DNA-based Dog Identification

Final Project

Authors
Asensio, Alejandro
Burgos, Óscar

Center
Institut Provençana
Carrer Sant Pius X, núm 8
08901 L'Hospitalet de Llobregat,
Barcelona
Curs 2018-2019

Accomplished on
29th May 2019

Studies
CFGS Desenvolupament d’Aplicacions
Web -
perfil professional Bioinformàtica
(DAWBIO)

Module
M12 Project

Table of Contents

Abstract

The main idea of our project arose from the detection of a problem that we saw increased in our municipality lately: the large amount of dog droppings in the city, caused by the social incivility of its owners.

In order to find a solution, the local council spends large amounts of money each year trying to alleviate this incivility and have a more clean and livable city, with sensitizing campaigns.

Our idea and business model passes through the collaboration with the city council, being we in charge of the management of the process as a company of contracted services and also the application of a new municipal law.

This idea is based on the detection of the dog that has left the excrement in the street without its owner picking it up. To be able to proceed with this detection, we will need to analyze these excrements through the STR of the dog's DNA and thus be able to identify it through the stored identification data of the dogs registered in the municipality.

Currently, dog owners are required to register their pets at the town hall and implant a chip for their identification control.

This project aims to use bioinformatics’ tools along with web developing to identify a certain dog by its **Short Tandem Repeats (STR)** pattern, located among the different chromosomes of its DNA. That identification may have several uses, and one of them is helping towns and cities to maintain the **streets clean** from dogs' droppings.

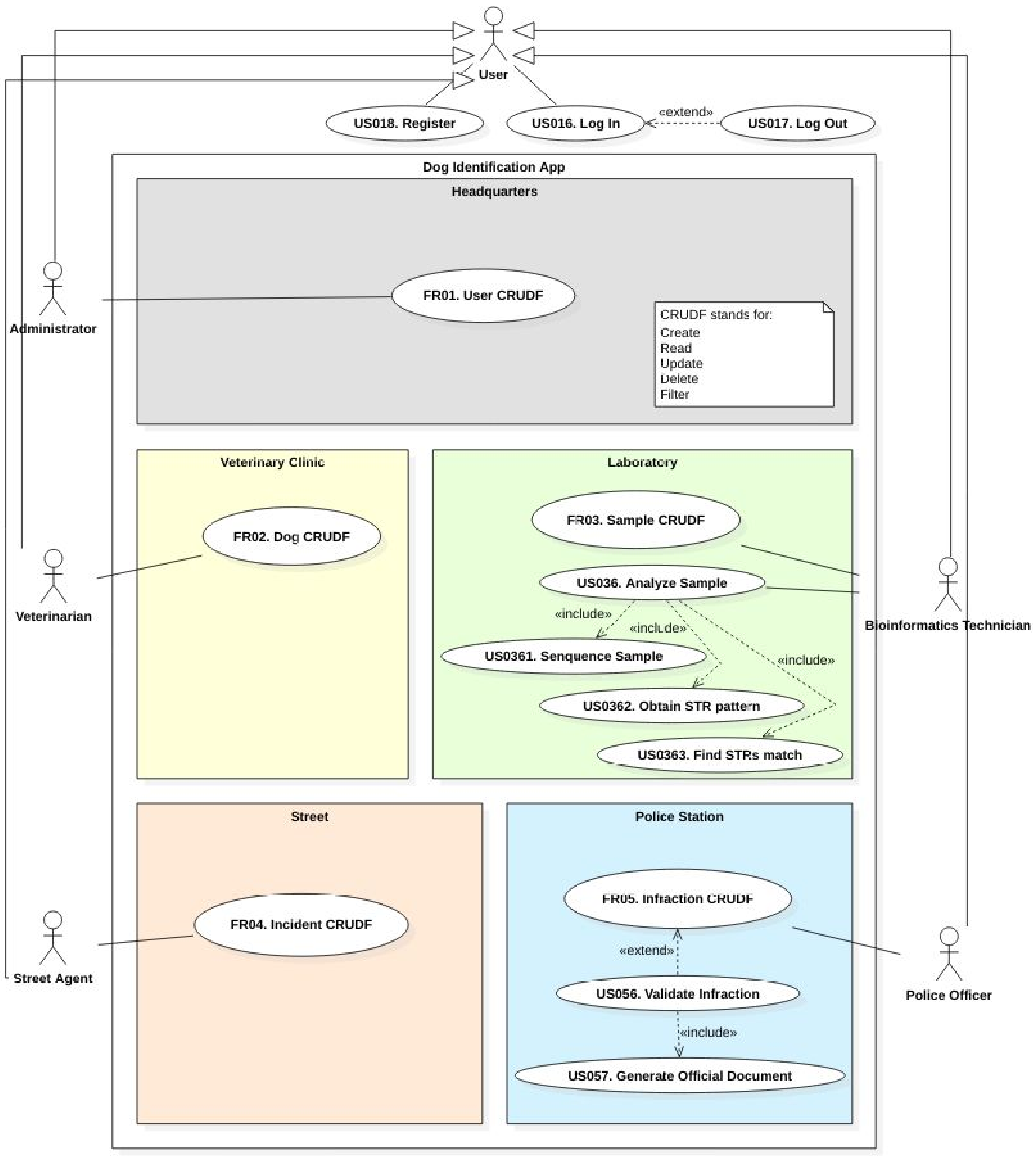
To accomplish that mission, we are building a web application with **Laravel 5.8** for a RESTful API backend, Angular 7 for the frontend, Python for the scripts, and MySQL as relational database for persistence.

The following points describe the expected workflow of our web app:

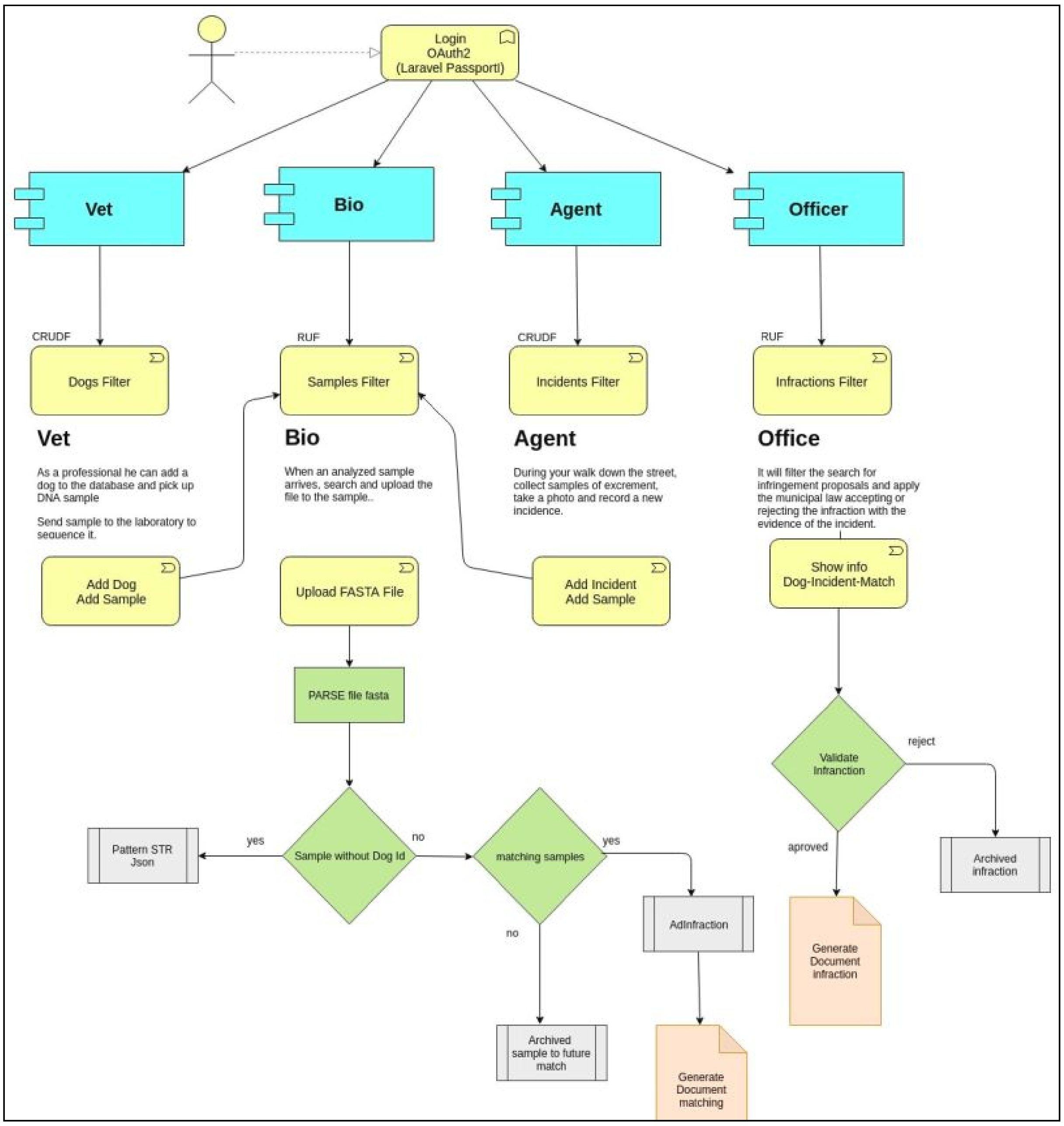
1. Veterinarian registers a new Dog with chip number, owner's DNI and saliva sample into a Kit.
2. Bioinformatics Technician sequences the DNA from the saliva sample, getting the fasta file and the STR pattern.
3. Street Agent takes a sample of a dog's droppings found in street, introduces the sample into the Kit, and registers the Incident into the app, storing the location, a photo as evidence and the Kit's barcode number.
4. Bioinformatics Technician sequences the DNA from the droppings sample, getting the fasta file and the STR pattern, which will reveal if it matches with some existing saliva sample. This method allows us to uniquely identify a certain dog. If there is no match, a new "Suspected non-registered Dog" log is automatically stored in our database.

5. A Police Officer validates the evidences: photo + location + STR matching patterns. If everything is considered as correct, a new Infraction Proposal is generated and sent to Town Administration, which will proceed with the corresponding monetary fine to the dog's owner.

Use Case Diagram



Workflow



Vet

As a professional he can add a dog to the database and pick up DNA sample

Send sample to the laboratory to sequence it.

Bio

When an analyzed sample arrives, search and upload the file to the sample..

Agent

During your walk down the street, collect samples of excrement, take a photo and record a new incidence.

Office

It will filter the search for infringement proposals and apply the municipal law accepting or rejecting the infraction with the evidence of the incident.

Functional Requirements

This document has the purpose of defining the Functional Requirements (FR) of the project. In order to accomplish that, we are going to break them into User Stories (US), which will be implemented afterwards, in the code developing stage.

Define user roles

As the Product Owner, I want to define the basic user roles for the application, so that I can decide which tasks are performed by which role(s). The roles are:

- Application Administrator, **admin**, who has permission to manage the users, as well as all the actions inside the application; for instance, the massive load of Kits for samples (barcode as id).
- Veterinarian, **vet**, who is able to create or modify a Dog entry, injecting the chip into the dog's skin and taking a sample of dog's saliva into the Kit.
- Bioinformatics Technician, **bio**, who is able to create a Sample entry, which contains its DNA sequence info, such as the fasta and STR's files; as well as determine if two samples are from the same dog (match).
- Street Agent, **agent**, who is able to create an Incident entry, taking the dog droppings sample into the Kit and uploading a photo of them.
- Police Officer, **officer**, who is able to validate, invalidate or ask for more evidences about the droppings street Incident. By validating the Incident,
- Final User, **dog**, who is able to log in *from the dog perspective*, this is, and list the Infractions of its human owner, as well as modify some customizable fields, such as the alias, the password or the profile photo.

In this project we are implementing the CRUD (Create, Read, Update, Delete) system for every object type in our app, plus the Filter function; so we call it **CRUDF**.

FR00. Database and Backend Set Up

US000. Create Database

As an **admin**, I want to create the MySQL database structure and the needed tables, so that I can store the information of the app.

US001. Create Backend and Main App Navbar

As an **admin**, I want to create the backend project using the Laravel framework and the main navbar in blade syntax, so that I can start developing and placing the functionalities on the app.

FR01. User Management

US011. List Users

As an **admin**, I want to list all the users, so that I can check their info and know the total amount of users in the app.

