page will show some feedback (Image 34).

|  |
| --- |
| Image 30: Cancel Request |

When the time of the ride arrives, a confirmation code pop up will show on the driver ride detail page (Image XXXI) to enter a code shown to the passenger ride detail page (Image XXXII) to confirm the ride.

|  |  |  |
| --- | --- | --- |
| Image 31: Enter Code | Image 32: Confirm Code | Image 33: Future Rides |

When a ride is canceled the arrow at the bottom of the screen can be pressed to show the feedback menu ( Image 35) to write an optional explanation on why the ride was canceled.

|  |  |
| --- | --- |
| Image 34: Canceled Ride | Image 35: Canceled Ride Reason |

## **3.6 User Profile**

The user profile (Image 36) will show the score of the user as well as the full name, profile picture and age. Current rides held by the user will be displayed with useful information for each of them. If there are no current rides held by the user some feedback will be shown (Image 37). If the user clicks on a ride the edit ride menu will be shown (Image 23). On the bottom of the page a “Edit Profile” button is displayed and when pressed the user is redirected to the edit profile menu (Image 38).

|  |  |
| --- | --- |
| Image 36: Profile | Image 37: Profile With No Rides |

The edit profile menu (Image 38) will show different data for the user to fill or modify to have a complete and correct user profile.

|  |
| --- |
| Image 38: Edit Profile |

## **3.7 Balance**

The balance menu (Image 39) displays all the current credits of the user. Available credits show the credits available for the user to freely spend or withdraw, blocked credits show the credits retained by the application because a ride has been booked with them and total will add the sum of available and blocked credits. Payment methods (credit/debit cards) can be added/removed by pressing on the +/remove buttons on the corresponding section. If the “+” payment method button is pressed, the user will be redirected to a formulary to add a new card (Image 40). If the “Deposit” button is pressed the user will be redirected to the deposit page (Image 41). If the “Withdraw” button is pressed the user will be redirected to the withdraw page (Image 43).

|  |  |
| --- | --- |
| Image 39: Balance | Image 40: New Card |

The deposit page (Image 41) shows a menu to select a payment method and an amount to deposit. Some information about the new balance is shown, and when the button “Deposit” is pressed feedback of the completed operation will be shown (Image 42).

|  |  |
| --- | --- |
| Image 41: Deposit | Image 42: Deposit Completed |

The withdrawal page (Image 43) shows a menu to select a card and an amount to deposit. Some information about the new balance is shown, and when the button “Withdraw” is pressed feedback of the completed operation will be shown (Image 44). If the credit balance is not enough an error will be displayed (Image 45).

|  |  |  |
| --- | --- | --- |
| Image 43: Withdraw | Image 44: Withdraw Completed | Image 45: Withdraw Failed |

The request menu (Image 46) shows the driver the requests of the seats of the rides offered. These requests can be accepted by swiping the offer to the right (Image 47) or rejected by swiping to the left (Image 48).

|  |  |  |
| --- | --- | --- |
| Image 46: Ride Requests | Image 47: Ride Request Accepted | Image 48: Ride Request Rejected |

## **3.8 Messages**

The message menu (Image 49 shows all the possible chats to open or interact with the other users of the car. On the top left corner you can change between the riders page (Image 49), where the possible chats are the passengers of the user rides, and the passenger page (Image 50), to chat with the drivers of already accepted rides. Red bubbles are shown in both rides and specific users with a number to show the amount of unread messages. Pressing a user will open a chat with that current user (Image 52)

The new message menu (Image 51) shows the possible chats for a specific ride. If the ride has been fulfilled with everything done and paid a tic will appear on the right of the user and the user profile picture will be shown in gray scale. If not there will be a yellow clock indicating the ride is not yet fulfilled or something is not done yet, and the user will be able to press the profile to access that person specific chat (Image 52).

|  |  |  |
| --- | --- | --- |
| Image 49: Message menu | Image 50: Message menu 2 | Image 51: New message menu |

The chat page (Image 52) is used to chat with other app users. The current user profile of the other chat user is shown at the top, and messages are shown in bubbles in top to down chronological order, with the user messages in the right and the other interactor’s messages in the left.

|  |
| --- |
| Image 52: Chat |

# **4. Conclusions**

Lately we are living in uncertain times where big historic events seem to happen on a daily basis. This has made some people forget about an important problem that has been happening in our world for a very long time: Pollution. Nevertheless, the UAM has managed to bring back to the table such an important subject with yearly editions of the EcoUAM event (As mentioned in section 1.1). In the eleventh edition a carsharing proposal for the UAM community was awarded a first prize in that competition, and we, the PINGS Software development team, have been employed to carry out the design of the application.

The goal of this project was to design a software that would offer an alternative more conscious method of transport for the UAM community. To do so, some guidelines had been given to us by our client, the UAM. However the design and interactivity of the application were entirely left to our control. By analyzing our competitors and a performing few brainstorming sessions, we decided to end up with the design presented earlier (In section 3) where we made the application very intuitive to encourage the new generation to use it without needing any kind of previous preparation.We went through a few stages of design so we could polish the visual aspect of the application, ending up with the same functionality as the one exposed in the presentation but a brand new more clean interface. To do so, we have been working with the online application “frame.com” which offers a free plan that allows cooperative work for mobile application mockups.

The application designed is a move forward towards a necessary change that could not only benefit UAM and its community but also the world. It is undeniable that the world can not afford to keep on its path of pollution, a change is needed, and the sooner the better. The fact that a public institution such as the UAM is willing to create an application that encourages change will be a wake up call to other institutions or enterprises to also join the movement.

Young people are the future and what better way to make them conscious about climate change than proposing an environment-healthy solution to their every-day transport? This easy and intuitive application is not only for people that want a cheap way to get to their daily place of work or study but also a way to communicate to the youth and tell them that we believe in them to make the change the past generations were not able to make.

Another benefit of the application will be for social purposes. People from the UAM that use the app will find other people that work or study in the same place as them and will allow the UAM community to make connections between them so they can grow together.

This application also gives a chance to some people that need it, providing affordable transport to people that might not be well communicated with the UAM and can not drive. In a nutshell it makes public education a little bit more public.

Last but not least another benefit for the UAM will be a new line of loaned credit provided by the application. Since the payments work with internal money that you have to purchase before using and when a driver gets paid, he receives the internal money not real one, this will allow the UAM to manage the app generated money as if it was a bank, by using the money that all the users have introduced in the app as a line of credit and having a financial cushion for the withdrawals.

There are times when some things escape our control and the best way to handle them is to make room for someone else with more resources or expertise in that field to handle them. This is the case on some of the aspects of the proposed application (mentioned in section 2.1) which are:

* Creating a personal GPS service
* In-app calls
* Customer support
* Bonification credits/status

When this project was assigned to us, we performed a brainstorming session on what ideas could be implemented to make this application even better. We came up with a few good proposals of future features for the application:

* Written reviews for drivers and passengers
* Possibility to add friends in the app and see their rides
* In-app group chat with drivers and the other passengers
* Page that offers statistics on pollution (CO2 saved by using the app, rides you have created…)
* Feature to save a frequent filters preset of search
* Allow paying an extra fee for the driver to deviate from the route to drop you off or pick you up
* Alert for when a driver is posting a ride similar to an already existing one suggesting him to pick the one existing to save CO2
* Possibility to pay for more than one seat on the same ride

# 

# 

# 

# References

## Services and applications used

Editing:

Website Name: Cacoo by Nulab

URL: <https://cacoo.com>

Last access: 23/02/2022

Website Name: Framer

URL: <https://www.framer.com>

Last access: 28/02/2022

Website Name: Google Fonts

URL: <https://fonts.google.com/>

Last access : 22/02/2022

Website Name: Google Docs

URL: <https://docs.google.com/document/>

Last access: 02/03/2022

Team Organising:

Website Name: Discord

URL: <https://discord.com>

Last access: 22/02/2022

Website Name: Trello

URL: <https://trello.com/>

Last access:

Website Name: Google Drive

URL: <https://drive.google.com/drive/>

Last access: 02/03/2022

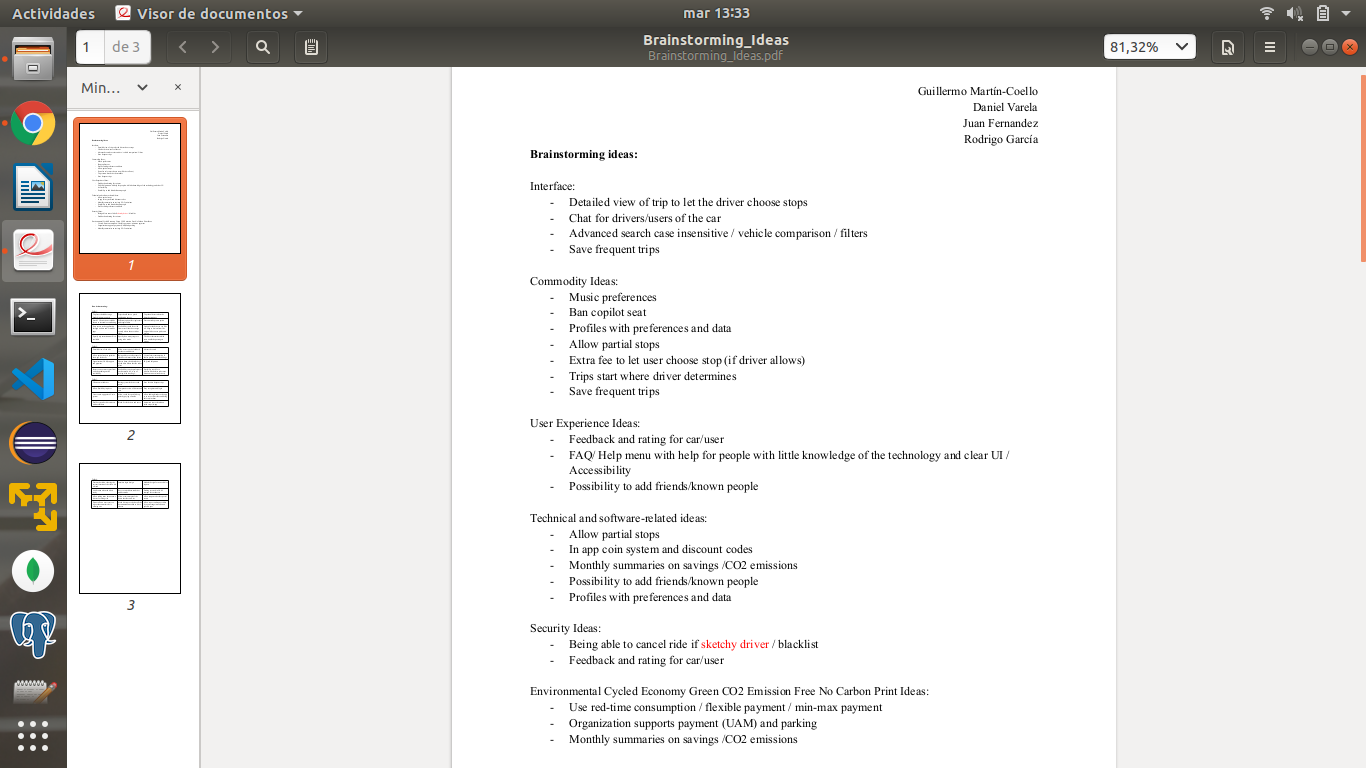
## Documents used

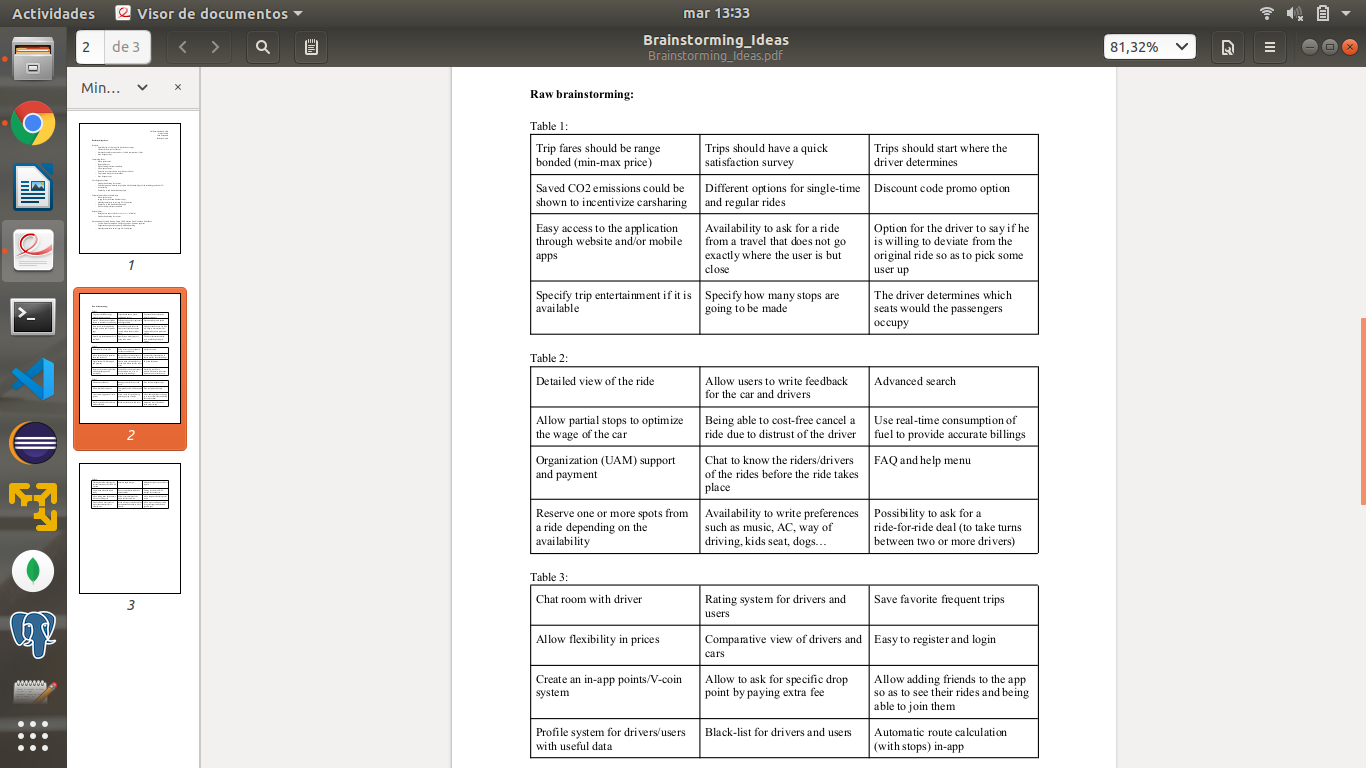
Unidad 4 - Análisis.pdf - Authors: INGS teacheing department.

