

1S-2020

# SOFTWARE ENGINEERING

## Second Evaluation Project



OMI & PALI  
GROUP #3

## **TEAM**

Karla B. Burgos Gayrey

Francesca Helena Man Ging Pow Hing

Jonathan Israel Quintana Solorzano

Danny Paul Ríos Benavides

Brenda Michelle Bermello Burgos

## CONTENT

<b>1. INTRODUCTION .....</b>	4
<b>2. USER STORIES .....</b>	5
<b>3. SYSTEM USERS .....</b>	8
<b>3.1. ADMINISTRATOR.....</b>	8
<b>3.2. SELLER.....</b>	9
<b>3.3. BUYER .....</b>	9
<b>4. PRODUCT .....</b>	9
<b>4.1. WEB DASHBOARD .....</b>	10
<b>4.2. APP MOBILE .....</b>	10
<b>5. SCRUM METHODOLOGY .....</b>	10
<b>5.1. SCRUM ROLES.....</b>	11
<b>5.2. SESSION RULES.....</b>	12
<b>5.2.1. SPRINT PLANNING .....</b>	12
<b>5.2.2. DAILY MEETING .....</b>	12
<b>5.3. PRODUCT BACKLOG.....</b>	13
<b>5.4. DEPENDENCIES.....</b>	17
<b>5.5. SPRINTS .....</b>	19
<b>5.6. SPRINT BACKLOG .....</b>	19
<b>6. PROJECT MANAGEMENT .....</b>	22
<b>6.1. TOOLS.....</b>	22
<b>6.1.1. CLICK UP .....</b>	22
<b>7. PREEMPTIVE ERROR DETECTION .....</b>	24
<b>7.1. CODACY .....</b>	24
<b>8. TESTING.....</b>	28
<b>9. PROFILING.....</b>	28
<b>10. AUTHORIZATION FRAMEWORK .....</b>	32
<b>11. SERVER DEPLOYMENT GUIDE .....</b>	37
<b>12. WEBAPP DEPLOYMENT GUIDE .....</b>	38
<b>13. BUILD AUTOMATION TOOL .....</b>	39
<b>14. CONTINUOUS INTEGRATION TOOL.....</b>	41

<b>15. ANNEXES .....</b>	45
<b>15.1. DOCUMENTATION .....</b>	45
<b>15.2. GITHUB-SERVER .....</b>	45
<b>15.3. GITHUB-WEBAPP .....</b>	45
<b>15.4. GITHUB-MOBILEAPP .....</b>	45
<b>15.5. GITHUB-CODING STANDARD DOCUMENTATION (AIRBNB) .....</b>	45
<b>15.6. DRIVE PRE-RECORDER-VIDEO .....</b>	45
<b>15.7. CLIENT ACCEPTANCE FORM (FIRST PARTIAL) .....</b>	47
<b>15.8. CLIENT ACCEPTANCE FORM (SECOND PARTIAL) .....</b>	48
<b>15.9. MOBILE SCREENS.....</b>	49
<b>15.10. WEB SCREENS.....</b>	63
<b>15.11. WEBAPP WINDOW FLOW.....</b>	77
<b>15.12. MOBILE APP WINDOW FLOW.....</b>	79
<b>16. BIBLIOGRAPHY .....</b>	80

# Omi&Pali

## 1. INTRODUCTION

The “Omi & Pali” project was presented for the first time in 2019 during the semester corresponding to the second term of the school year, during Software Engineering I. A primary information survey was carried out that includes the identification of the users that intervene in the system at the enterprise level, the logic of the internal and external processes that require automation and the possibility of implementing a software development with the available resources of the company. The functional and non-functional requirements were established in the first meetings with the participation of the company owner, they were validated, and a work methodology was used according to the recognized needs.

Omi&Pali is a coffee shop located in Babahoyo, Los Ríos, aimed at families who want something sweet and fresh to share their moments, just to relax and disconnect from the world for a while. Currently, they take orders through WhatsApp and coordinate them with the delivery man to get a little closer to their customer's house or work. The implementation of an application seeks to increase the efficiency of its processes and a closer relationship with its customer as well as faster and organized saving of information. Being able to serve more clients and provide them a better service.

## 2. USER STORIES

To obtain the estimated points we rely on 3 factors: Complexity, effort, and uncertainty. Each factor will have a maximum score of 3 where 1 is low weight, 2 medium weight and 3 high weight, having as 1 minimum value of estimated points 3 and a maximum value of estimated points 9.

Administrator	
User Story: 1	
History Name: Add product	Business Priority: High
Estimated points: 5 As administrator, I WANT to insert a new product and the available stock of that product to be able to fill the catalog of the mobile application.	
User Story: 2	
History Name: Product display.	Business Priority: Medium
Estimated points: 4 As an administrator I WANT to view the products to know the detailed information.	
User Story: 3	
History Name: Search product.	Business Priority: Medium
Estimated points: 5 As an administrator, I WANT to search for products and to know if the product is registered.	
User Story: 4	
History Name: Modify Product.	Business Priority: High
Estimated points: 6 As an administrator, I WANT to modify a product TO maintain the information recorded in the mobile application.	
User Story: 5	
History Name: Delete product.	Business Priority: Medium
Estimated points: 5 As administrator, I WANT to delete a product TO keep the product catalog displayed in the mobile app up to date.	
User Story: 6	
History Name: Create promotions.	Business Priority: Low
Estimated points: 5 As an administrator, I WANT to create promotions of a product TO be able to fill the catalog of promotions of the mobile application.	
User Story: 7	
History Name: Edit promotions.	Business Priority: Low
Estimated points: 5 As an administrator, I WANT to edit a promotion of a product TO be able to update the catalog of promotions of the mobile application.	
User Story: 8	
History Name: Delete promotions.	Business Priority: Low
Estimated points: 5 As an administrator, I WANT to delete promotions of a product TO be able to update the catalog of promotions of the mobile application.	
User Story: 9	
History Name: Sales report generation.	Business Priority: Medium
Estimated points: 6 As administrator, I WANT to view sales report to know the buying habits of our customers.	
User Story: 10	
History Name: Manual assignment of orders.	Business Priority: High
Estimated points: 5 As administrator, I WANT to assign orders that are on hold TO the delivery men to manage the delivery of the order.	
User Story: 11	
History Name: Manual assignment of orders.	Business Priority: High
Estimated points: 5 As an administrator, I WANT to view the orders that have already been assigned to a delivery person TO have better control in a sale.	

<b>User Story: 12</b>	
<b>History Name:</b> Display assigned orders.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 5
AS an administrator, I WANT to view the orders that have already been assigned to a delivery person TO have better control in a sale.	

<b>User Story: 13</b>	
<b>History Name:</b> Display dispatched orders.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 5
AS an administrator, I WANT to view the orders that have already been assigned to a delivery person TO have better control in a sale.	

<b>User Story: 14</b>	
<b>History Name:</b> Display news by customer.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 4
AS an administrator, I WANT to see the news presented by customers TO know the improvements that must be made and have better control.	

<b>User Story: 15</b>	
<b>History Name:</b> Create new to deliveryman.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 4
AS an administrator, I WANT to create a novelty for deliveryman TO have better control over the dealers.	

<b>User Story: 16</b>	
<b>History Name:</b> Display news by deliveryman.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 4
AS an administrator, I WANT to see the news presented by deliveryman TO know the improvements that must be made and have better control.	

<b>User Story: 17</b>	
<b>History Name:</b> Create new to customer.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 4
AS an administrator, I WANT to create a novelty for customer TO have better control over the customer.	

<b>User Story: 18</b>	
<b>History Name:</b> Create User.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 5
AS an administrator, I WANT to create new user account to have better management of the activities to be performed on the web.	

<b>User Story: 19</b>	
<b>History Name:</b> Display Users.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 4
AS an administrator, I WANT to see registered users to be able to keep a record of control.	

<b>User Story: 20</b>	
<b>History Name:</b> Edit User.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 5
AS administrator, I WANT to edit a user's information to keep the recorded information up to date.	

<b>User Story: 21</b>	
<b>History Name:</b> Delete User.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 5
AS an administrator, I WANT to delete a user's account to keep the recorded information up to date.	

<b>User Story: 22</b>	
<b>History Name:</b> View registered customers.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 5
As administrator, I WANT to view registered customers so I can view detailed information for each account.	

<b>User Story: 23</b>	
<b>History Name:</b> Display conflicting clients.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 5
As administrator, I WANT to view conflicting clients so I can see the detailed information of each conflicting client.	

<b>User Story: 24</b>	
<b>History Name:</b> Edit account.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 5
As an administrator, I WANT to edit my account to keep the recorded information up to date.	

<b>User Story: 25</b>	
<b>History Name:</b> Display dispatched orders.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 4
As administrator, I WANT to display the list of orders shipped to get the details of the finished orders.	

Client

<b>User Story: 26</b>	
<b>History Name:</b> Create account.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 4
As a customer, I WANT to create new user account to be able to make purchases in the mobile app.	

<b>User Story: 27</b>	
<b>History Name:</b> Edit account.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 5
As a customer, I WANT to edit my account to keep the recorded information up to date.	

<b>User Story: 28</b>	
<b>History Name:</b> Add product to favorites.	
<b>Business Priority:</b> Low	<b>Estimated points:</b> 5
As a customer, I WANT to add a product to the favorites list TO have quick access to products that are more my liking.	

<b>User Story: 29</b>	
<b>History Name:</b> Display products.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 4
As a customer I WANT to view the products available to know that they are available in the store.	

<b>User Story: 30</b>	
<b>History Name:</b> Add products to the shopping cart.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 6
As customer, I WANT to add products to the shopping cart TO have a list of what I want to buy.	

<b>User Story: 31</b>	
<b>History Name:</b> Edit product quantity in the shopping cart.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 6
As a customer, I WANT to remove product from the shopping cart in case you no longer want it to avoid buying it.	

<b>User Story: 32</b>	
<b>History Name:</b> Remove products from the shopping cart.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 5
AS a customer, I WANT to remove product from the shopping cart in case you no longer want it to avoid buying it.	

<b>User Story: 33</b>	
<b>History Name:</b> View shopping cart.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 6
AS a customer, I WANT to view the shopping cart to know which product I have added to the Cart to make the purchase.	

<b>User Story: 34</b>	
<b>History Name:</b> Confirm Order.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 5
As a customer, I WANT to confirm my order to place the order and make the payment.	

<b>User Story: 35</b>	
<b>History Name:</b> View payment method.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 6
AS customer, I WANT to see the payment methods available TO check if I can make the payment with any of the payment methods arranged.	

<b>User Story: 36</b>	
<b>History Name:</b> Monitor Order.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 3
AS a customer I, WANT to monitor my order TO consult all the detailed information of the order stages.	

<b>User Story: 37</b>	
<b>History Name:</b> View FAQs	
<b>Business Priority:</b> Low	<b>Estimated points:</b> 3
AS a customer, I WANT to view the FAQs TO know if any of my concerns are answered.	

<b>User Story: 38</b>	
<b>History Name:</b> Add news.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 3
AS a customer, I WANT to add news regarding a dealer SO THAT you have better control in the store.	

#### Dealer

<b>User Story: 39</b>	
<b>History Name:</b> Display assigned orders.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 3
AS a dealer, I WANT to view the orders that have been assigned to me TO have detail of the purchase that the customer made.	

<b>User Story: 40</b>	
<b>History Name:</b> Finish purchase.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 6
AS a delivery man, I WANT to complete a purchase TO be aware that the order placed by the customer has been delivered.	

<b>User Story: 41</b>	
<b>History Name:</b> Enter new customer.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 5
AS a dealer, I WANT to enter novelty to a client SO THAT better control is given.	

#### General

<b>User Story: 42</b>	
<b>History Name:</b> Sign in.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 7
AS administrator/user/client, I WANT to log in TO access the features arranged for each role.	

<b>User Story: 42</b>	
<b>History Name:</b> Sign in.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 7
AS administrator/user/client, I WANT to log in TO access the features arranged for each role.	

<b>User Story: 43</b>	
<b>History Name:</b> Recover password.	
<b>Business Priority:</b> Medium	<b>Estimated points:</b> 4
AS a customer, I WANT to recover my password TO continue watching and making purchases in the mobile app.	

<b>User Story: 44</b>	
<b>History Name:</b> Log out.	
<b>Business Priority:</b> High	<b>Estimated points:</b> 5
AS administrator/user/client, I WANT to log in to finish my activities and prevent someone else from using the system, without being authorized or not accessing.	

## 3. SYSTEM USERS

### 3.1. ADMINISTRATOR

The administrator of the place will have control over the inventory of the products that they offer, he will be able to create, modify, eliminate and read the records of all the movements register in the base. Will have the option to manually assign the orders to a specific delivery man and verify if any alert has been reported to them or, other cases, they have reported any alert about a customer. This role can create promotions that will be launched to buyers, control the level of daily sales and manage system reports.

### **3.2. SELLER**

The role of the seller will have limited access to the functionalities of the system, will be able to manage the inventory and sales, as well as the orders that are placed. However, we cannot manually assign orders for deliveries, or launch product promotions. this access will be limited to the sales record.

### **3.3. BUYER**

This user will have their own interface. The administrator and the seller are users just in the web application; however, the buyer will only have jurisdiction in the mobile application. He can view the menu of the store, mark his favorite products, order them at home or reserve them for pick-up at the local. The app will be offered the possibility to pay by transfer or cash, track your order and save it for an occasional repeat. It has his own profile that shows the data that has been registered in the cafeteria base, which will have security measures to guarantee the privacy and integrity of customer information.

## **4. PRODUCT**

The development of a Web Board for the business owner and a mobile application was proposed for communication with local customers and a sell strategy. Both ideas differentiable by the access of the roles and the specific functions that are handled in each one separately.



#### **4.1. WEB DASHBOARD**

The web application will be modified for the administration of the business owner, it will be able to manage general inventory processes, create promotions for customers, generate sales reports, manage users, etc.<sup>1</sup>

#### **4.2. APP MOBILE**

The mobile application will be available in the App Store and/or Play store. It is specifically aimed at the business customer, it focuses on highlighting the menu offered by the cafeteria and the possibility of ordering and purchasing online, either under the modality of home delivery or the reservation of the product to be picked up directly from the local.

### **5. SCRUM METHODOLOGY**

The use of Scrum was allowed thanks to the opening of the Business and the commitment that was manifested when starting the project proposal (OBS, s.f.). There was high flexibility to changes, mainly in the design of the application wireframes, transparent communication helped the internal collaboration of the team, improving productivity and increasing experience in useful technology (Yadiz, s.f.). The risks to substantially complicated changes affecting the scope of the project were reduced by maintaining constant deliveries and customer feedback at each stage.

---

<sup>1</sup> The specification of functional and non-functional requirements was presented in the delivery part A of the project approach.

## 5.1. SCRUM ROLES

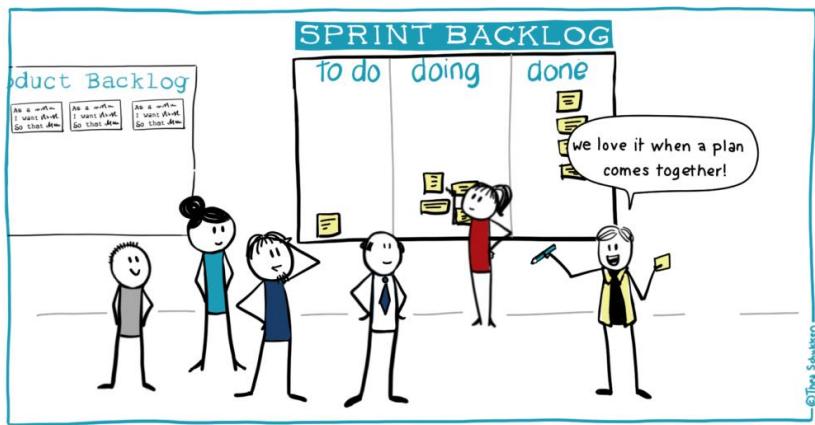
The Scrum team is made up of 5 members, who are currently studying Engineering in Computational Sciences at the Escuela Superior Politécnica del Litoral. Each team member contributes to the growth and development of the project from different roles, governed by the basic rules of the work methodology, with specific responsibilities that measure its effectiveness in a different way. The sum of all the established functions makes up the entire work team.

NAME	ROLE	START DATE	RESPONSABILITIES
<b>Francesca Man Ging</b>	Scrum master	October 2019	<ul style="list-style-type: none"><li>• Design tasks to the team.</li><li>• Advise and strengthen the team.</li></ul>
<b>Jonathan Quintana</b>	Product Owner	October 2019	<ul style="list-style-type: none"><li>• Lead the daily meetings.</li><li>• Keep the communication between the client and the team.</li></ul>
<b>Karla Burgos</b>	Developer	October 2019	<ul style="list-style-type: none"><li>• Develop the product, managing to deliver an increase to the software.</li></ul>
<b>Danny Rios</b>	Developer	October 2019	<ul style="list-style-type: none"><li>• Develop the product, managing to deliver an increase to the software.</li></ul>
<b>Brenda Bermello</b>	Developer	May 2020	<ul style="list-style-type: none"><li>• Develop the product, managing to deliver an increase to the software.</li></ul>

## 5.2. SESSION RULES

### 5.2.1. SPRINT PLANNING

The sprint planning meeting is generally held on Sundays, which is the day when all team members have full-time availability. It lasts approximately 1 to 2 hours, the Backlog tasks are explained in priority order o dependency, grouped and the sprint backlog is formed.