US012. Filter Users

As an **admin**, I want to filter the users list, so that I can search for an specific user.

US013. Add User

As an **admin**, I want to create a new user with an specific role, so that I can give the correct permissions to use the app.

US014. Modify User

As *any user role*, I want to modify the editable fields of my user, such as my password, so that I can maintain my account updated and secure.

US015. Delete User

As an **admin**, I want to delete a wrong user entry, so that I can maintain clean the users table in database.

US016. Log In

As *any user role*, I want to log in the app, so that I can use its functionalities, depending on my role.

US017. Log Out

As *any user role*, I want to log out from the app, so that I can leave the site securely.

US018. Register

As a *new user*, I want to register into the app, so that I can be assigned an specific role by the **admin** and start contributing to the app.

FR02. Dog Management

US021. List Dogs

As a **vet**, I want to list all the dogs, so that I can check their info and know the total amount of dogs in the app.

US022. Filter Dogs

As a **vet**, I want to filter the dogs list, so that I can search for an specific dog.

US023. Add Dog

As a **vet**, I want to create a new dog entry, so that I can register its new chip number, body features and the Kit's barcode; that Kit is made for taking a saliva sample from dog. This is the START of our workflow.

US024. Modify Dog

As **vet**, I want to modify a dog entry, so that I can update any change what is needed. For example, if a dog dies, we are not deleting it, but modifying its status field.

US025. Delete Dog

As a **vet**, I want to delete a wrong dog entry, so that I can maintain clean the dogs table in database.

FR03. Sample Management

US031. List Samples

As a **bio**, I want to list all the samples, so that I can check their info and know the total amount of samples in the app.

US032. Filter Samples

As a **bio**, I want to filter the samples list, so that I can search for an specific sample.

US033. Add Sample

As a **bio**, I want to create a new sample entry, so that I can register its sequenced DNA as a fasta file and the obtained STR's, which identifies the dog uniquely.

US034. Modify Sample

As a **bio**, I want to modify a sample entry, so that I can update any change what is needed. For example, if the sequentiation process has to be repeated, the fasta file must be overwritten.

US035. Delete Sample

As a **bio**, I want to delete a wrong sample entry, so that I can maintain clean the samples table in database.

US036. Analyze Sample

As a **bio**, I want to analyze a sample, so that I can **sequence** it, obtain its **STR pattern** and find a possible **STR match** among our database. This user story is complex, so it has been split in three sub-tasks.

US0361. Sequence Sample

As a **bio**, I want to sequence a sample, so that I can store it as a FASTA file as an attribute of the sample register.

US0362. Obtain STR Pattern

As a **bio**, I want to obtain the STR pattern of a sample, so that I can identify a unique dog by that pattern of Short Tandem Repeats of DNA.

US0363. Find STR Match

As a **bio**, I want to find a possible STR match, so that I can uniquely identify a certain dog that was previously registered in our database. If the match occurs, an Infraction proposal is automatically emitted to the Police Officers; if no match is found, this anonymous sample is stored and our workflow reaches its END.

FR04. Incident Management

US041. List Incidents

As an **agent**, I want to list all the incidents, so that I can check their info and know the total amount of incidents in the app.

US042. Filter Incidents

As an **agent**, I want to filter the incidents list, so that I can search for an specific incident.

US043. Add Incident

As an **agent**, I want to create a new incident entry, so that I can register the location, an attatched photo of the evidence, along with the Kit barcode, where the dog droppings sample is taken.

US044. Modify Incident

As an **agent**, I want to modify an incident entry, so that I can update the attatched photo or re-scan the Kit barcode.

US045. Delete Incident

As an **agent**, I want to delete a wrong incident entry, so that I can maintain clean the incidents table in database.

FR05. Infraction Management

US051. List Infractions

As an **officer**, I want to list all the infraction proposals, so that I can check their info and know the total amount of infraction proposals in the app.

US052. Filter Infractions

As an **officer**, I want to filter the infraction proposals list, so that I can search for an specific infraction proposal.

US053. Add Infraction

As an **officer**, I want to add an infraction proposal entry manually.

US054. Modify Infraction

As an **officer**, I want to modify an infraction proposal entry manually. so that I can fix any possible wrong information.

US055. Delete Infraction

As an **officer**, I want to delete a wrong infraction entry, so that I can maintain clean the infractions table in database.

US056. Validate Infraction

As an **officer**, I want to validate an Infraction proposal. In one hand, I may want to **approve** an existing infraction proposal entry, so that I can confirm the evidences; in this case, the infraction status changes to "approved". In the other hand, I may want to **reject** an existing infraction proposal entry, so that I can refuse the evidences as non-conclusive (maybe due to a non-accurate droppings photo or an inconsistency between owner address and Incident address); in this case, the infraction status changes to "rejected".

US057. Generate Official Document

As an **officer**, I want to emit an automated generation of an offical document, so I can proof my approval or rejection of an Infraction proposal. That document will be sent to the corresponding Town Administration. At this point, our workflow reaches its END.

FR06. Notification Management (*pending*)

US061. Receive Notifications

As *any user role*, I want to receive a notification when a certain task is pending to be done by my user role, so that I can get my work done on time and when it's needed. For example, when a sample is sent to the lab, the **bio** role will receive a notification to analyze that sample. Other example is: when an infraction proposal is created and is pending to be validated, the **officer** role will receive a notification to accept or reject that infraction proposal.

Non Functional Requirements

This section has the purpose of defining the Non Functional Requirements (NFR) of the project. In other words, the infrastructure we need to build our application.

NFR01. Hardware

We need a computer for each one of us, with a minimum requirements: a decent processor like an Intel i5/i7 and 8GB RAM.

NFR02. Operating System

We need a Linux-based operating system, such as **Ubuntu**, with a set of software tools to implement the source code and test the application.

NFR03. Server

We need a Linux-based server, such as **Ubuntu Server**, with a set of software tools to deploy the application and perform tests in a real environment.

NFR04. Software

We need certain packages to write and test our code, such as programming language compilers and/or interpreters (**PHP**, **Python**), frameworks (**Laravel**), IDE (**Netbeans**, **Visual Studio Code**), web browsers (**Firefox**, **Chromium**) and version control system (**Git**, **GitHub**), database engine (**MySQL**), Python libraries (**exrex**).

Project Planning and Developing Methodology

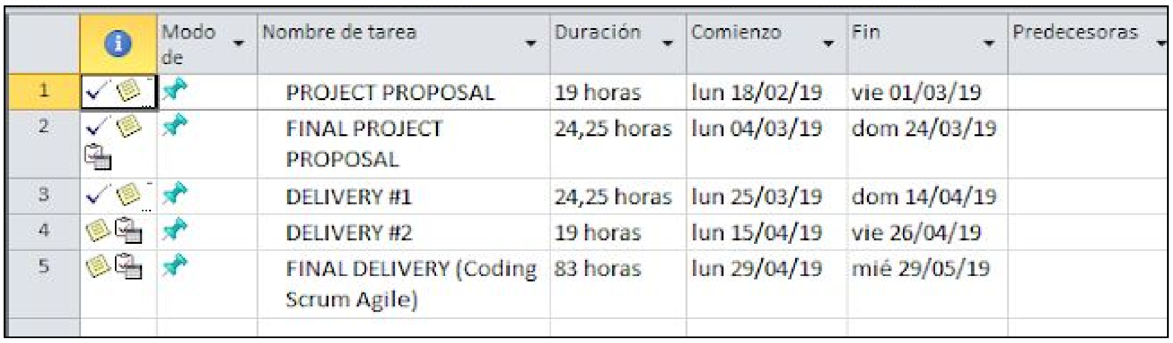
This project has been planned with two methodology types:

- **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

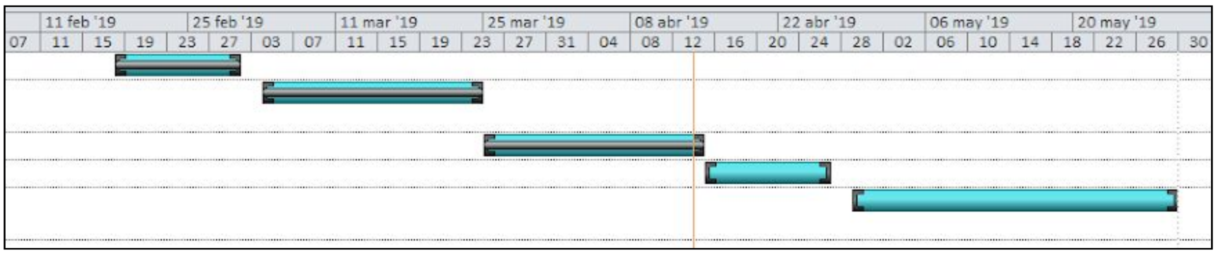
The first actions to define are the tasks' names, duration, date of start and date of end. This has been made with **Microsoft® Project** software.



	ℹ	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	

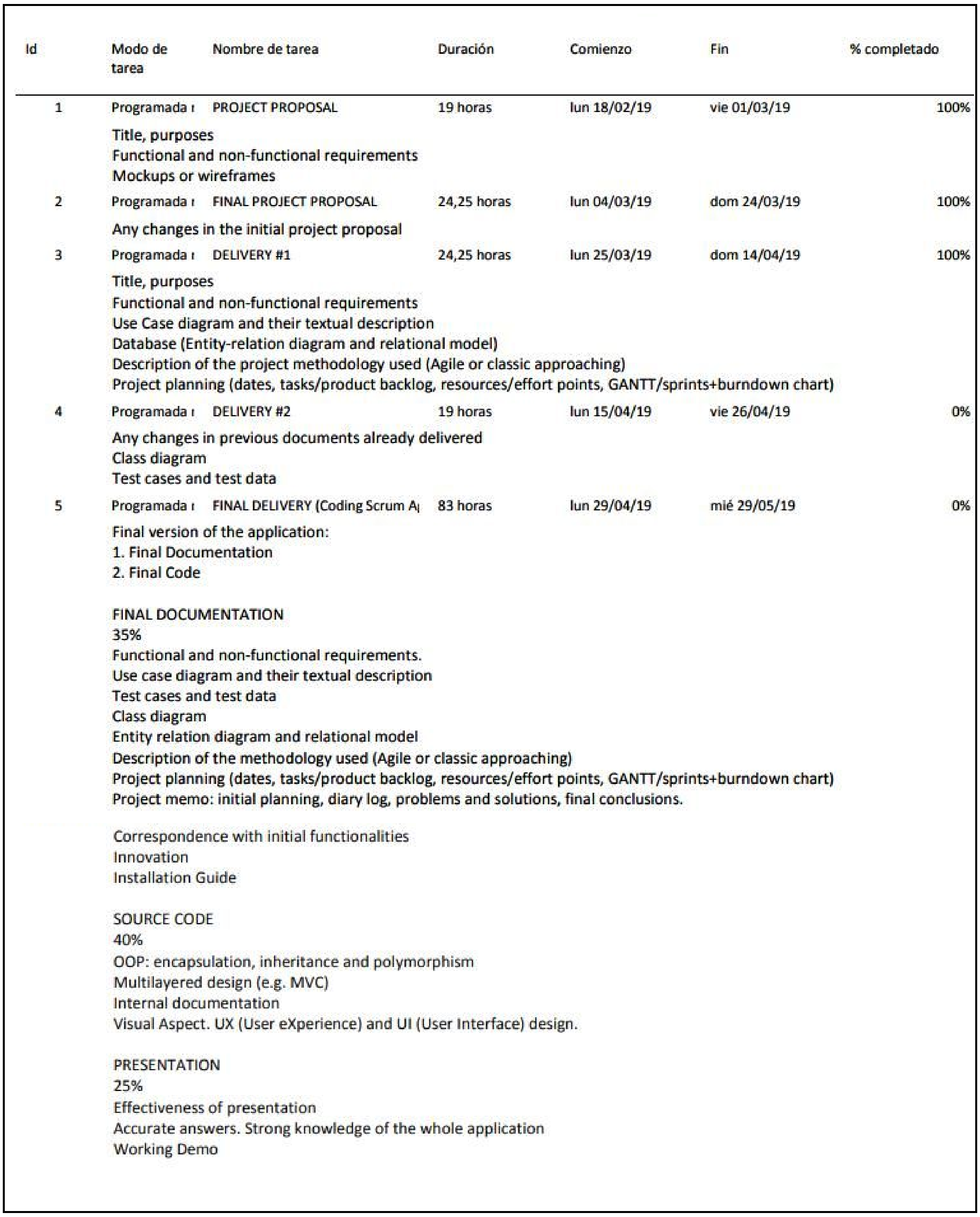
Timeline Diagram

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.