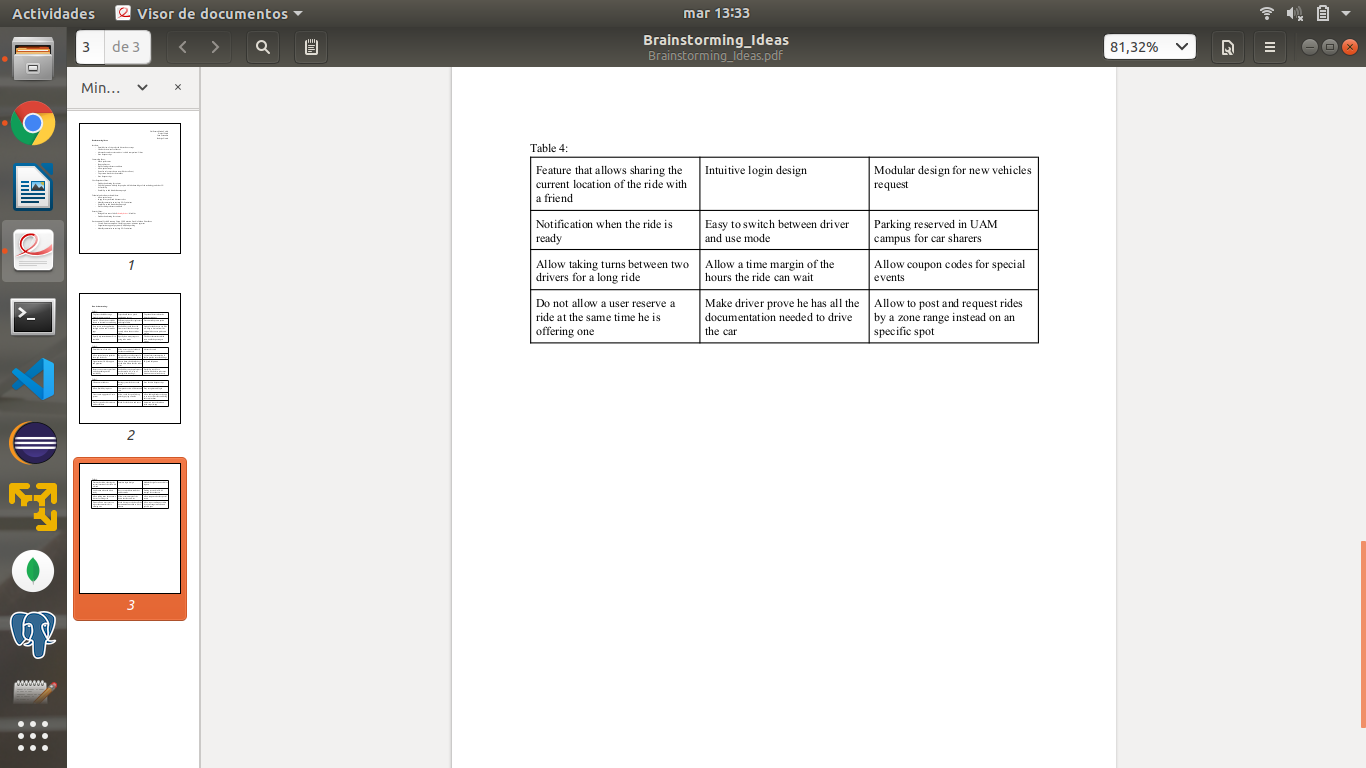
SEPRO\_PRACTICAL\_ASSIGNMENT1\_2021-2022\_Sections.pdf - Authors: María Elena Gómez.