1. The previous sprint must be completed in its entirety.
2. All members should be online 10 minutes early the meeting hour.
3. All members must agree to the tasks assigned to them
4. All members must complete the tasks assigned within the time limit set for the sprint
5. In case of presenting a problem with the assignment of tasks, an agreement will be reached between the members.

### 5.2.2. DAILY MEETING

The daily meetings have a short duration, they are held at the end of the day and the difficulties that were encountered in the progress of the assigned task are shared.



1. All members should be online 5 minutes early the meeting hour.
2. In the case that a member cannot attend to the daily meeting, must notify and sends its report with anticipation.
3. All members should fill the daily meeting report before it starts.
4. The maximum time for the meeting is 15 minutes.
5. In case that a member needs to extend his topic, we will schedule a meeting to discuss it.

### 5.3. PRODUCT BACKLOG<sup>2</sup>

Below is a summary list of the web and mobile system requirements according to their priority

PRIORITY	HOMEWORK
Normal	Registry implementation with firebase
Normal	Graphical interface layout of the order history screen and order tracking
Normal	Graphic interface layout of the home screen system, product tabs and shopping cart

PRIORITY	HOMEWORK
Normal	Registry implementation with firebase
Normal	Graphical interface layout of the order history screen and order tracking
Normal	Graphic interface layout of the home screen system, product tabs and shopping cart

---

<sup>2</sup> This is just a superficial view of the backlog; the full product backlog is attached in the project documentation into the repository.

<b>Normal</b>	Layout of the graphical interface of the system Login screen, registration, change of password and orders
<b>High</b>	Refactoring of mobile app screens, Phone screen
<b>High</b>	Refactoring of mobile app screens, Orders screen
<b>High</b>	Refactoring of mobile app screens, registration screen
<b>High</b>	Refactoring of mobile app screens, Password change screen
<b>High</b>	Refactoring of mobile app screens, Login screen
<b>Normal</b>	Graphical interface layout of the user profile screen and edit profile
<b>High</b>	Upload API to pre-Production server
<b>High</b>	Refactoring of mobile app screens, shopping cart screen
<b>High</b>	Refactoring of mobile app screens, Home screen
<b>High</b>	Payment file refactoring
<b>High</b>	Refactoring of mobile app screens, User Profile screen
<b>High</b>	Refactoring of mobile app screens, Edit Profile screen
<b>High</b>	Security file refactoring
<b>High</b>	Refactoring of order file in API
<b>High</b>	JWT refresh token settings
<b>High</b>	Implementation of security in the API
<b>High</b>	Access control to user data
<b>Normal</b>	JWT encryption implementation

<b>High</b>	User file refactoring in API
<b>High</b>	Role file refactoring in API
<b>High</b>	Novel file refactoring in API
<b>High</b>	Login file refactoring in API
<b>Normal</b>	Creation of order screens, reports, and modification of menus
<b>Normal</b>	Implementation of registry to the system API
<b>Normal</b>	Refresh token settings in mobile app with interceptors
<b>Normal</b>	Mobile application login functionality
<b>High</b>	Implementation of a registered cell number confirmation system.
<b>Normal</b>	Implementation of guards for access control
<b>High</b>	Products select screen.
<b>High</b>	Product modification screen.
<b>High</b>	Disposal of products.
<b>High</b>	Add a new product
<b>High</b>	Beverage load
<b>High</b>	Cookie load
<b>High</b>	Ice cream load
<b>High</b>	Promotions loading
<b>High</b>	Candy load

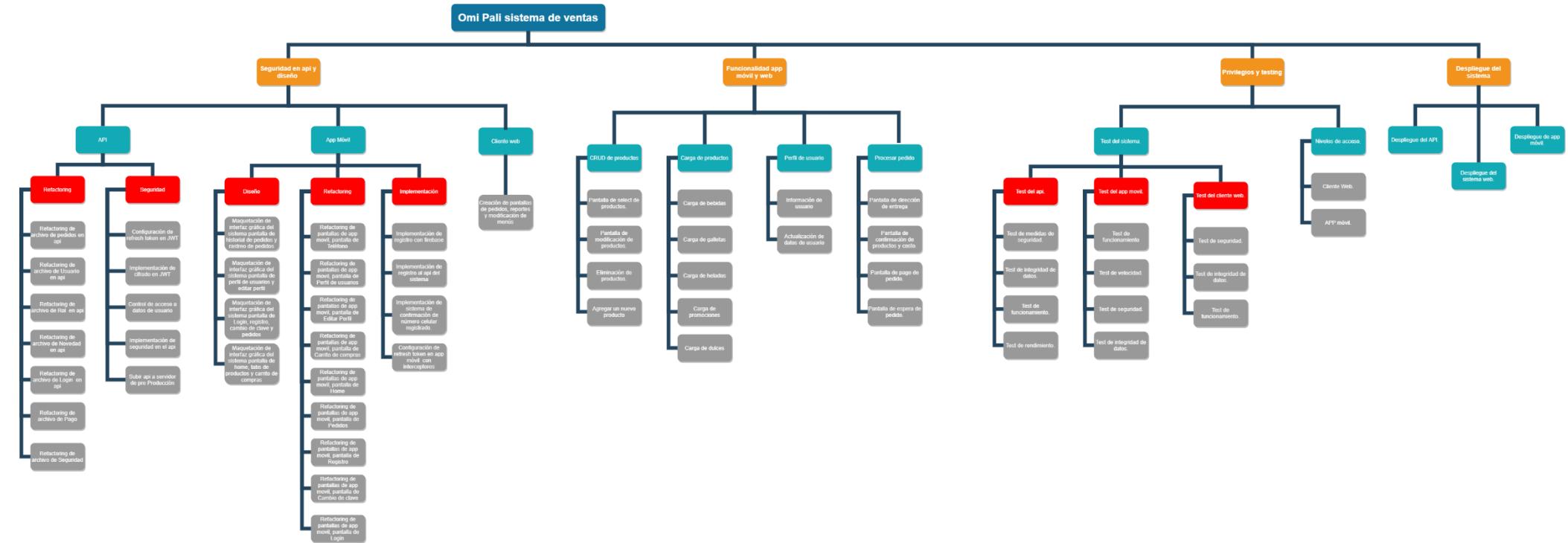
<b>Normal</b>	User information
<b>Normal</b>	Updating user data
<b>Normal</b>	Delivery address screen
<b>Normal</b>	Product and cost confirmation screen.
<b>Normal</b>	Order payment screen.
<b>Normal</b>	Order waiting screen.
<b>High</b>	Security measures test.
<b>High</b>	Data integrity test.
<b>High</b>	Functional test.
<b>High</b>	Performance test.
<b>High</b>	Function test
<b>High</b>	Speed test.
<b>High</b>	Security test.
<b>High</b>	Data integrity test.
<b>High</b>	Security test.
<b>High</b>	Data integrity test.
<b>High</b>	Functional test.
<b>High</b>	Web Client.
<b>High</b>	Mobile APP.

<b>High</b>	API deployment.
<b>High</b>	Deployment of the web system.
<b>High</b>	Deployment of mobile app.

#### 5.4. DEPENDENCIES<sup>3</sup>

---

<sup>3</sup> The attached file for task dependencies is in the repository.



## **5.5. SPRINTS**

The deliverable of this sprint will be the API with the security measures that are stipulated in the tasks implemented and tested together with the design of all the screens of the mobile application and the function of the login and registration of users and the web system template raised and uploaded to the repository.

## **5.6. SPRINT BACKLOG<sup>4</sup>**

With a previous presentation of the product backlog, the PO presented the tasks prioritized and grouped into items of higher or lower priority for the subsequent scrum team analysis. The team determined which items can be completed during the sprint that was starting. In the sprint planning meeting, the team moved the items that corresponded from the product backlog to the sprint backlog. In this way, subtasks are added to each of the general backlog tasks and active members are assigned to these functions.

For this part, the established tasks consisted of the development of the web part of the system and modifications in certain parts of the mobile system.

- **Request:** those requirements are based on defined categories, in this case, the requirement A consists in the general operation of the web application, while the requirement B is the tests and deployments of the systems
- **Sprint hours:** Are the hours established for the sprint by the workgroup.
- **Days:** these are the days that the working group had to carry out the sprint task.
- **Task:** It reference to the tasks that the working group must end for the sprint delivery.
- **Status:** Are the status of each task in the sprint. The states are complete, in progress, and not started.

---

<sup>4</sup> The document is attached in the project repository

- **Who:** Indicates the name of the person assigned to the task.

The attached image shows the sprint backlog of this deliverable, the time in total hours that was considered to perform the tasks of this sprint was 185 in total. The start date of the sprint was 07/24/2020 and its end date was 08/28/2020, without considering weekends.

SPRINT BACKLOG						SPRINT	DURATION	START	END
REQUEST	TASK	WHO	STATE (Complete/Uninitiated/in progress)	DAY	24/07/2020	27/07/2020 - 3/08/2020	4/08/2020 - 10/08/2020	11/08/2020 - 17/08/2020	18/08/2020 - 28/08/2020
				Hours	185	158	119	77	35
Request A	Mobile application login functionality	Francesco Man Ging	Complete		10	0	0	0	0
Request A	Web application login functionality	Karla Burgos	Complete		0	10	0	0	0
Request A	CRUD novelty	Brenda Bermello	Complete		0	3	0	0	0
Request A	CRUD user	Karla Burgos	Complete		4	2	0	0	0
Request A	CRUD client	Brenda Bermello	Complete		4	4	0	0	0
Request A	CRUD order	Danny Rios	Complete		0	3	0	0	0
Request A	Creation of promotions	Karla Burgos	Complete		0	3	4	0	0
Request A	WebApp settings	Brenda Bermello	Complete		0	2	4	0	0
Request A	Search dynamically in tables	Brenda Bermello	Complete		4	0	3	4	0
Request A	Changes api products	Danny Rios	Complete		0	2	0	5	0
Request A	Recover password for webApp users	Karla Burgos	Complete		2	5	2	0	0
Request A	Dashboard statistics	Danny Rios	Complete		0	0	3	6	0
Request A	Manual order assignment	Danny Rios	Complete		3	5	8	0	0
Request A	Deliveryman security	Danny Rios	Complete		0	0	4	0	0
Request A	Manual assignment of orders	Danny Rios	Complete		0	0	2	4	0
Request A	Deliveryman interface	Danny Rios	Complete		0	0	4	4	0
Request A	Reports generation	Brenda Bermello	Complete		0	0	0	4	0
Request A	Incoming orders in dashboard UI	Danny Rios	Complete		0	0	2	4	3
Request A	Load products	Jonathan Quintana	Complete		0	0	0	4	3
Request A	User profile	Francesco Man Ging	Complete		0	0	0	4	2
Request A	Process order	Danny Rios	Complete		0	0	6	3	2
Request B	System test	Brenda Bermello	Complete		0	0	0	0	5
Request B	Platform review	Karla Burgos	Complete		0	0	0	0	4
Request B	Access levels	Karla Burgos	Complete		0	0	0	0	4
Request B	Api Deployment	Jonathan Quintana	Complete		0	0	0	0	4
Request B	Mobile app deployment	Jonathan Quintana	Complete		0	0	0	0	4
Request B	Web app deployment	Jonathan Quintana	Complete		0	0	0	0	4
Request A		General Refactoring		Hours left	158	119	77	35	0
Request B		Implementation requirements for the api rest and the mobile application							

## **6. PROJECT MANAGEMENT**

### **6.1. TOOLS**

Project management tools personalize the registration and planning of the work carried out by the Scrum team, adapting to the needs that arise throughout the development of the project. A total of 3 management tools were tested, each offering different options to improve control of established tasks. The first tool that was used was Jira but its limited number of allowed users for project development and its non-intuitive interface were the main reasons why the group decided to change the project control tool. Then, the information was changed to Asana and although the interface was much easier to use, there were few functionalities allowed under the free account. It allowed the assignment and states controls of tasks, but it did not show any graphical report on the performance of the team, everything had to be registered under a paid account. Finally, doing exploratory research, the team moved to Click Up.

#### **6.1.1. CLICK UP**

Public site view: <https://share.clickup.com/l/h/5-16530724-1/65526a3c6aa0669>

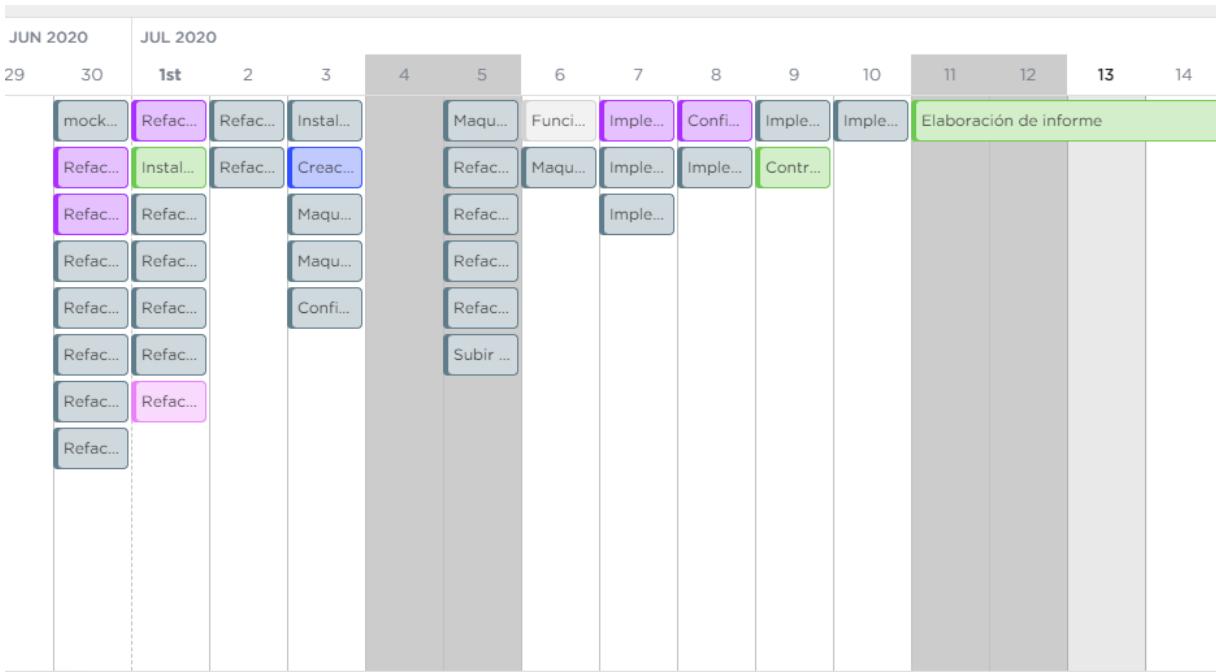
In Click Up, tasks were assigned to each team member. It has three main states like:

- In progress
- Finished
- Closed
- Approved

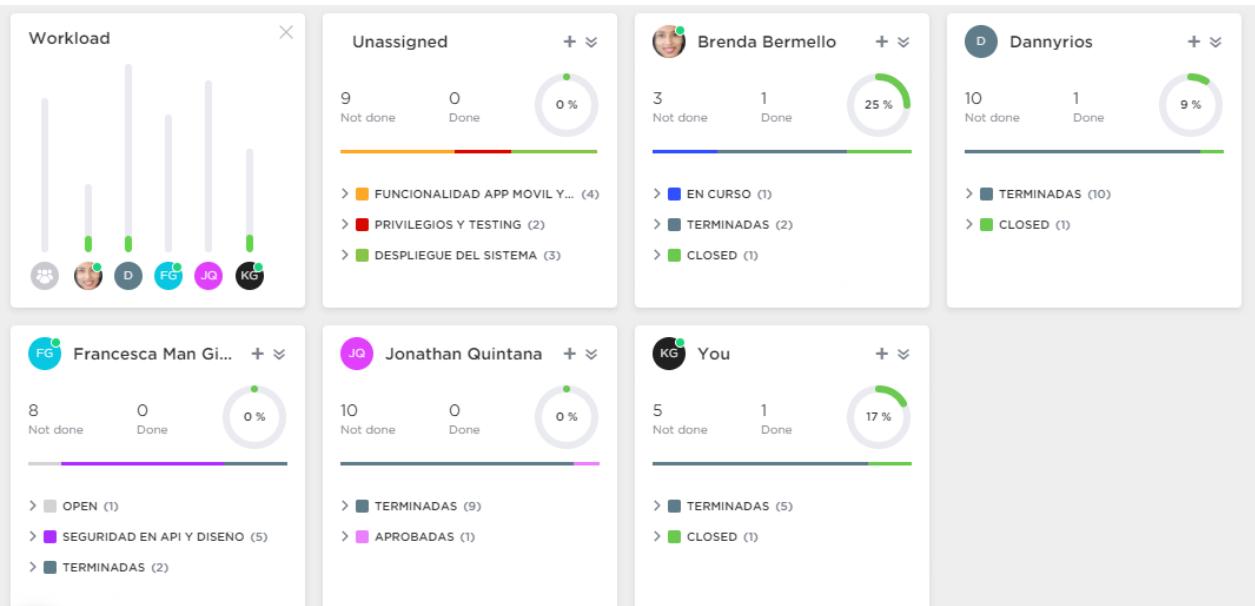
Team members are the ones who change the state of their tasks, based on their willingness and preference to start. The progress made during the day is reported at the daily meeting.

Labels are handled for better display on the boards. The priorities are defined based on dependence on a higher task and there is the possibility of timing the hours spent on the task in question.

Displays a better view of task completion time, considering the days it took.