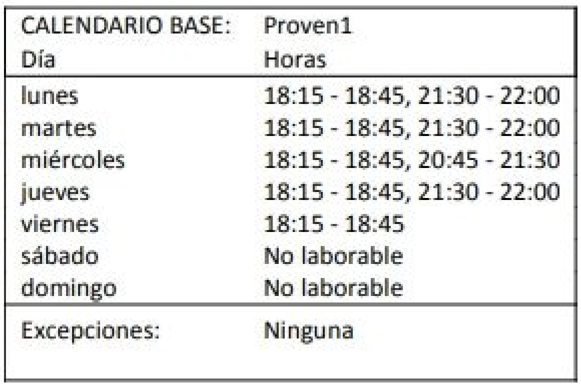
Higher Level Tasks



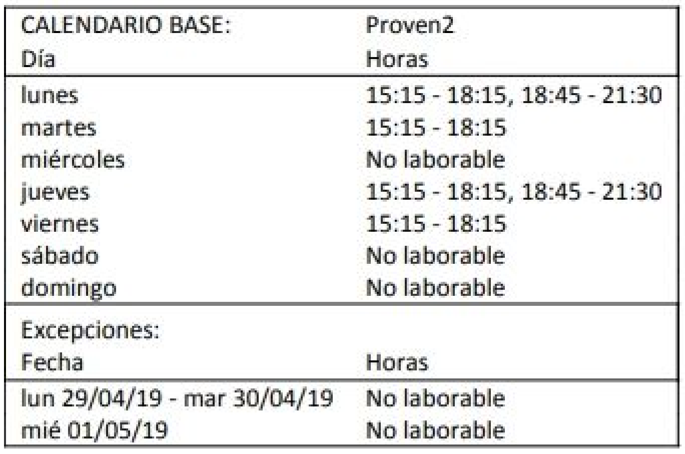
Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	% completado
1	Programada r	PROJECT PROPOSAL	19 horas	lun 18/02/19	vie 01/03/19	100%
	Title, purposes Functional and non-functional requirements Mockups or wireframes					
2	Programada r	FINAL PROJECT PROPOSAL	24,25 horas	lun 04/03/19	dom 24/03/19	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.

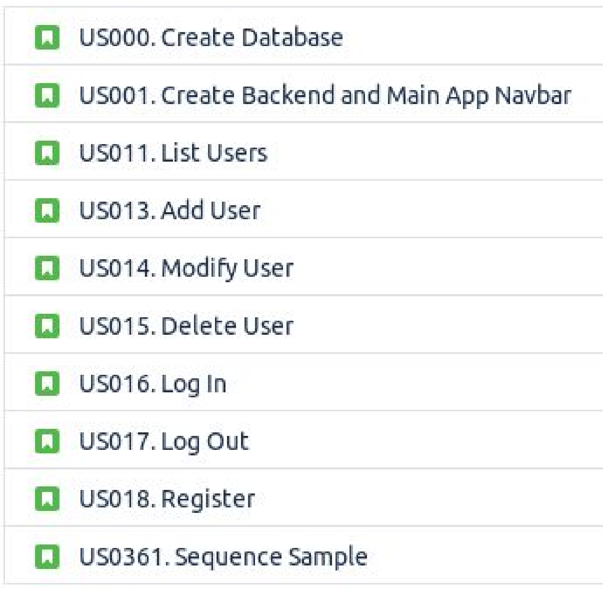


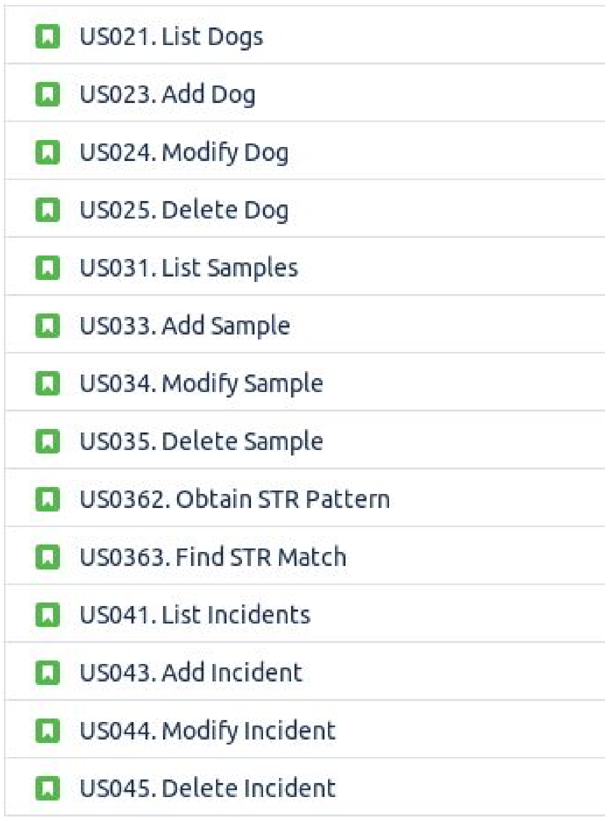
CALENDARIO BASE:	Proven1
Día	Horas
lunes	18:15 - 18:45, 21:30 - 22:00
martes	18:15 - 18:45, 21:30 - 22:00
miércoles	18:15 - 18:45, 20:45 - 21:30
jueves	18:15 - 18:45, 21:30 - 22:00
viernes	18:15 - 18:45
sábado	No laborable
domingo	No laborable
Excepciones:	Ninguna

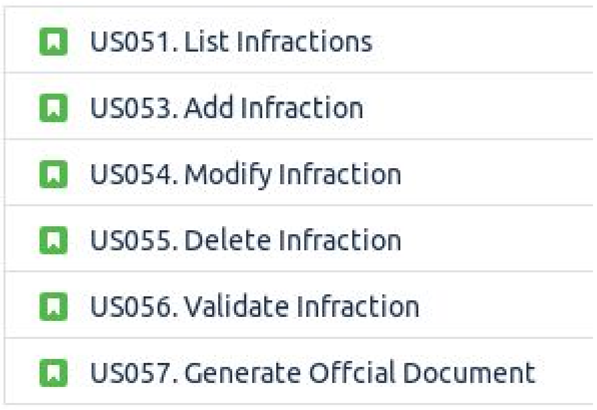


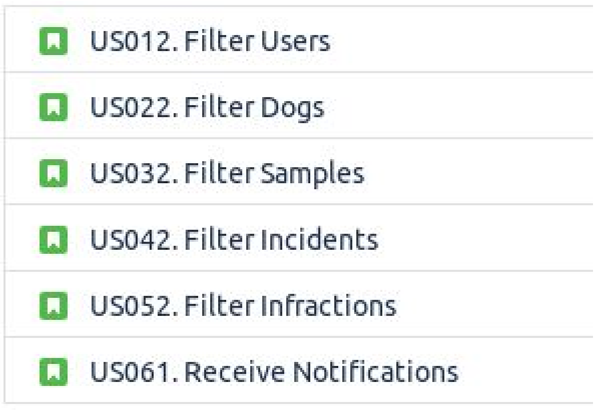
CALENDARIO BASE:	Proven2
Día	Horas
lunes	15:15 - 18:15, 18:45 - 21:30
martes	15:15 - 18:15
miércoles	No laborable
jueves	15:15 - 18:15, 18:45 - 21:30
viernes	15:15 - 18:15
sábado	No laborable
domingo	No laborable
Excepciones:	
Fecha	Horas
lun 29/04/19 - mar 30/04/19	No laborable
mié 01/05/19	No laborable

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

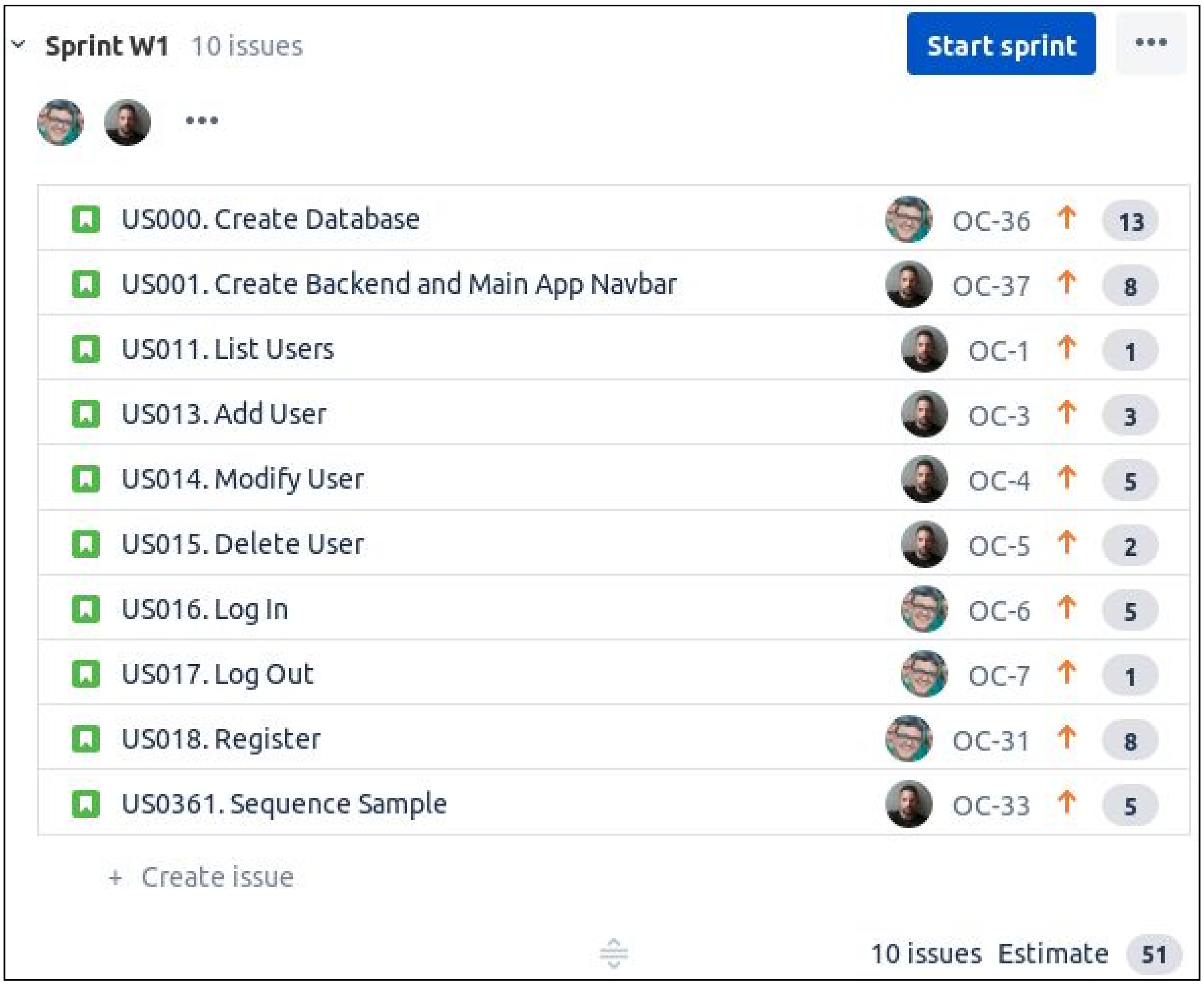

- US051. List Infractions
- US053. Add Infraction
- US054. Modify Infraction
- US055. Delete Infraction
- US056. Validate Infraction
- US057. Generate Offcial Document