# **Appendices**

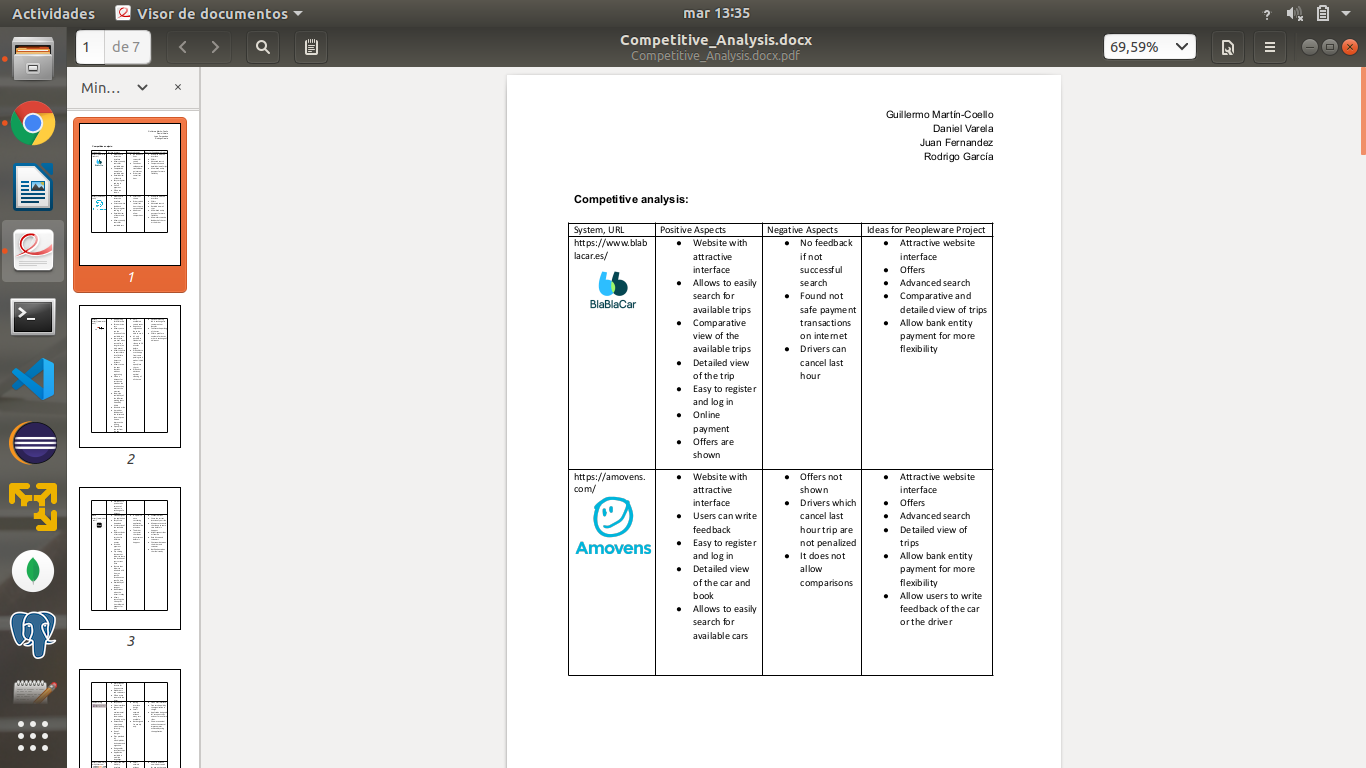
## **Appendix 1: Brainstorming**

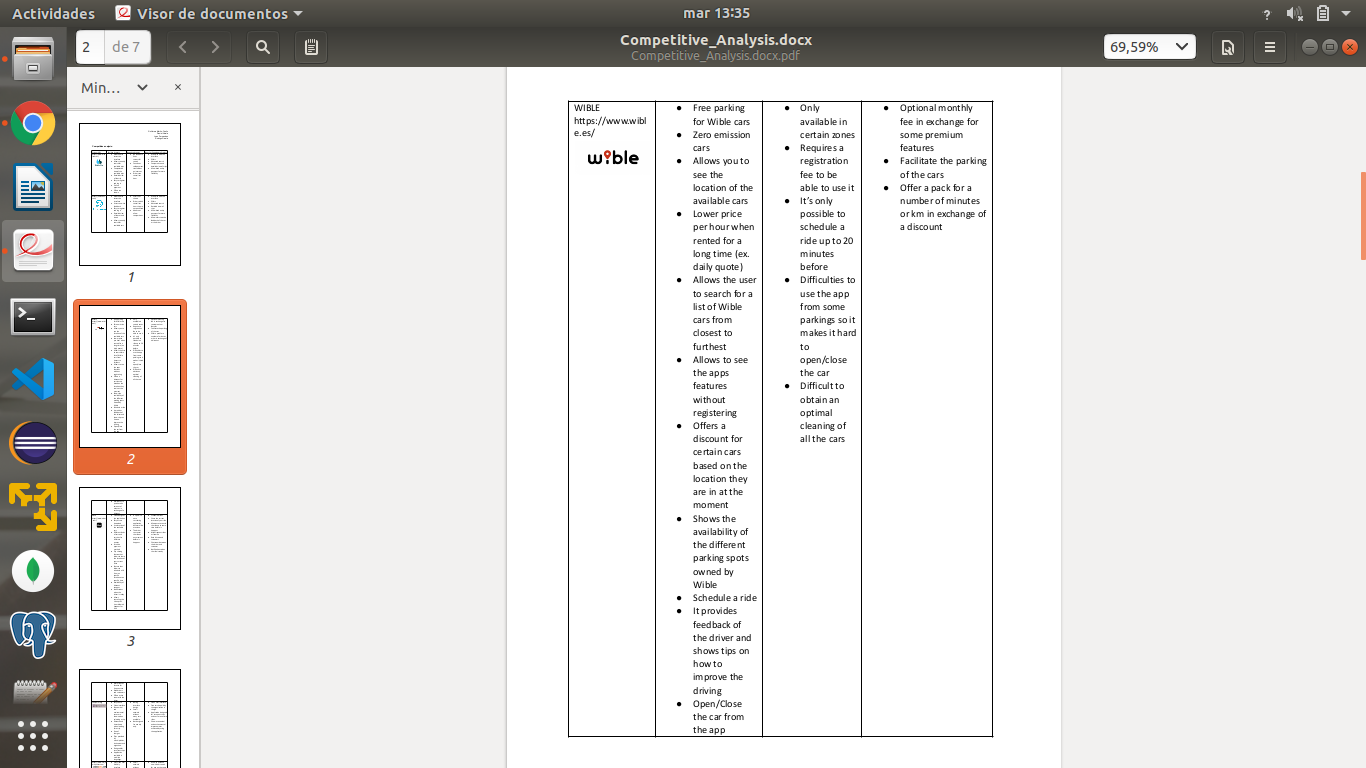
****

****

****

## **Appendix 2:Competitive analysis**





## 

## 

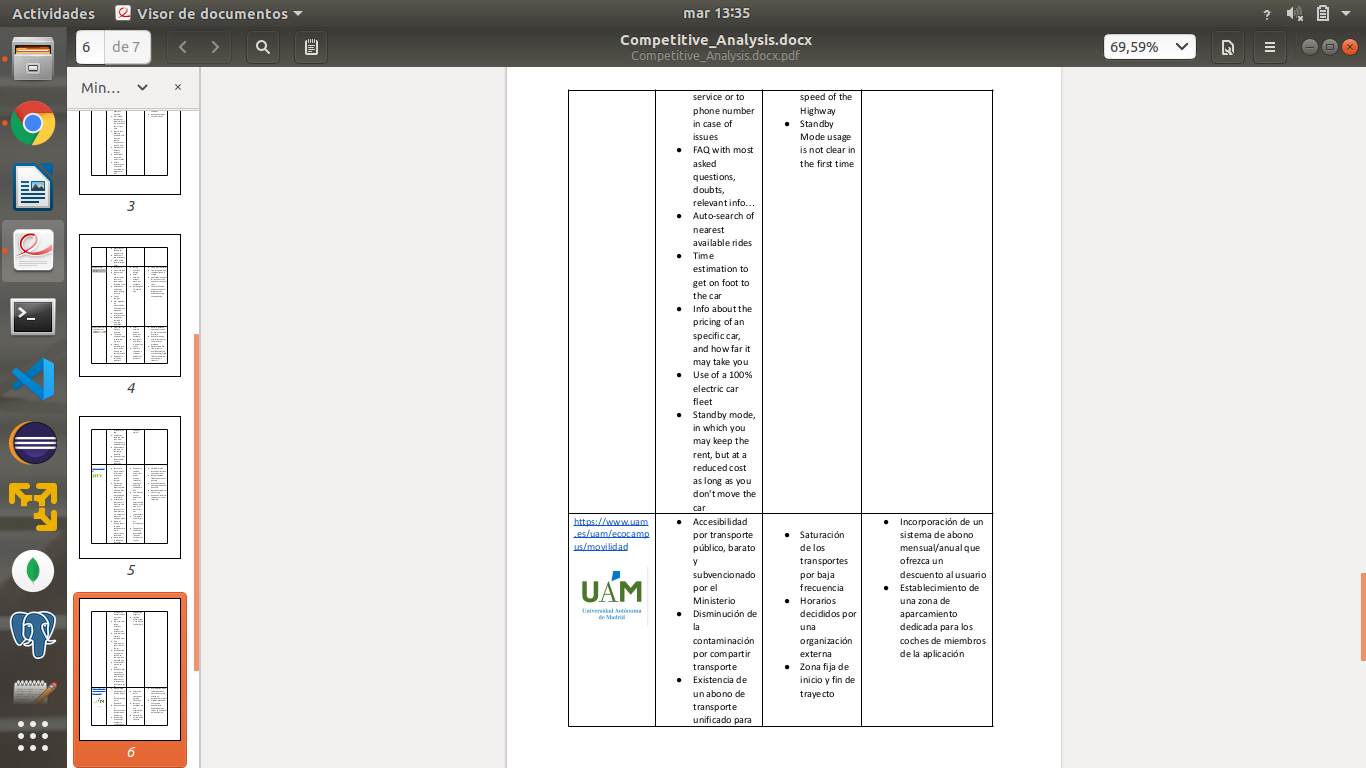
