Project Planning and Developing Methodology

This project has been planned with two methodology types:

- **Classic**, implemented with Microsoft® Project.
- **Agile**, implemented with Atlassian® Jira and Scrum Poker.

For the first stages (documentation tasks), we are using a classic methodology; and for the last stage (coding tasks), we are using an agile methodology.

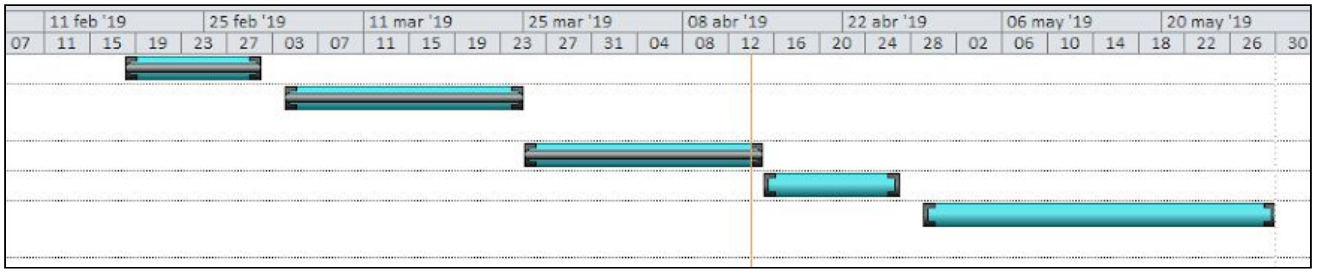
Global Tasks

The first actions to define are the tasks' names, duration, date of start and date of end. This has been made with **Microsoft® Project** software.

	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		PROJECT PROPOSAL	19 horas	lun 18/02/19	vie 01/03/19	
2		FINAL PROJECT PROPOSAL	24,25 horas	lun 04/03/19	dom 24/03/19	
3		DELIVERY #1	24,25 horas	lun 25/03/19	dom 14/04/19	
4		DELIVERY #2	19 horas	lun 15/04/19	vie 26/04/19	
5		FINAL DELIVERY (Coding Scrum Agile)	83 horas	lun 29/04/19	mié 29/05/19	

Timeline Diagram

This is the **GANTT** classic methodology to represent the tasks to be performed along the project's time period.



The last task (coding stage) have been planned with an agile methodology. See below in this document.

Higher Level Tasks

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	% completado
1	Programada r	PROJECT PROPOSAL	19 horas	lun 18/02/19	vie 01/03/19	100%
	Title, purposes Functional and non-functional requirements Mockups or wireframes					
2	Programada r	FINAL PROJECT PROPOSAL	24,25 horas	lun 04/03/19	dom 24/03/19	100%
	Any changes in the initial project proposal					
3	Programada r	DELIVERY #1	24,25 horas	lun 25/03/19	dom 14/04/19	100%
	Title, purposes Functional and non-functional requirements Use Case diagram and their textual description Database (Entity-relation diagram and relational model) Description of the project methodology used (Agile or classic approaching) Project planning (dates, tasks/product backlog, resources/effort points, GANTT/sprints+burndown chart)					
4	Programada r	DELIVERY #2	19 horas	lun 15/04/19	vie 26/04/19	0%
	Any changes in previous documents already delivered Class diagram Test cases and test data					
5	Programada r	FINAL DELIVERY (Coding Scrum A	83 horas	lun 29/04/19	mié 29/05/19	0%
	Final version of the application: 1. Final Documentation 2. Final Code FINAL DOCUMENTATION 35% Functional and non-functional requirements. Use case diagram and their textual description Test cases and test data Class diagram Entity relation diagram and relational model Description of the methodology used (Agile or classic approaching) Project planning (dates, tasks/product backlog, resources/effort points, GANTT/sprints+burndown chart) Project memo: initial planning, diary log, problems and solutions, final conclusions. Correspondence with initial functionalities Innovation Installation Guide SOURCE CODE 40% OOP: encapsulation, inheritance and polymorphism Multilayered design (e.g. MVC) Internal documentation Visual Aspect. UX (User eXperience) and UI (User Interface) design. PRESENTATION 25% Effectiveness of presentation Accurate answers. Strong knowledge of the whole application Working Demo					

Calendar

The working time values of the Proven1 calendar have been used for stages 1,2,3 and 4 (without specific time for project in teaching hours) and specifications of the Proven 2 calendar for stage 5 with time in class for project.