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

Sprints Planning

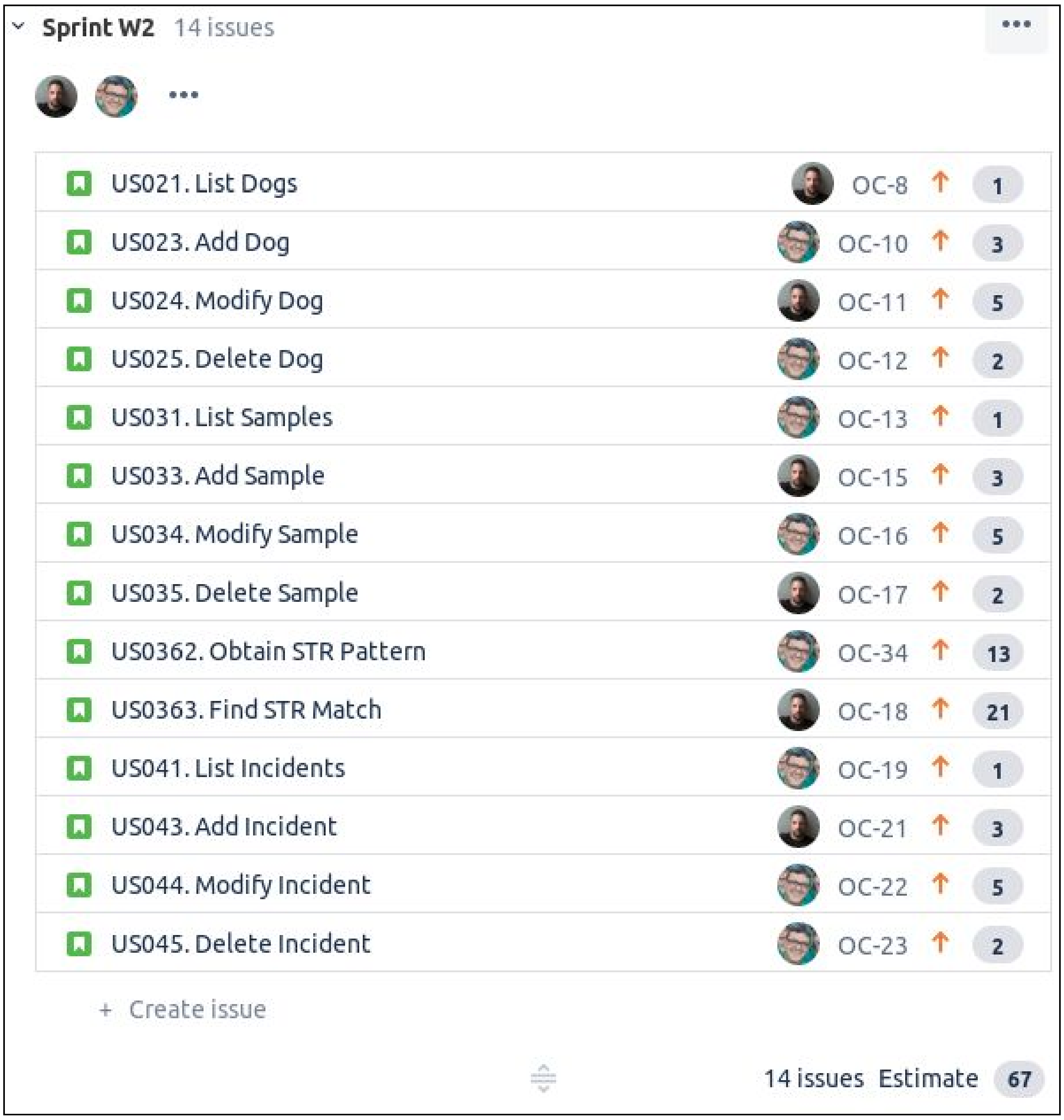
We are going to do 4 sprints, and each sprint is going to take 1 week.

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



US000. Create Database	OC-36	13
US001. Create Backend and Main App Navbar	OC-37	8
US011. List Users	OC-1	1
US013. Add User	OC-3	3
US014. Modify User	OC-4	5
US015. Delete User	OC-5	2
US016. Log In	OC-6	5
US017. Log Out	OC-7	1
US018. Register	OC-31	8
US0361. Sequence Sample	OC-33	5

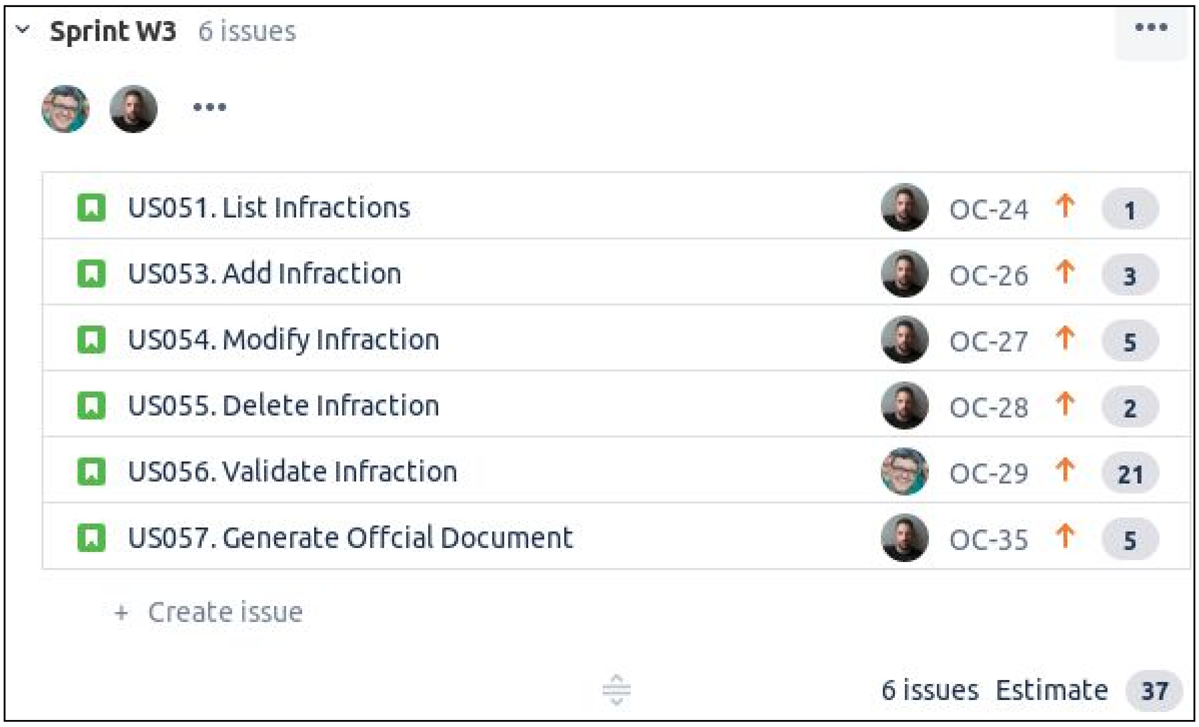
10 issues Estimate 51



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

14 issues Estimate 67

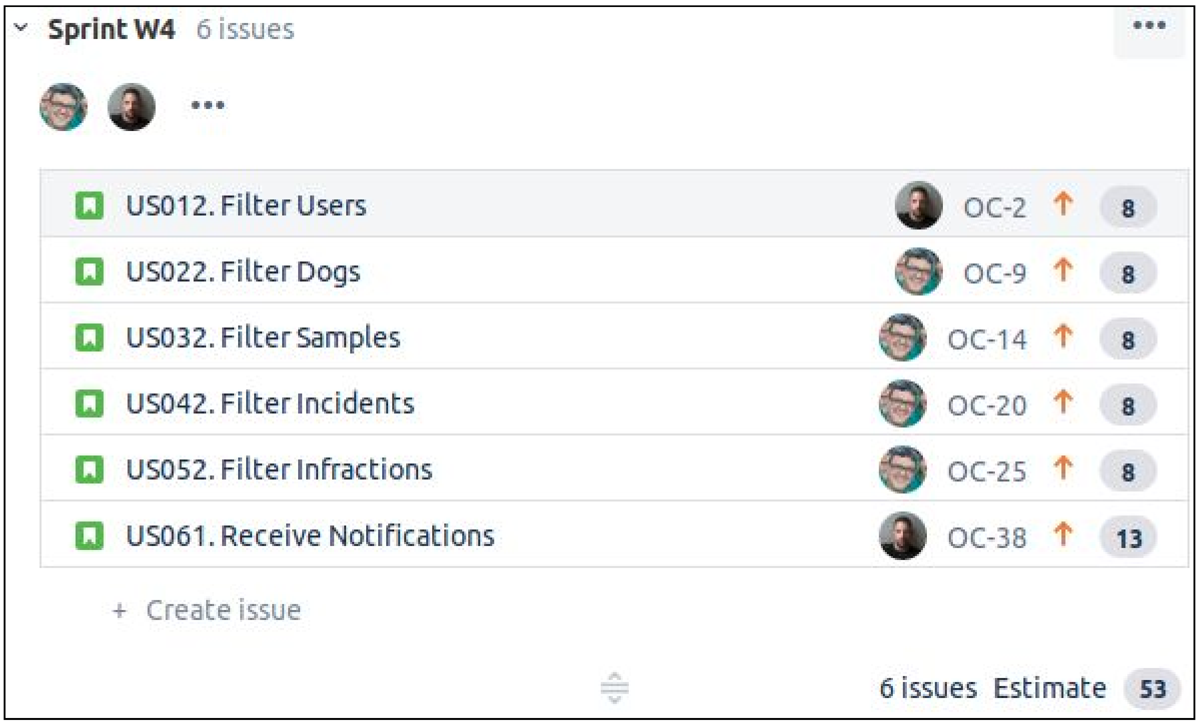


Sprint W3 6 issues

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

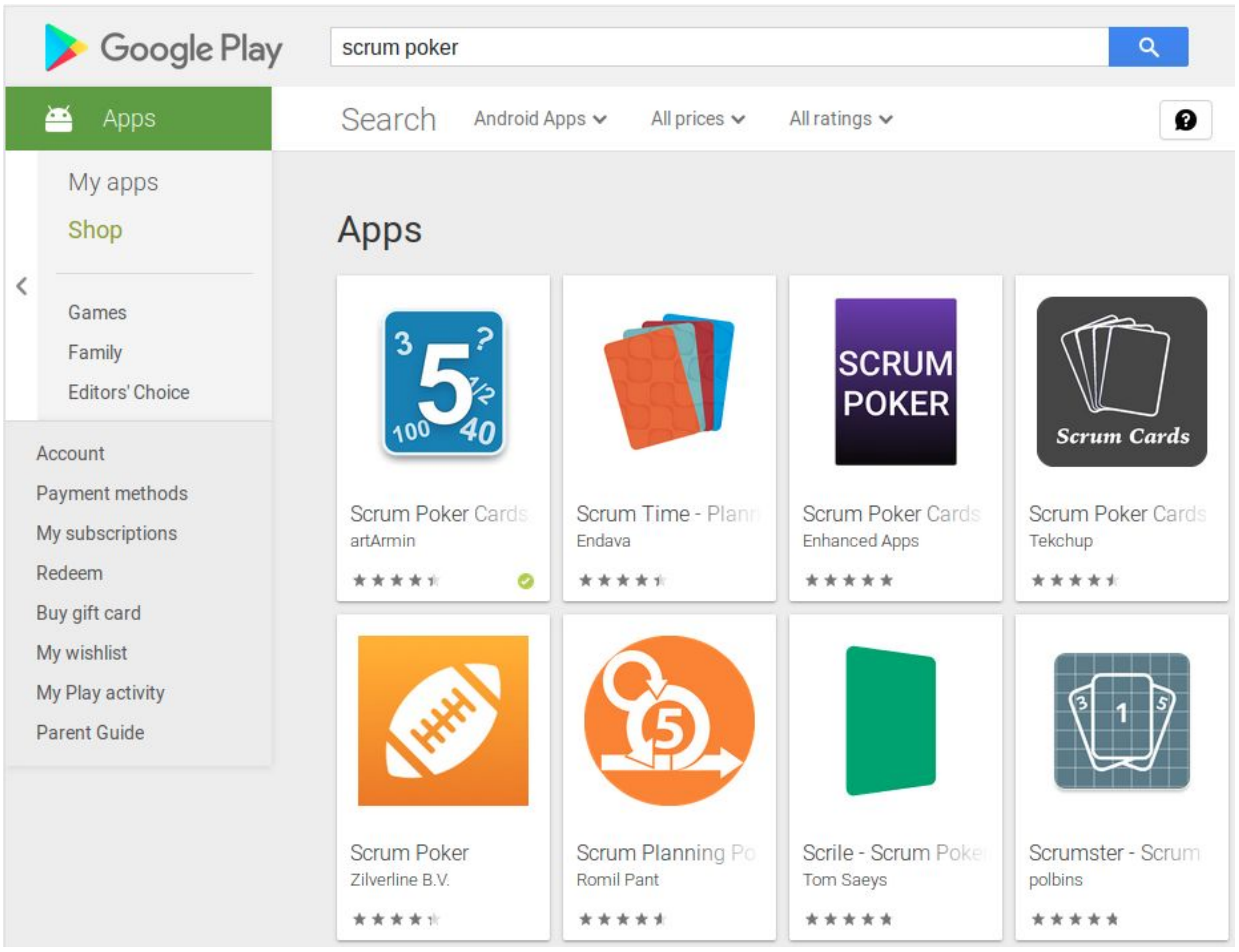
+ Create issue

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.