It shows the comparative performance of the work team. It compares them by tasks assigned and hours invested in each task.



## 7. PREEMPTIVE ERROR DETECTION

### 7.1. CODACY

To face the problem of preemptive error detection, we decided to use Codacy which automatically identifies problems through static code review analysis. We chose this tool because of the ease with which it can be used; by accessing directly from the github repository and allowing us to choose the rules in our case we choose PMD 6.26.0 and PMD (legacy) 5.8.1.

When making our code of the web part be reviewed by this tool, we obtained types of results: Code Style, Error prone and Security.

webSoftwareII master

Badge

B Repository certification

Quality evolution

Last 7 days Last 31 days

Issues 126% =	Complex Files 0% =	Duplicated code 1% =	Coverage -
---------------	--------------------	----------------------	------------

1792 total issues

Category	Total
Security	94
Error Prone	105
Code Style	1593
Compatibility	0
Unused Code	0
Performance	0

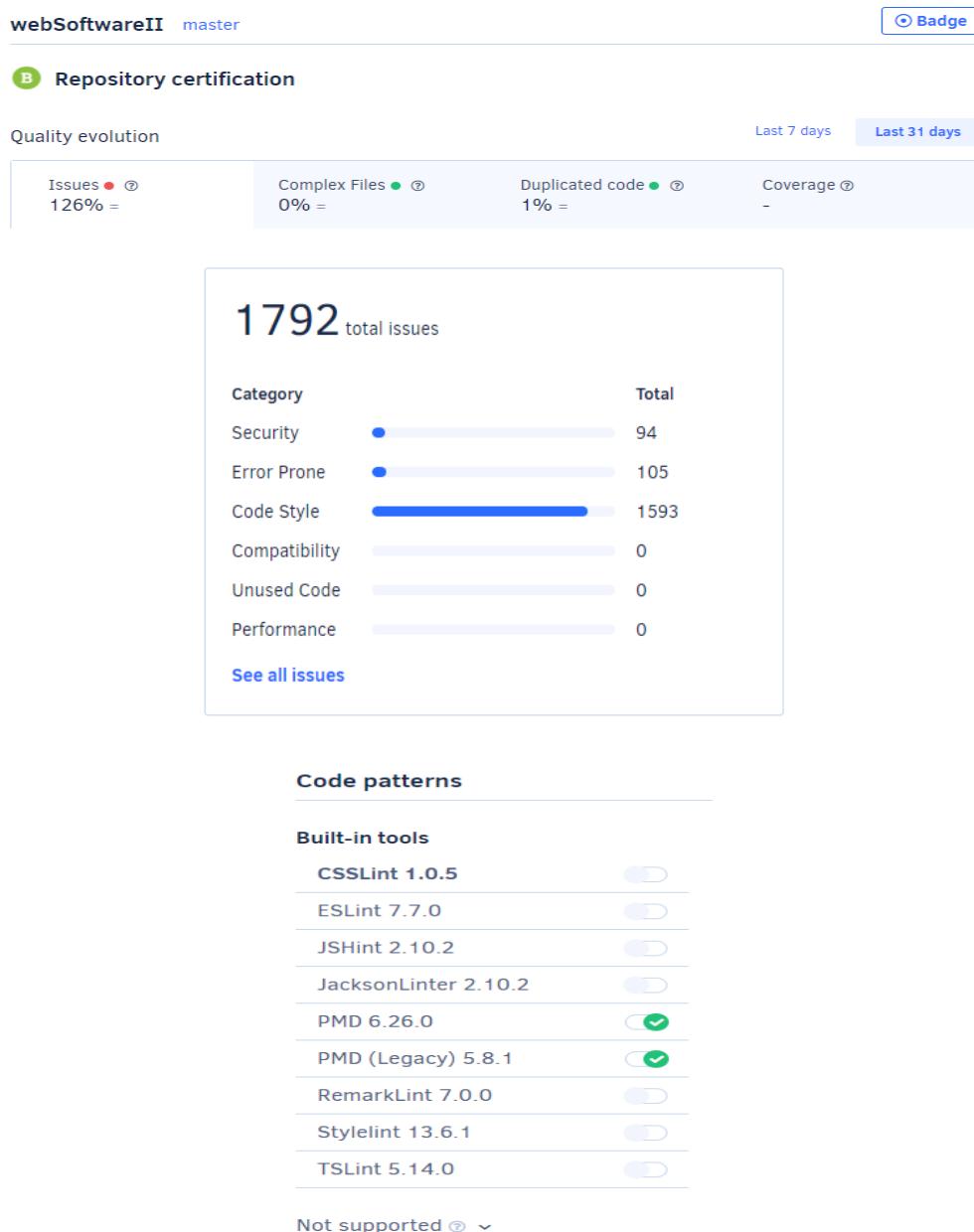
[See all issues](#)

Code patterns

Built-in tools

CSSTint 1.0.5	OFF
ESLint 7.7.0	OFF
JSHint 2.10.2	OFF
JacksonLinter 2.10.2	OFF
PMD 6.26.0	ON
PMD (Legacy) 5.8.1	ON
RemarkLint 7.0.0	OFF
Stylelint 13.6.1	OFF
TSLint 5.14.0	OFF

Not supported ▾



Code Style problems were based on inserted extra lines, missing lines, required spaces, poorly defined structures, for which we proceeded to identify each of items presented to make their respective change.

<code>src/app/components/navbar/navbar.component.css</code>	<code>src/app/pages/products/products.component.css</code>
Expected no more than 1 empty line (max-empty-lines) 18 }	Expected single space before "(" (block-opening-brace-space-before) 22 td{
<code>src/app/pages/assigned/assigned.component.css</code>	Expected single space before "(" (block-opening-brace-space-before) 32 .button-save{
Expected "#91CCDD" to be "#91ccdd" (color-hex-case) 30 background-color:#91CCDD;	Expected single space before "(" (block-opening-brace-space-before) 45 .save{
Expected single space before "(" (block-opening-brace-space-before) 49 .datesDivFooter{	<code>src/app/pages/promotions/promotions.component.css</code>
Unexpected missing end-of-source newline (no-missing-end-of-source-newline) 79 }	Expected single space before "(" (block-opening-brace-space-before) 29 .save{
<code>src/app/pages/delivery-notification/delivery-notification.component.css</code>	Unexpected extra semicolon (no-extra-semicolons) 31 border-color: DARKMAGENTA;;
Expected single space before "(" (block-opening-brace-space-before) 22 .required{	<code>src/app/pages/users/users.component.css</code>
<code>src/app/pages/delivery-order/delivery-order.component.css</code>	Expected single space before "(" (block-opening-brace-space-before) 39 .required{
Unexpected whitespace at end of line (no-eol-whitespace)	
<code>src/app/components/sidebar/sidebar.component.css</code>	<code>src/app/pages/dispatched/dispatched.component.css</code>
Unexpected empty line before declaration (declaration-empty-line-before) 5 box-shadow: 0 4px 20px 0px rgba(0, 0, 0, 0.14), 0 7px 10px -5px rgba(145, 141, 141, 0.8);	Expected "#91CCDD" to be "#91ccdd" (color-hex-case) 30 background-color:#91CCDD;
<code>src/app/pages/current-promotion/current-promotion.component.css</code>	Expected single space before "(" (block-opening-brace-space-before) 58 .divSimbology{
Expected single space before "(" (block-opening-brace-space-before) 36 .Form-Label{	Expected single space before "(" (block-opening-brace-space-before) 74 .tituloCampo{
Unexpected missing end-of-source newline (no-missing-end-of-source-newline) 43 }	<code>src/app/pages/login/login.component.css</code>
<code>src/app/pages/dashboard/dashboard.component.css</code>	Expected whitespace after /** (comment whitespace-inside) 145 /* Utility */;
Expected single space before "(" (block-opening-brace-space-before) 101 .subtitleCard{	Expected whitespace after /** (comment whitespace-inside) 236 /* Input */;
Expected single space before "(" (block-opening-brace-space-before) 107 .buttonSearch{	Unexpected empty line before declaration (declaration-empty-line-before) 348 display: -webkit-box;
<code>src/app/pages/deliveryman/deliveryman.component.css</code>	Expected whitespace before /** (comment whitespace-inside) 397 /* { responsive */;
Expected newline after "," (selector-list-comma-newline-after) 44 .buttonSearch	

Error prone problems were located in the typescript files, in which most of them gave errors due to the use of console.log, which proceeded to eliminate all the console.logs present in the project.

documentation/js/material-dashboard.js	src/app/pages/configuration/configuration.component.ts
Unnecessary semicolon. 146 };	Unexpected console statement. 62 console.log(data);
src/app/core/services/auth/auth.service.ts	Unexpected console statement. 180 console.log("Ingreso correcto");
Unexpected console statement. 104 console.log({	src/app/pages/current-promotion/current-promotion.component.ts
Unnecessary semicolon. 142 doLoginUser(tokens: Token, cedula: string) {	Unexpected console statement. 381 console.log("Guardado");
src/app/core/services/deliverman/deliverman-reporter.service.ts	src/app/pages/customers/customers.component.ts
Unexpected console statement. 58 console.log('recibe esta novedad: ', novedad);	Unexpected console statement. 63 console.log(err);
src/app/core/services/interceptor/token-interceptor.service.ts	src/app/pages/deliveryman/deliveryman.component.ts
Unexpected console statement. 30 console.log(req.url);	Unexpected console statement. 80 console.log(this.data);
Unexpected console statement. 34 console.log("entra aqui")	src/app/pages/login/login.component.ts
src/app/pages/old-promotion/old-promotion.component.ts	Unexpected console statement. 114 console.log(err);
Unexpected console statement. 98 console.log(err);	src/app/components/navbar/navbar.component.ts
src/app/pages/on-hold/on-hold.component.ts	Unexpected console statement. 140 console.log("estas en cerrar sesion")
Unexpected console statement. 241 console.log(this.pedido.cantidades);	src/app/core/services/categoria/categoria.service.ts
src/app/pages/promotions/promotions.component.ts	Unexpected console statement. 88 console.log("Document successfully deleted!");
Unexpected console statement. 148 console.log(producto);	src/app/core/services/deliverman/deliverman-reporter.service.ts
Unexpected console statement. 194 console.log("Guardado");	Unexpected console statement. 88 console.error('Error removing document: ', error);
src/app/pages/user-notification/user-notification.component.ts	src/app/core/services/interceptor/token-interceptor.service.ts
Unexpected console statement. 134 console.log(listaAdmin[j].esCliente);	Unexpected console statement. 31 console.log("Hello men")
Unexpected console statement. 144 console.log(data);	src/app/core/services/product/producto.service.ts
- - - - -	Unexpected console statement. 82 console.log(producto)

Regarding security problems, insertion was presented in the variables and objects necessary to carry out our project.

```

src/app/core/utils/seguiridad.ts
Generic Object Injection Sink
111 data += clave + ":" + json[clave] + ":";

src/app/pages/conflicting/conflicting.component.ts
Generic Object Injection Sink
111 this.getAddresses(environment.variables.nombreClientes[i]['direccion']);

Function Call Object Injection Sink
112 unique.push(coleccion[i]);

src/app/pages/current-promotion/current-promotion.component.ts
Function Call Object Injection Sink
113 this.allFiles.push(file[i]);

src/app/pages/dashboard/dashboard.component.ts
Function Call Object Injection Sink
114 if(this.maze.get(this.listadoProductosApi[1]) == null) {
115   this.allFiles.push(file[i]);
}

src/app/pages/delivery-notification/delivery-notification.component.ts
Generic Object Injection Sink
116 if(environment.variables.nombreClientes[i]['redes'] === listaC[i].idCliente) {
117   lista[i].nombre = environment.variables.nombreClientes[i]['correo'];
}

src/app/components/navbar/navbar.component.ts
Generic Object Injection Sink
118 return this.listTitles[item].title;

src/app/core/services/seguiridad.service.ts
Generic Object Injection Sink
119 data += clave + ":" + json[clave] + ":";

src/app/pages/assigned/assigned.component.ts
Generic Object Injection Sink
120 this.listadoPartidores[1].pedidos[1].idPedido;
121 this.listadoPartidores[1];
122 cantidad: cantidades[1];

src/app/pages/conflicting/conflicting.component.ts
Generic Object Injection Sink
123 if(environment.variables.nombreClientes[i]['redes'] === listaC[i].idCliente) {
124   lista[i].nombre = environment.variables.nombreClientes[i]['correo'];
}

```

```

src/app/pages/dashboard/dashboard.component.ts
Generic Object Injection Sink
125 lista[i].direccion = this.direccion;
126 this.listadoCuidadosApi[1] = this.productsIntrantesApi[1].cuidadosString().split(',');
127 this.productosMap = this.maze.get(this.listadoProductosApi[1]) + parseInt(this.listadoCuidadosApi[1]);
128 Function Call Object Injection Sink
129 this.maze.set(this.listadoProductosApi[1], this.productosMap);
130 this.maze.set(this.listadoCuidadosApi[1], this.productosMap);

src/app/pages/delivery-notification/delivery-notification.component.ts
Generic Object Injection Sink
131 if(environment.variables.nombreClientes[i]['redes'] === listaC[i].idUsuarioReportado);

src/app/pages/delivery-order/delivery-order.component.ts
Generic Object Injection Sink
132 "products": this.products[1].redes,
133 "cuidados": cuidadoses[1];

```

## 8. TESTING

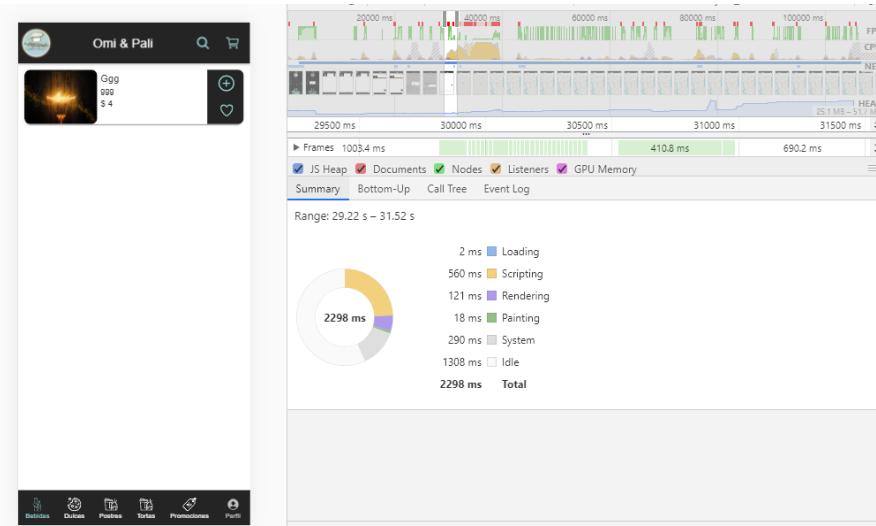
## 9. PROFILING

Profiling is the dynamic analysis of programs, it is the study of the system behavior in real-time (atinternet, s.f.), as the program runs. It helps to determine which section or requirement takes more time than should, what function must be optimized, how to decrease its memory resources, or sometimes both. To identify the functions and requirements that need an adjustment in their response time, the main functions of use were analyzed both in the mobile app and in the web client.

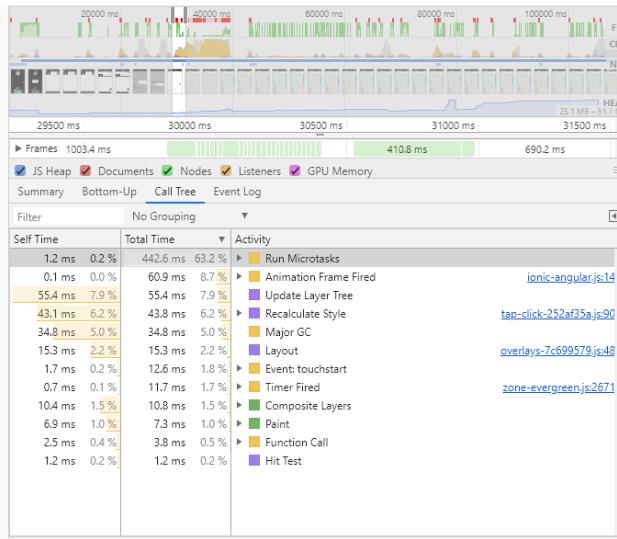
### 9.1. MOBILE APP

In the mobile app the login process was outlined, where the user provides his ID and password to enter the application menu. The different menu option tabs of the available products were navigated to enter an order, which was selected. An order was entered with a home delivery method, placing a new address, not saved in the database. When entering a new address, the location map is loaded where we select the location, if the user wants it, from where he wants to receive the order. Finally, when creating a new order and the customer can track the status of the order from his user profile.

