ARQUITECTURA DE SOFTWARE

# PLANNING SPRINT 1

SEBASTIAN LOPEZ

JUAN SEBASTIAN MENDOZA MONCADA

JOSEPH IMANOL REYES CHAPARRO

DOCENTE:

CARLOS EDUARDO MUJICA REYES

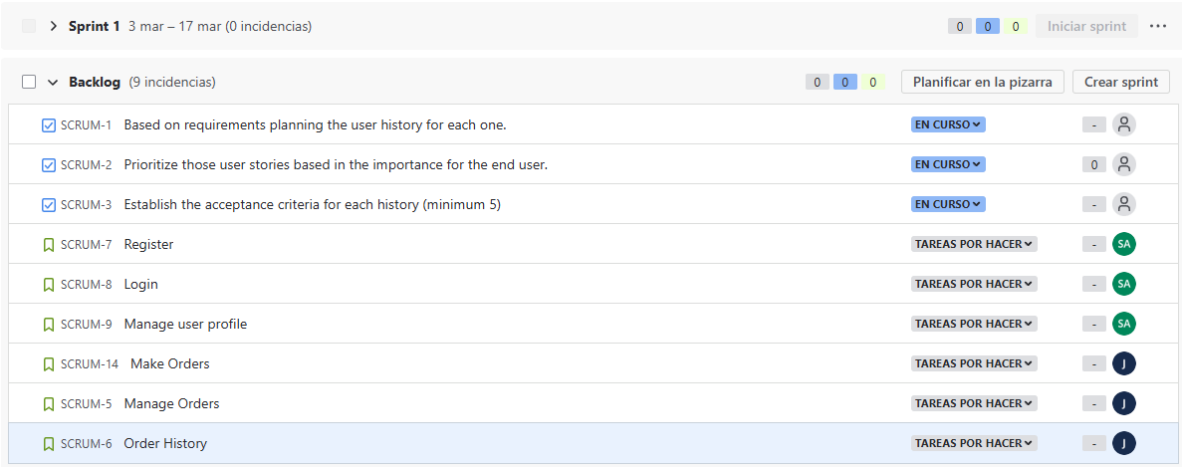
UNIVERSIDAD MANUELA BELTRAN

2025

# Product Backlog

Figure 1

Backlog board




Note: On this board, three user area requirements and three order area requirements were established, at the same time the 2-week sprint is proposed.

**Selected user stories**

**Manage Orders**

As a User with the necessary credentials to operate the module, I want to manage the orders that the system shows to control them, as those orders who are already checked as those who are in wait, all this to operate the system, know what the client wants and warning if something happens with the order.

## Acceptance criteria

1. The order must be seen as checked or in wait to be checked.
2. The orders made by the client must be shown in the screen of the User that attended the order.
3. The order would show the total amount, total IVA, and the product that is being purchased.
4. The order will have the next data of the client.
    a. The name of the client.
    b. The Address of the client.
    c. The date that the client made the purchase.
    d. And in some cases, depending on the amount of purchase the ID card.
5. The user with the credentials can warning by message to client if something happens with the order or the product.

**Order History**

As a User, I want to have access to the history of the orders that I made, with this have access to the information, different total ammounts, access to the information of the product that was purchased and the state of the order.

## Acceptance criteria

1. The Client can access to the history of the orders that were made and access to the information.
2. The information shown must be all related to the orders or purchase that the client made.
3. The orders must be seen, It doesn't matter if these were a long time ago purchased.

4. The Client can see in the History of orders the time that the product arrived, and when the order was made.
5. The orders must be seen by the state, if the order was already made and attended, or if the order is in wait these must be shown with the respective state.

### Make Orders

As a User, I want to make orders at any moment, select a product and if I want to purchase It, go to the section of payment to do it.

## Acceptance criteria

1. The Client at the moment to pay has to give all the information that is needed to procees the buying.
2. The Client has the information of the total ammount to pay for the order before the payment is processed.
3. The order will appear in the History of Orders only if the payment process is valided.
4. The order will be send to the user with credentials to be attend and managed once the payment process is valided.
5. The system will show a message of success or disaprove depending of the state of the payment.

### Register

As a user, I want to register with my email, password and a unique username.

## Acceptance criteria

1. The system only receives alphanumeric data without special characters
2. The email entered must have an @ character indicating domain
3. The password must have 10 characters
4. The password must have a special character
5. If the nickname is already registered, the user must be informed to change it
6. There must be 2 password fields for the user to repeat it correctly.

### Login

As a user I want to be able to log in with my nickname/email.

## Acceptance criteria

1. The profile must be blocked for 5 minutes.
2. The email entered must be previously registered in the database
3. The password must correspond to the email
4. The user must authenticate using a captcha.
5. The login can only be attempted 3 times, after which the profile must be blocked.

### Manage user profile

As a user, I want to be able to change my email, password and profile nickname at any time, and at the same time I want to have the possibility of deleting the profile.

## Acceptance criteria

1. There cannot be any blank spaces
2. To change the password you must authenticate by email
3. You can only change the data every 2 weeks
4. The password cannot be changed to the current one
5. The new email must not have an account assigned

### Prioritization methods

For the prioritization of requirements, 2 methods were implemented, due to the need to prioritize the requirements based on the needs of the sprint.

The first factor for prioritization was based on the functionalities of the system. Let me explain, this factor is based on achieving that the main problem for which the ERP is designed is solved, as is the case of orders.

Then we use the dependency factor to know which would be the next requirements to prioritize, which are registration and login, since this becomes the first basis of user interaction with the ERP.

# Definition of Done

We decided to make a meeting via Online to define what is going to be our “Definition of Done”, We found different perspectives between us, but we achieve a conclusion to stablish the project when is finished, we are based on the time that we have to create documentation, schemas and also code, we take into account the fact of the tests and the acceptance criteria of different user stories, also, we defined guided by our strategy to define which user stories is more difficult than other, the strategy was used by Planning Pocker and our analysis from that activity.