App Store Preview

This app is only available on the App Store for iOS devices.

Scrum Poker Planning (cards)

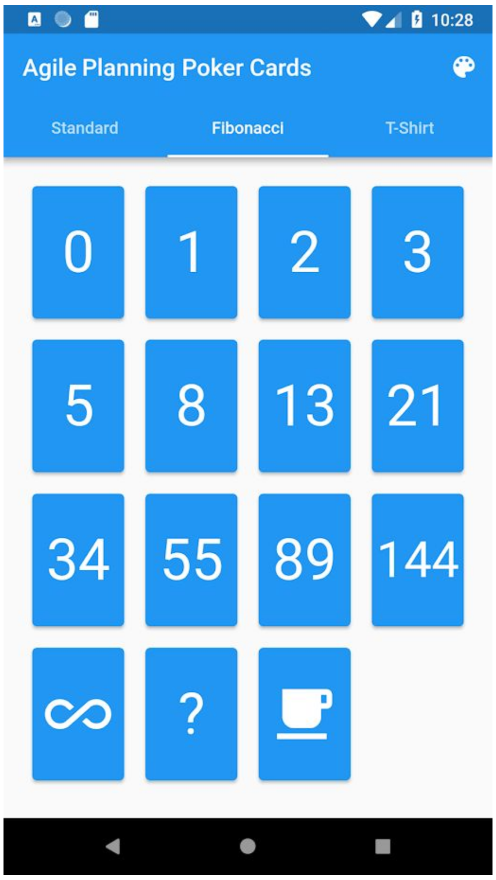
4+

Ruslan Balkarov

★★★★★ 4.9, 7 Ratings

Free

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.

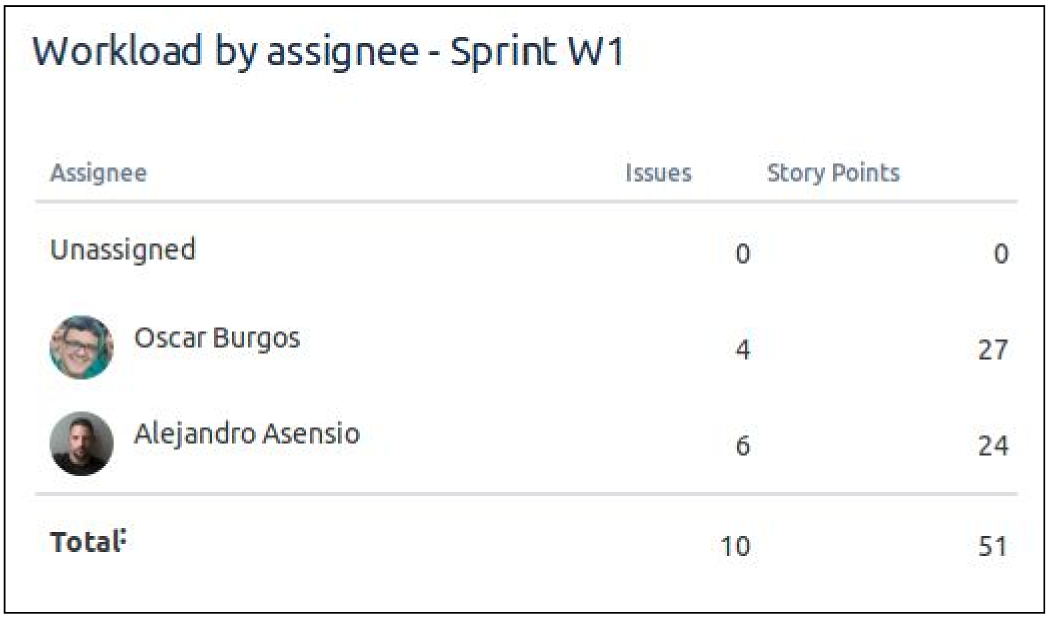
For each sprint, we have assigned approximately the same amount of user stories per each developer, regarding the effort points assigned by Scrum Poker previously.

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

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

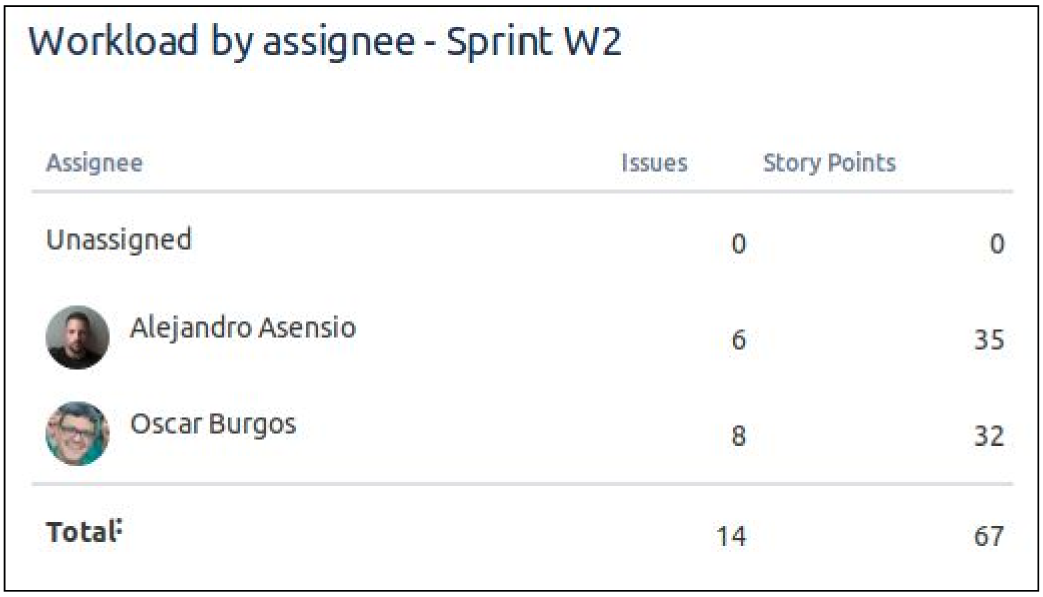
We have 1 month to develop de app, so if we divide the total effort points by 4 weeks, the resulting planning should have 4 sprints of approximately:

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



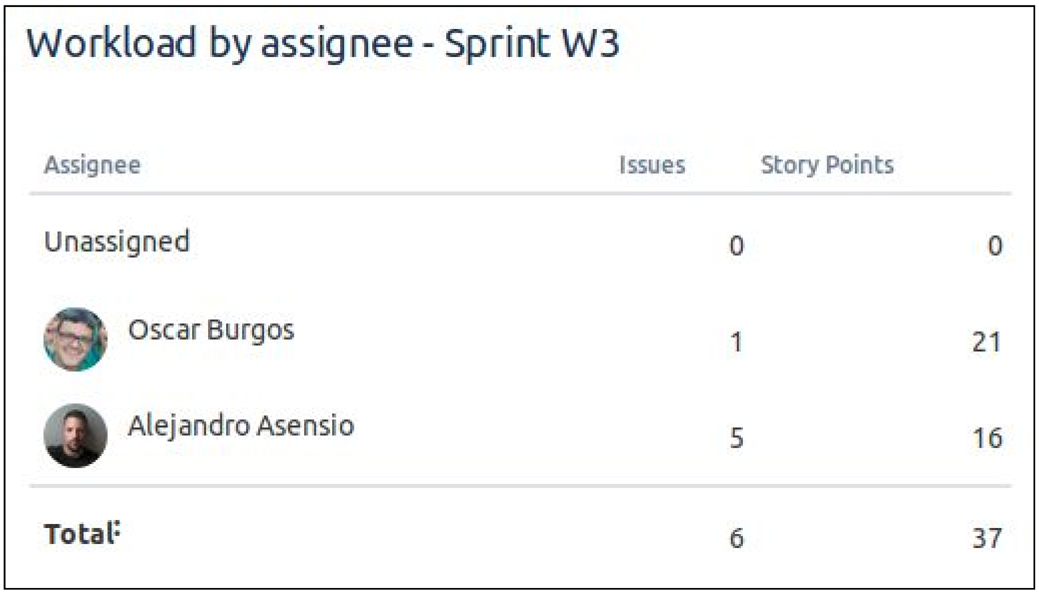
Workload by assignee - Sprint W1

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



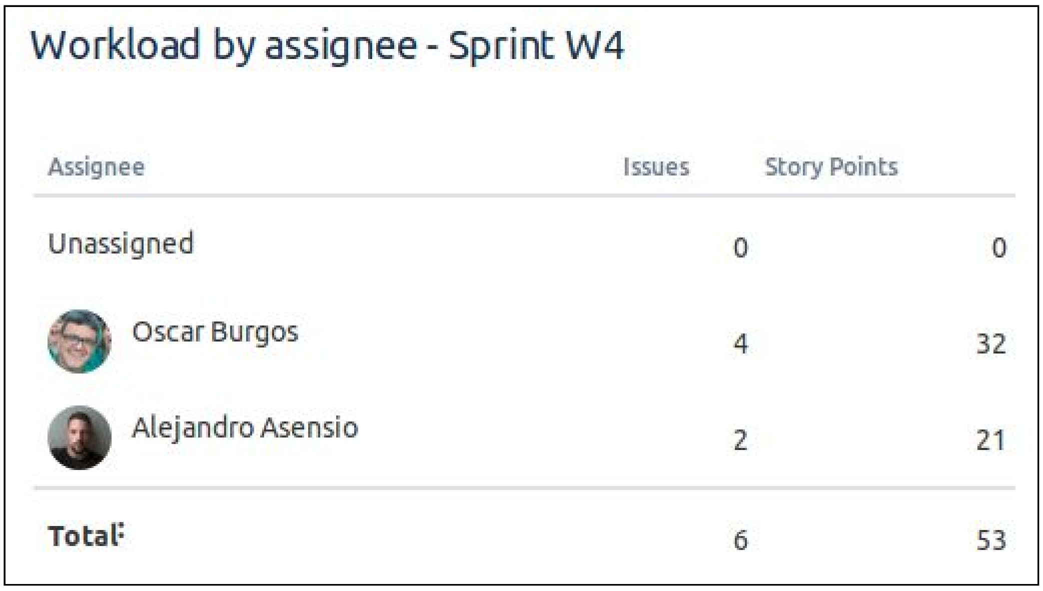
Workload by assignee - Sprint W2

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



Workload by assignee - Sprint W3

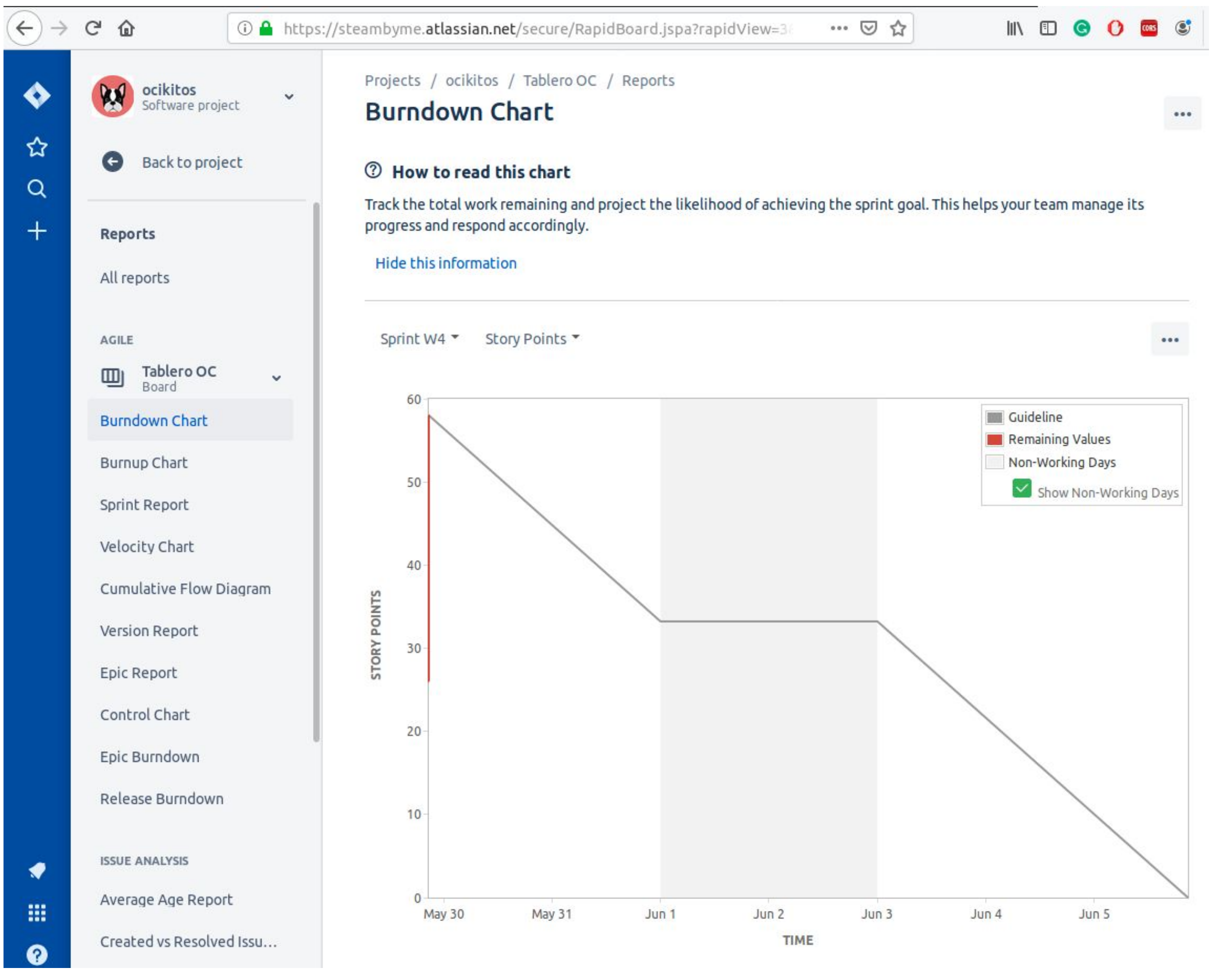
Assignee	Issues	Story Points
Unassigned	0	0
Oscar Burgos	1	21
Alejandro Asensio	5	16
Total:	6	37



Workload by assignee - Sprint W4

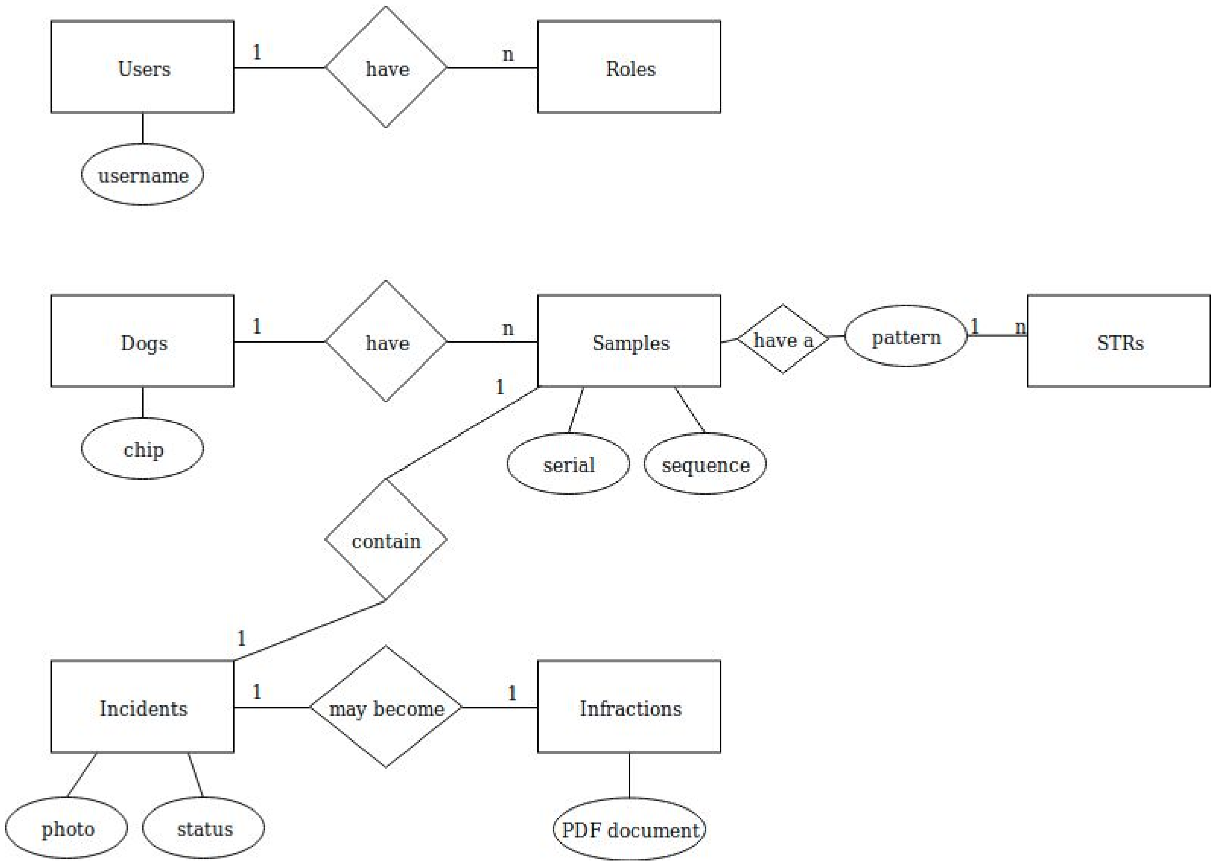
Assignee	Issues	Story Points
Unassigned	0	0
Oscar Burgos	4	32
Alejandro Asensio	2	21
Total:	6	53

Burndown Chart

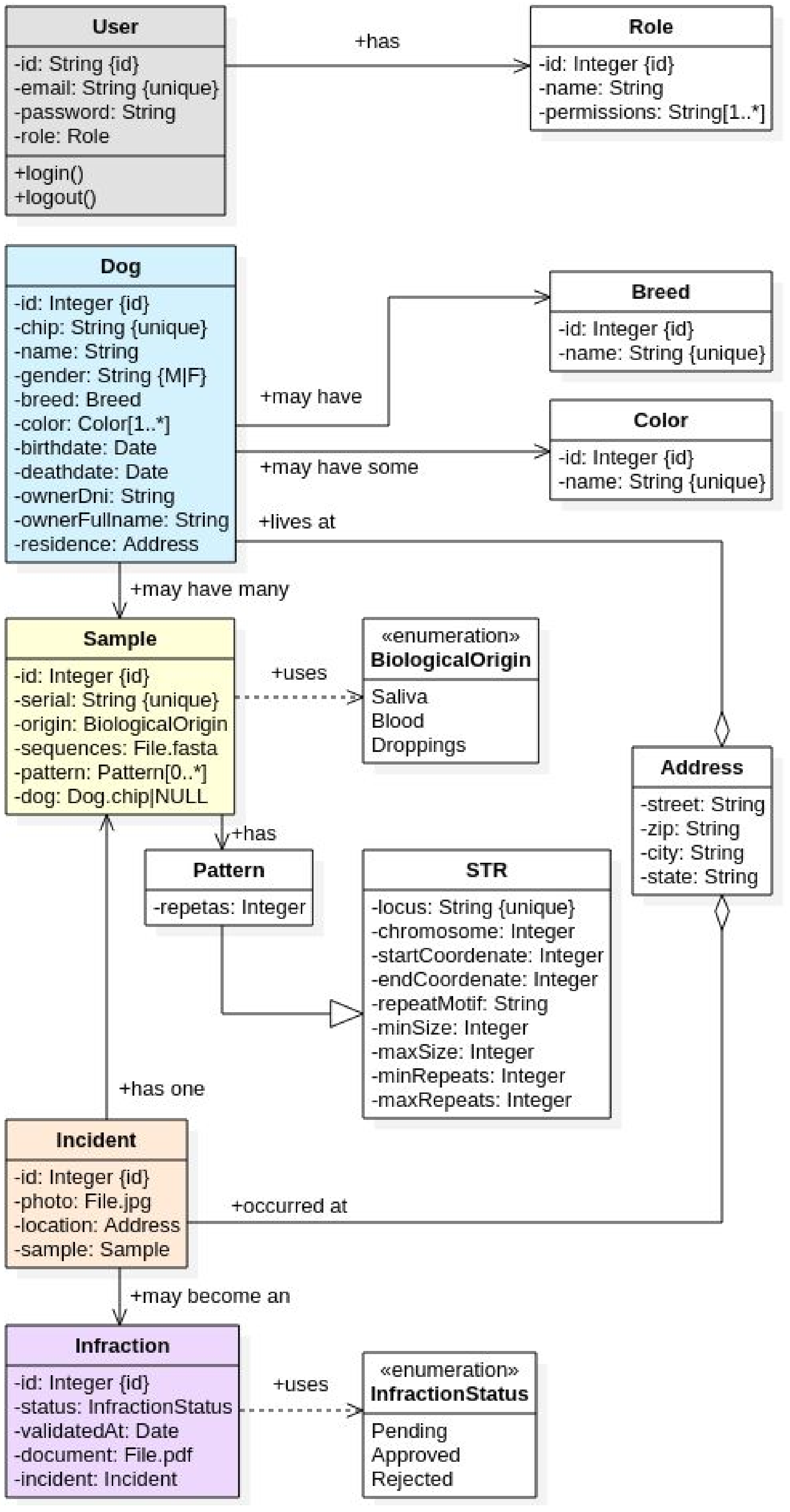


Track the total work remaining and project the likelihood of achieving the sprint goal. This helps your team manage its progress and respond accordingly.

Entity-Relationship Diagram



Class Diagram



Use Case Specification

To understand better the following **Use Case Specifications**, first check the Use Case Diagram that illustrates them.

These Use Cases are specified by the initials US, which refers to **User Stories**. Those User Stories will be used later to make the project planning with an agile methodology (Scrum).

There are some use cases grouped by the same type of action.

Id	US011, US021, US031, US041, US051
Name	List users|dogs|samples|incidents|infractions
Description	List all elements of the corresponding class in table format.
Normal Flux	
Actors	User with the required role for each object data management.
Pre-conditions	To be logged in the app.
Activation	Click on list button.
Description	A list of users|dogs|samples|incidents|infractions appears in table format.
Post-conditions	User can click on a row to check the detailed info about that object.
Alternative Flux 1	
Description	There is no elements in database to list.
Post-conditions	A descriptive info message is shown to the user.
Alternative Flux 2	
Description	There is some database error while retrieving data.
Post-conditions	A descriptive warning message is shown to the user.

Id	US012, US022, US032, US042, US052
Name	Filter users|dogs|samples|incidents|infractions
Description	Filter all elements of the corresponding class in table format.
Normal Flux	

Actors	User with the required role for each object data management.
Pre-conditions	To be logged in the app.
Activation	Click on filter button.
Description	User types in or selects the filter criteria to get a list of filtered users|dogs|samples|incidents|infractions in table format.
Post-conditions	User can click on a row to check the detailed info about that object or change the filtering criteria.
Alternative Flux 1	
Description	There is no elements in database that match the filtering criteria.
Post-conditions	A descriptive info message is shown to the user.
Alternative Flux 2	
Description	There is some database error while retrieving data.
Post-conditions	A descriptive warning message is shown to the user.

Id	US013, US023, US033, US043, US053
Name	Add user|dog|sample|incident|infraction
Description	Add a new object of a given class to its corresponding table in database.
Normal Flux	
Actors	User with the required role for each object data management.
Pre-conditions	To be logged in the app.
Activation	Click on add button.
Description	User completes the according form to add a new user|dog|sample|incident|infraction.
Post-conditions	A success message is shown to the user.
Alternative Flux 1	
Description	There is some form field with validation errors.
Post-conditions	As many info messages as validation errors are shown to the user.
Alternative Flux 2	
Description	There is some error while sending data to database.

Post-conditions	A descriptive warning message is shown to the user.