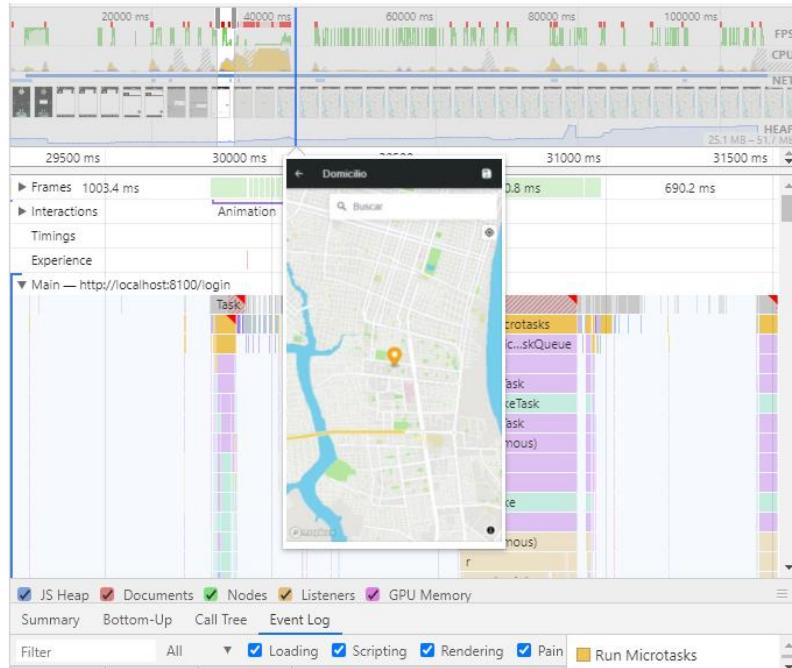
The profiling tool offered by the Google Chrome browser was used, it analyzes the reaction time and shows a summary in the developer tools. The image clearly shows the reaction time that the mobile application had to the navigation tests it was subjected to. The total time it took to complete order was 2298ms.



Sorting the time of the services, the longest reaction time was running the small individual tasks for each component took a time of 442.6 ms, almost 63% of the total.



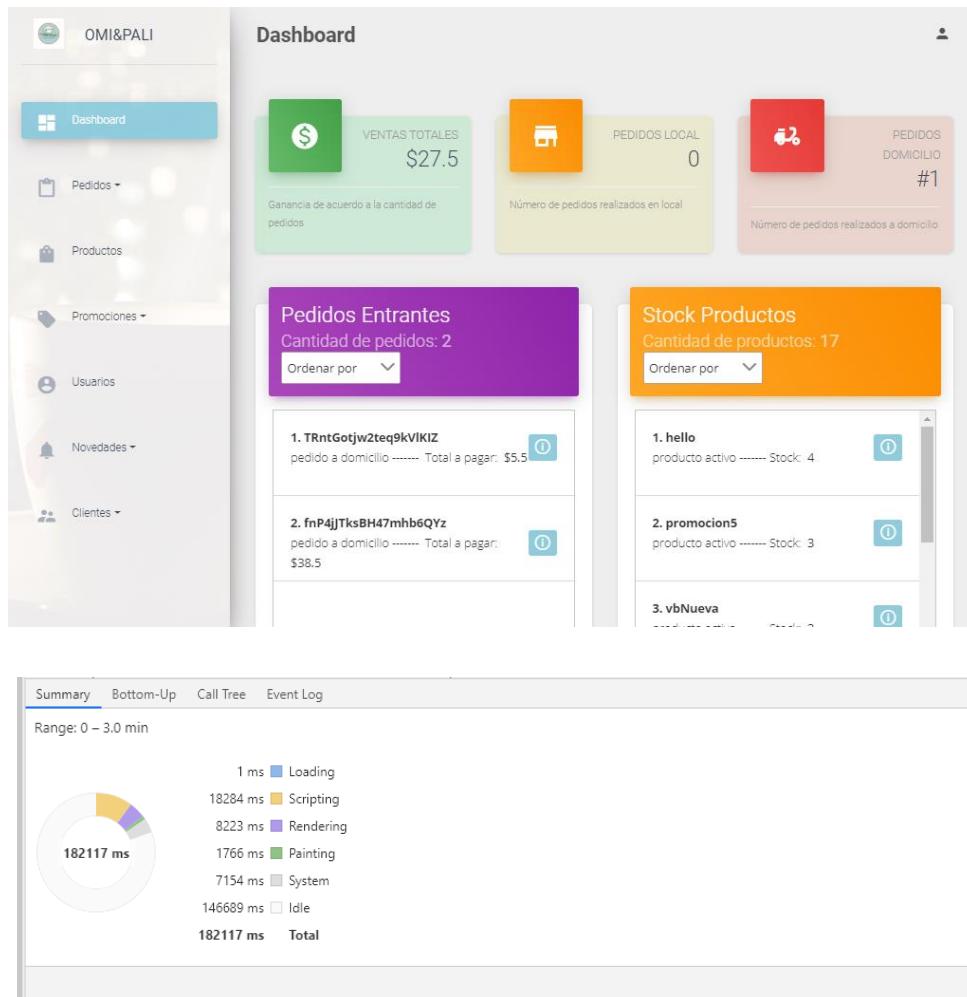
Loading the application frames took 1003.4 ms, which is reflected in the summary initially presented. The map component took the longest time to respond to the request, however, it did not significantly affect the efficiency of the app.



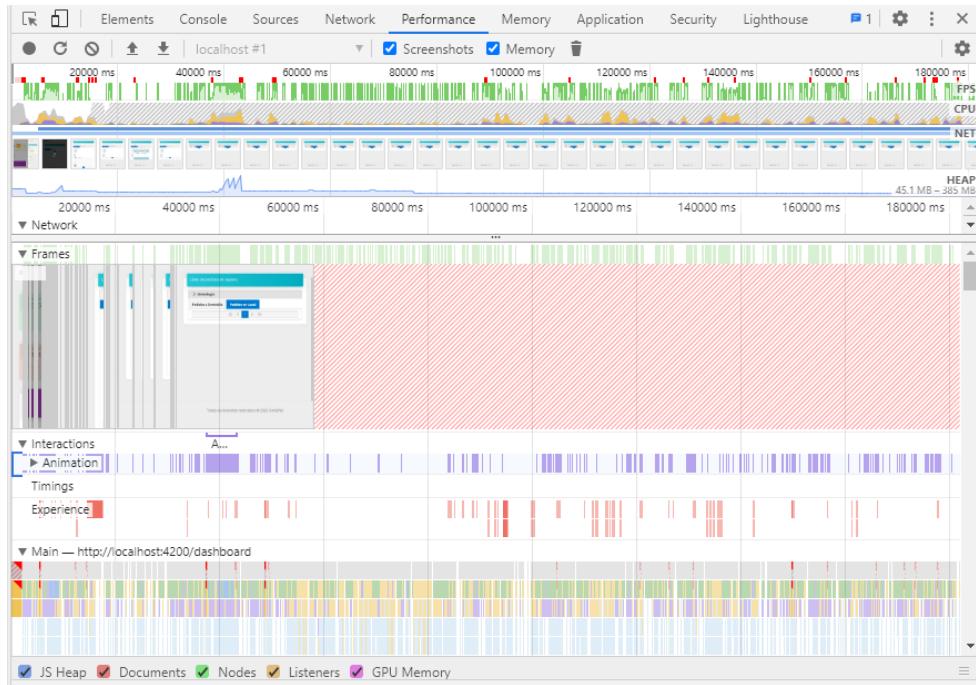
## 9.2. WEB CLIENT

In the web client, the processes of assigning orders to a delivery person, sending the termination link to close the home order, and the general CRUDs of the system

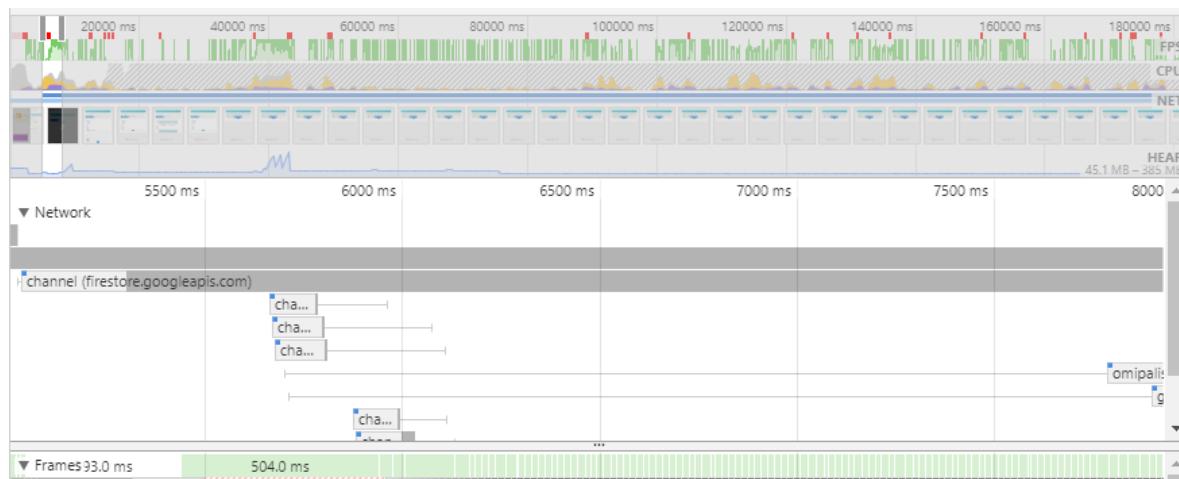
were analyzed. The general process that the administrator and/or seller had in the daily use of the application was simulated.



Processes were carried out for the assignment of distributors, review of the details of the order, and its completion. A new product was entered on the menu, it was updated in stock and the test record was deleted. This was the procedure to analyze the CRUD of users, news, promotions, and orders. En promedio todo el proceso de revisión se realizó en 182117 ms, sin embargo 146689 ms es de Idle (The session idle timeout).



There are two different types of requests, to firebase and to the rest API. In time comparisons and measurements, requests to firebase take considerably less time than API requests.



## 10. AUTHORIZATION FRAMEWORK

To restricting access and redirecting to the login page, when the user is not logged in, we use Router Guards. These guards let usallow implementing policies governing possible route transitions in an Angular application. As Router Guards are just simple class providers, a proper interface must be implemented.

```

1 import { Injectable } from "@angular/core";
2 import {
3   CanActivate,
4   ActivatedRouteSnapshot,
5   RouterStateSnapshot,
6   UrlTree,
7   Router,
8 } from "@angular/router";
9 import { Observable } from "rxjs";
10 import { map } from "rxjs/operators";
11 import { isNullOrUndefined } from "util";
12 import { AuthService } from "../services/auth/auth.service";
13
14 @Injectable({
15   providedIn: "root",
16 })
17 export class AuthGuard implements CanActivate {
18   constructor(private authService: AuthService, private router: Router) {}
19
20   canActivate(): boolean {
21     if (this.authService.isAuthenticated) {
22       return true;
23     }
24     this.router.navigateByUrl("/login");
25     return false;
26   }
27 }

```

The deliveryman interface is a little bit simpler; it was configured the next way.

```

1 import { Injectable } from '@angular/core';
2 import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot, UrlTree, Router } from '@angular/router';
3 import { Observable } from 'rxjs';
4 import { AuthDeliverymanService } from '../services/deliveryman/auth-deliveryman.service';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class AuthDeliverymanGuard implements CanActivate {
10
11   constructor(private authDeliveryman : AuthDeliverymanService, private router: Router) { }
12
13
14   canActivate(): boolean {
15     if(this.authDeliveryman.getLoggedInFirebase()) {
16       return true;
17     } else {
18       this.router.navigate(['deliveryman']);
19       return false;
20     }
21   }
22 }

```

AuthGuard implements canActivate() which tells the app router whether it can or cannot activate a particular route. To attach given guard to the route that it should protect, we just need to place its reference in canActivate property of that route as presented below.

```

42  {
43    path: "dashboard",
44    redirectTo: "dashboard",
45    pathMatch: "full",
46    canActivate: [AuthGuard],
47  },
48  {
49    path: "configuration",
50    redirectTo: "configuration",
51    pathMatch: "full",
52    canActivate: [AuthGuard],
53  },

```

The Json Web Token (JWT) persisted after user logs into the application, it can be used it to authorize outgoing requests. One approach could be to simply update every service that communicates with API to enrich requests with additional HTTP Header. This will result in a lot of duplicated code comparing to approach with HTTP Interceptor. Creating an HTTP interceptor is like creating a Router Guard.

We need to have a class that implements a specific interface with the required method. If we have a token, we set an appropriate HTTP header. This code also contains error handling logic, which will be described later in this article.

```

17  export class TokenInterceptorService implements HttpInterceptor {
18    private isRefreshing = false;
19    private refreshTokenSubject: BehaviorSubject<any> = new BehaviorSubject<any>(
20      null
21    );
22
23    constructor(private authService: AuthService) {}
24
25    intercept(
26      req: HttpRequest<any>,
27      next: HttpHandler
28    ): Observable<HttpEvent<any>> {
29
30      console.log(req.url);
31      const url: string = "https://omipalif2.herokuapp.com/api/login/token";
32      if (req.url.search(url) === -1) {
33        if (this.authService.token) [
34          req = this.addToken(req, this.authService.token.token)];
35        }
36      return next.handle(req).pipe(
37        catchError((error) => {
38          if (error instanceof HttpErrorResponse && error.status === 401) {
39            return this.handleError401Error(req, next);
40          }
41          return throwError(error);
42        })
43      );
44    }
45  }
46 }

```

```

48  private addToken(request: HttpRequest<any>, token: string) {
49    return request.clone({
50      setHeaders: {
51        // "Content-Type": "application/json",
52        Authorization: `Bearer ${token}`,
53      },
54    });
55  }
56
57  private handle401Error(request: HttpRequest<any>, next: HttpHandler) {
58    if (!this.isRefreshing) {
59      this.isRefreshing = true;
60      this.refreshTokenSubject.next(null);
61      return this.authService.refreshToken().pipe(
62        switchMap((token: any) => {
63          this.isRefreshing = false;
64          this.authService.token.token = token.access_token;
65          this.authService.token.refreshToken = token.newRefreshToken;
66          this.refreshTokenSubject.next(token.access_token);
67          return next.handle(this.addToken(request, token.access_token));
68        })
69      );
70    } // tslint:disable-next-line: unnecessary-else
71    else {
72      return this.refreshTokenSubject.pipe(
73        filter((token) => token != null),
74        take(1),
75        switchMap((jwt) => {
76          return next.handle(this.addToken(request, jwt));
77        })
78      );
79    }
80  }
81 }

```

The app uses an AuthService. This will be the place where we implement all the logic to handle logging in and out. Below you can find the source of that service and we will analyze it step by step. Here we use HttpClient to execute post call to the server and apply some operators with pipe () method.

The side effect we want to execute is to store these tokens calling doLoginUser. Finally, in case of error, we show the alert and return observable of false

```

22  export class AuthService {
23    private readonly JWT_TOKEN = "JWT_TOKEN";
24    private readonly REFRESH_TOKEN = "REFRESH_TOKEN";
25
26    loggedUser: string;
27    token: Token;
28    isAuthenticated: boolean = false;
29    dataUser: Usuarios;
30
31    isAdmin: boolean = false;
32
33    constructor(
34      private httpClient: HttpClient,
35      private router: Router,
36      private AAuth: AngularFireAuth
37    ) {}
38

```

```

52  public loginToApi(email: string, clave: string) {
53    const body = {
54      email,
55      clave,
56    };
57    return this.httpClient.post(environment.rutas.urlLogin, body);
58  }
59

```

Implementation of the logout method is basically the same, apart from the fact, that inside of the request's body we send refreshToken. This will be used by the server to identify who is attempting to log out. Then, the server will remove the pair.

```

97  refreshToken() {
98    console.log({
99      id: this.dataUser.email,
100     refreshToken: this.token.refreshToken,
101   });
102   return this.httpClient
103     .post<any>(environment.rutas.urlToken, {
104       id: this.dataUser.cedula,
105       refreshToken: this.token.refreshToken,
106     })
107     .pipe(
108       tap((token: Token) => {
109         this.storeJwtToken(token.token);
110       })
111     );
112   }
113

```

```

161  getRefreshToken() {
162    return localStorage.getItem(this.REFRESH_TOKEN);
163  }
164
165 /**
166 * @async
167 * @method
168 * @public
169 * @version 1.0.0
170 * @returns {String} JSON user access token.
171 * @desc this method storage the user authorization token in the localStorage. <br> Creation Date: 08/10/28
172 * @author Karla Burgos Gayrey <kbburgos@espol.edu.ec>
173 */
174 private storeJwtToken(jwt: string) {
175   localStorage.setItem(this.JWT_TOKEN, jwt);
176 }
177
178 /**
179 * @async
180 * @method
181 * @public
182 * @version 1.0.0
183 * @returns {String} JSON user information.
184 * @desc this method storage the user authorization token in the localStorage. <br> Creation Date: 08/10/28
185 * @author Karla Burgos Gayrey <kbburgos@espol.edu.ec>
186 */
187
188 private storeTokens(tokens: Token, cedula: string, rol: any) {
189   localStorage.setItem(this.JWT_TOKEN, tokens.token);
190   localStorage.setItem(this.REFRESH_TOKEN, tokens.refreshToken);
191   localStorage.setItem("cedula", cedula);
192   localStorage.setItem("rol", rol);
193 }

```

```

204  private removeTokens() {
205    localStorage.removeItem(this.JWT_TOKEN);
206    localStorage.removeItem(this.REFRESH_TOKEN);

```

Even when the roles are not centralized, the web and mobile app use the same authentication and security methodology.

## 11. SERVER DEPLOYMENT GUIDE

1. git clone <https://github.com/jquintanas/restApiSoftware1>
2. cd restApiSoftware1
3. Inside the directory run npm i
  - 3.1. This is to install all the necessary dependencies so that the server can run.
4. Go to the folder config and inside this to the file config.json
  - 4.1. This file oversees storing the data for the connection to the database, depending on the environment in which it will be used, the fields of the .json file must be filled.
5. Execute the command npm run build
  - 5.1. This command is to build the code that will be used to raise our server.
  - 5.2. Once our server has been raised, this command will have a transpiration server raised in observer mode so that after saving each change, the code .ts is transpiled to .js automatically.
6. Execute the command npm run dev
  - 6.1. This command is for development, it should be emphasized that if it were in production mode, the npm start command must be executed to start our server.
  - 6.2. If we stay in the development environment, the command will create a new server in observer mode so that each time the code is transpired, the server will auto start, and the changes will be reflected.