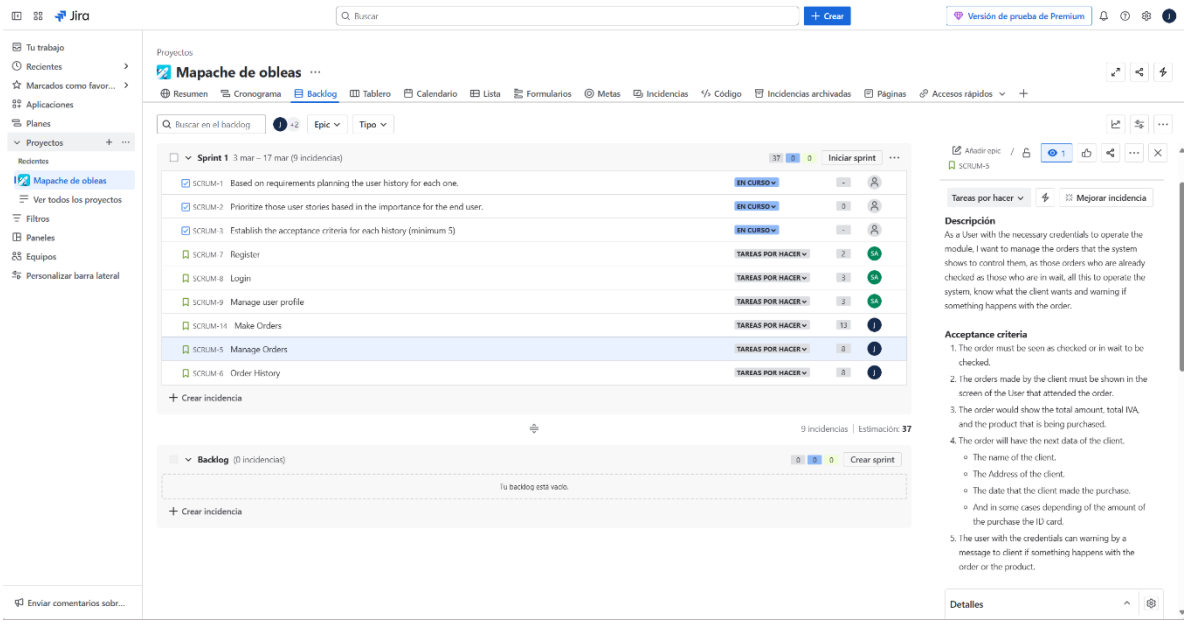
1. All the User Stories must be done, this is going to be based on the check of acceptance criteria.
2. Each User Storie is completed just when the acceptance criteria is valid and check, it has no exceptions, at least 80% of them must be finished.
3. The Sprints during the project are done when the code and test from that code are valid and successful.
4. At the moment we don’t include the production or the functionalities in real areas.
5. We can continue to another Sprint when we create the code part of the Sprint and in the test, we achieve at least three acceptance criteria.
6. Based on the difficulty of different User Stories, we will take the more difficult to do at the beginning, if all the most difficult User Stories are working properly the Sprint and therefore the Project could be determined as Done.

In conclusion is not possible to take this project into a real area of use, but is important to define that the main modules and functions at least 50% of them are working properly, we decided this because a matter of time to achieve It.

# Sprint Backlog

Figure 2

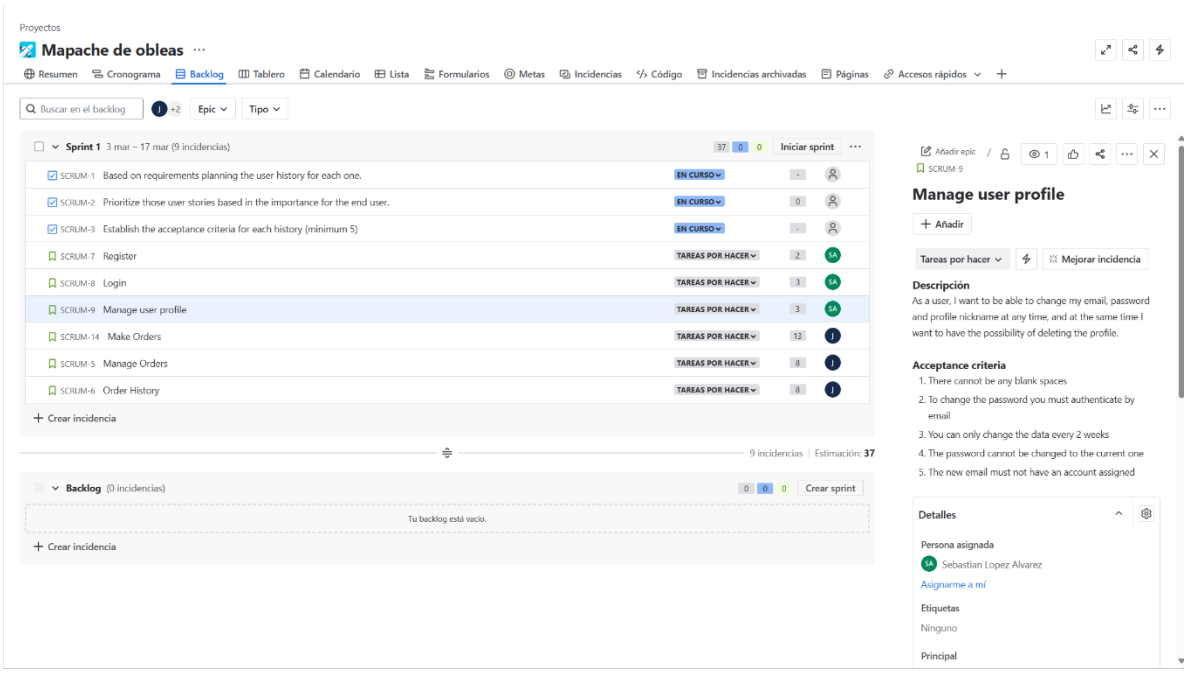
Jira Sprint One




Note: As we can see this is the project in Jira, the respective User Stories have the respective description and the Acceptance Criteria with minimum 5 of them.