## 

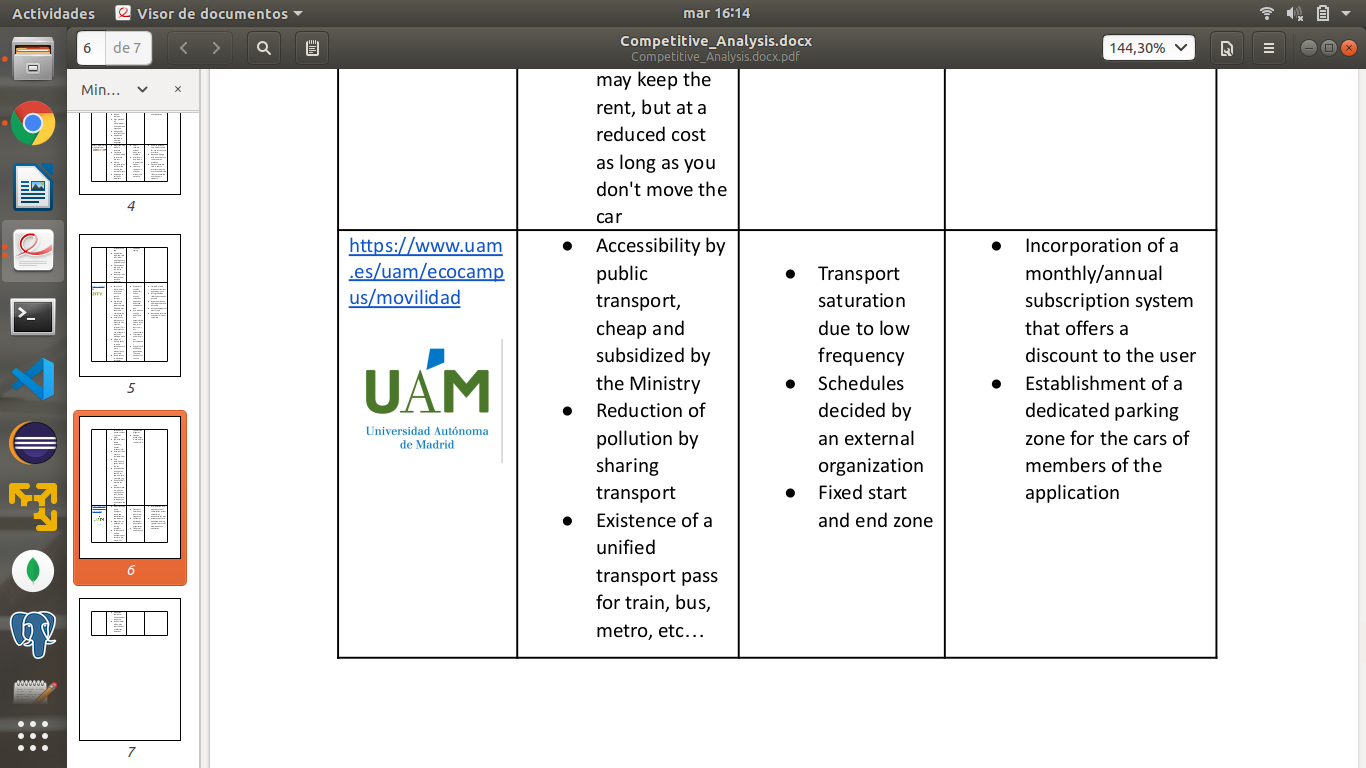
## 

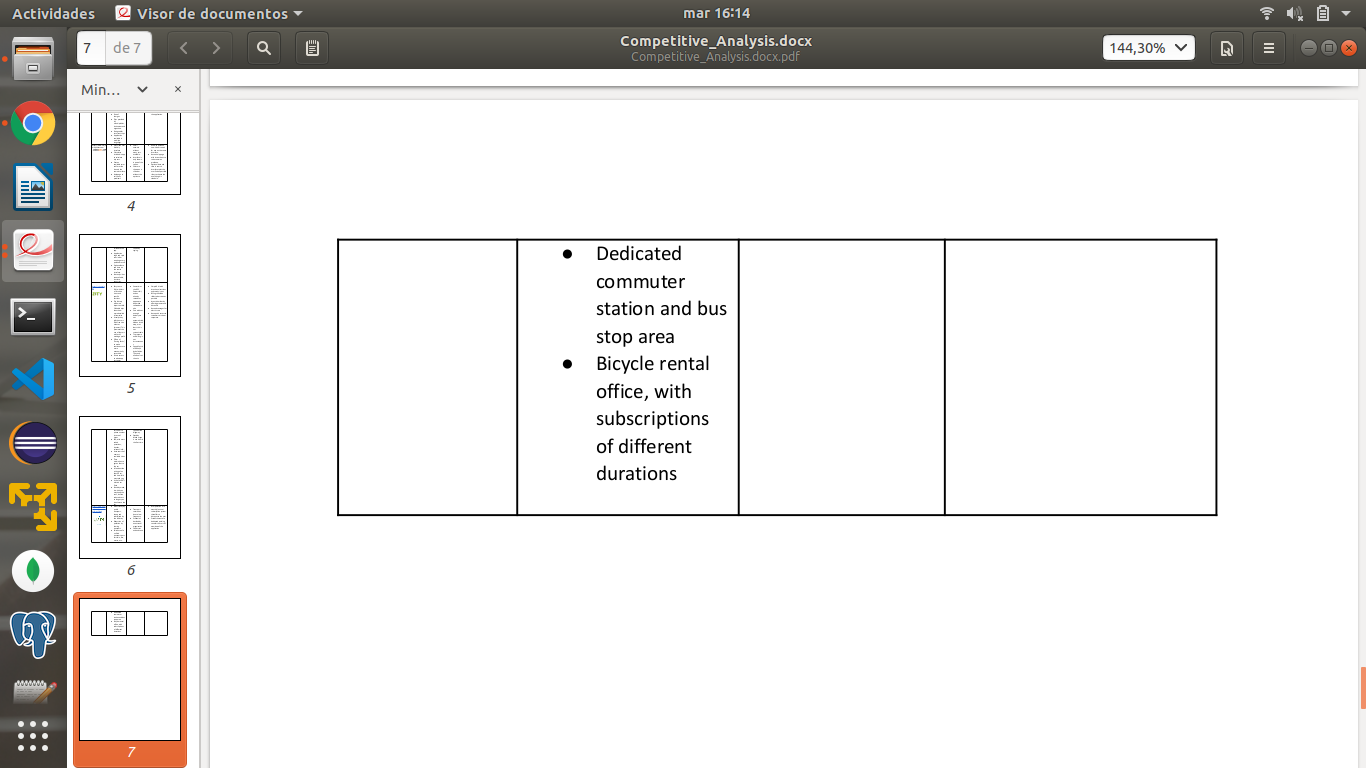
## 

## 

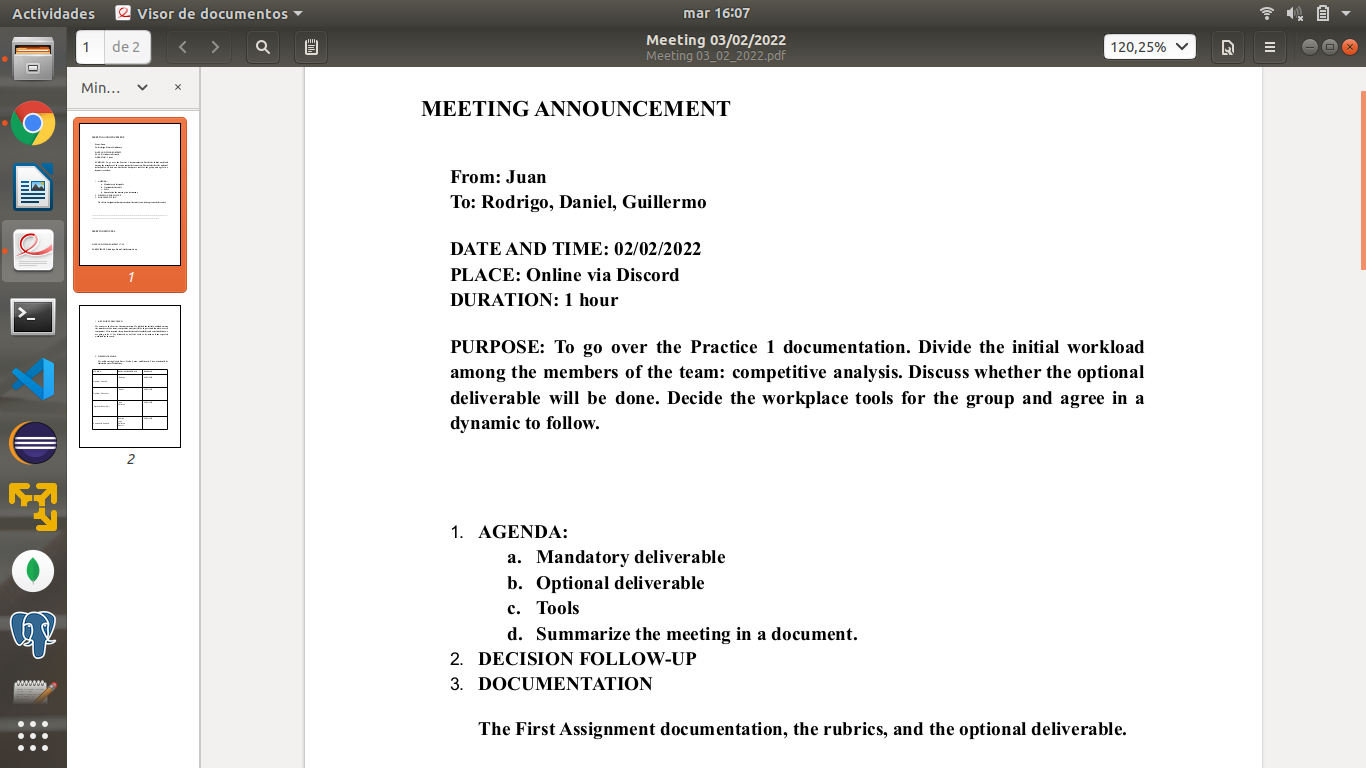
## 

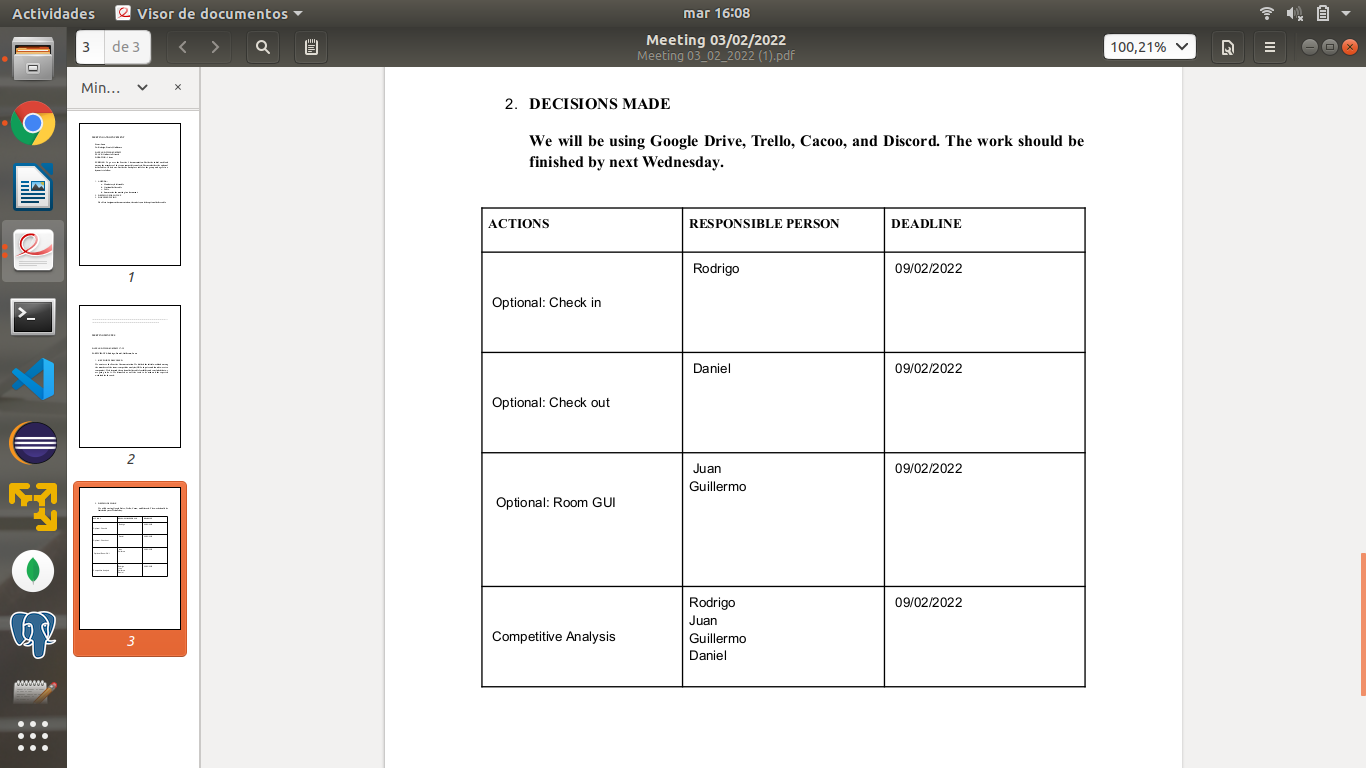


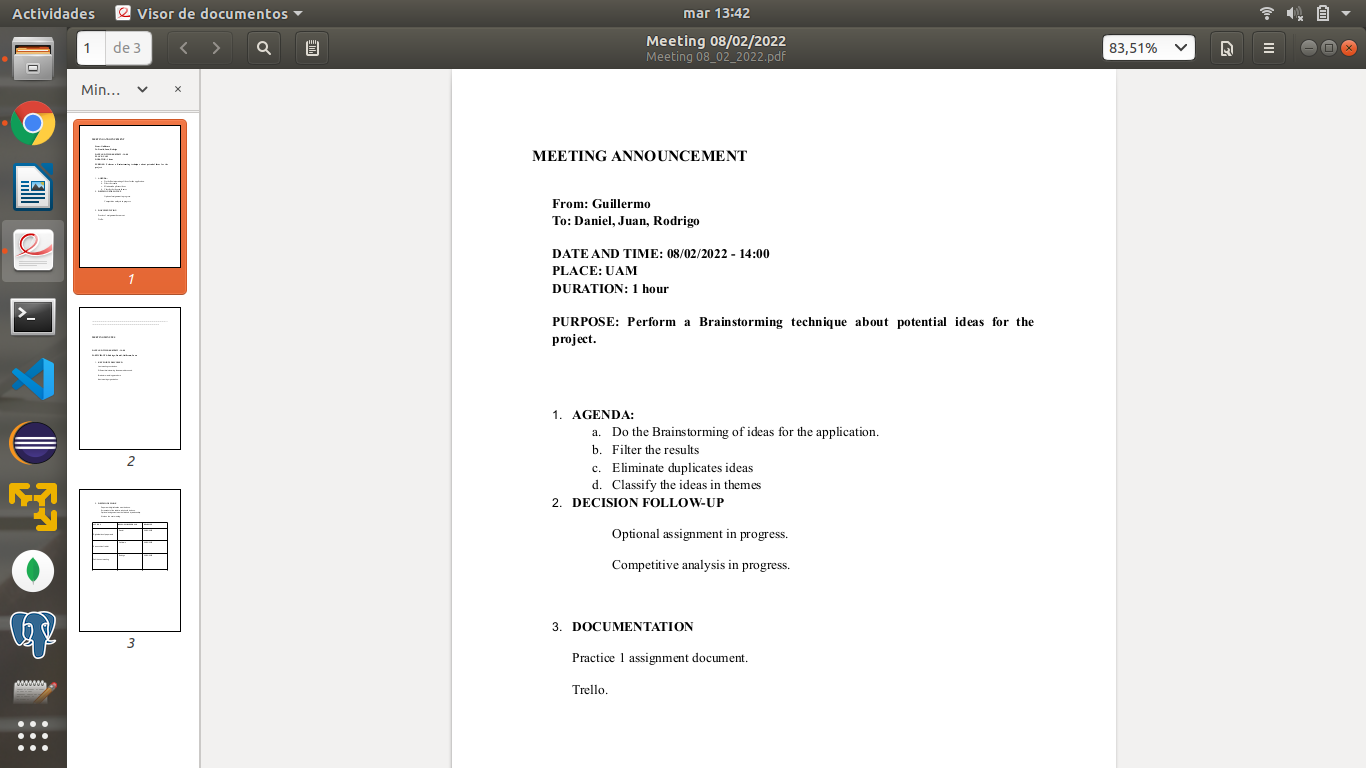


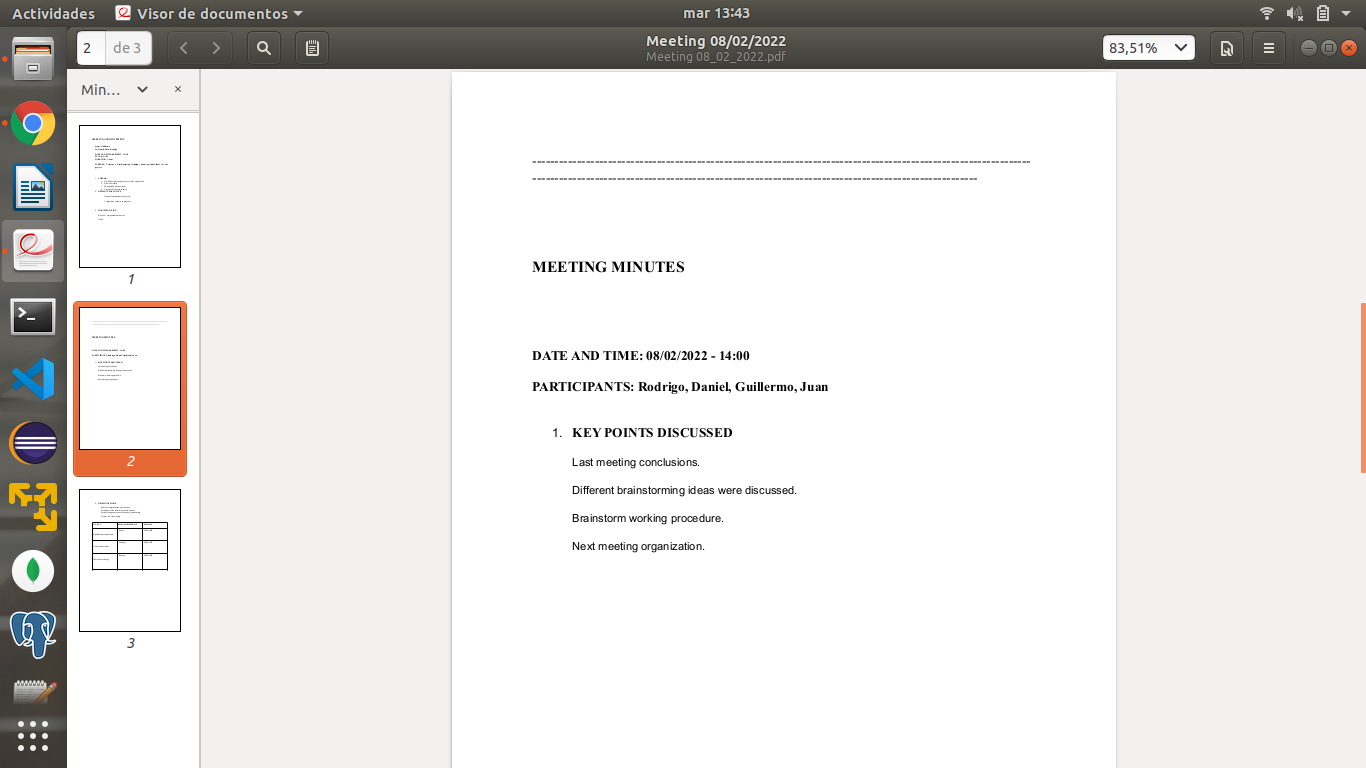