## **PRE-REQUIREMENTS**

1. Have node js installed
2. Have typeScript installed

## **12. WEBAPP DEPLOYMENT GUIDE**

1. git clone <https://github.com/kbburgos/webSoftwareII>
2. cd webSoftwareII
3. Inside the directory run npm i
  - 3.1. This is to install all the necessary dependencies so that the server can run.
4. Go to the folder config and inside this to the file config.json
  - 4.1. This file oversees storing the data for the connection to the database, depending on the environment in which it will be used, the fields of the .json file must be filled.
5. Execute the command ng serve
  - 5.1. This command is to build and run the code that will be used to raise our webApp.

## **PRE-REQUIREMENTS**

1. Have node js installed
2. Have typeScript installed
3. Have angular cli installed

## 13. BUILD AUTOMATION TOOL

Two tools for continuous integration will be used for automatic deployment, testing, building and documentation.

Git for version control and that once a new change is loaded, continuous integration tasks are performed automatically.

Git CI will carry out the construction of the code, that is, it will transpile the .ts code to .js so that the api can be deployed, it will execute the tests and build the documentation automatically after each change that is uploaded after passing the integration, it will be sent to heroku CI for automatic implementation of the new changes.

### Git Continuous Integration Configuration File

```
39 lines (24 sloc) | 764 Bytes
1 # This workflow will do a clean install of node dependencies, build the source code and run tests across different versions of node
2 # For more information see: https://help.github.com/actions/language-and-framework-guides/using-nodejs-with-github-actions
3
4 name: Node.js CI
5
6 on:
7   push:
8     branches: [ despliegue ]
9   pull_request:
10    branches: [ despliegue ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16
17     strategy:
18       matrix:
19         node-version: [10.x, 12.x, 14.x]
20
21     steps:
22       - uses: actions/checkout@v2
23       - name: Use Node.js ${{ matrix.node-version }}
24         uses: actions/setup-node@v1
25         with:
26           node-version: ${{ matrix.node-version }}
27       - run: npm ci
28       - run: npm run build --if-present
29       - run: npm test
30       - run: npm run docs
```

## Heroku pipeline configuration

The screenshot shows the Heroku Pipeline configuration page. At the top, there's a navigation bar with 'HEROKU' and a user icon. Below it, the pipeline name 'apimppal' is displayed. The pipeline consists of three stages: 'REVIEW APPS', 'STAGING', and 'PRODUCTION'. The 'REVIEW APPS' stage has a button 'Give review' and a note about using the new Review Apps feature. The 'STAGING' stage contains an app named 'apimppapi'. The 'PRODUCTION' stage contains an app named 'apimppal' with a note 'Auto-deploy - deploque' and a deployment log entry from 11:34 PM. A 'Configure permission' button is visible at the top right.

## Heroku deployment configuration with git

The screenshot shows the Heroku app overview page for 'apimppal'. At the top, there's a navigation bar with 'HEROKU' and a user icon. Below it, the app name 'apimppal' is shown along with deployment details. The page includes sections for 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. Under 'Deploy', there's a note about free-grained control with Heroku Teams. The 'Latest activity' section shows a deployment by 'jonax300\_jonu@hotmail.com' at 11:34 PM yesterday, followed by build success and log enable events. The 'Collaborator activity' section shows a deployment by 'jonax300\_jonu@hotmail.com' at 11:24 PM yesterday. A 'More' button is located at the bottom right.

The image contains two screenshots of the Heroku dashboard. The top screenshot shows the deployment method section, which includes options for Heroku Git, GitHub Connected, and Container Registry. It also displays a message about the app being connected to GitHub and provides links to manage the pipeline and review apps. The bottom screenshot shows the Activity feed, listing several recent events such as deployments, build successes, and log enablements.

## 14. CONTINUOUS INTEGRATION TOOL

Two tools will be used for continuous integration for deployment, testing and automatic construction.

Git for version control and that once a new change is uploaded, the continuous integration tasks are carried out automatically.

Git CI will carry out the construction of the code, that is, it will transpile the code .ts to .js so that the api can be deployed, it will execute the tests and it will build the documentation automatically after each change that is uploaded after passing the integration, it will be sent to heroku CI for a self-deploy of the new changes.

Git continuous integration configuration file

```

webSoftwareII/node.js.yml at master · jquintanas/webSoftwareII · GitHub
Raw Blame History
30 lines (24 sloc) 759 Bytes
1 # This workflow will do a clean install of node dependencies, Build the source code and run tests across different versions of node
2 # For more information see: https://help.github.com/actions/language-and-framework-guides/using-nodejs-with-github-actions
3
4 name: Node.js CI
5
6 on:
7   push:
8     branches: [ master ]
9   pull_request:
10    branches: [ master ]
11
12 jobs:
13   build:
14
15     runs-on: ubuntu-latest
16
17     strategy:
18       matrix:
19         node-version: [10.x, 12.x, 14.x]
20
21     steps:
22       - uses: actions/checkout@v2
23       - name: Use Node.js ${{ matrix.node-version }}
24         uses: actions/setup-node@v1
25         with:
26           node-version: ${{ matrix.node-version }}
27       - run: npm ci
28       - run: npm run build --if-present
29       - run: npm test
30       - run: npm postinstall

```

Actions · jqquintanas/webSoftwareII · GitHub

jqquintanas / webSoftwareII

forked from mburges/webSoftwareII

Code Pull requests Actions Projects Wiki Security Insights Settings

All workflows

**Create node.js.yml**

Node.js CI #1: Commit 7d990c6 pushed by jqquintanas

Event: master Status: In progress 25 minutes ago

Contact GitHub Pricing API Training Blog About

## Heroku CI pipeline configuration

webomipali · Pipeline | Heroku · Brave

HEROKU

Pipeline Tests Access Settings

REVIEW APPS + New app STAGING + Add app PRODUCTION + Add app

This pipeline is using the new Review Apps

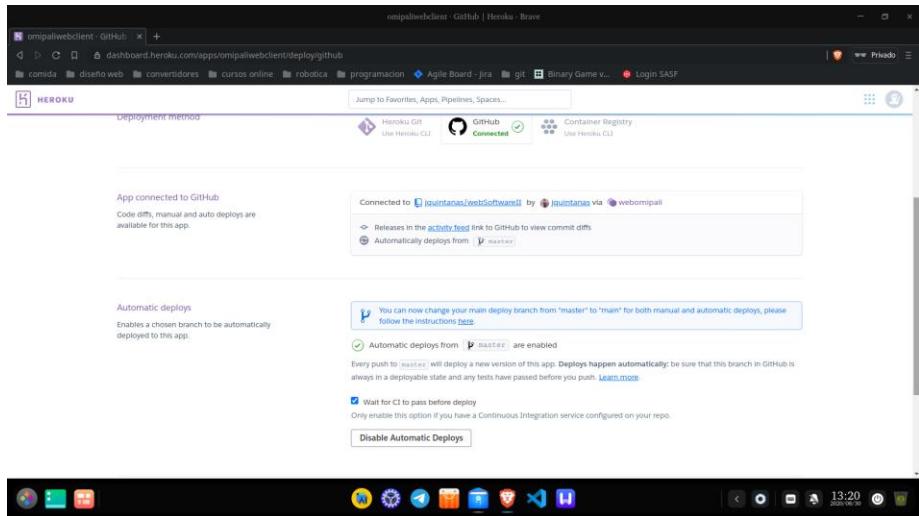
Docs Give us feedback

omnipilotwebclient Auto-deploy: manual Deployed today at 12:29 PM

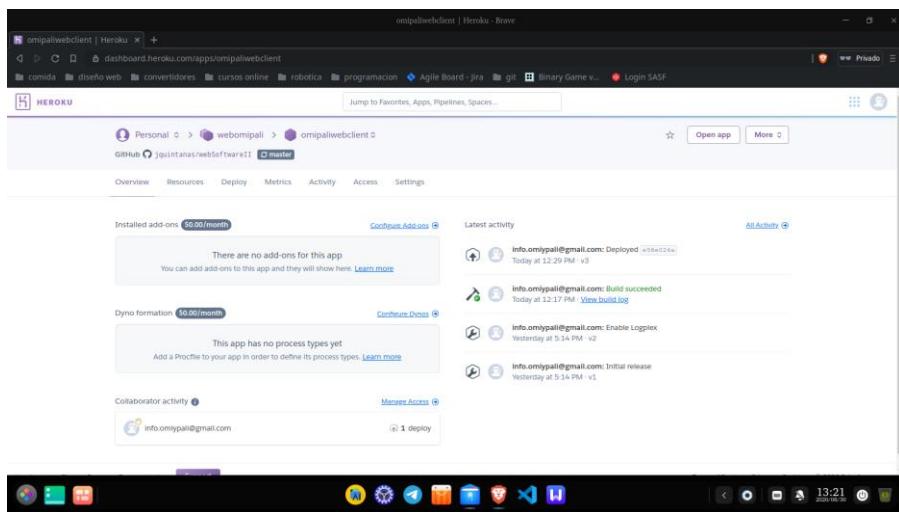
webomipali-master-cnbko7codyh master Deployed today at 1:18 PM

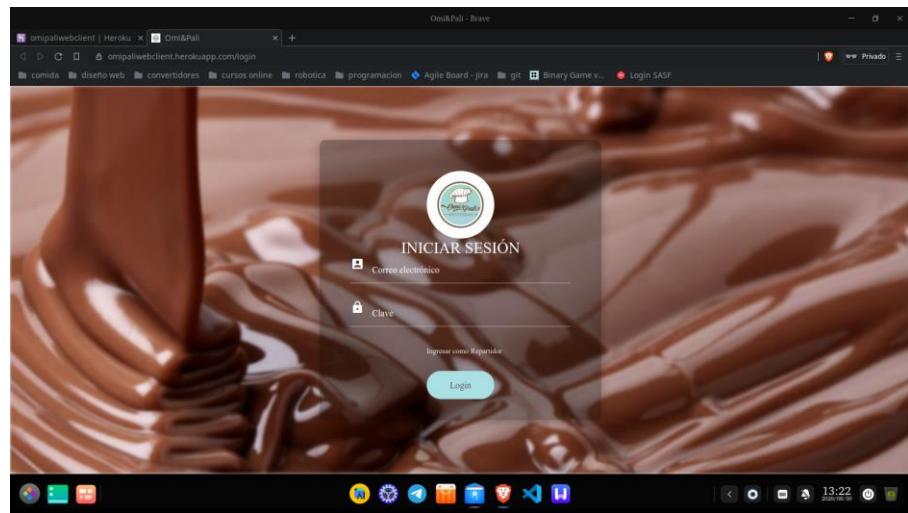
Terms of Service Privacy Cookies © 2020 Salesforce.com

## Heroku deployment configuration with git



## App deployment screenshots





## **15. ANNEXES**

### **15.1. DOCUMENTATION**

The code documentation could be found in the repository:

[https://jquintanas.github.io/restApiSoftware1/index.html#toc8\\_anchor](https://jquintanas.github.io/restApiSoftware1/index.html#toc8_anchor)

### **15.2. GITHUB-SERVER**

<https://github.com/jquintanas/restApiSoftware1>

### **15.3. GITHUB-WEBAPP**

<https://github.com/kbburgos/webSoftware1>

### **15.4. GITHUB-MOBILEAPP**

<https://github.com/jquintanas/ionicSoftware1>

### **15.5. GITHUB-CODING STANDARD DOCUMENTATION (AIRBNB)**

<https://github.com/airbnb/javascript>

### **15.6. DRIVE PRE-RECORDER-VIDEO**

<https://drive.google.com/drive/folders/1FHu7gjrJFBwn5KTfz15ebFiKNuLxkIg2?usp=sharing>

## **15.7. DRIVE MOBILE APP APK**

<https://drive.google.com/drive/folders/17WEuZO6VZMkFrDwgf5Es90DwYAuGD7h?usp=sharing>

## **15.8. DRIVE DOCUMENTATION (EVIDENCE)**

<https://drive.google.com/drive/folders/11iVKkD6q5DorN3eDOeLrxMFVs6f4D4ql?usp=sharing>

## 15.9. CLIENT ACCEPTANCE FORM (FIRST PARTIAL)

### DATOS DE CONTROL DEL DOCUMENTO

Tabla 1 Historia del documento

Fecha	Versión
08/07/2020	1.0

Tabla 2 Personas involucradas

Nombre	Rol
Francesca Helena Man Ging Pow Hing	Scrum Master
Jonathan Israel Quintana Solorzano	Product Owner
Danny Paul Ríos Benavides	Desarrollador
Karla Belén Burgos Gayrey	Desarrollador
Brenda Michelle Bermello Burgos	Desarrollador

Tabla 3 Evaluación de la funcionalidad del entregable

Funcionalidad	¿Cumple?	Observaciones
Inicio de sesión	Si	Al iniciar sesión en la parte móvil se redirige directamente al menú de bebidas.
Gestión de compra (lista de favoritos)	Si	
Mostrar productos disponibles.	Si	
Añadir productos al carrito de compra.	Si	
Confirmar pedido.	Si	
Monitoreo de pedido.	Si	No hay enlace directo entre detalles y carrito de compra.
Pago de pedido	Si	

#### 4. Propuestas de mejoramiento

- Agregar botón de enlace entre la pantalla detalles de pedido y el carrito de compra
- Modificar el redirecccionamiento desde el login hasta la sección dulces.

## 15.10. CLIENT ACCEPTANCE FORM (SECOND PARTIAL)

Guayaquil 27 agosto, 2020

**ACTA DE ENTREGA DE PROYECTO**

**DATOS**

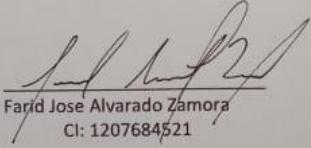
EMPRESA	GRUPO #4
PROYECTO	APP MOVIL: REALIZAR UNA COMPRA EN LINEA CON RASTREO DE PEDIDO APP WEB: MANEJO DE STOCK, PRODUCTOS Y COORDINACION DE ENTREGA DE PEDIDOS REALIZADOS EN EL APP MOVIL
FECHA DE INICIO	OCTUBRE 2019
CLIENTE	OMI & PALI
REPRESENTANTE	FARID JOSE ALVARADO ZAMORA

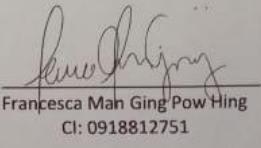
**GRUPO #4**

ROL	NOMBRE
SCRUM MASTER	FRANCESCA MAN GING
PRODUCT OWNER	JONATHAN QUINTANA
DESAROLLADOR	DANNY RIOS
DESAROLLADOR	KARLA
DESAROLLADOR	BRENDA BERMELLO

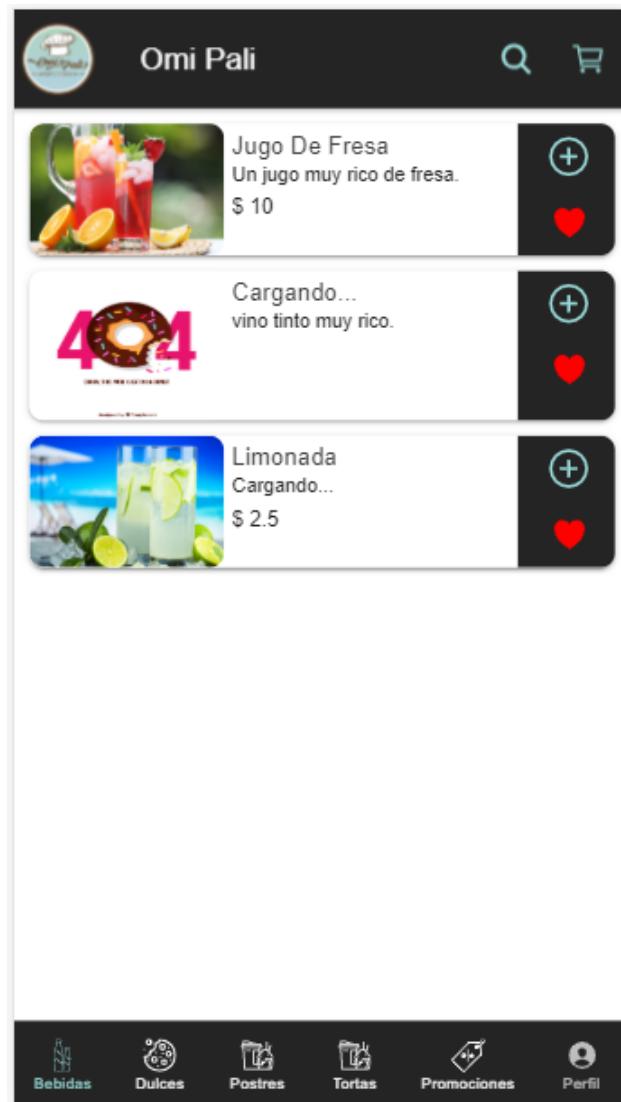
**ACEPTACION DE LOS PRODUCTOS ENTREGABLES**