Figure 3

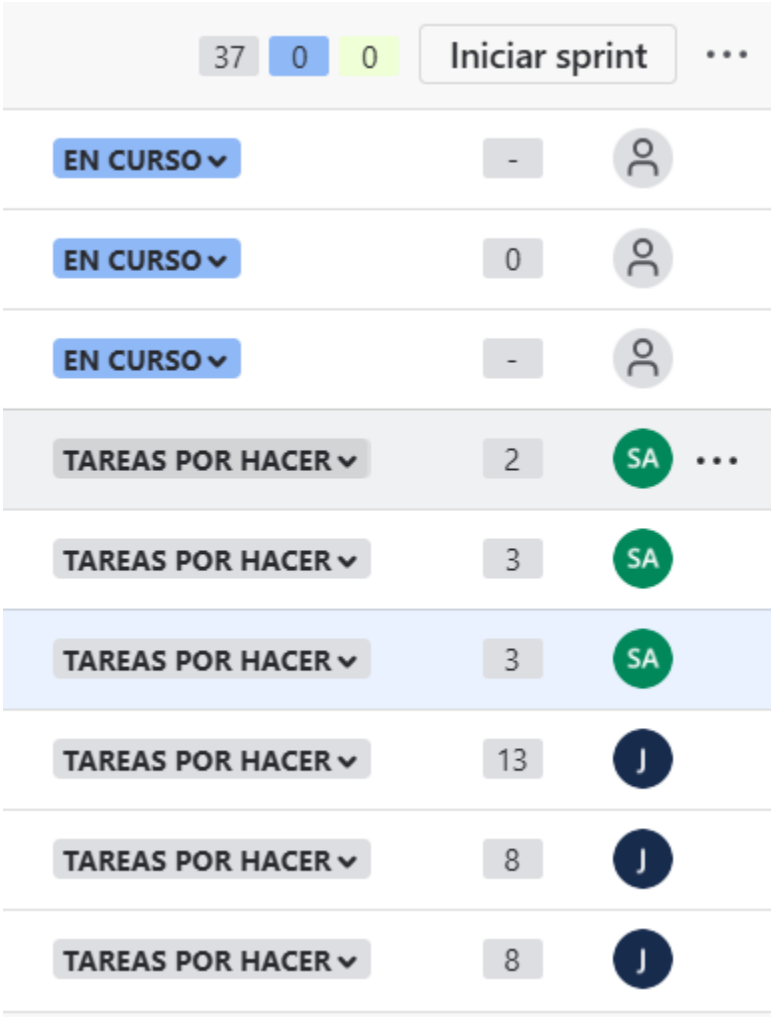
Jira Sprint One




Note: As we can see this is the project in Jira, the respective User Stories have the respective description and the Acceptance Criteria with minimum 5 of them, this is another example, all of the User Stories have this Schema, visit the link to the project to see all the Left User Stories.

Figure 4

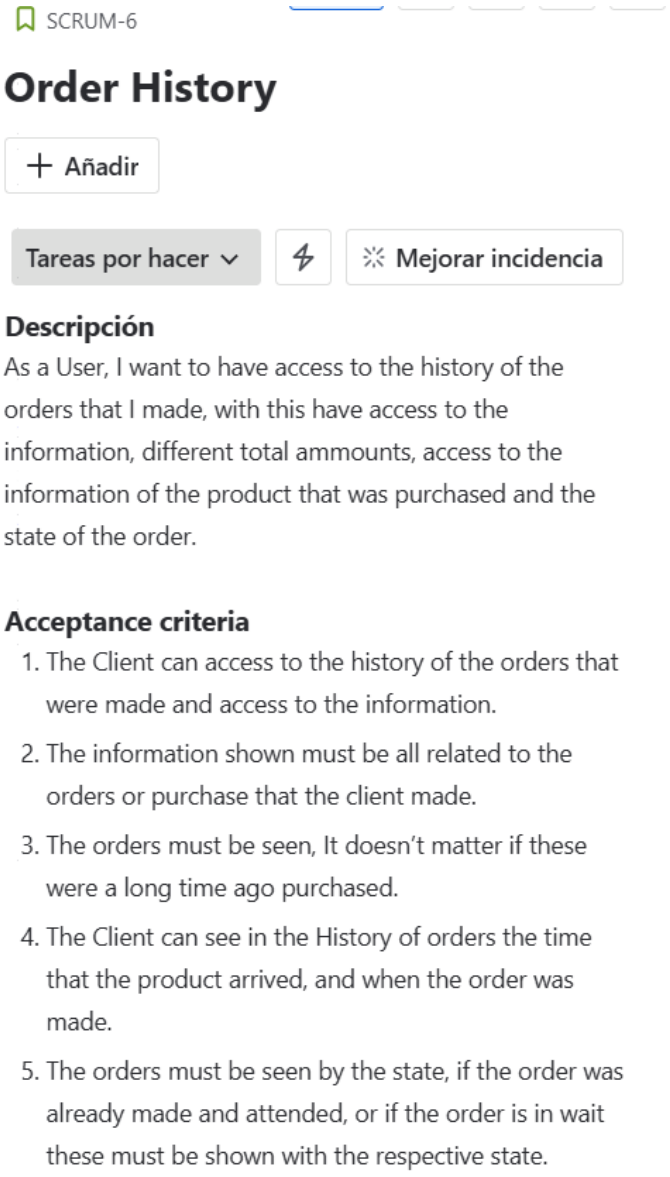
Jira Sprint One Score User Stories




Note: We can see the Points assigned to the different User Stories, this is about the complexity and the developers assigned who are going to develop the User Stories, the State of them.