## **Appendix 3: Meeting minutes**



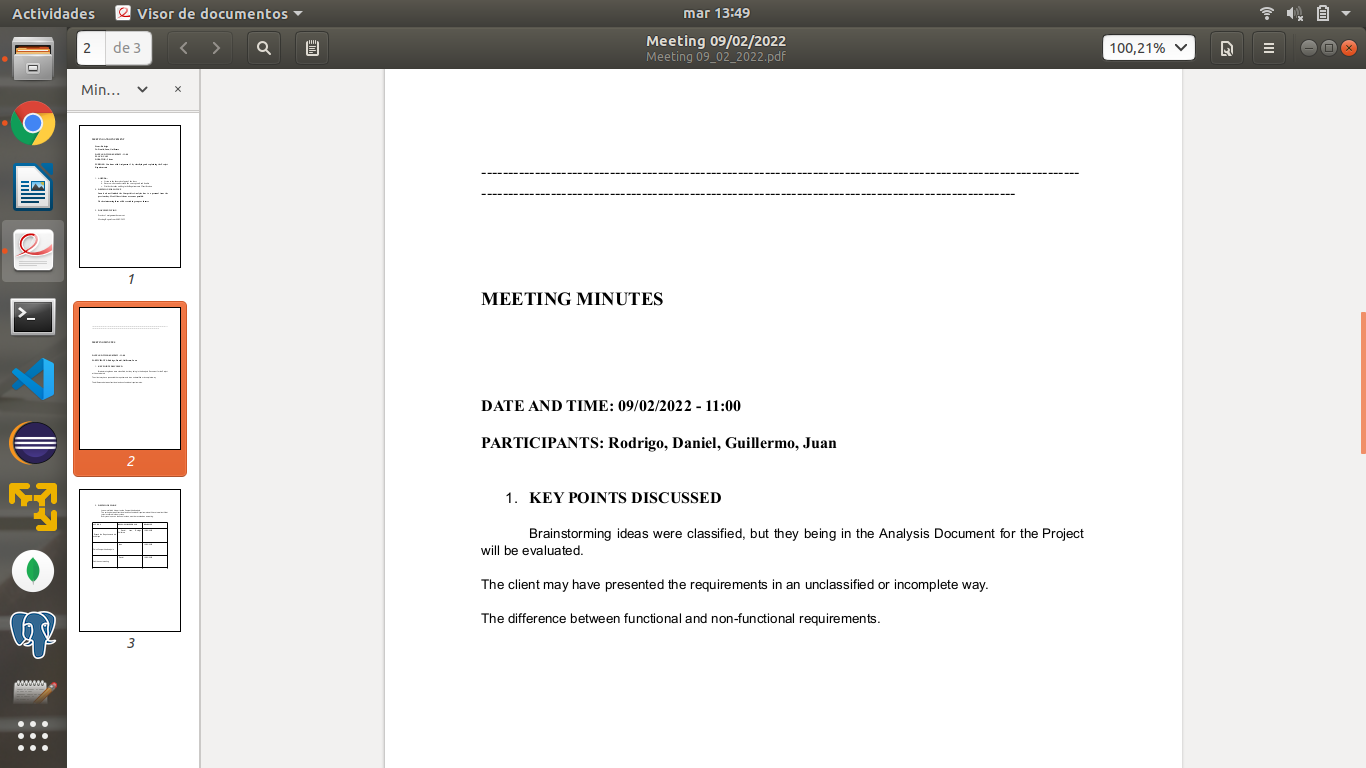


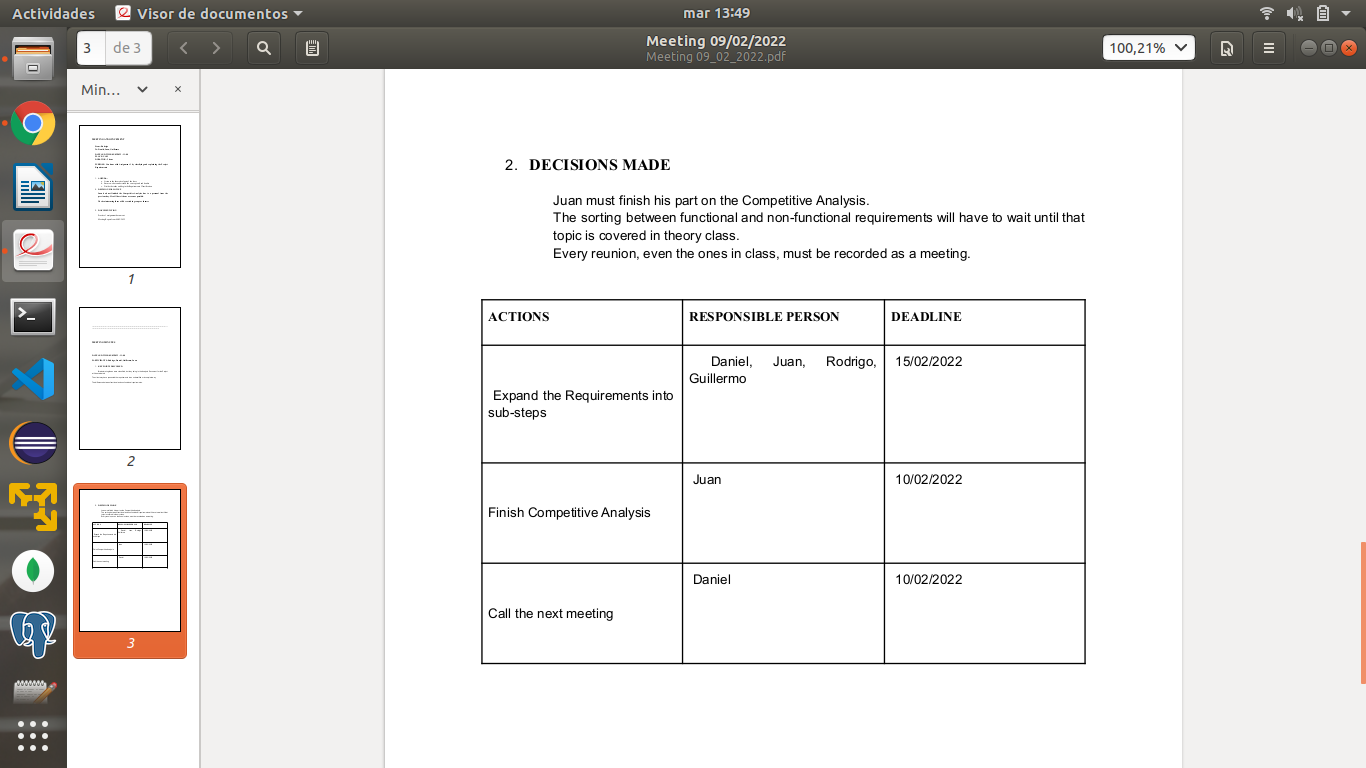


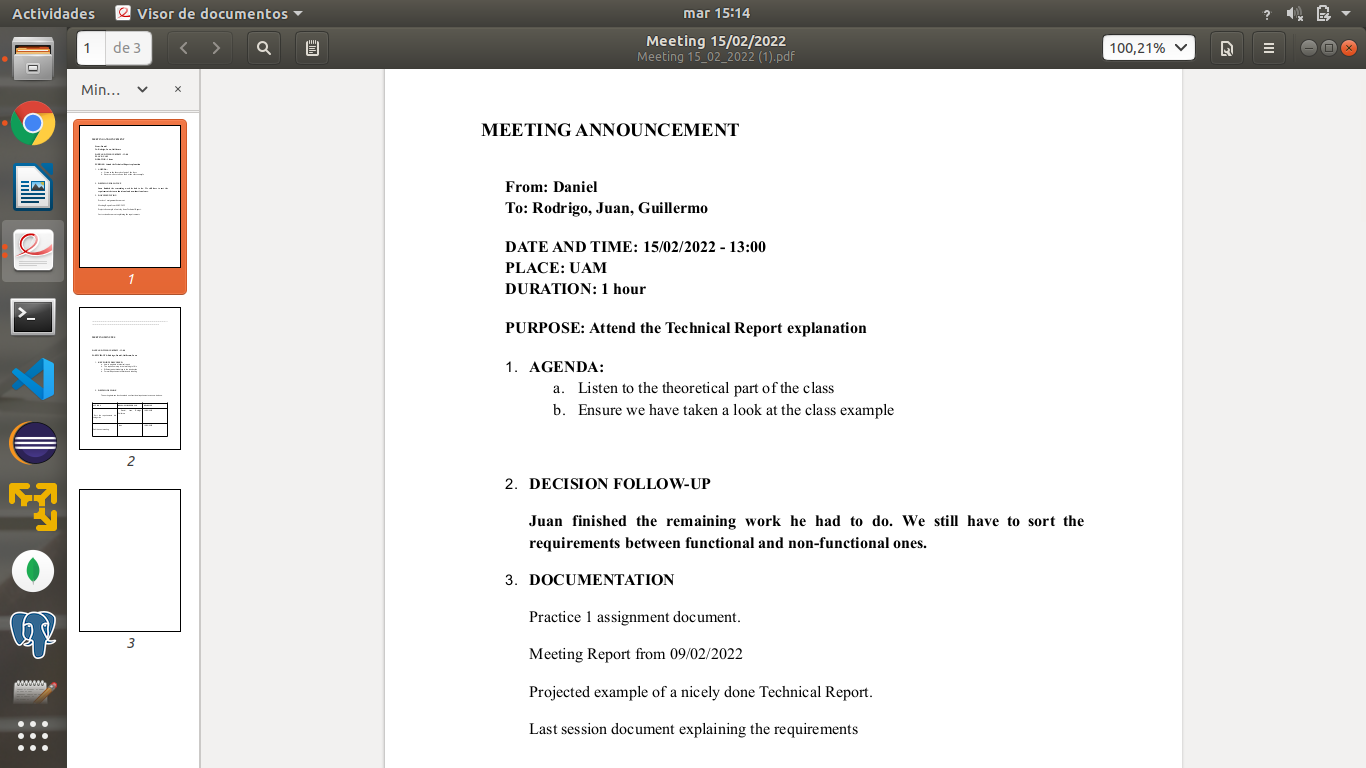


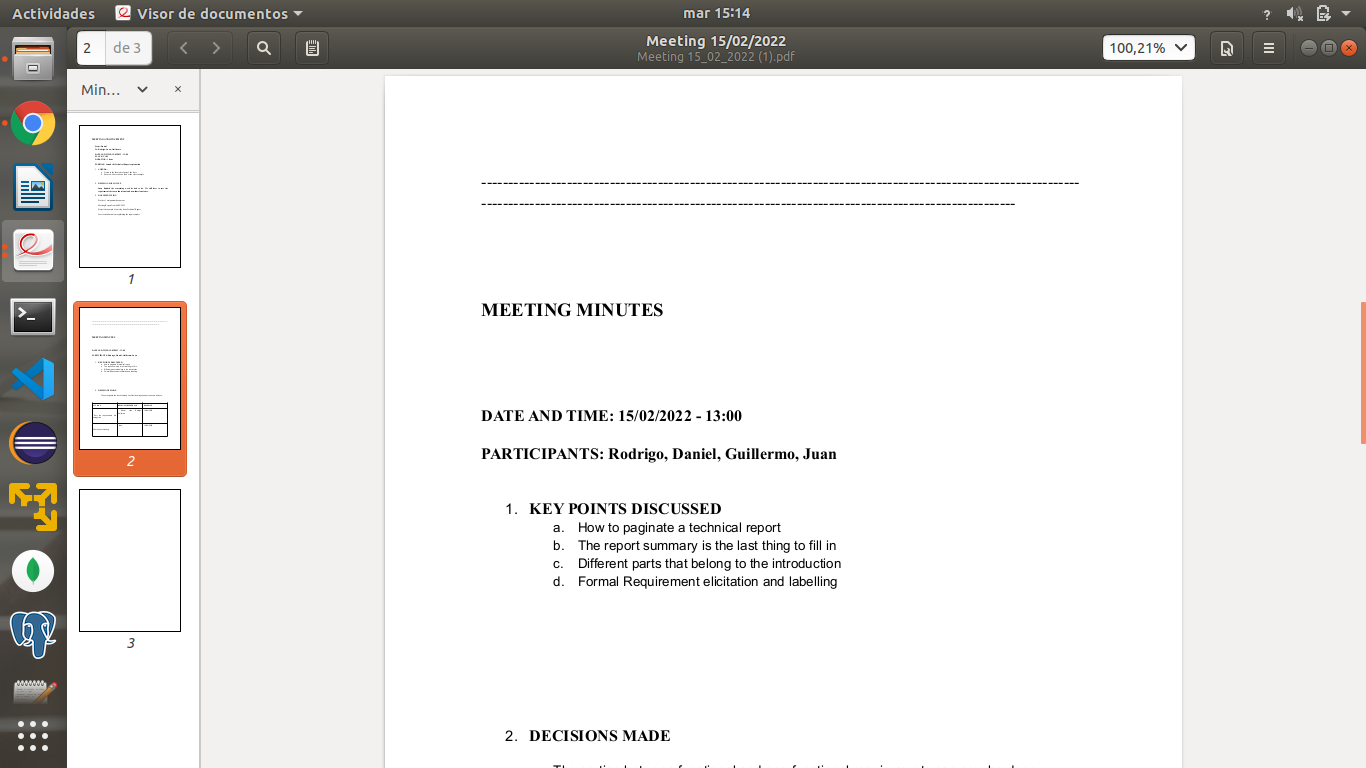


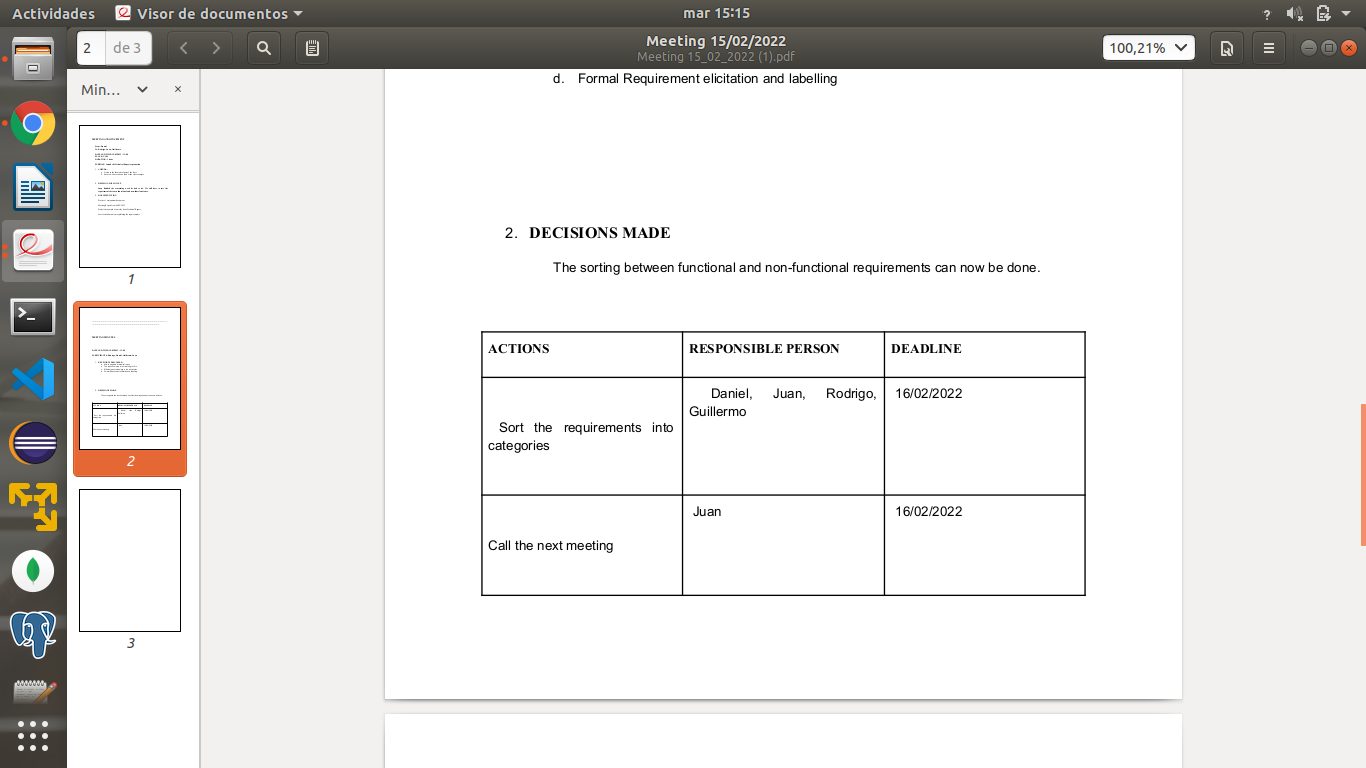