CALENDARIO BASE:	Proven1
Día	Horas
lunes	18:15 - 18:45, 21:30 - 22:00
martes	18:15 - 18:45, 21:30 - 22:00
miércoles	18:15 - 18:45, 20:45 - 21:30
jueves	18:15 - 18:45, 21:30 - 22:00
viernes	18:15 - 18:45
sábado	No laborable
domingo	No laborable
Excepciones:	Ninguna

CALENDARIO BASE:	Proven2
Día	Horas
lunes	15:15 - 18:15, 18:45 - 21:30
martes	15:15 - 18:15
miércoles	No laborable
jueves	15:15 - 18:15, 18:45 - 21:30
viernes	15:15 - 18:15
sábado	No laborable
domingo	No laborable
Excepciones:	
Fecha	Horas
lun 29/04/19 - mar 30/04/19	No laborable
mié 01/05/19	No laborable

Costs

The project costs will be determined in the final delivery when we have evaluated all the resources directly and indirectly dedicated.

Product Backlog

- US000. Create Database
- US001. Create Backend and Main App Navbar
- US011. List Users
- US013. Add User
- US014. Modify User
- US015. Delete User
- US016. Log In
- US017. Log Out
- US018. Register
- US0361. Sequence Sample

- US021. List Dogs
- US023. Add Dog
- US024. Modify Dog
- US025. Delete Dog
- US031. List Samples
- US033. Add Sample
- US034. Modify Sample
- US035. Delete Sample
- US0362. Obtain STR Pattern
- US0363. Find STR Match
- US041. List Incidents
- US043. Add Incident
- US044. Modify Incident
- US045. Delete Incident

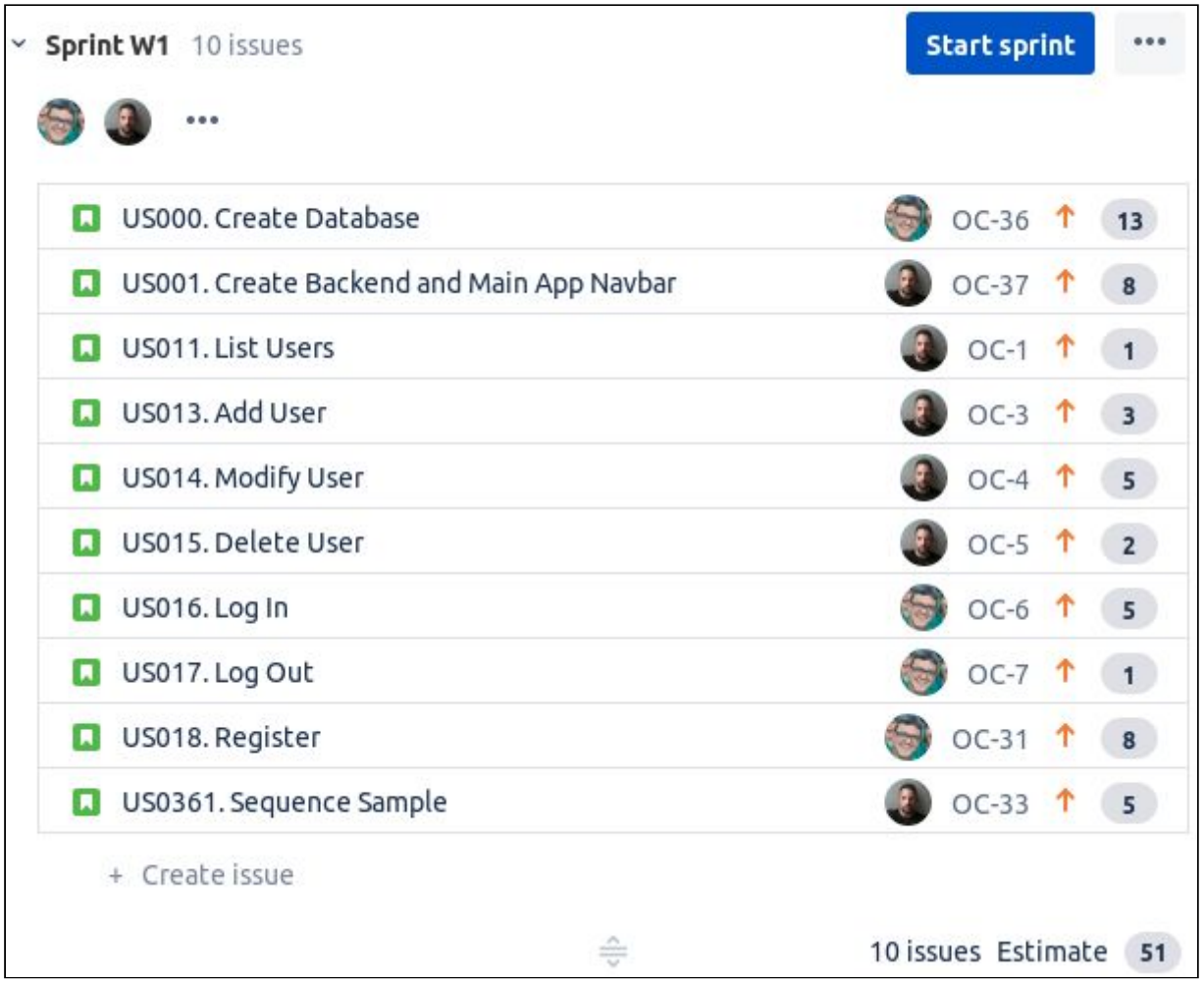
- US051. List Infractions
- US053. Add Infraction
- US054. Modify Infraction
- US055. Delete Infraction
- US056. Validate Infraction
- US057. Generate Offcial Document