Figure 5

Jira Sprint One



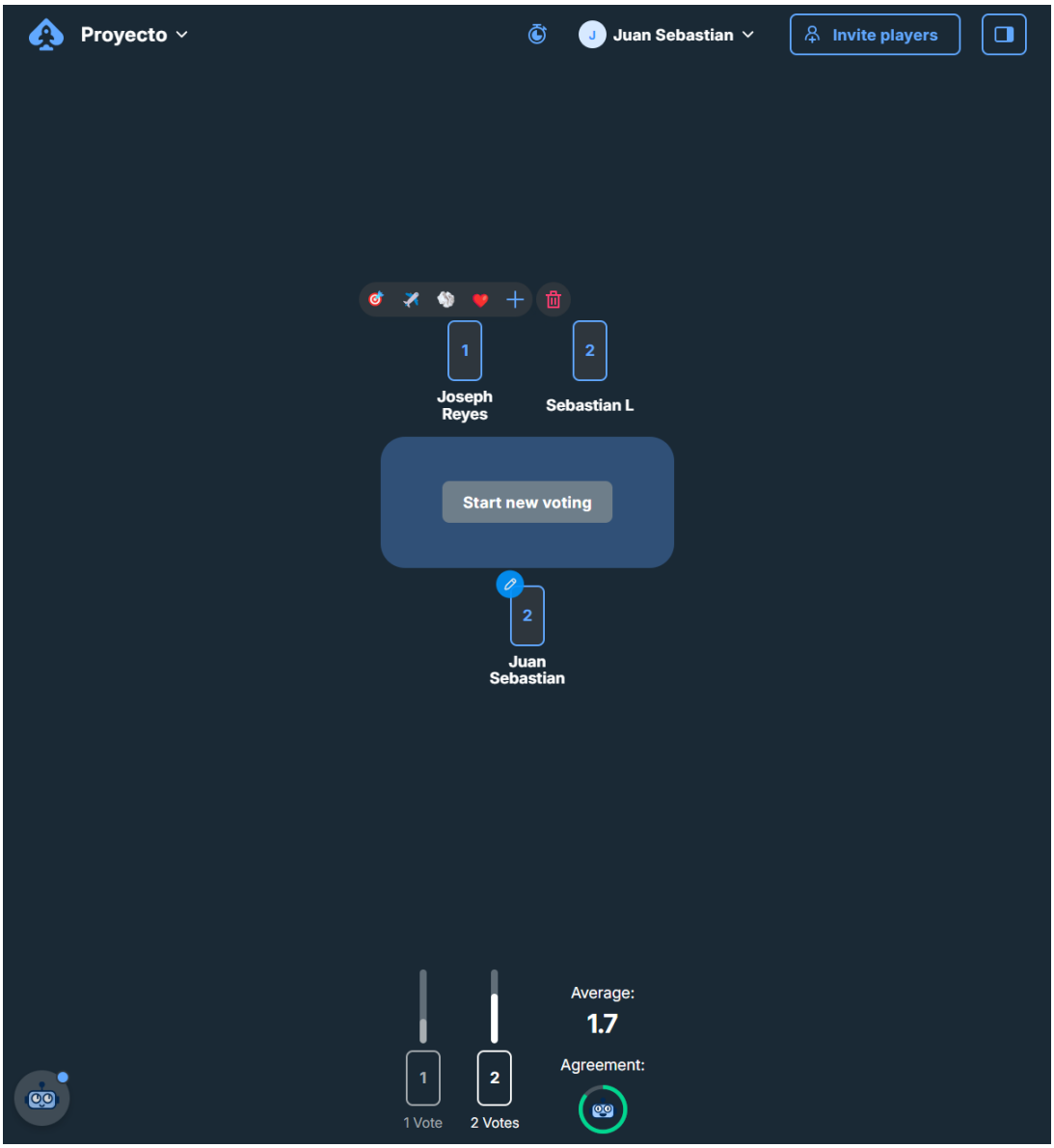

Note: We can see the example of the User Storie of Order History, in this case is the description and the acceptance criteria.

## Planning Pocker Strategy

We decided to use this Strategy to determinate the complexity of each User Storie, we argue to seek a conclusion, and we use for this exercises the Fibonacci Serie.