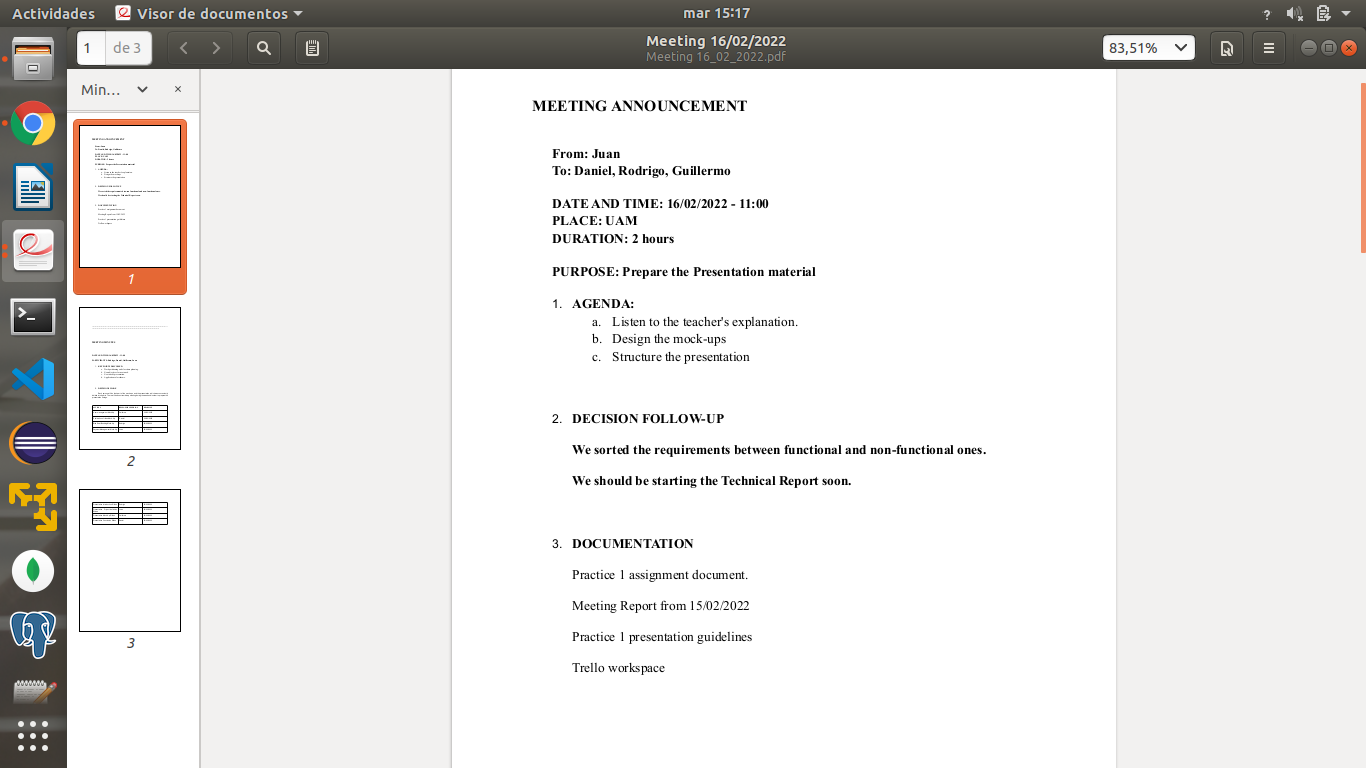


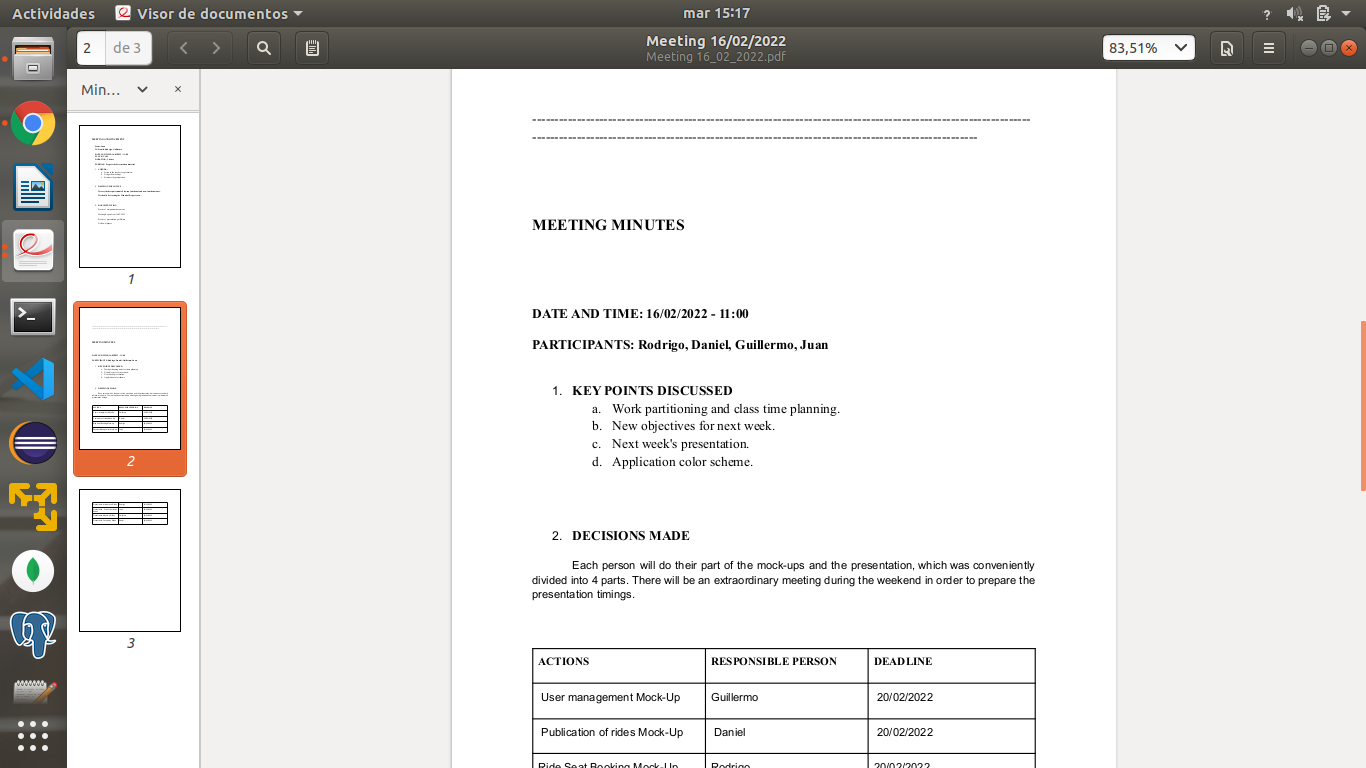


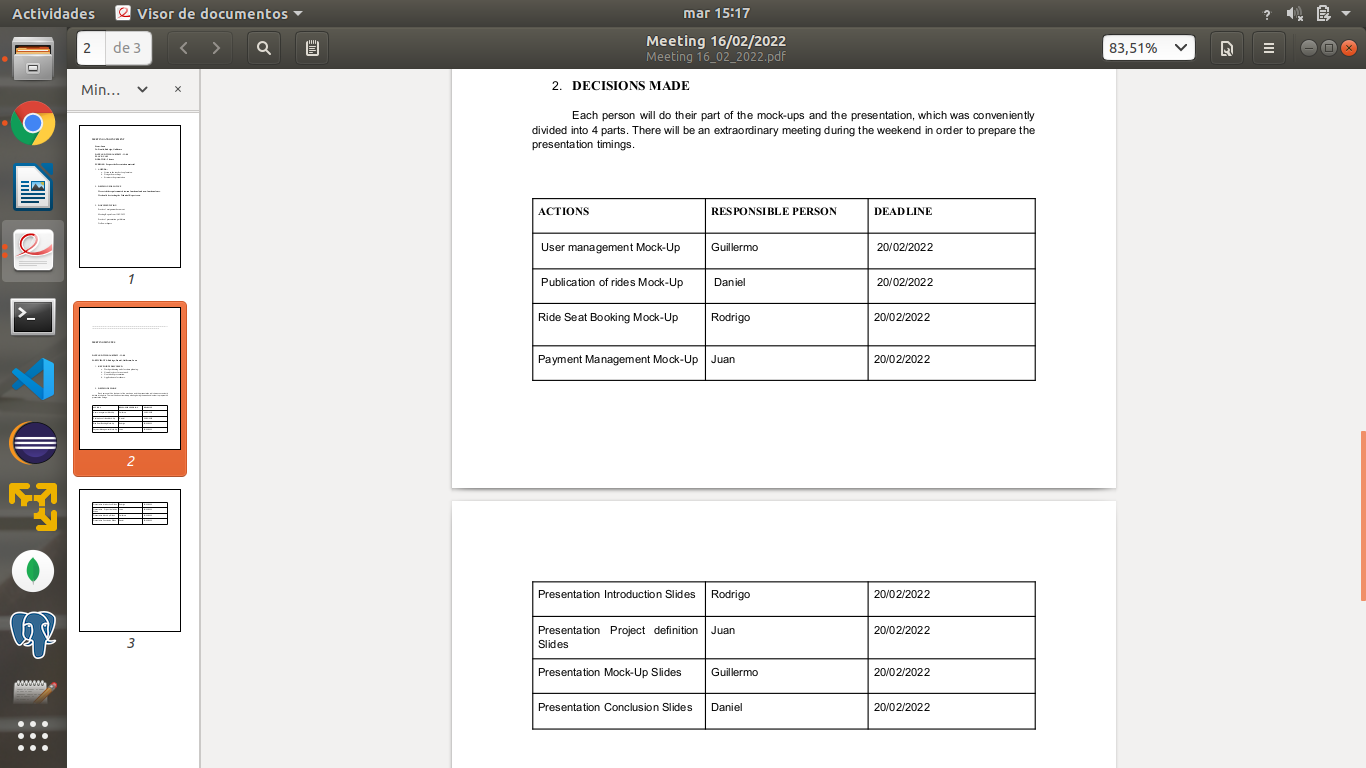


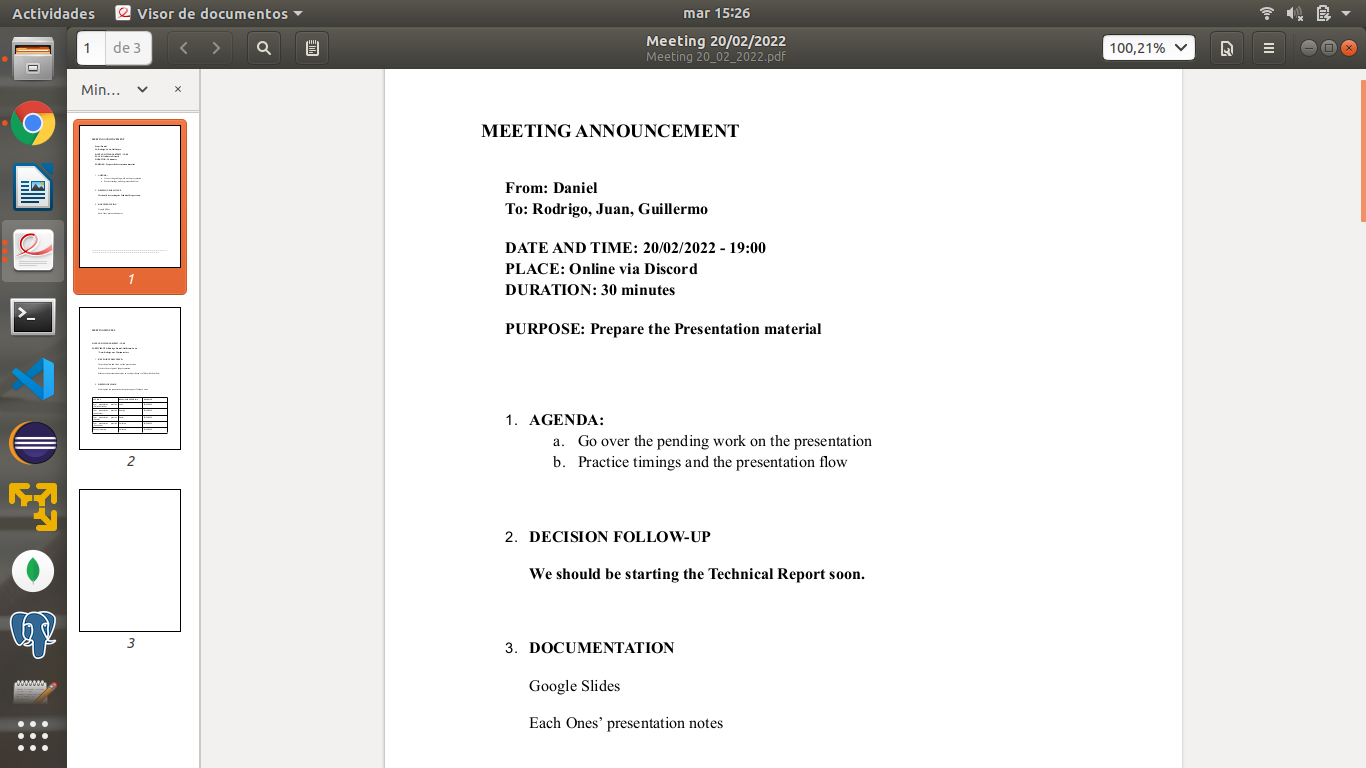


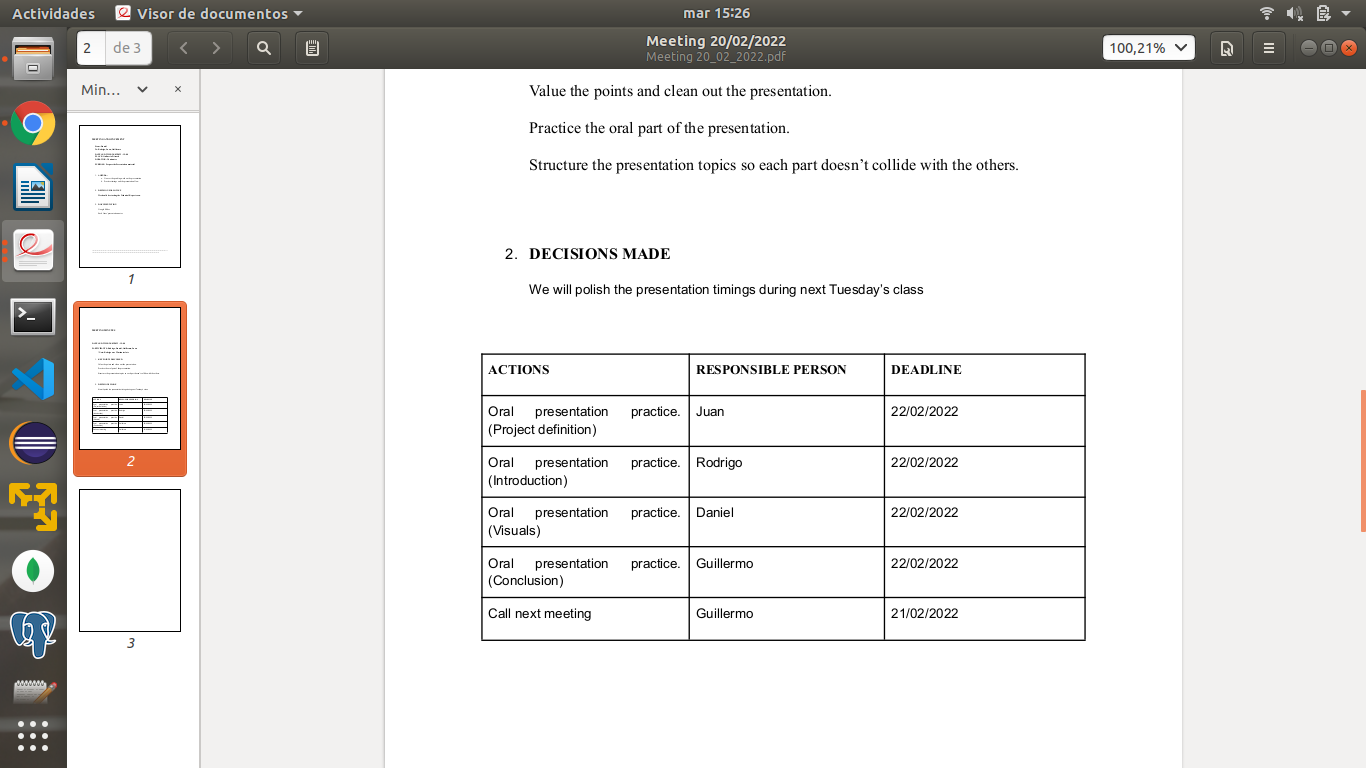
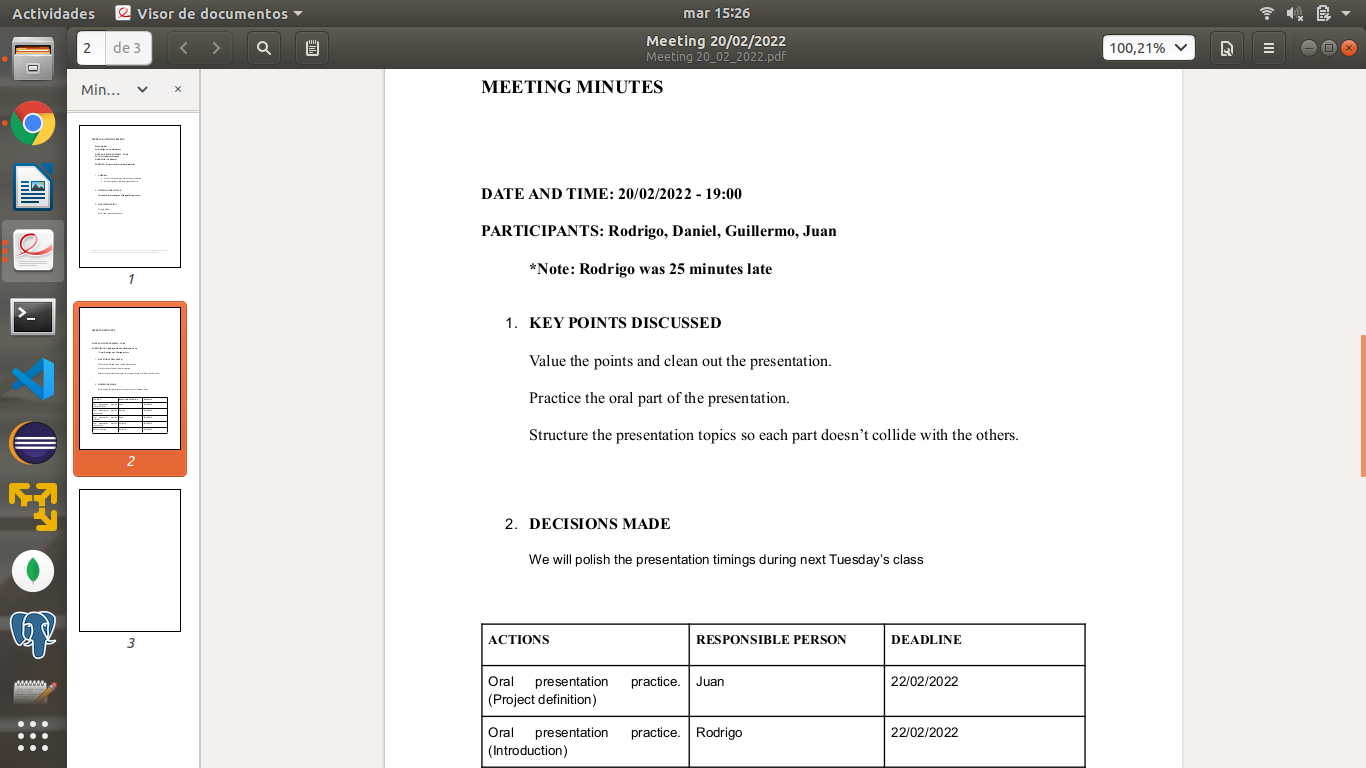