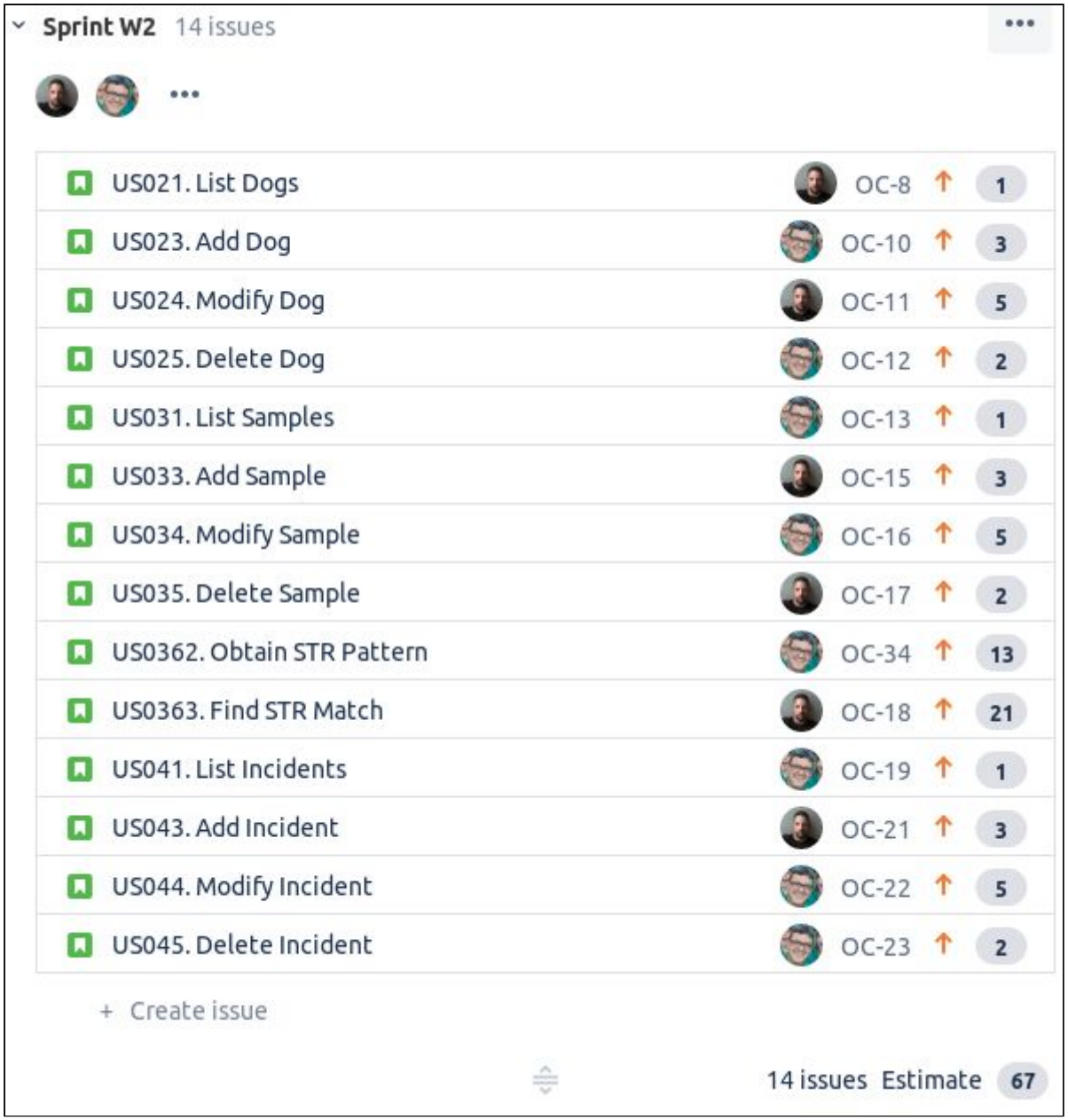
- US012. Filter Users
- US022. Filter Dogs
- US032. Filter Samples
- US042. Filter Incidents
- US052. Filter Infractions
- US061. Receive Notifications