Figure 6

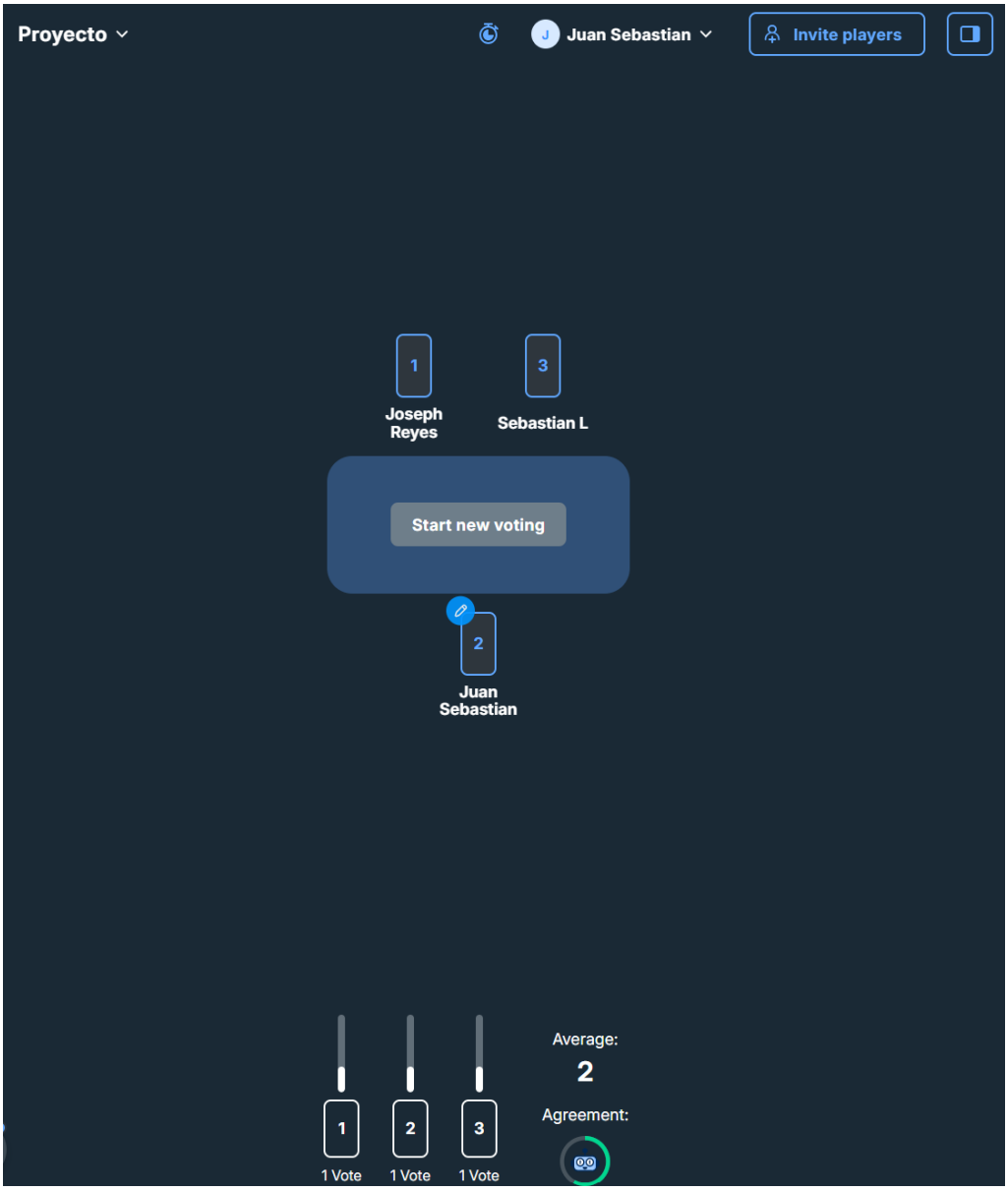
Register




Note: In this figure we can see the activity to put points to different User Stories, this one is about Register User, we argue about the points, we reached to a result, leave the User Storie with 2 points.

Figure 7

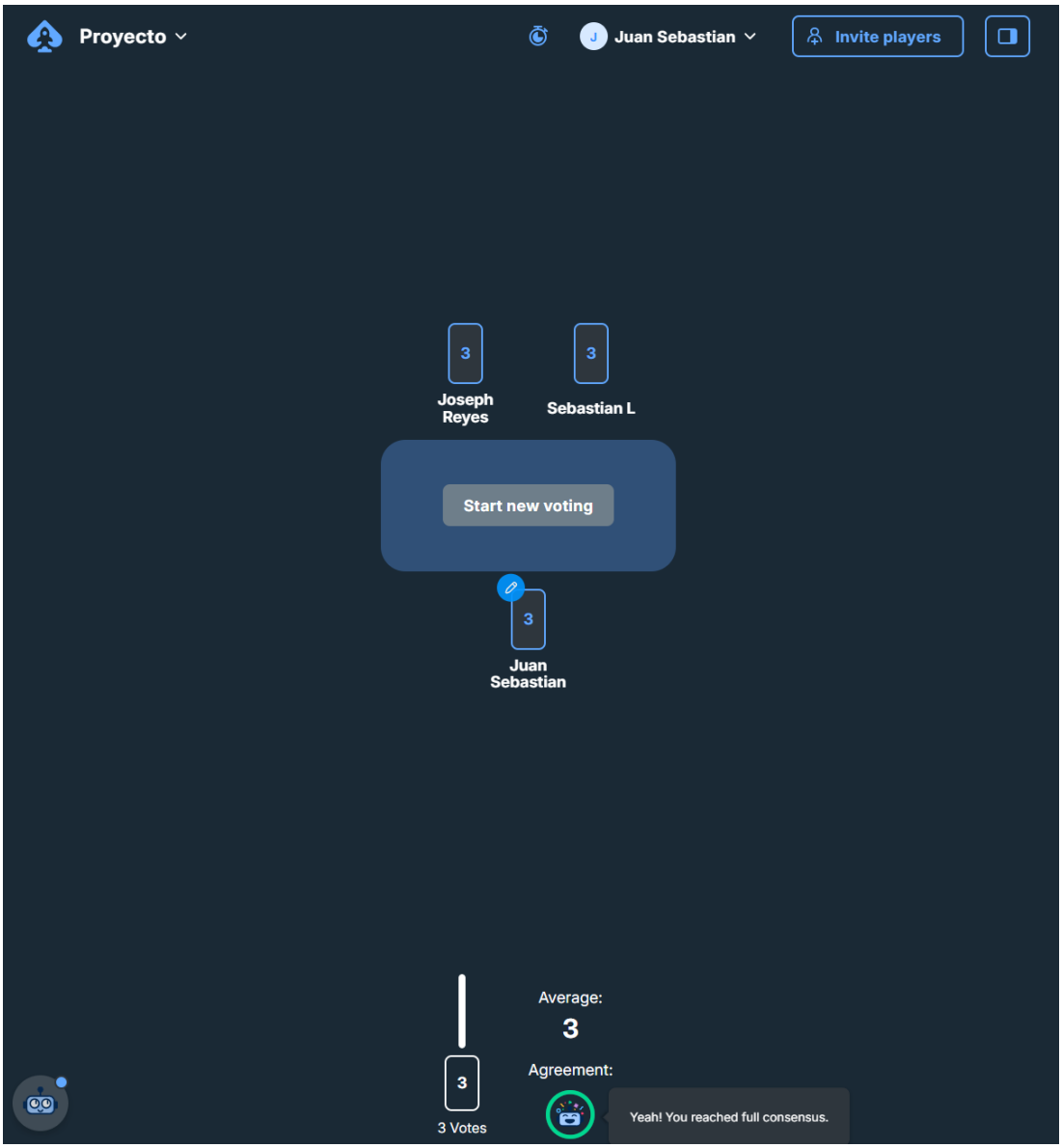
First decision in Log-In




Note: In this figure we can see the activity to put points to different User Stories, this one is about Log-In User, we didn't have an agreement at first but after talking we could reach to let the points to this User Storie in 3 points for the complexity of different users, but is still way easy in comparation to another User Stories.