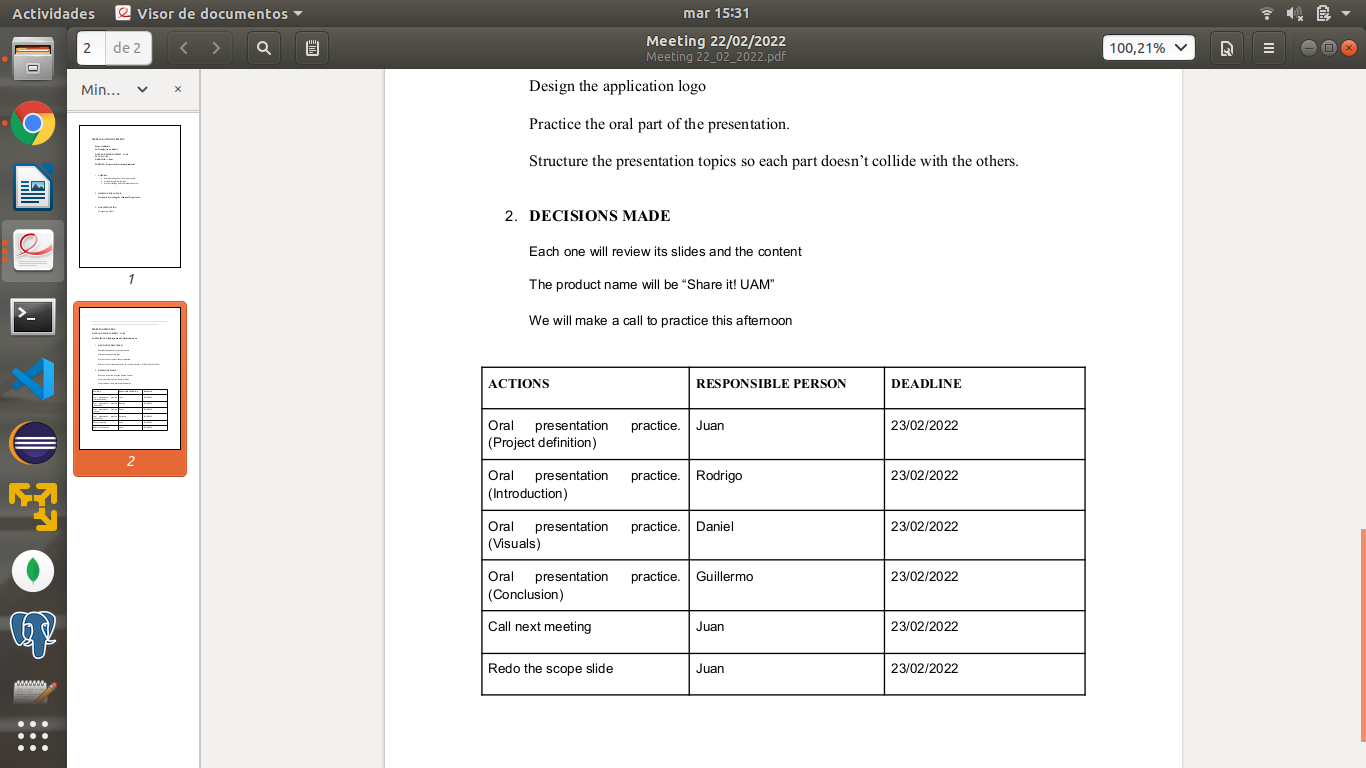
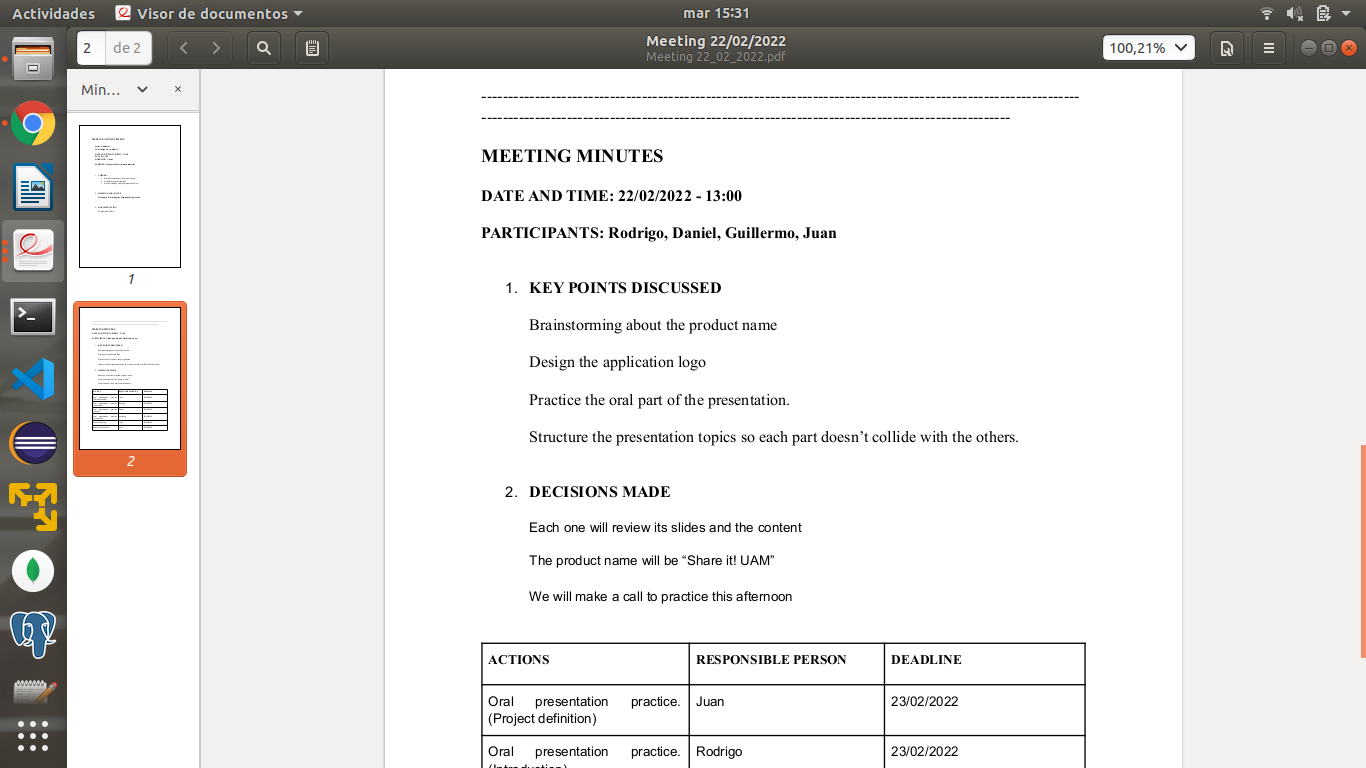
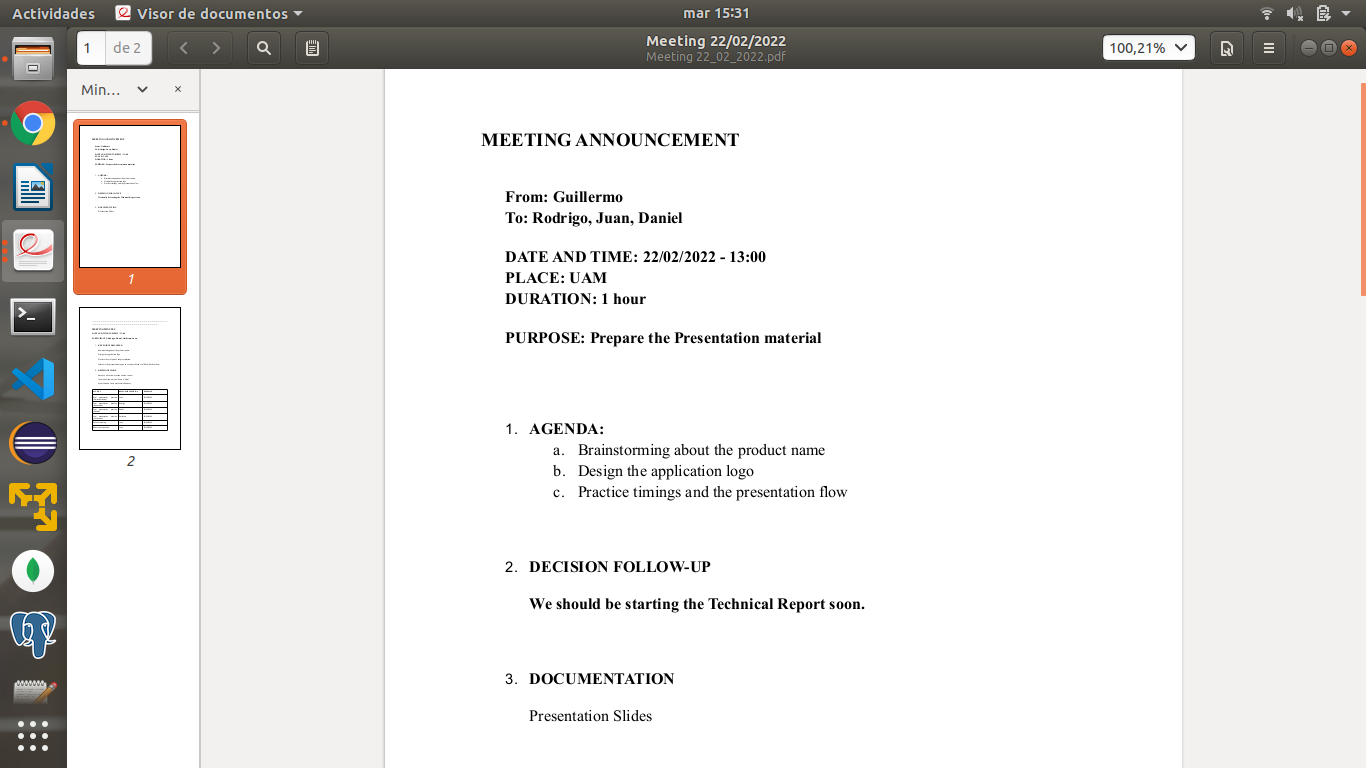