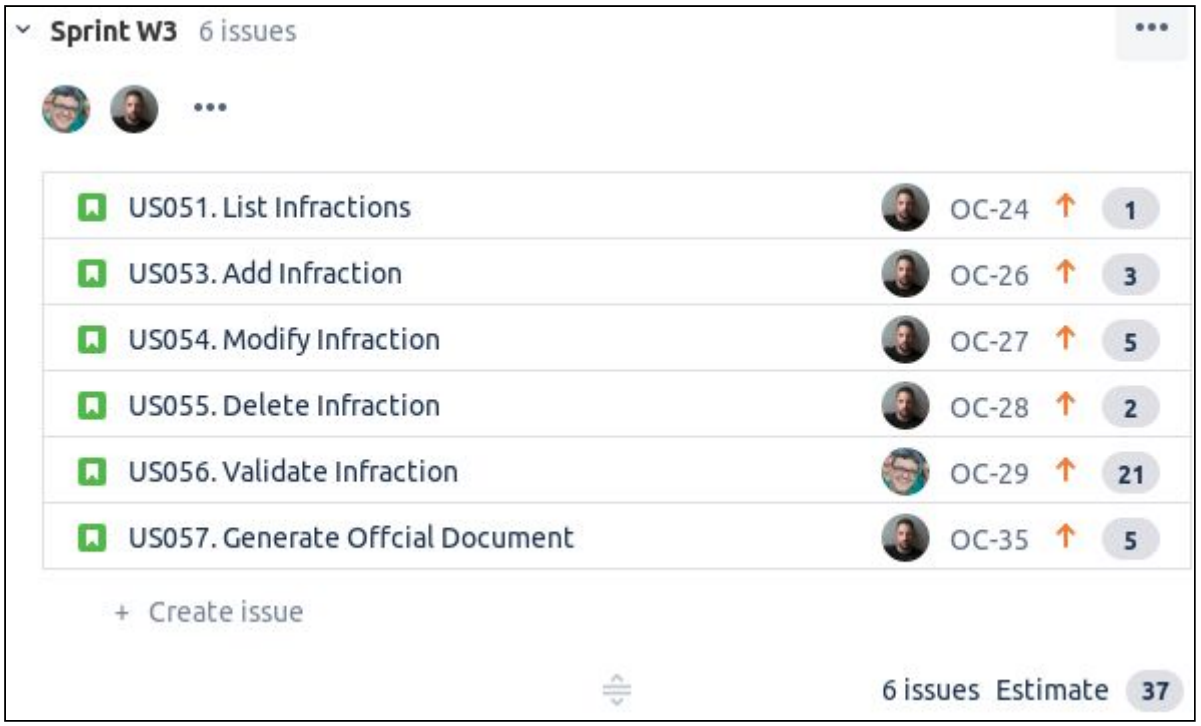
Sprints Planning

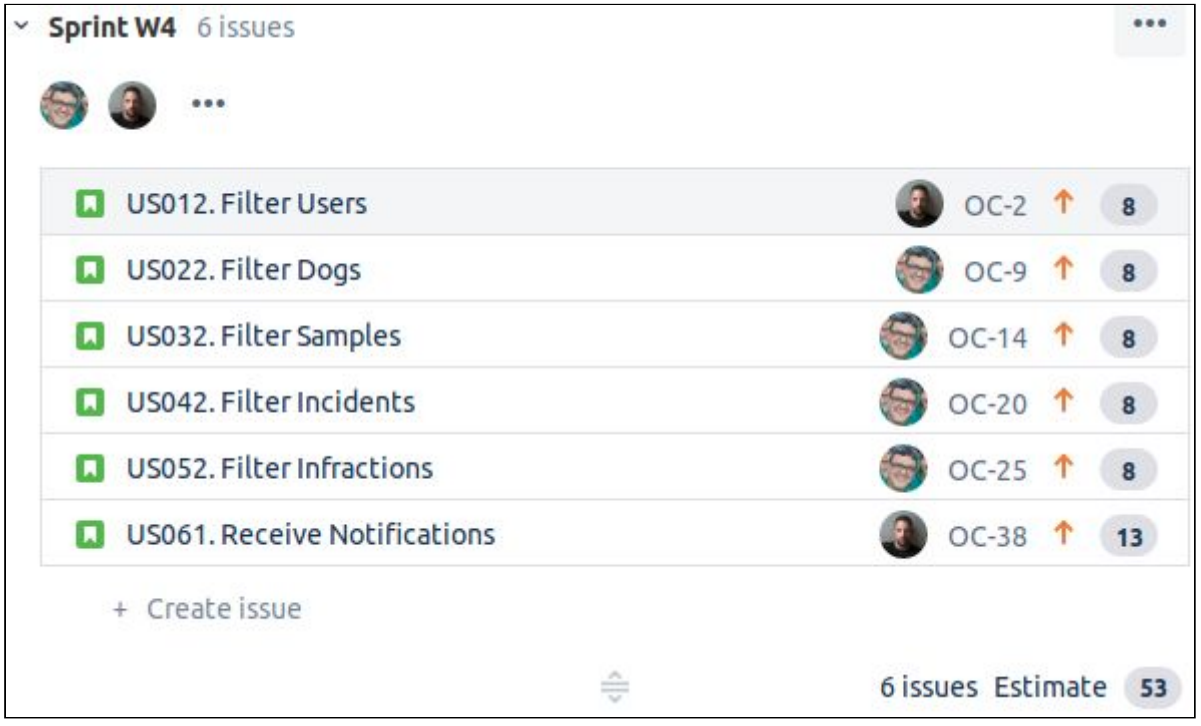
We are going to do 4 sprints, and each sprint is going to take 1 week.

First, the Product Backlog (this is, the User Stories) is created at Atlassian® Jira.