Figure 8

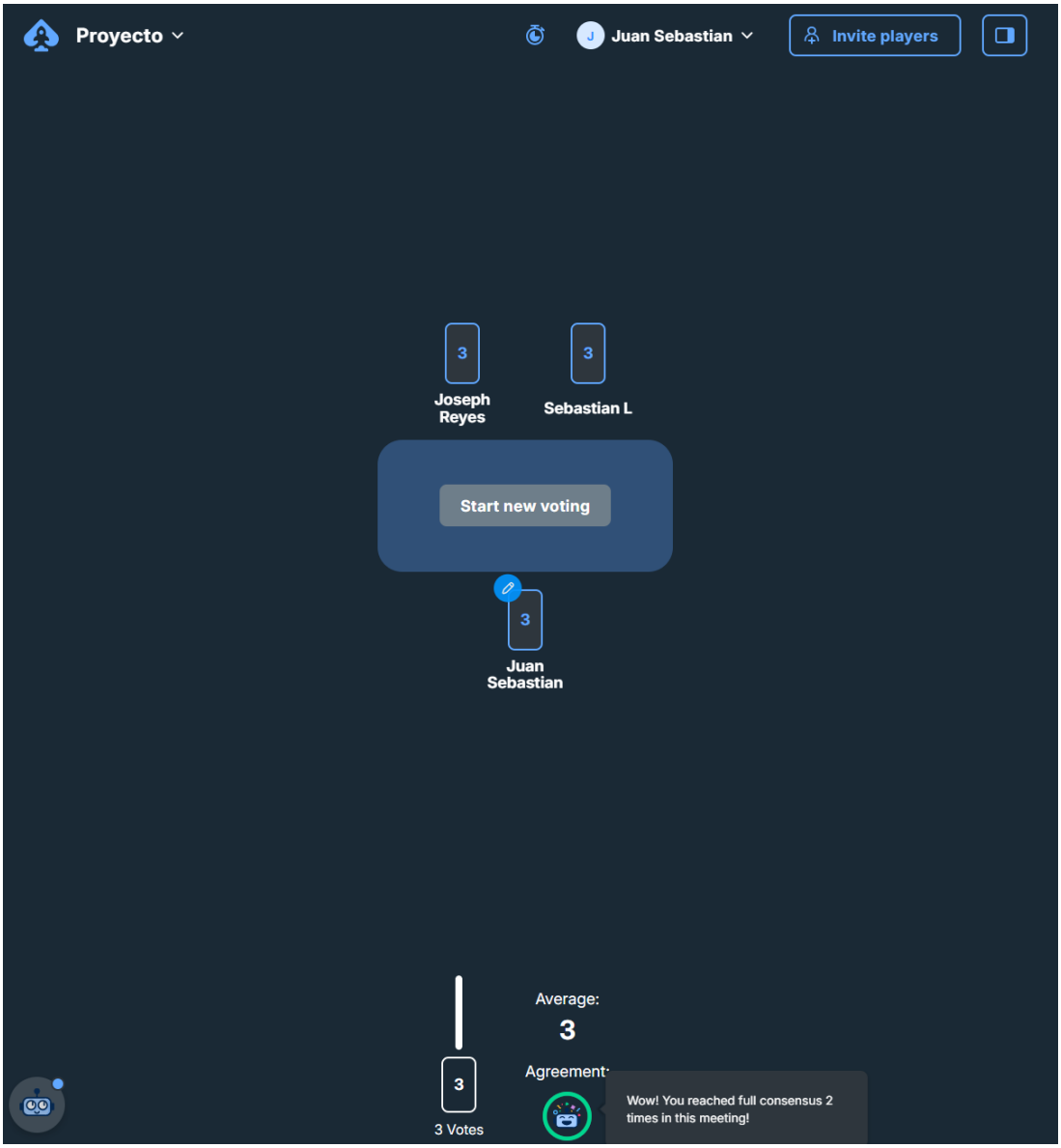
Final Decision in Log-In




Note: In this figure we can see the activity to put points to different User Stories, this one is about Log-In User, this one was the final decision after arguing and concluding the level of complexity of this User Story.