Id	US014, US024, US034, US044, US054
Name	Modify users|dogs|samples|incidents|infractions
Description	Modify an object of the corresponding class.
Normal Flux	
Actors	User with the required role for each object data management.
Pre-conditions	To be logged in the app.
Activation	Click on modify button.
Description	User completes the according form to modify an existing user|dog|sample|incident|infraction.
Post-conditions	A success message is shown to the user.
Alternative Flux 1	
Description	There is some form field with validation errors.
Post-conditions	As many info messages as validation errors are shown to the user.
Alternative Flux 2	
Description	There is some error while sending data to database.
Post-conditions	A descriptive warning message is shown to the user.

Id	US015, US025, US035, US045, US055
Name	Delete user|dog|sample|incident|infraction
Description	Delete an existing object of a given class from its corresponding table in database.
Normal Flux	
Actors	User with the required role for each object data management.
Pre-conditions	To be logged in the app.
Activation	Click on delete button.
Description	User confirms that is sure to delete the selected user|dog|sample|incident|infraction.
Post-conditions	A success message is shown to the user.

Alternative Flux 1	
Description	The selected object cannot be deleted due to dependency reasons.
Post-conditions	A descriptive warning message is shown to the user.
Alternative Flux 2	
Description	There is some error while sending the deletion action to database.
Post-conditions	A descriptive warning message is shown to the user.

Id	US016
Name	Login
Description	Log in the app with username and password.
Normal Flux	
Actors	Any existing user.
Pre-conditions	To be registered in the app.
Activation	Click on login button.
Description	User types in the username and password and submits the form.
Post-conditions	The user is redirected to the dashboard according with its role.
Alternative Flux 1	
Description	The username doesn’t exist.
Post-conditions	A descriptive info message is shown to the user.
Alternative Flux 2	
Description	The password is not valid for the given username.
Post-conditions	A descriptive info message is shown to the user.

Id	US017
Name	Logout
Description	Log out from the app.
Normal Flux	
Actors	Any existing user.

Pre-conditions	To be logged in the app.
Activation	Click on logout button.
Description	User clicks in the logout button to end its active session in the app.
Post-conditions	The user is redirected to the home page.
Alternative Flux 1	
Description	There is some error while sending the logout action to the backend.
Post-conditions	A descriptive warning message is shown to the user.

Id	US018
Name	Register
Description	Register a new user in the app.
Normal Flux	
Actors	Any future worker in our app.
Pre-conditions	Not to be registered yet in the app.
Activation	Click on register button.
Description	User completes the required fields and submits the form.
Post-conditions	The user is automatically logged in and redirected to the welcome page.
Alternative Flux 1	
Description	The typed username already exists.
Post-conditions	A descriptive info message is shown to the user.
Alternative Flux 2	
Description	The fields password and password repeat don’t match.
Post-conditions	A descriptive info message is shown to the user.

Id	US036 (includes US0361, US0362, US0363)
Name	Analyze Sample (includes Sequence Sample, Obtain STR pattern, Find STR match)
Description	Analyze a new or an existing sample.

Normal Flux	
Actors	User with Bioinformatics Technician (“bio”) role.
Pre-conditions	To be logged in the app as a bio user.
Activation	Click on analyze sample button.
Description	Firstly, the sequence machine gives us the FASTA file with the raw DNA. Secondly, the STR pattern is obtained (number of repeats of each target locus). Thirdly, the previous STR pattern is tested against all database to find a potential matching.
Post-conditions	If matching occurs, a success message is shown to the user and the existing dog details are shown. If matching doesn’t occur and the sample comes from the veterinarian clinic, the current sample is stored as the first sample of the new dog, as well as its STR pattern; otherwise (the sample comes from the street agent), the sample is stored as an anonymous sample.
Alternative Flux 1	
Description	There is some errors while obtaining STR pattern.
Post-conditions	A descriptive info message is shown to the user.
Alternative Flux 2	
Description	There is some errors while looking for STR matching.
Post-conditions	A descriptive info message is shown to the user.

Id	US056
Name	Validate Infraction
Description	Validate (approve or reject) an infraction proposal.
Normal Flux	
Actors	User with Police Officer (“oficer”) role.
Pre-conditions	An existing DNA sample has been match with another one, that means, a dog has been identified.
Activation	Click on validate infraction button.
Description	User checks all the evidences of the infraction and submits the form with the “approve” button.
Post-conditions	An automated official document of approval is generated and a descriptive success message is shown to the user.

Alternative Flux 1	
Description	User checks all the evidences of the infraction and submits the form with the "reject" button.
Post-conditions	An automated official document of rejection is generated and a descriptive success message is shown to the user.
Alternative Flux 2	
Description	There is some error while sending the validation action to the backend.
Post-conditions	A descriptive info message is shown to the user.

Id	US057
Name	Generate Official Document
Description	Generate a PDF document to leave proof of the Police Officer approval or rejection of an infraction.
Normal Flux	
Actors	This is an automatic process after each time an officer validates an infraction.
Pre-conditions	An existing infraction proposal has to be with "pending" status.
Activation	An officer clicks on "approve" or "reject" button of the validate infraction form.
Description	User checks all the evidences of the infraction and submits the form with the "approve" button.
Post-conditions	An automated official document of approval is generated and a descriptive success message is shown to the user.
Alternative Flux 1	
Description	There is some error while sending the validation action to the backend.
Post-conditions	A descriptive info message is shown to the user.

Test Data

This document is the Test Data content. Note that the coloured tables are the main models of the app, while the grayed-out ones exist to give support.

ROLES									
id (PK)	name (UNIQUE)	permissions							
0	admin	all							
1	vet	dogs							
2	bio	samples							
3	agent	incidents							
4	officer	infractions							
USERS									
username (PK)	password	fullname	email (UNIQUE)	role_id (FK)					
aasensio	aasensio1.	Alejandro Asensio	aasensio@ocikitos.com	0					
obp	obp1.	Óscar Burgos	obp@ocikitos.com	1					
eaguayo	eaguayo1.	Elisabet Aguayo	eaguayo@happydogs.com	1					
abarcelo	abarcelo1.	Ainhoa Barceló	abarcelo@bsc.com	2					
paco	paco1.	Francisco Regaña	paco@ciutatneta.bcn	3					
mgonz	mgonz1.	Marisa González	mgonz@urbana.lh	4					
BREEDS									
id (PK)	name								
1	Border Collie								

2	Husky								
3	German Shepherd								
4	St. Bernard								
5	Basset Hound								
COLORS									
id (PK)	name								
1	white								
2	black								
3	brown								
4	lightbrown								
5	grey								
6	lightgrey								
DOGS									
chip (PK)	name	gender	breed_id (FK)	color_id (FK)	birthdate	death date	owner_dni	owner_fullname	
1001	Lassie	female	1	3	1951-05-06	1963-07-05	69456058Z	Michael Jordan	
1002	Laika	female	2	1, 2	1942-07-30	1957-11-03	47231677G	Elvis Presley	
1003	Rin-tin-tin	male	3	1, 4	1949-11-18	1959-06-19	29808652P	Juana de Arco	
1004	Beethoven	male	4	1, 3	1991-05-11	2007-03-10	58325882K	Albert Einstein	
1005	Poochy	male	5	2, 3	2001-09-27	2001-10-12	94542968L	Homer Simpson	
STRS									
locus (PK)	chromosome	start_coordenate	end_coordenate	repeat_motif	min_size	max_size	min_repeats	max_repeats	annealing_temp

FH2001	23	50961325	50961475	GATA	119	160	30	40	51
FH2004	11	32161381	32161621	AAAG	233	325	58	81	64
FH2010	24	5196383	5196605	ATGA	222	243	56	61	57
FH2054	12	37914504	37914739	GATA	139	177	35	44	57
FH2088	15	53905651	53905779	TTTA/TTCA	95	138	12	17	56
FH2107	3	83830247	83830574	GAAA	292	426	73	107	54
FH2309	1	85772974	85773377	GAAA	340	428	85	107	52
FH2328	33	19158127	19158477	GAAA	171	213	43	53	58
FH3377	3	78748898	78749090	GAAAA	184	305	37	61	54
PEZ02	17	13276076	13276209	GGAA	104	144	26	36	60
PEZ05	12	60326434	60326541	TTTA	92	116	23	29	57
PEZ16	27	10305692	10305995	GAAA	281	332	70	83	57
PEZ17	4	71904833	71905038	GAAA	191	225	48	56	59
PEZ21	2	36438658	36438751	AAAT	83	103	21	26	52
VWF.X	27	41977918	41978074	AGGAAT	151	187	25	31	57

SAMPLES

serial (PK)	origin	sequence	pattern	dog_id (FK)
2001	saliva	2001.fasta	2001.pattern.txt	1001
2002	blood	2002.fasta	2002.pattern.txt	1002
2003	droppings	2003.fasta	2003.pattern.txt	1003
2004	saliva	2004.fasta	2004.pattern.txt	1004

2005	droppings	2005.fasta	2005.pattern.txt	NULL					
INCIDENTS									
id (PK)	location	photo	sample_serial						
3001	Carrer Rovires 69 num. 25	20190425012059.jpg	2001						
3002	Carrer Rovires 69 num. 26	20190425012147.jpg	2002						
3003	Carrer Rovires 69 num. 27	20190425020323.jpg	2003						
3004	Carrer Rovires 69 num. 28	20190426165241.jpg	2004						
3005	Carrer Rovires 69 num. 29	20190427132321.jpg	2005						
INFRACTIONS									
id (PK)	status	validated_at	document	incident_id (FK)					
4001	pending	NULL	4001.pdf	3001					
4002	approved	2019-03-26	4002.pdf	3002					
4003	rejected	2019-03-26	4003.pdf	3003					
4004	approved	2019-03-26	4004.pdf	3004					
4005	pending	NULL	4005.pdf	3005					

Conclusions

In the future, we may consider using the graphic user interface **Voyager admin tool for Laravel** [https://laravelvoyager.com/] to manage changes on the backend more quickly and consistently.

Personal Conclusions by Alejandro

This Project began with a lot of passion regarding the topic that my colleague Oscar and me have chosen: do a social improvement in our town and cities.

At mid-term tempo, we began to see that the time was going to be the limitation for our excitement. Working under pressure is tough but when the things start to be done, we could see the end of the tunnel.

In web developing terms, the previous words translate to facing OAuth2 issues because we wanted to use the official package Laravel Passport. And we finally made it. Then, Angular client had to consume the API with that particular authentication protection, handling the access token back and forth.

This little section meant to be an assertive and constructive feedback. That being said, I think that this project is a challenging opportunity to put in order all the knowledge accomplished in these two years, but I sincerely think that it would be nicer without other subjects (and their respective exams) to attend. I hope that lines will help the next generations of students, and I wish them all luck in this adventure of coding.

Personal Conclusions by Oscar

Once again I do not want to be reiterative in the demand for the extension of the time allotted for the realization of it, since it is a very important summary of the knowledge acquired during these last two years.

This project has allowed me to acquire new knowledge and skills in overcoming challenges that we have encountered at each step of the evolution of its development, reflecting a future work environment.

The beginning of this cycle, "Development of web programming with a specialty in bioinformatics", I raised it as a personal challenge; because of age and the acquisition of knowledge in biology, programming had been with me for many years but this time it would allow me to update my training. And at the beginning of this project I considered tackling my two biggest handicaps of this cycle; On the one hand, use the frontend technology with the Angular 7 framework and on the other hand the use of the English language. Not only have I managed to improve the learning outcomes stipulated in the M06 module, but I have also increased them by using new content that has not been possible to learn in the module.

It has also allowed me to work, in coordination and collaboration, with a partner to achieve a common goal; taking the opportunity to thank him for the involvement and patience with me at all times in the face of difficulties.

Thanks to my parents, classmates, project partner, teachers and especially to Conxi, Laia and Tania… Blessed patience.

Bibliography

Works Cited

https://www.mybib.com/b/noD5oN

"Angular Materials." *Angular Materials*, material.angular.io/.

asciimoo. "Asciimoo/Exrex." *GitHub*, 14 Jan. 2019, github.com/asciimoo/exrex. Accessed 6 May 2019.

barryvdh. "Barryvdh/Laravel-Cors." *GitHub*, 26 Feb. 2019, github.com/barryvdh/laravel-cors. Accessed 22 May 2019.

"Canis Lupus Familiaris (ID 85) - Genome - NCBI." *Nih.Gov*, 2016, www.ncbi.nlm.nih.gov/genome/85. Accessed 7 May 2019.

"Dog STRs." *Nist.Gov*, National Institute of Standards and Technology, 2018, strbase.nist.gov/dogSTRs.htm. Accessed 24 Apr. 2019.