Sprint W2 14 issues
US021. List Dogs OC-8 1
US023. Add Dog OC-10 3
US024. Modify Dog OC-11 5
US025. Delete Dog OC-12 2
US031. List Samples OC-13 1
US033. Add Sample OC-15 3
US034. Modify Sample OC-16 5
US035. Delete Sample OC-17 2
US0362. Obtain STR Pattern OC-34 13
US0363. Find STR Match OC-18 21
US041. List Incidents OC-19 1
US043. Add Incident OC-21 3
US044. Modify Incident OC-22 5
US045. Delete Incident OC-23 2
+ Create issue
14 issues Estimate 67

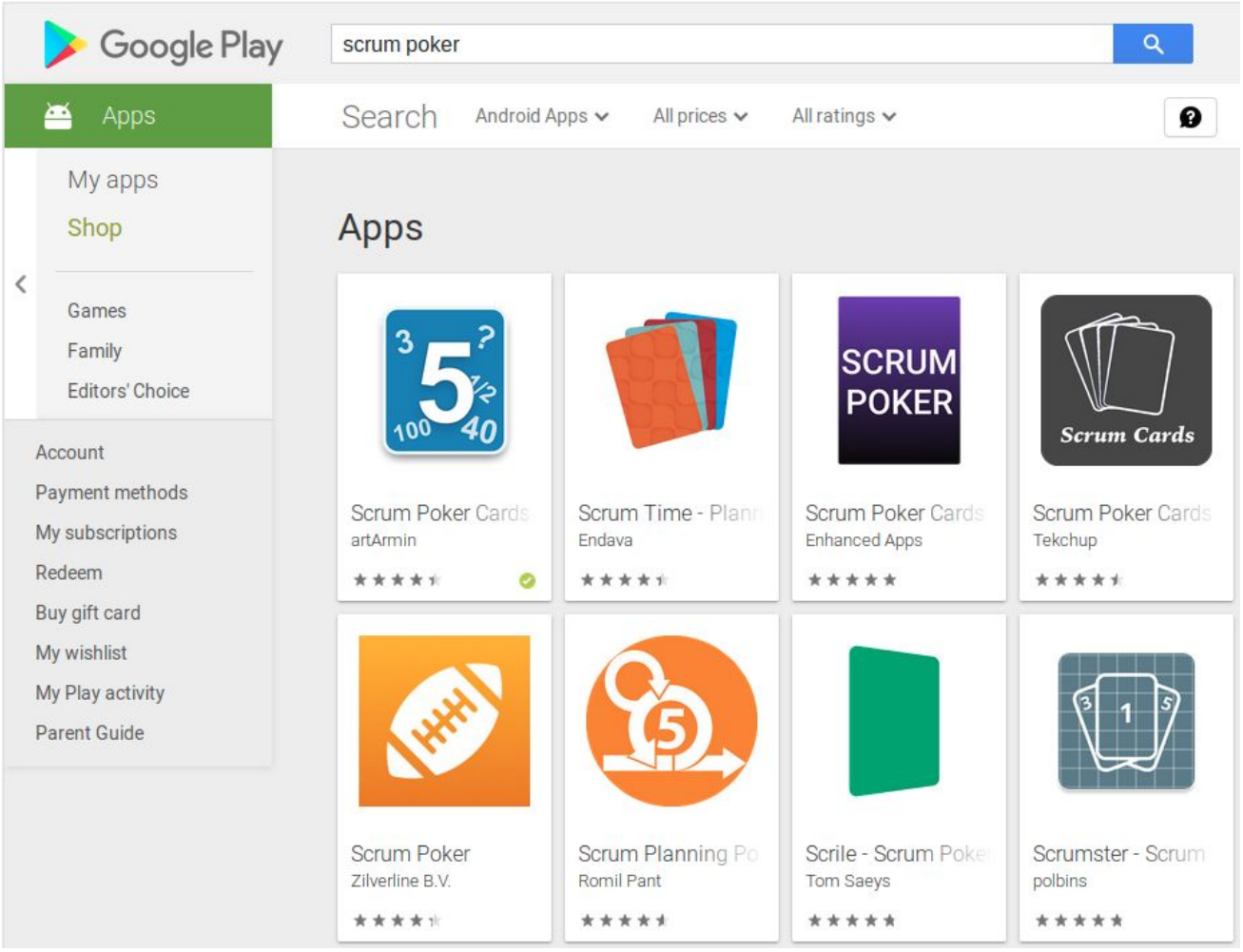

Sprint W3 6 issues
US051. List Infractions OC-24 1
US053. Add Infraction OC-26 3
US054. Modify Infraction OC-27 5
US055. Delete Infraction OC-28 2
US056. Validate Infraction OC-29 21
US057. Generate Offcial Document OC-35 5
+ Create issue
6 issues Estimate 37