Figure 9

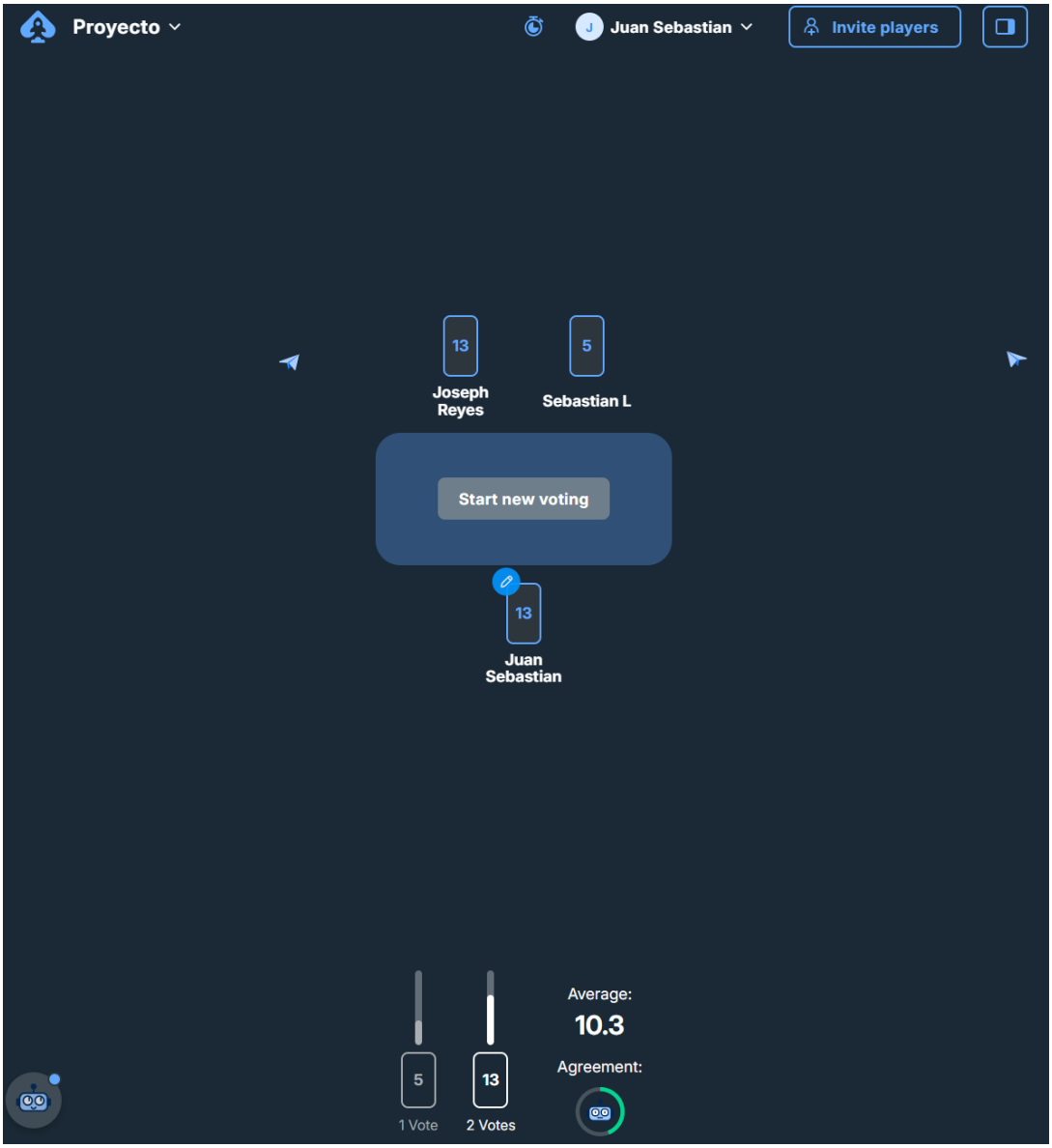
Manage User Profile




Note: In this figure we can see the activity to put points to different User Stories, this one is about Manage User Profile, in this one we at first could be in an agreement, 3 points is the level of complexity to this User Story.

Figure 10

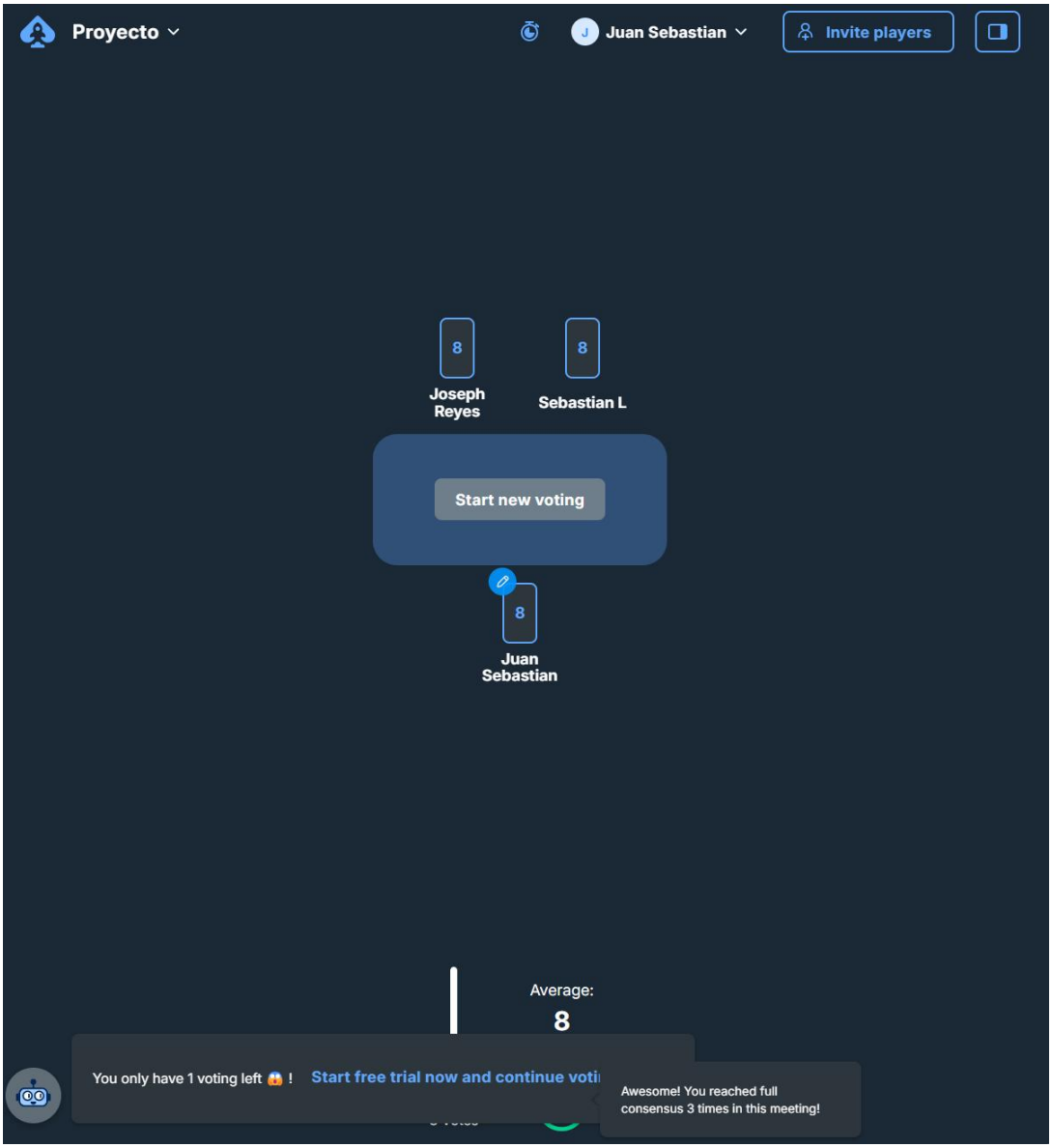
Manage Orders




Note: In this figure we can see the activity to put points to different User Stories, this one is about Manage Orders, we had different ways to see the project and the way to see the module of Orders, we were talking about how is going to be possible to do it, also, in any case if some of us already worked with the technologies necessary to do this User Storie, we concluded that the points to this User Story are 8 points.