ENTREGABLE	SI O NO	OBSERVACION
DISEÑO APP MOVIL	SI	
APP MOVIL: CUMPLE CON LOS REQUERIMIENTOS ESTABLECIDOS	SI	
DISEÑO APP WEB	SI	
APP WEB: CUMPLE CON LOS REQUERIMIENTOS ESTABLECIDOS	SI	

  
Farid Jose Alvarado Zamora  
CI: 1207684521

  
Francesca Mah Ging Pow Hing  
CI: 0918812751

## 15.11. MOBILE SCREENS



Omi & Pali

Subtotal: \$ 7

Imagen	Nombre	Descripción	Precio
	Producto1	Nueva	\$ 3
	Ggg	999	\$ 4

Bebidas Dulces Postres Tortas Promociones Perfil

Omi Pali

Subtotal : \$ 7

Imagen	Nombre	Precio
	Producto1	\$3.00
	Ggg	\$4.00

Subtotal (2 productos): \$ 7

COMPRAR

## Pedido

Confirmación del pedido

Datos del cliente

Nombres	Jonathan Perez
Cédula	0924995426
Teléfono	0939468609

Dirección de Entrega

Elija el tipo de entrega

Carrito

Producto	Cantidad	Precio
producto1	1	\$3.00
ggg	1	\$4.00

Cubiertos

Necesita cubiertos para su pedido

Pago

Elija el método de pago

## Pedido

Confirmación del pedido

Datos del cliente

Nombres	Jonathan Perez
Cédula	0924995426
Teléfono	0939468609

Dirección de Entrega

Elija el tipo de entrega

Carrito

Producto		
producto1		
ggg	1	\$4.00

Domicilio

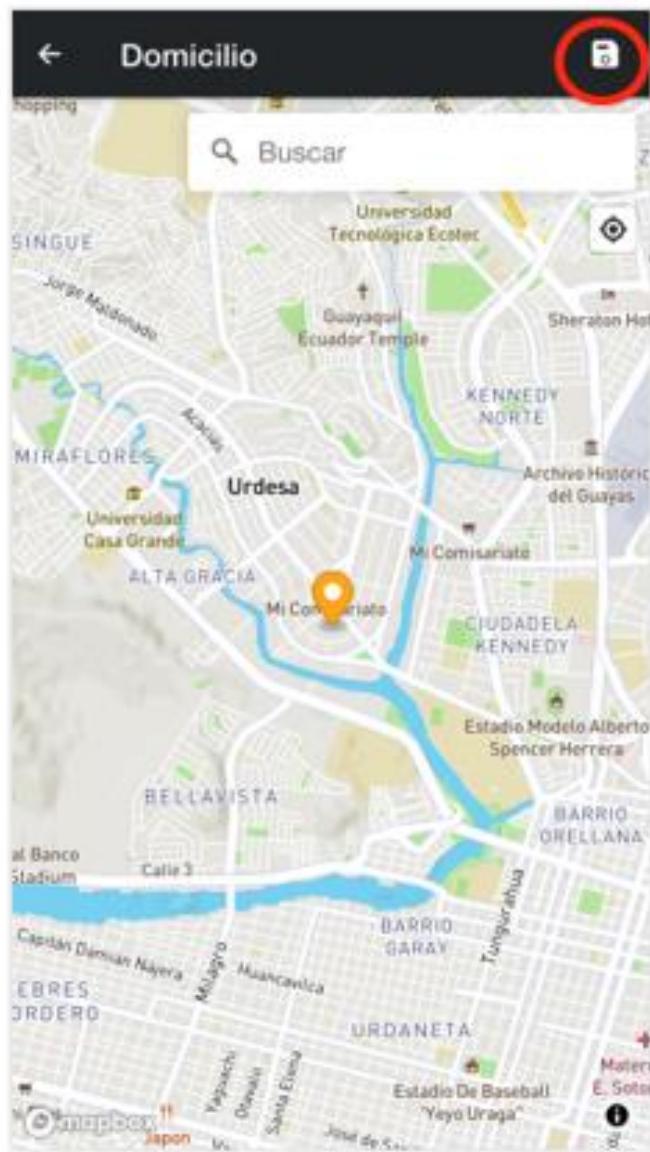
Retiro en Local

Cubiertos

Necesita cubiertos para su pedido

Pago

Elija el método de pago



**Pedido**

Referencia de la dirección añadida  
Casa blanca tres pisos

Tiempo estimado de espera 10 - 20 minutos

**Carrito**

Producto	Cantidad	Precio
producto1	1	\$3.00
999	1	\$4.00

**Cubiertos**

Necesita cubiertos para su pedido No +

**Pago**

Elija el método de pago Depósito +

**Resumen**

Subtotal	\$7.00
Envío	\$1.50
<b>Total</b>	<b>\$8.50</b>

**REALIZAR COMPRA**

**Pedido**

Referencia de la dirección añadida  
Casa blanca tres pisos

Tiempo estimado de espera 10 - 20 minutos

**Carrito**

Producto	Cantidad	Precio
producto1	1	\$3.00
999	1	\$4.00

**Cuentas Bancarias**

Banco Pichincha  
Cuenta de Ahorros #45789657479  
FARID ALVARADO Cl:1207684521  
Omiypali@gmail.com

Banco Guayaquil  
Cuenta de Ahorros #45789657479  
FARID ALVARADO Cl:1207684521  
Omiypali@gmail.com

**Pago**

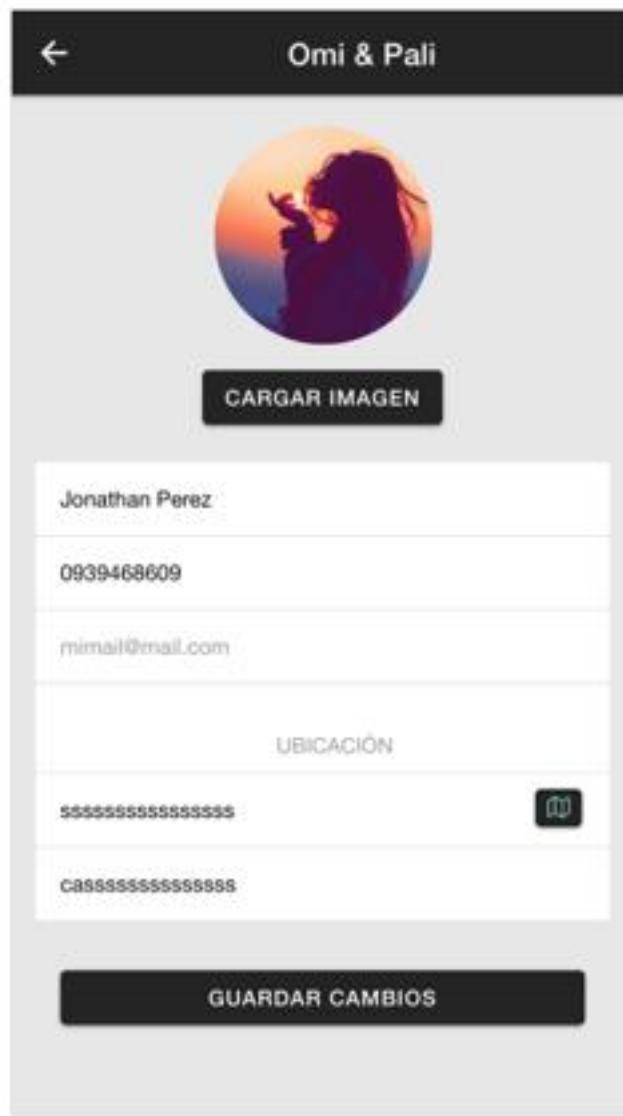
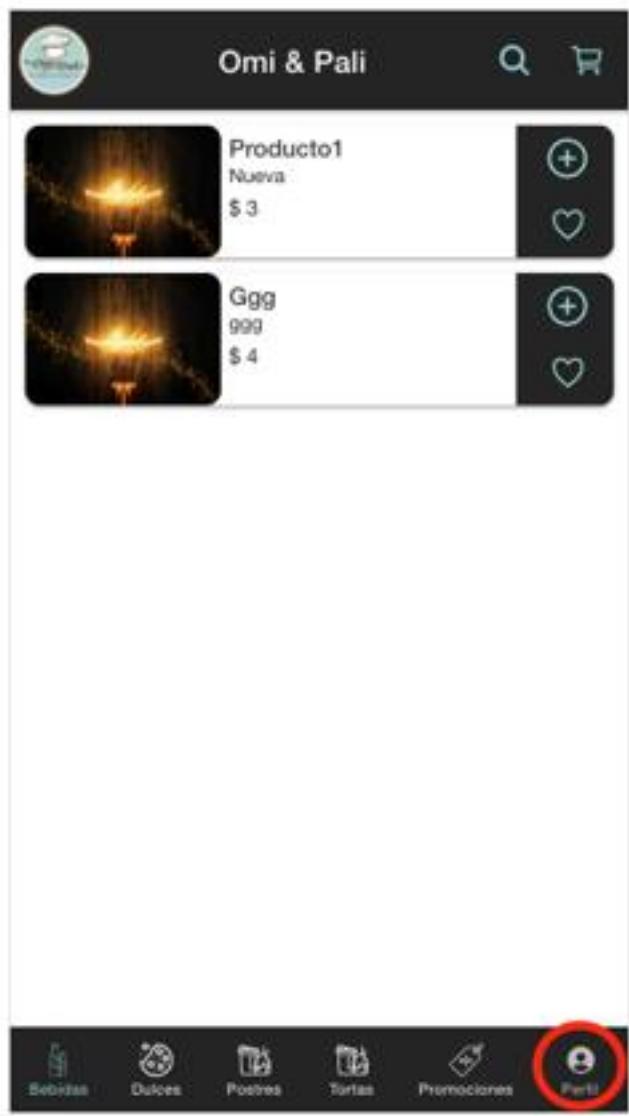
Elija el método de pago Depósito +

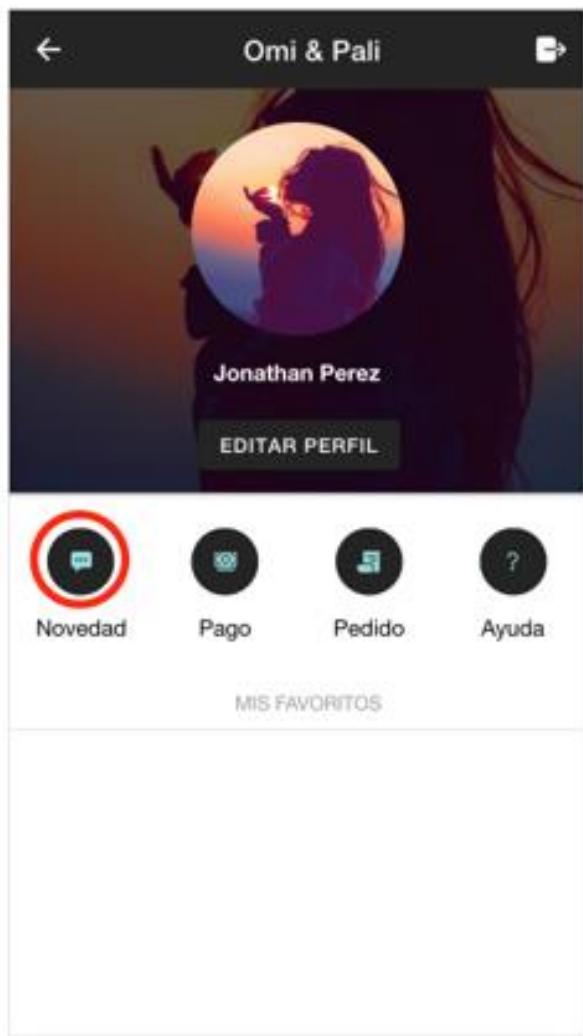
**Resumen**

Subtotal	\$7.00
Envío	\$1.50
<b>Total</b>	<b>\$8.50</b>

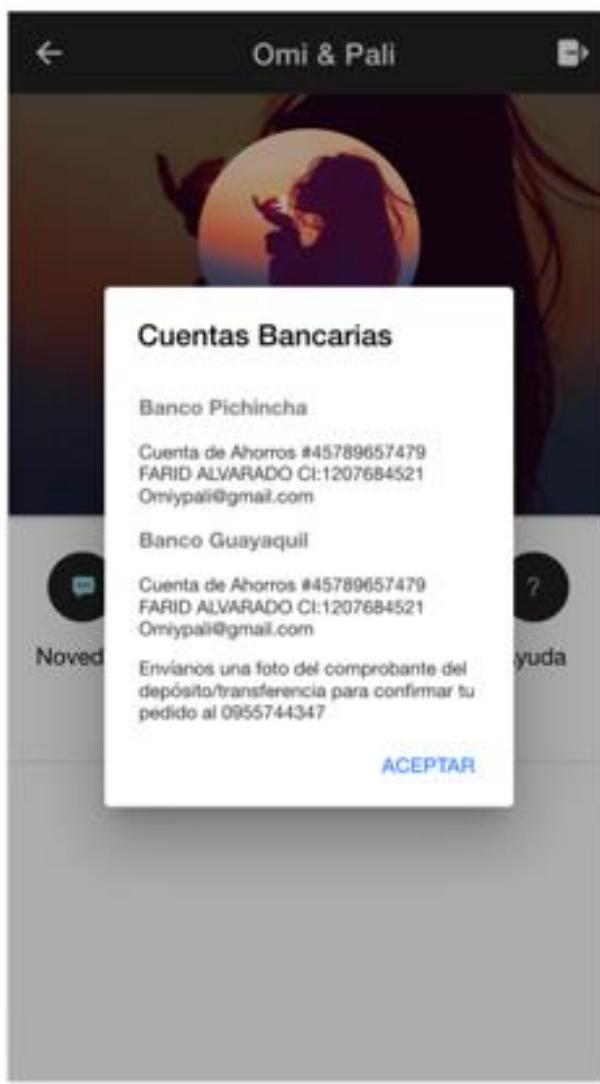
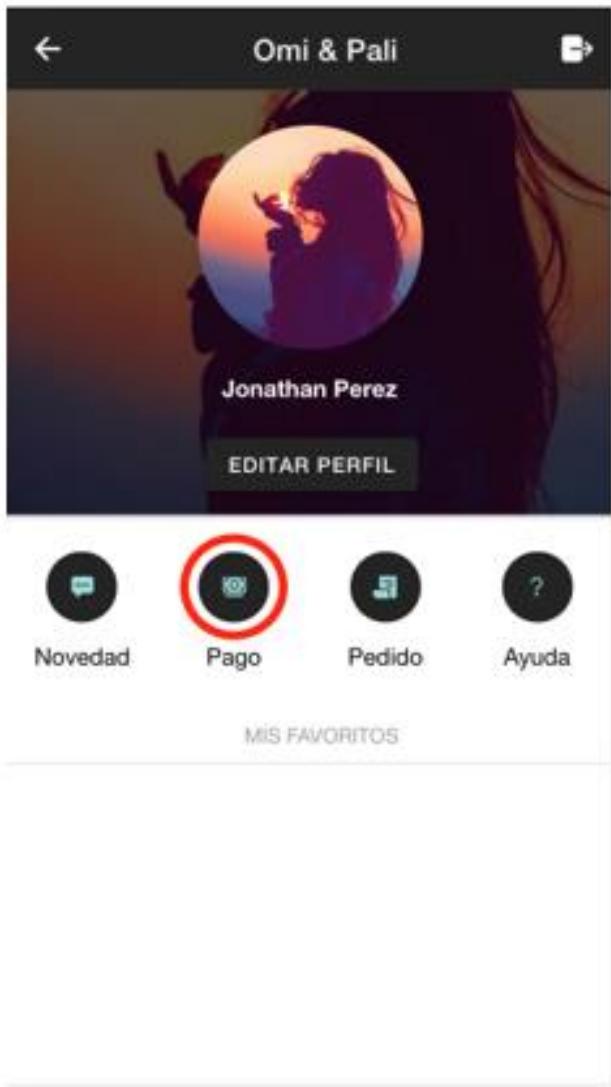
**ACEPTAR**