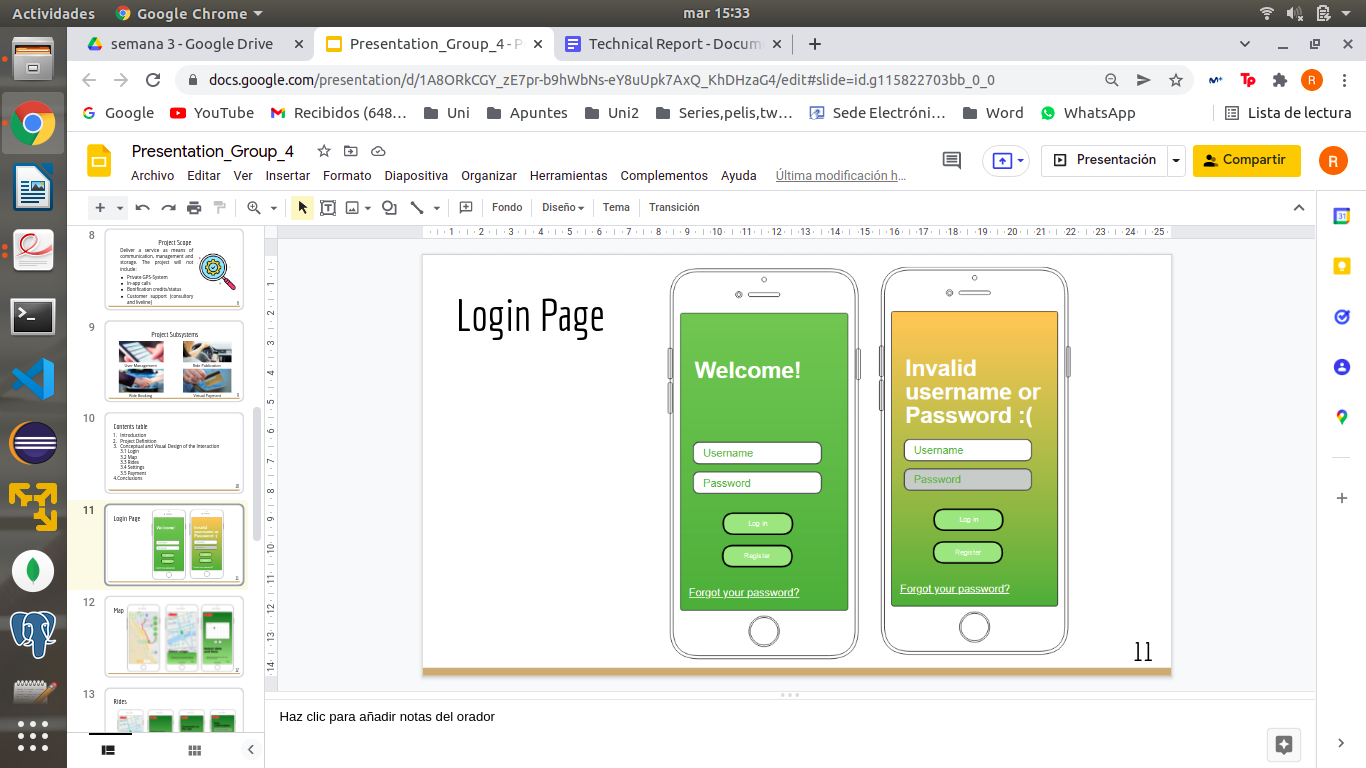




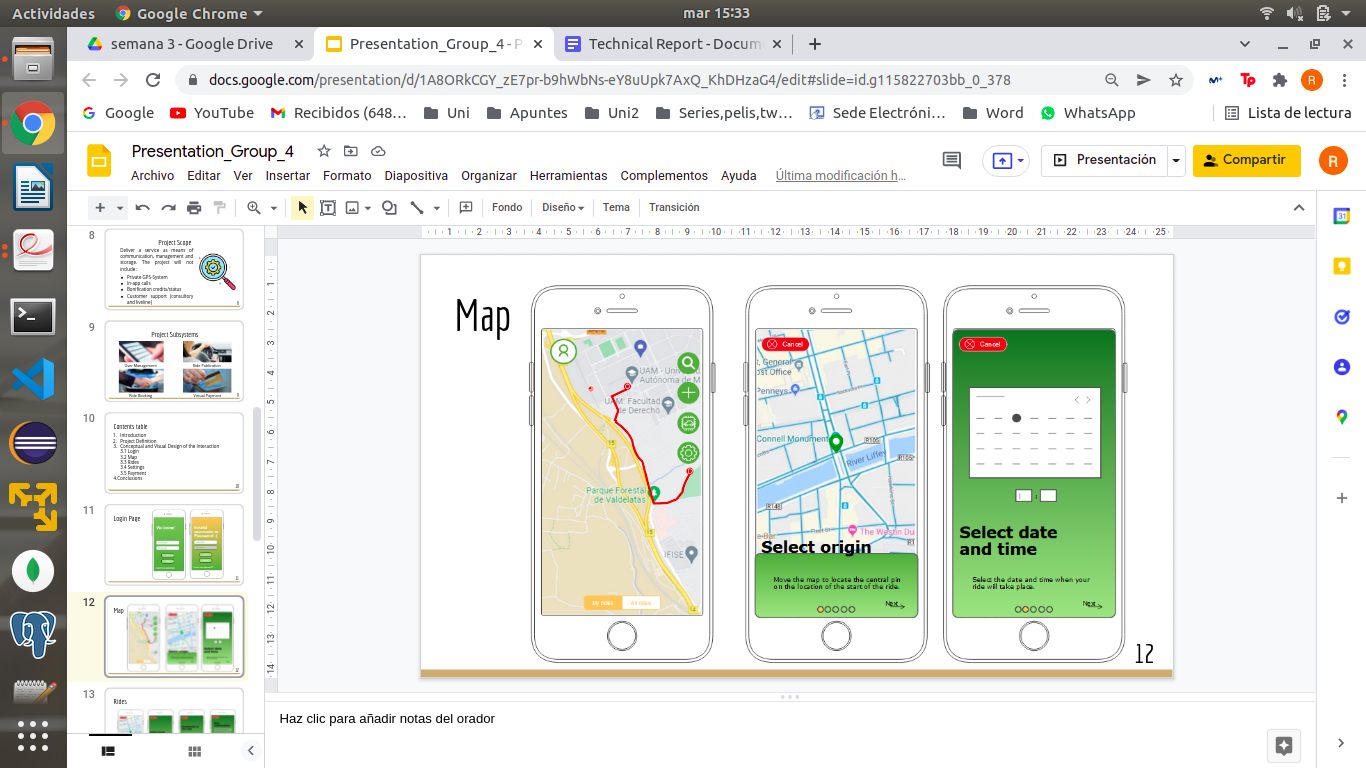


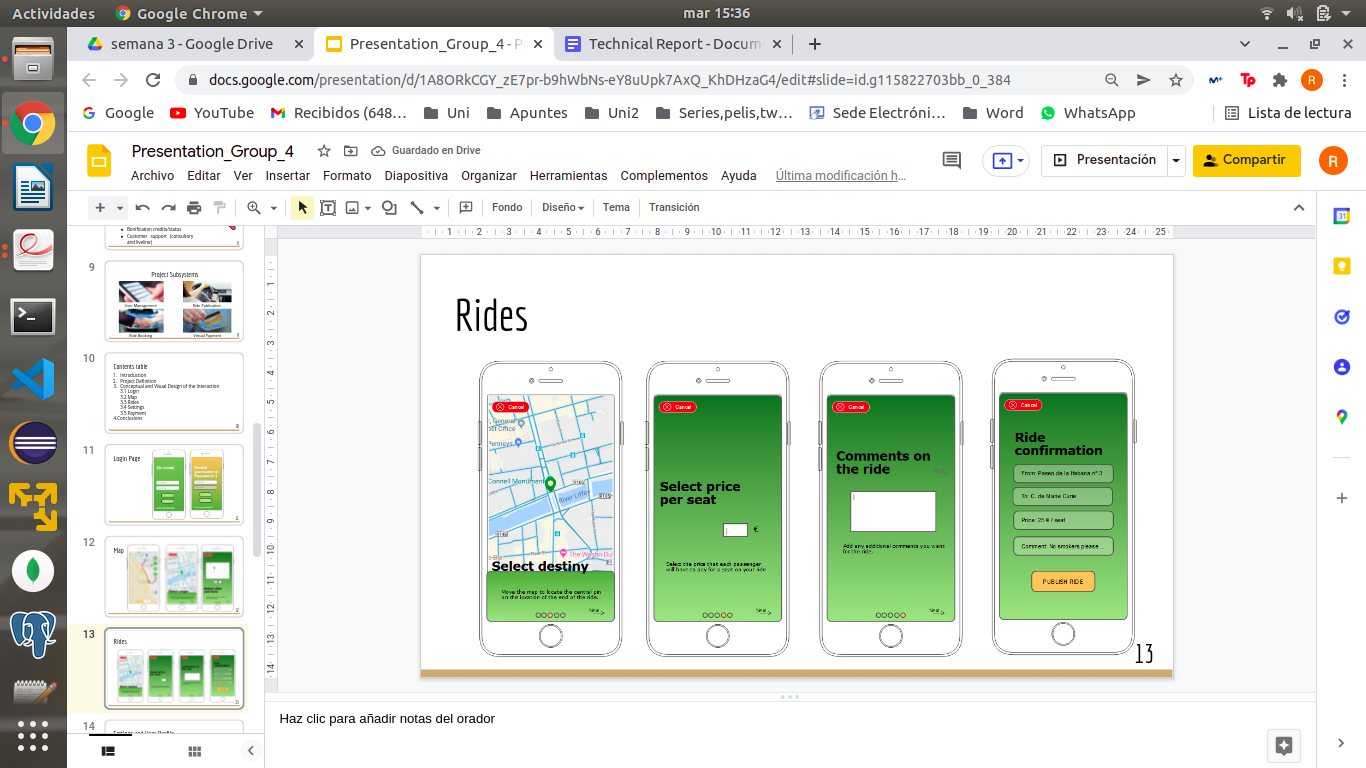
## **Appendix 4: Old mock-ups**

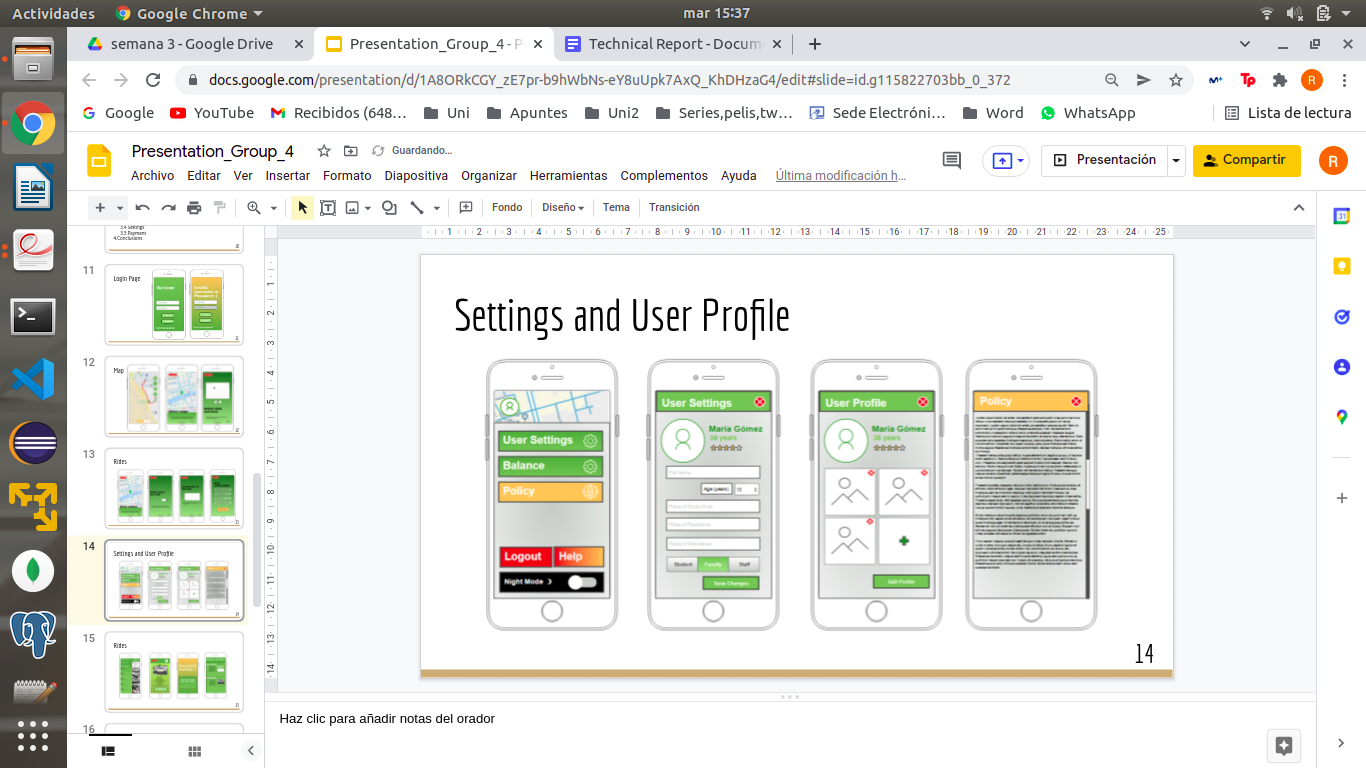
Login Page

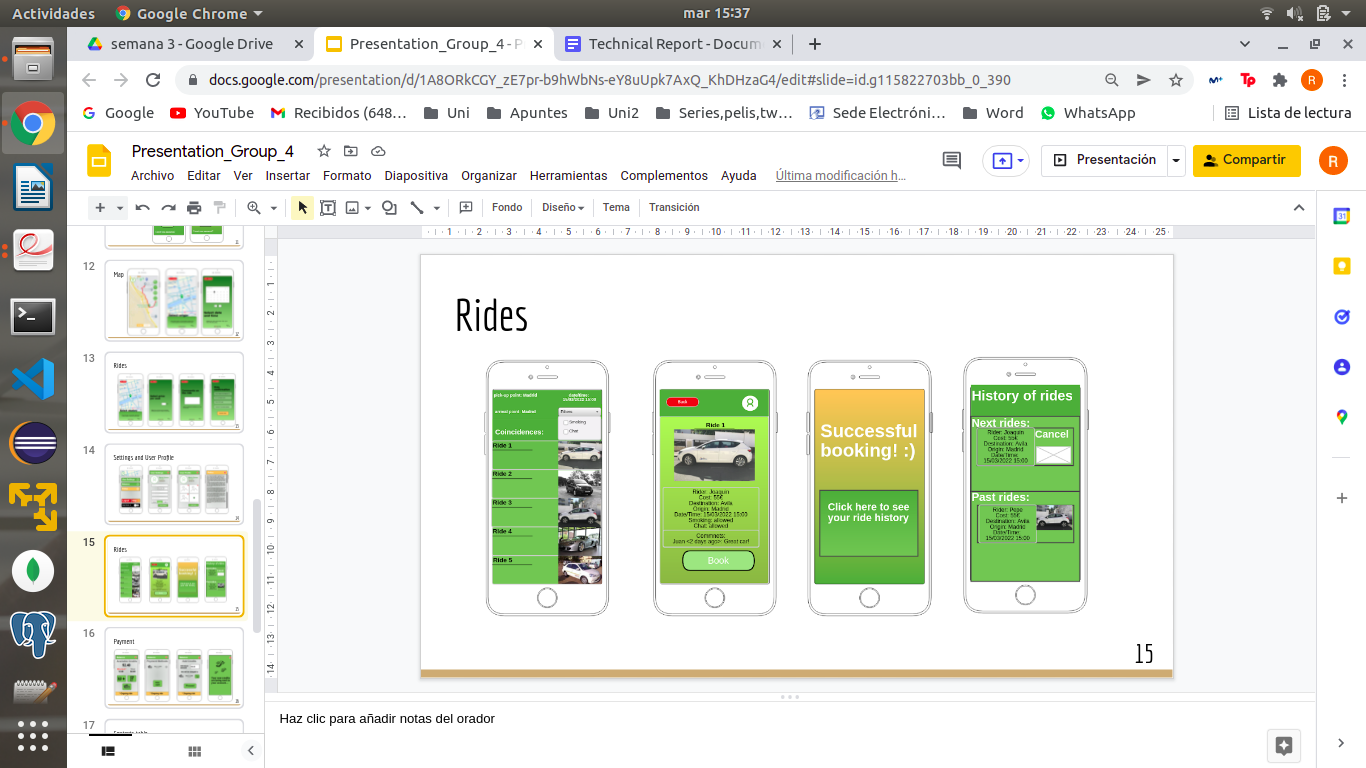


Map



Rides

Settings and user profile

Rides

Payment