Figure 11

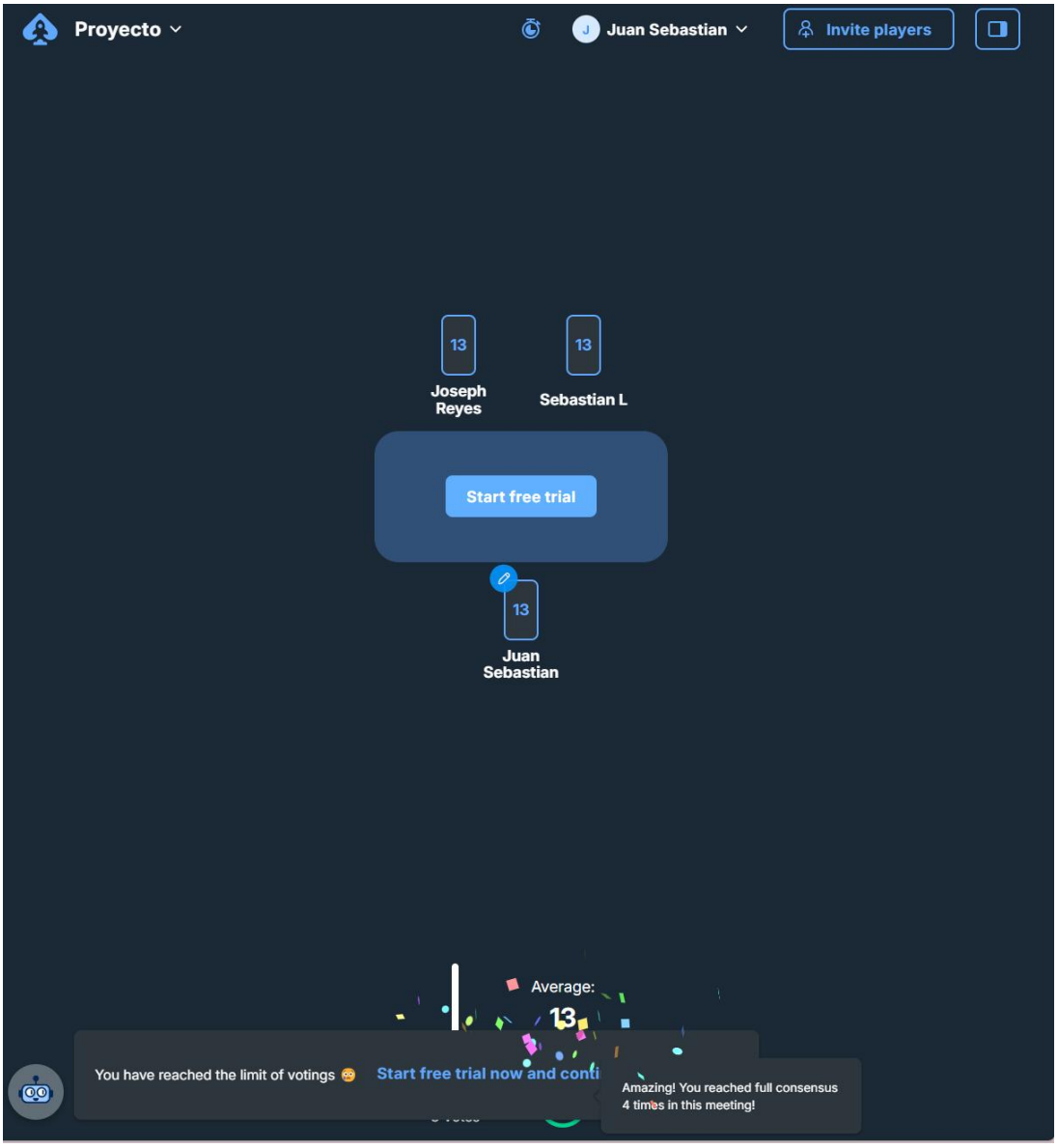
Order History




Note: In this figure we can see the activity to put points to different User Stories, this one is about Order History, this one we reached to a conclusion very easily, so the points selected to this one were about 8 points.

Figure 12

Make Orders




Note: In this figure we can see the activity to put points to different User Stories, this one is about Make Orders, this one we reached to a conclusion very easily, so the points selected for this one were about 13 points being the hardest User Story above all on this First Sprint.

## Roles

| | Strategies | Manager |
|---|---|---|
| Technical aspects | Conduct code reviews and document best practices. | Jodhep Reyes |
| Communication | Organize weekly meetings, maintain an open communication channel (WhatsApp) | Juan Mendoza |
| Quality assurance | Implement tests, review compliance with quality standards and ensure bug fixes before deployment. | Sebastian Lopez |

## References

Atlassian. (s. f.). *Jira | Software de seguimiento de proyectos e incidencias | Atlassian*. https://www.atlassian.com/es/software/jira

Scrumio. (2023, 13 noviembre). *La Definición de Hecho (Definition of Done)*. Scrumio |. https://www.scrumio.com/scrum/definicion-de-hecho#:~:text=La%20Definici%C3%B3n%20de%20Hecho%20o%20Definici%C3%B3n%20de%20Terminado,que%20el%20trabajo%20de%20desarrollo%20asociado%20ha%20concluido.