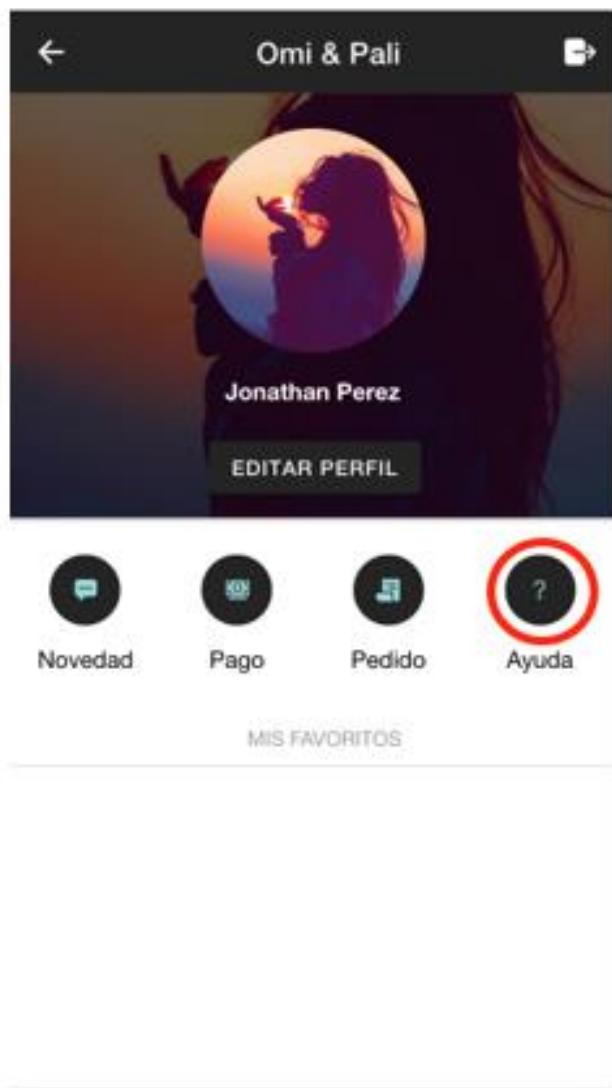
**REALIZAR COMPRA**

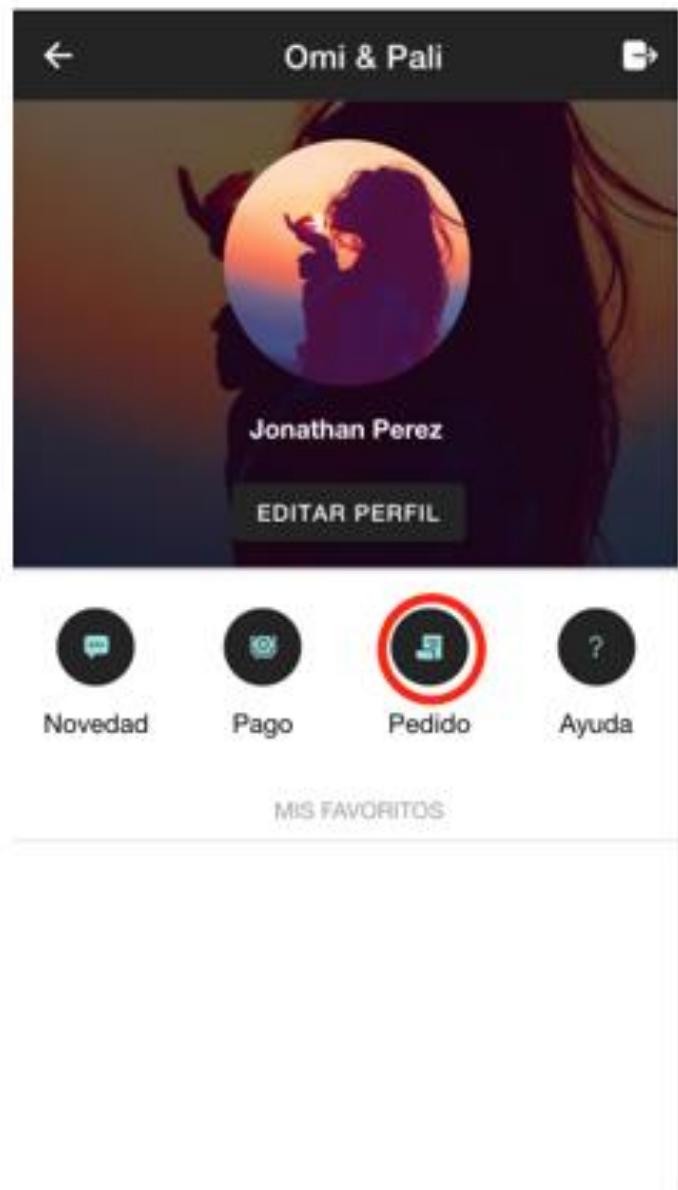


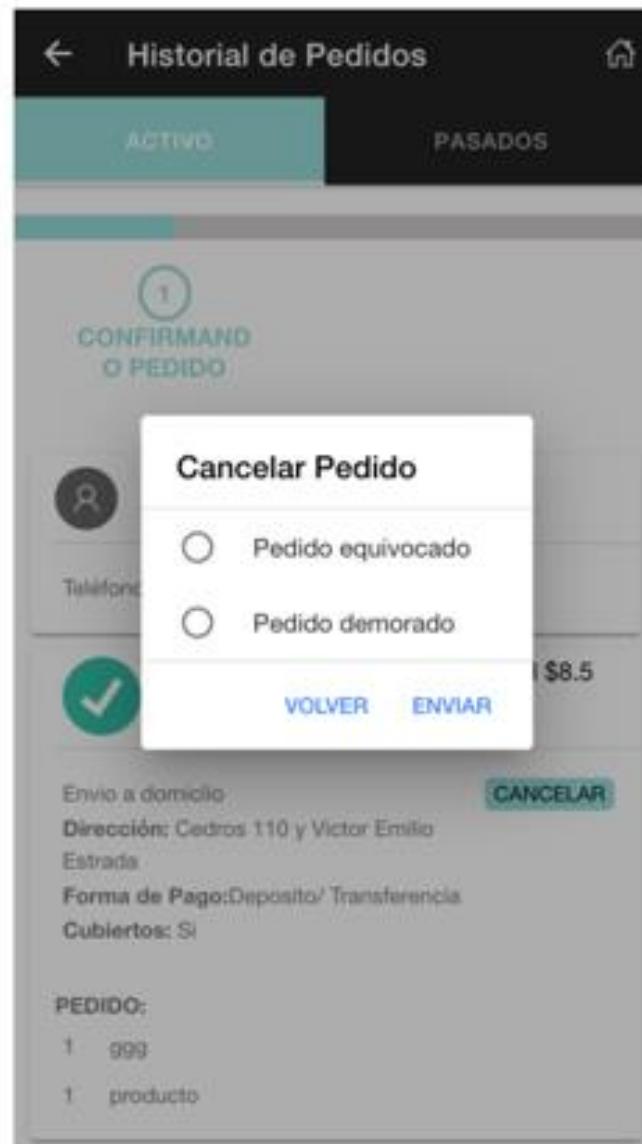


- ## Novedades
- NOVEDAD #2
- USUARIO REPORTADO:  
0968645215
- DESCRIPCIÓN:  
pedEquivocado
- NOVEDAD #1
- USUARIO REPORTADO:  
0968645215
- DESCRIPCIÓN:  
pedEquivocado
- NOVEDAD #4
- USUARIO REPORTADO:  
0910314558
- DESCRIPCIÓN:  
pedEquivocado
- NOVEDAD #3
- USUARIO REPORTADO:  
0968645215
- DESCRIPCIÓN:





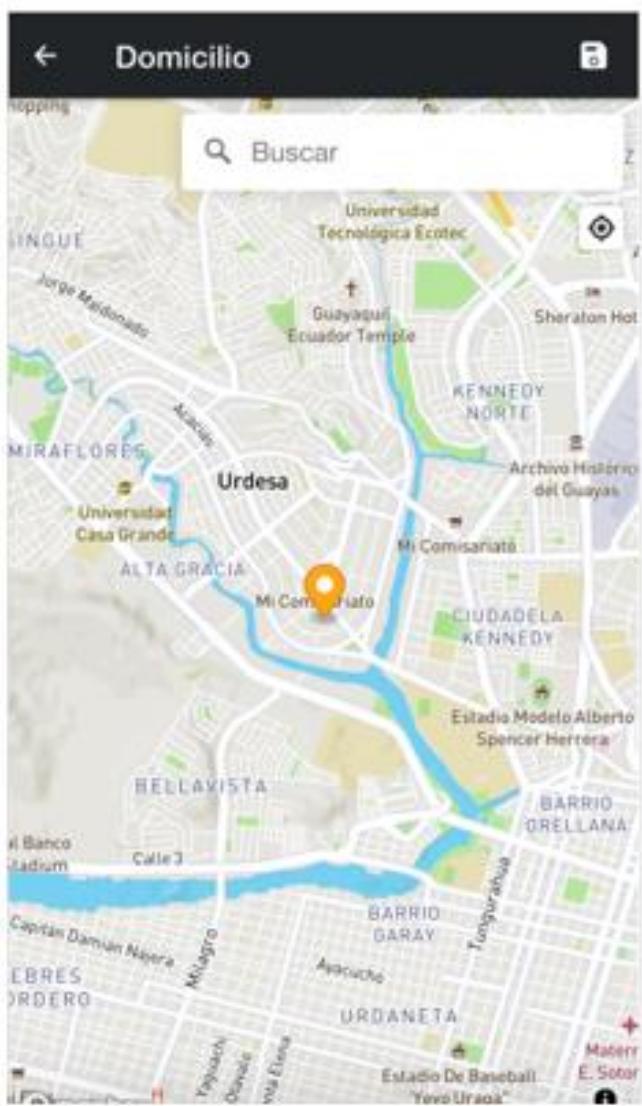






The image shows the registration screen titled "Nueva Cuenta" (New Account). The title is at the top left, with a back arrow icon. Below it is a section titled "Datos Personales" (Personal Data) with a horizontal line. The form contains six input fields: "Cédula" (ID), "Nombres" (Names), "Apellidos" (Last Name), "Teléfono" (Phone), "Correo Electrónico" (Email), and "Contraseña" (Password). Below these is a "Confirmar Contraseña" (Confirm Password) field. At the bottom is a large green button labeled "SIGUIENTE" (Next). The background features a dark theme with a central image of a chocolate cake.

Datos Personales	
Cédula	
Nombres	
Apellidos	
Teléfono	
Correo Electrónico	
Contraseña	
Confirmar Contraseña	
<b>SIGUIENTE</b>	



← Nueva Cuenta

Datos Personales

Dirección de domicilio

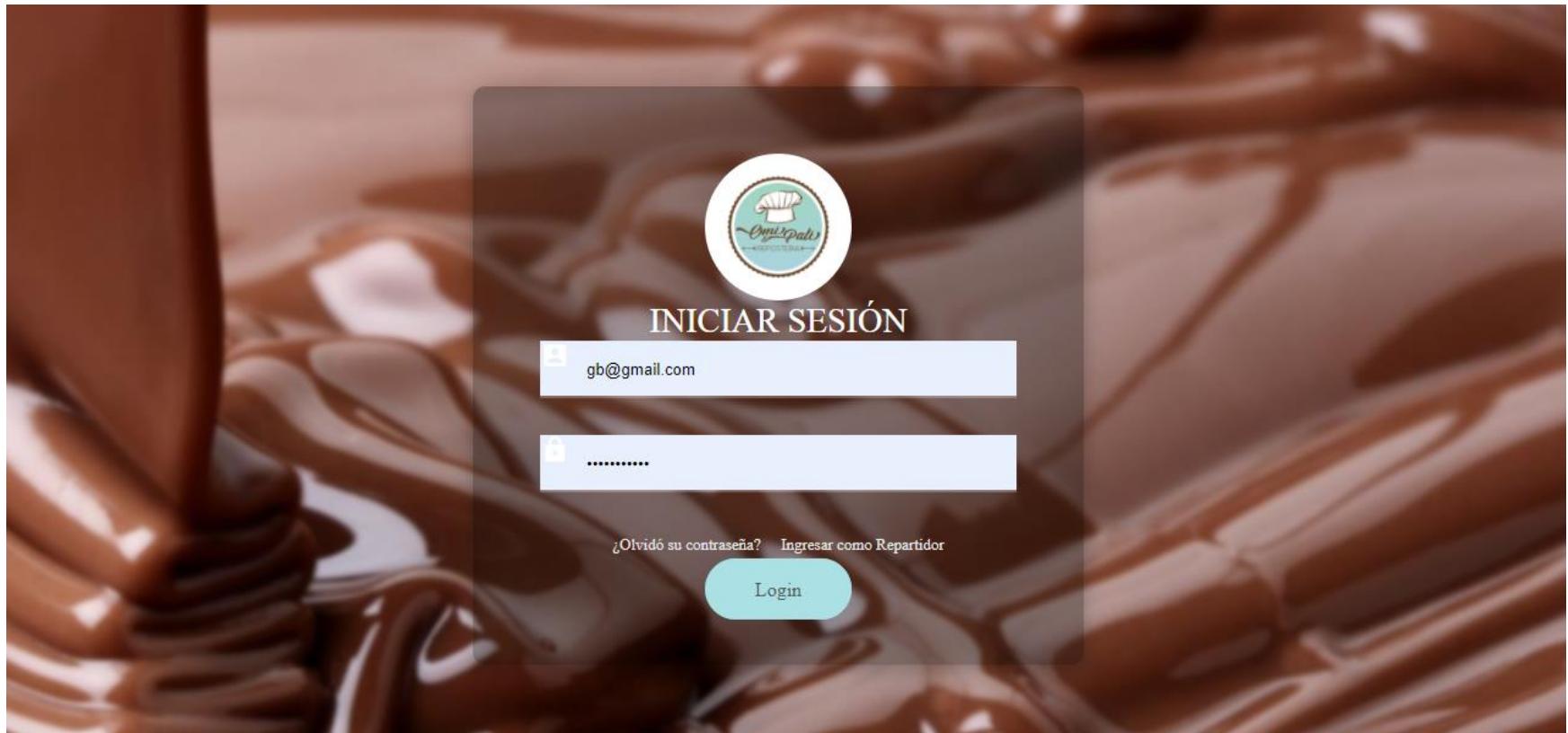


Puede indicar alguna referencia de su domicilio

REGISTRARSE



## 15.12. WEB SCREENS



OMI&PALI

Dashboard

Pedidos

Productos

Promociones

Usuarios

Novedades

Clients

## Dashboard

**VENTAS TOTALES** \$27.5

Ganancia de acuerdo a la cantidad de pedidos

**PEDIDOS LOCAL** 0

Número de pedidos realizados en local

**PEDIDOS DOMICILIO** #1

Número de pedidos realizados a domicilio

### Pedidos Entrantes

Cantidad de pedidos: 2

Ordenar por ▾

1. TRntGotjw2teq9kVlKIZ	pedido a domicilio ----- Total a pagar: \$5.5	(i)
2. fnP4jjTksBH47mhb6QYz	pedido a domicilio ----- Total a pagar: \$38.5	(i)

### Stock Productos

Cantidad de productos: 17

Ordenar por ▾

1. hello	producto activo ----- Stock: 4	(i)
2. promocion5	producto activo ----- Stock: 3	(i)
3. vbNueva	----- Stock: 2	(i)

OMI&PALI

Dashboard

Pedidos ▾

Productos

Promociones ▾

Usuarios

Novedades ▾

Clients ▾

## Pedidos en espera

Lista de pedidos en espera

Simbología

Pedidos a Domicilio Pedidos en Local

PEDIDO	CLIENTE	PRODUCTOS	CUBIERTOS	TOTAL A PAGAR	ACCIONES
EjGh1cvjQGn8SpGpkk8Q	0924995426	<a href="#">ver detalle</a>	si	8.5	

1

OMI&PALI

Dashboard

Pedidos ▾

Productos

Promociones ▾

Usuarios

Novedades ▾

Clientes ▾

## Pedidos asignados

Lista de Pedidos a Domicilio Asignados

> Simbología					
PEDIDO ↑↓	CLIENTE ↑↓	PRODUCTOS ↑↓	CUBIERTOS	TOTAL A PAGAR	DETALLES
18S0JHbESEV158LR6a	0918812751	<a href="#">ver detalle</a>	si	24.5	
eSmMIQYPQQVDL8MYhyL1	0924995426	<a href="#">ver detalle</a>	si	8.5	

1

OMI&PALI

Dashboard

Pedidos

Productos

Promociones

Usuarios

Novedades

Clientes

## Pedidos despachados

### Lista de Fedidos Despachados

PEDIDO ↑↓	PRODUCTOS ↑↓	CUBIERTOS	ENTREGA	TOTAL A PAGAR
1	<a href="#">ver detalle</a>	si	Domicilio	\$27.5

1

OMI&PALI

Dashboard

Pedidos

Productos

Promociones

Usuarios

Novedades

Clientes

## Productos

+Nuevo

NOMBRE ↑↓	DESCRIPCIÓN ↑↓	CATEGORÍA ↑↓	PRECIO ↑↓	STOCK ↑↓	FOTO	ACCIONES
producto	Nueva	dulces	3	3		
nuevoProducto	nuevo producto	tortas	23	23		
fiesta23	fiesta23	dulces	23	23.5		
nuevo prueba modal	prueba de modal	dulces	34.5	10		
888	888	bebidas	4	4		

1

OMI&PALI

Dashboard

Pedidos

Productos

Promociones

Usuarios

Novedades

Clientes

## Activas

+Nuevo

Nombre ↑↓	Descripcion ↑↓	Precio ↑↓	Stock ↑↓	FOTO	ACCIONES
hello	hell4	3	4		
promocion5	promocion5	34	3		
vbNueva	vb	3	3		
promocion122	promocion1 un dulce muy rico	2	3		
sd	sd3	3	3		

1 2

OMI&PALI

Inactivas

Lista de Promociones Inactivas

Nombre ↑↓	Descripcion ↑↓	Precio ↑↓	Stock ↑↓	FOTO	ACCION

Nombre ↑↓ Descripcion ↑↓ Precio ↑↓ Stock ↑↓ FOTO ACCION

1

Dashboard

Pedidos ▾

Productos

Promociones ▾

Usuarios

Novedades ▾

Ciudades ▾ Clientes ▾

OMI&PALI

Dashboard

Pedidos ▾

Productos

Promociones ▾

**Usuarios**

Novedades ▾

Clientes ▾

## Usuarios

### Lista de Usuarios

+Nuevo

CÉDULA	NOMBRES	ROL	EMAIL	DIRECCIÓN	TELÉFONO	ACCIONES
0000111111	repartidor repartidor	Vend	repartidorOmiPali@gmail.com	sin dirección	1111111111	
0101017895	alanri avendano	Vend	av@gmail.com	guangala Mz E8 V21	5555555599	
0205896471	usuario1 nuevo	Admin	nuevo@gmail.com	nueva	0258025802	
0874158962	Gabriel Cantos	Admin	gb@gmail.com	Base Naval Sur	0968918111	
0952852036	Karla Burgos	Admin	kb@gmail.com	Guangala	0909080705	