Sprint W4 6 issues
US012. Filter Users OC-2 8
US022. Filter Dogs OC-9 8
US032. Filter Samples OC-14 8
US042. Filter Incidents OC-20 8
US052. Filter Infractions OC-25 8
US061. Receive Notifications OC-38 13
+ Create issue
6 issues Estimate 53

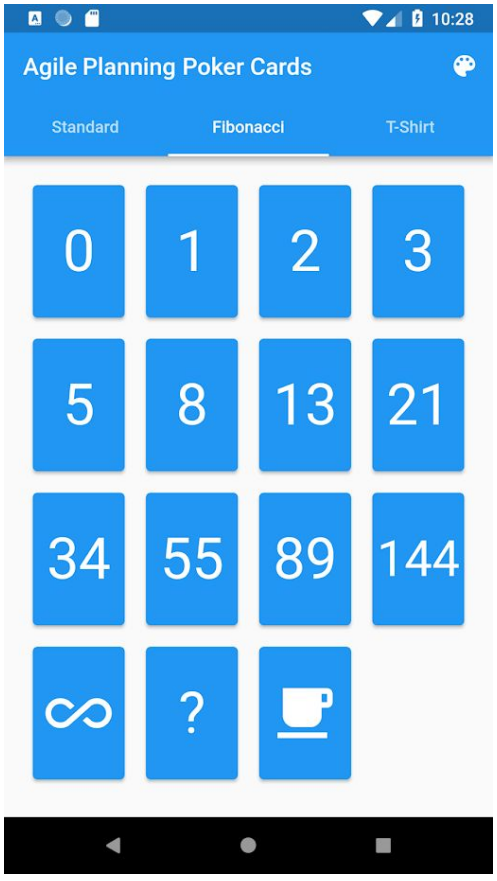
Effort Points

The developers, we have sit and assign the effort points to each user story using the agile methodology named **Scrum Poker**.

This can be played with real poker cards or, alternatively, using an app for smartphone. There are several Scrum Poker apps in both (Android) Google Play and (iOS) App Store.



Concretely, we have used the **Fibonacci** progression to vote the effort points that each one thinks for each user story.



After the voting, if the two of us chose the same score, it is assigned directly to that user story; otherwise, each of us must **explain why that score has been chosen**. Then, we can re-vote or assign the mean of the two scores.

For each sprint, we have assigned approximately the same amount of user stories per each developer, regarding the effort points assigned by Scrum Poker previously.

Counting down all the user stories, we have a total of:

51 + 67 + 37 + 53 = **208 effort points**

We have 1 month to develop de app, so if we divide the total effort points by 4 weeks, the resulting planning should have 4 sprints of approximately:

208 / 4 = **52 effort points per sprint**

Workload by assignee - Sprint W1

Assignee	Issues	Story Points
Unassigned	0	0
 Oscar Burgos	4	27
 Alejandro Asensio	6	24
Total:	10	51

Workload by assignee - Sprint W2

Assignee	Issues	Story Points
Unassigned	0	0
 Alejandro Asensio	6	35
 Oscar Burgos	8	32
Total:	14	67

Workload by assignee - Sprint W3

Assignee	Issues	Story Points
Unassigned	0	0
Oscar Burgos	1	21
Alejandro Asensio	5	16
Total:	6	37

Workload by assignee - Sprint W4

Assignee	Issues	Story Points
Unassigned	0	0
Oscar Burgos	4	32
Alejandro Asensio	2	21
Total:	6	53

We can not evaluate the burndown chart since we have not started the first sprint. From the first sprint we will make the daily meeting (daily scrum) to follow up and expose the problems that arise. At the end of the sprint we will do the sprint review to see the finished tasks and we will do a retrospective sprint where we will evaluate the improvements for the following sprints.