1

OMI&PALI

Dashboard

Pedidos

Productos

Promociones

Usuarios

Novedades

Clientes

## Novedades de clientes a repartidores

### Novedades por administrador

CÉDULA	REPARTIDOR	NOVEDAD
0910314558	Andrea Camila Fajardo Andrade	prueba
0968645215	Carlos Ramón Lozano Vergara	prueba2
0968645215	Carlos Ramón Lozano Vergara	Hola

+Nuevo

1 2

### Novedades por usuarios

CÉDULA	CLIENTE	CÉDULA	REPARTIDOR	DESCRIPCIÓN
0924995426	Jonathan Perez	0968645215	Carlos Ramón Lozano Vergara	pedEquívocado
0924995426	Jonathan Perez	0968645215	Carlos Ramón Lozano Vergara	pedEquívocado
0924995426	Jonathan Perez	0968645215	Carlos Ramón Lozano Vergara	pedEquívocado
0924995426	Jonathan Perez	0910314558	Andrea Camila Fajardo Andrade	pedEquívocado
0918812751	Francesca Man Ging	0968645215	Carlos Ramón Lozano Vergara	false

1 2 3

OMI&PALI

- Dashboard
- Pedidos ▾
- Productos
- Promociones ▾
- Usuarios
- Novedades ▾**
- Ciudades
- Clientes ▾

## Novedades de repartidores a clientes

### Novedades por administrador

CEDULA	CLIENTE	NOVEDAD

+Nuevo

1

### Novedades por repartidores

CÉDULA	REPARTIDOR	PEDIDO	CÉDULA	CLIENTE	NOVEDAD	FECHA
0968645215	Carlos Ramón Lozano Vergara	GZB0kBTa3CQF9UTd2hHT	0924995426	Jonathan Perez	Cliente falta respeto	Wed Aug 19 2020 22:34:07 GMT-0500 (hora de Ecuador)
0955744347	Danny Paúl Ríos Benavides	VcOB9YCAaLCNGrqDJWid	0924995426	Jonathan Perez	cliente molesto	Tue Aug 25 2020 22:09:50 GMT-0500 (hora de Ecuador)
0968645215	Carlos Ramón Lozano Vergara	3F1UjEdCvs7eEkfyDPQ5	0924995426	Jonathan Perez	cliente molesto	Thu Aug 20 2020 23:35:28 GMT-0500 (hora de Ecuador)
0955744347	Danny Paúl Ríos Benavides	252dcz	0968918012		cliente molesto	Wed Jul 08 2020 00:00:00 GMT-0500 (hora de Ecuador)

OMI&PALI

Clientes registrados

Lista de Clientes Registrados

CEDULA	NOMBRE	APELLIDO	EMAIL	DIRECCION	TELEFONO
0918812751	Francesca	Man Ging	fman@espol.edu.ec	Cedros 111 y Victor Emilio Estrada	0997850751
0924995426	Jonathan	Perez	mimail@mail.com	5555555555555555	0939468608
0924995427	Jonathan	Quintana	jonathan@gmail.com	5555555555555555	0939468608

1

- Dashboard
- Pedidos ▾
- Productos
- Promociones ▾
- Usuarios
- Noticias ▾
- Clientes ▾

OMI&PALI

Clientes conflictivos

Lista de clientes registrados

CEDULA	NOMBRE	APELLIDO	EMAIL	DIRECCION	TELEFONO
<input type="text"/>	<input type="text"/>	<input type="text"/>			
0924995426	jonathan	Perez	mimail@mail.com	ssssssssssssssss	0939468608
<span>&lt;</span> <span>&lt;</span> <span>1</span> <span>&gt;</span> <span>&gt;</span>					

Dashboard

Pedidos ▾

Productos

Promociones ▾

Usuarios

Noticias ▾

Clientes ▾

Salir 

### Finalizar Pedido

CEDULA_CLIENTE ↑↓	CLIENTE ↑↓	PEDIDO ↑↓	PRODUCTOS ↑↓	ACCIONES
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
0924995426	Jonathan Perez	eSmMIQYPQQVDL8MYhyL1	<a href="#">ver detalle</a>	 Finalizar

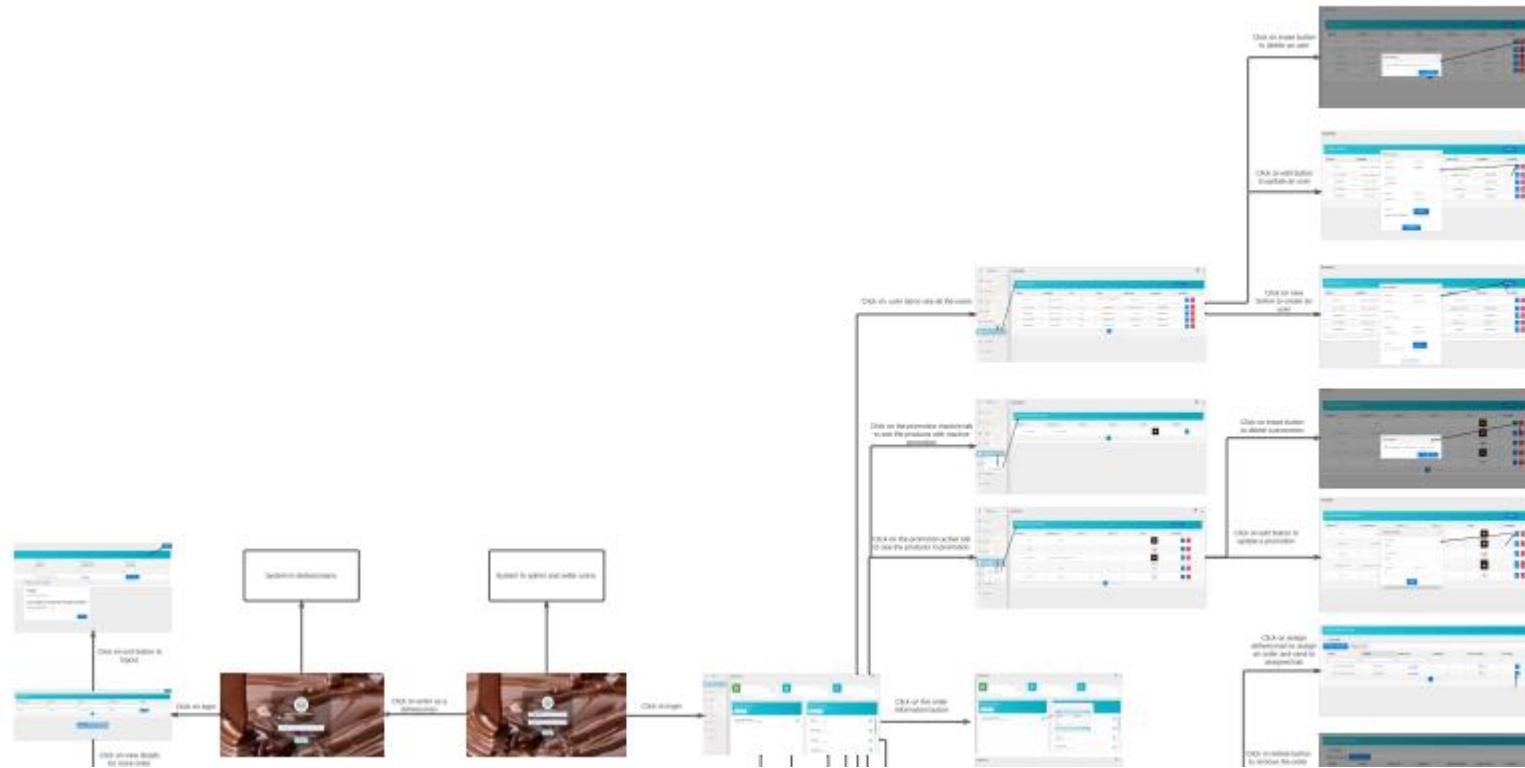


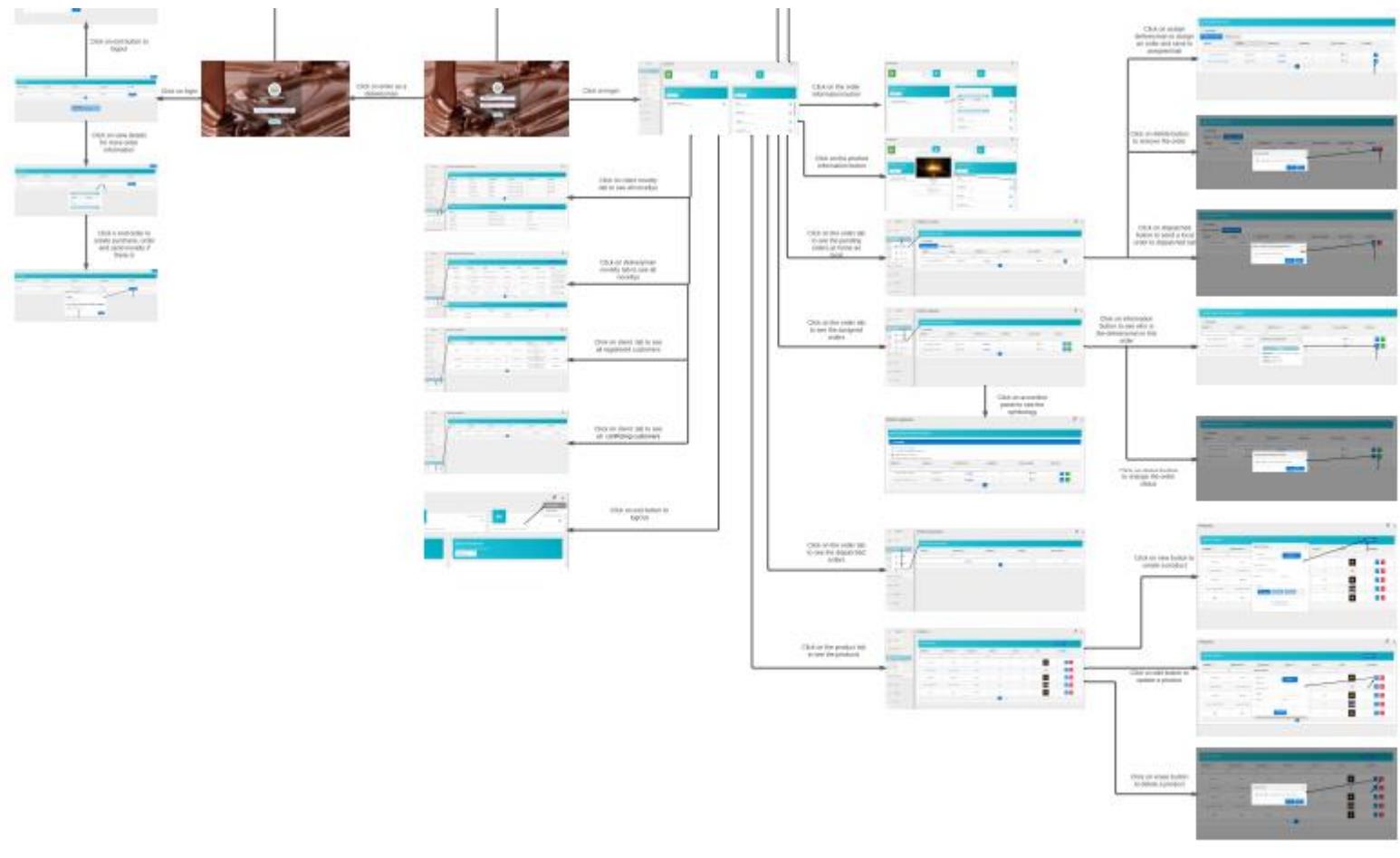
BIENVENIDO

Carlos Ramón Lozano Vergara

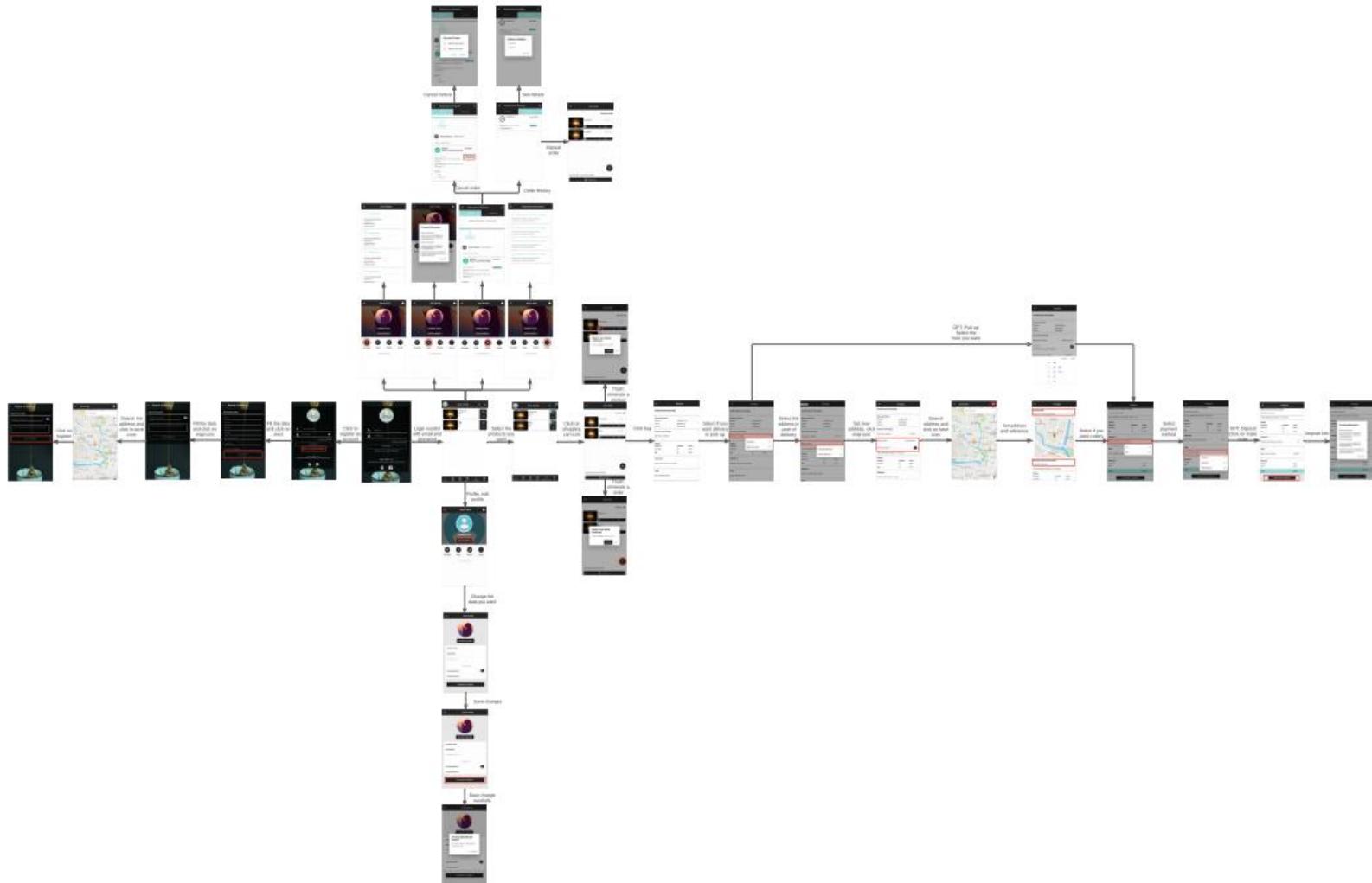


## 15.13. WEBAPP WINDOW FLOW





## 15.14. MOBILE APP WINDOW FLOW



## 16. BIBLIOGRAPHY

11 razones por las que Git triunfa como el sistema de control de versiones más popular entre los desarrolladores. (2019, 4 17). Retrieved from <https://www.sivsa.com/site/ventajas-controlador-versiones-git-en-programacion/>

atinternet. (n.d.). *ELABORACIÓN DE PERFILES*. Retrieved from <https://www.atinternet.com/es/glosario/elaboracion-de-perfiles/>

Mora, L. (n.d.). *Securizando tus arquitecturas*. Retrieved from <https://www.paradigmadigital.com/dev/securizando-tus-arquitecturas-de-microservicios-con-istio/>

OBS. (n.d.). *Project Management*. Retrieved from ¿Qué es un Project Manager y cuáles son sus tareas?: <https://obsbusiness.school/es/blog-project-management/salidas-profesionales-pm/que-es-un-project-manager-y-cuales-son-sus-tareas>

TRYQA. (n.d.). *What is Defect or bugs or faults in software testing?* Retrieved from <http://tryqa.com/why-is-testing-necessary/#:~:text=Software%20Testing%20is%20necessary%20because%20we%20all%20make%20mistakes.&text=Software%20testing%20is%20very%20important,made%20during%20the%20development%20phases.>

Wikipedia. (n.d.). *Npm*. Retrieved from <https://es.wikipedia.org/wiki/Npm>

Yadiz. (n.d.). Retrieved from <https://yodiz.com/help/agile-scrum-meetings/>

Bouchefra, A. (04 de 08 de 2020). *Techiediaries*. Obtenido de Techiediaries: <https://www.techiediaries.com/angular-firebase/angular-9-firebase-authentication-email-google-and-password/>

Google. (s.f.). *Firebase*. Obtenido de Firebase:  
<https://firebase.google.com/docs/auth/web/start>