U.S. Department of Commerce

Font. "Font Awesome 5 Icons." *Fontawesome.Com*, Font Awesome 5 Icons, 2017, fontawesome.com/icons. Accessed 28 May 2019.

"Genetic Data from 15 STR Loci for Forensic Individual Identification and Parentage Analyses in UK Domestic Dogs (Canis Lupus Familiaris)." *Https://Www.Research.Ed.Ac.Uk/Portal/Files/21730846/15_STR_loci.Pdf*, 26 Apr. 2019, www.research.ed.ac.uk/portal/files/21730846/15_STR_loci.pdf.

Google, INC. "Angular." *Angular.Io*, 2019, angular.io/. Accessed 28 May 2019.

"Home | MyDogDNA." *Mydogdna.Com*, 2015, mydogdna.com/. Accessed 29 Apr. 2019.

"Identificación Animal." *Perros.Com*, 2010, www.perros.com/foros/general/veterinaria/identificacion-animal.html. Accessed 19 May 2019.

"Laravel Voyager." *Laravelvoyager.Com*, 2019, laravelvoyager.com/. Accessed 12 May 2019.

Lucidchart. "UML Use Case Diagram Tutorial." *YouTube*, 7 Feb. 2018, www.youtube.com/watch?v=zid-MVo7M-E. Accessed 14 Apr. 2019.

Media, Traversy. "Full Stack Vue.Js & Laravel." *YouTube*, 19 Feb. 2018, www.youtube.com/watch?v=DJ6PD_jBtU0. Accessed 16 Apr. 2019.

---. "Laravel 5.5 API from Scratch Using Resources." *YouTube*, 29 Dec. 2017, www.youtube.com/watch?v=4pc6cgisbKE. Accessed 16 Apr. 2019.

---. "Laravel from Scratch." *YouTube*, 2019, www.youtube.com/playlist?list=PLillGF-RfqbYhQsN5WMXy6VsDMKGadrJ-. Accessed 15 Apr. 2019.

"Microchip Dog Id." *Http://Www.Realtrace.Com/Page-Sp/Regulacion*, 27 Apr. 2019, www.realtrace.com/page-sp/regulacion.

"NGX Barcode." *NGX Barcode*, www.npmjs.com/package/ngx-barcode.

Otto, Mark. "Bootstrap." *Getbootstrap.Com*, 2000, getbootstrap.com/.

Otwel, Taylor. "Laracasts." *Laracasts*, 2016, laracasts.com/series/whats-new-in-laravel-5-3/episodes/13. Accessed 24 May 2019.

Otwell, Taylor. "Laravel - The PHP Framework For Web Artisans." *Laravel.Com*, 2015, laravel.com/. Accessed 25 May 2019.

---. "Laravel Passport." *Laravel.Com*, 2019, laravel.com/docs/5.8/passport. Accessed 27 May 2019.

Parker, Heidi G. “Genomic Analyses of Modern Dog Breeds.” *Mammalian Genome*, vol. 23, no. 1–2, 10 Jan. 2012, pp. 19–27, www.ncbi.nlm.nih.gov/pmc/articles/PMC3559126/, 10.1007/s00335-011-9387-6. Accessed 29 Apr. 2019.

Savani, Hardik. “Build RESTful API In Laravel 5.8 Example.” *Itsolutionstuff.Com*, 2019, itsolutionstuff.com/post/build-restful-api-in-laravel-58-exampleexample.html.

The Apache Software Foundation. “VirtualHost Examples - Apache HTTP Server Version 2.4.” *Apache.Org*, 2019, httpd.apache.org/docs/2.4/vhosts/examples.html. Accessed 18 Apr. 2019.

Wictum, Elizabeth. “Developmental Validation of DogFiler, a Novel Multiplex for Canine DNA Profiling in Forensic Casework.” *Https://Www.Fsigenetics.Com/Article/S1872-4973(12)00174-3/Pdf*, 2013, www.fsigenetics.com/article/S1872-4973(12)00174-3/pdf.

Http://Www.Fci.Be/Es/, www.fci.be/es/.

Fédération Cynologique Internationale es la Organización Canina Mundial. *347 razas reconocidas *10 grupos de razas
26 Apr. 2019, www.research.ed.ac.uk/portal/files/21730846/15_STR_loci.pdf.

Annexes

Annex 1 – API Documentation

The backend of out project is made with **Laravel**, a widely known PHP framework. The version used in this project has been the version 5.8.

With Laravel, we have build a **RESTful API** with **OAuth2** authentication, using the official package **Laravel Passport.**

We decide to separate the backend from the frontend in order to, in the future, possibly build more client applications that consume the same database, with the same backend logic. This can be achieved developing a RESTful API that serves URL, some protected under OAuth2 authentication; this is, the user must be logged in the app to perform some actions.

Considering the client consumption side, there are two HTTP Headers that must be present in every HTTP Request to the API:

```
Accept:application/json
Authorization:'Bearer '.$accessToken
```

Users management:

```
POST http://apps.proven.cat/~dawbi1901/api/api/register
POST http://apps.proven.cat/~dawbi1901/api/api/login
```

Dogs, Samples, Incidents and Infractions management:

Create Dog
```
POST http://apps.proven.cat/~dawbi1901/api/api/dogs
```

HTTP Body:

```
chip:123456789012345
name:Dug
gender:male
breed_id:1
color_id:2
birthdate:2000-01-01
deathdate:
owner_dni:46477213D
owner_fullname:Alejandro Asensio
residence:Calle Falsa, 123
```

Read All Dogs

```
GET http://apps.proven.cat/~dawbi1901/api/api/dogs
```

Read One Dog (by its primary key 'id' or the whole Dog object)

```
GET http://apps.proven.cat/~dawbi1901/api/api/dogs/51
```

Update A Dog

```
PUT/PATCH
http://apps.proven.cat/~dawbi1901/api/api/dogs/51?chip=123456789012346&nam
e=Dig&gender=female&breed_id=1&color_id=2&birthdate=2000-01-02&deathdate=2
010-12-30&owner_dni=X6477213D&owner_fullname=Alejandro
Asensio&residence=Calle de la Piruleta S/N
```

HTTP Params to send for update:

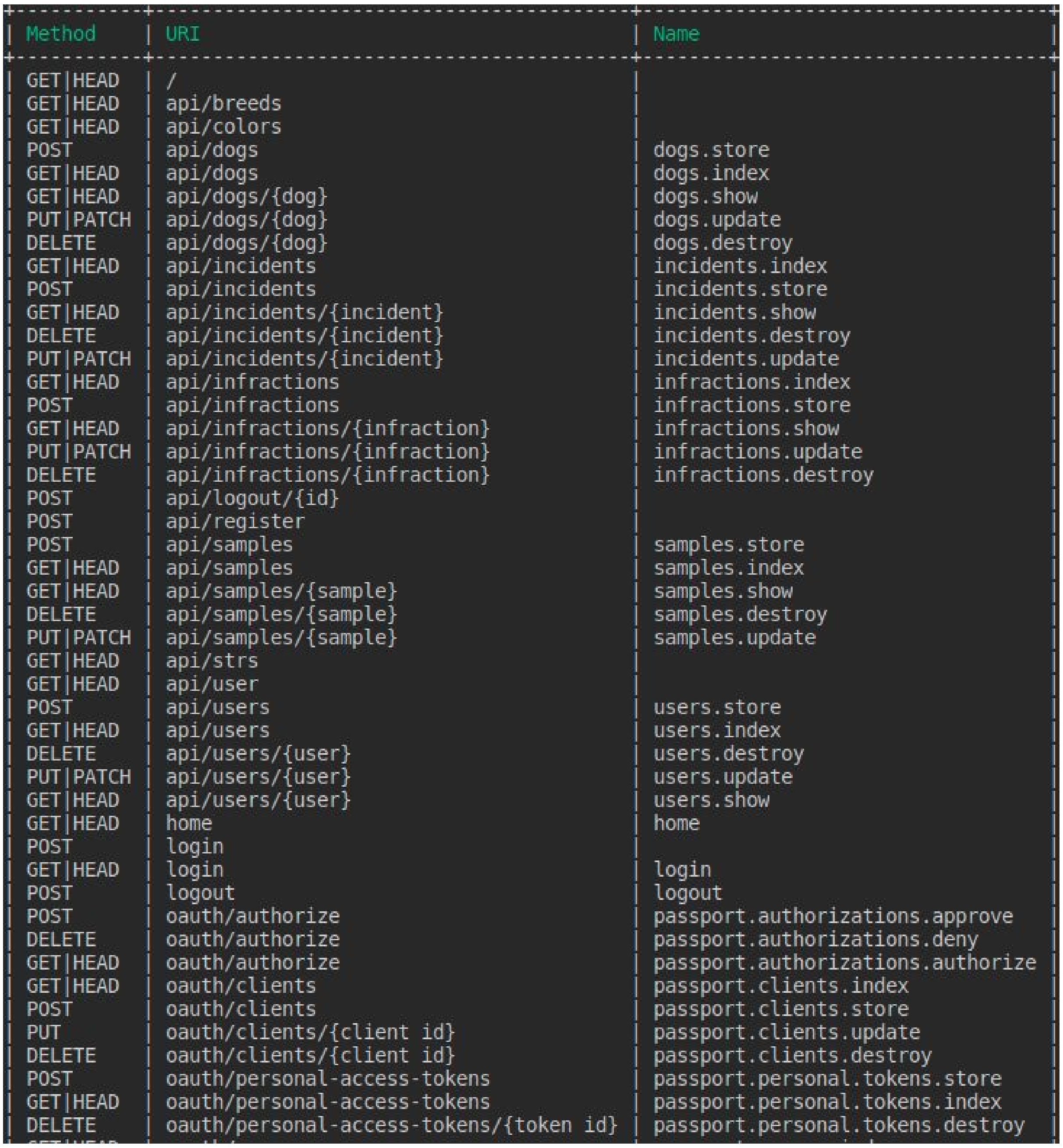
```
chip:123456789012346
name:Dig
gender:female
breed_id:1
color_id:2
birthdate:2000-01-02
deathdate:2010-12-30
owner_dni:X6477213D
owner_fullname:Alejandro Asensio
residence:Calle de la Piruleta S/N
```

Delete A Dog

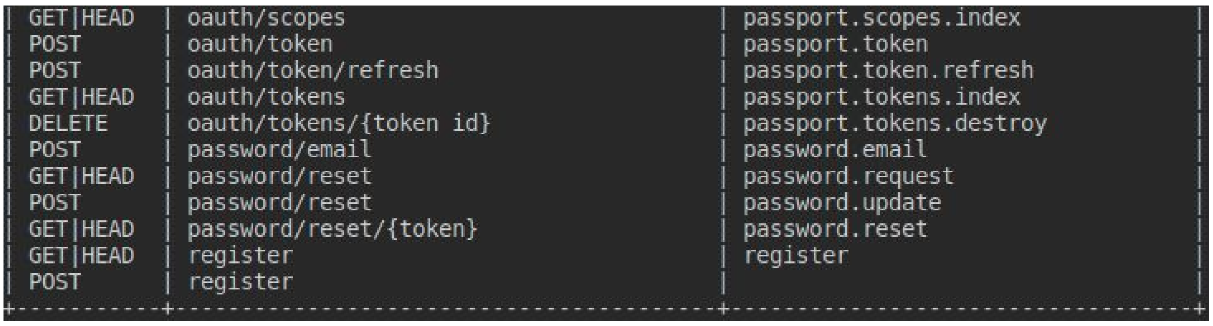
```
DELETE http://apps.proven.cat/~dawbi1901/api/api/dogs/51
```

API Routes

To be more accurate, here below are listed all the accepted routes by the API:



```
+-----------+------------------------------------------+--------------------------------------+
| Method    | URI                                      | Name                                 |
+-----------+------------------------------------------+--------------------------------------+
| GET|HEAD  | /                                        |                                      |
| GET|HEAD  | api/breeds                               |                                      |
| GET|HEAD  | api/colors                               |                                      |
| POST      | api/dogs                                 | dogs.store                           |
| GET|HEAD  | api/dogs                                 | dogs.index                           |
| GET|HEAD  | api/dogs/{dog}                           | dogs.show                            |
| PUT|PATCH | api/dogs/{dog}                           | dogs.update                          |
| DELETE    | api/dogs/{dog}                           | dogs.destroy                         |
| GET|HEAD  | api/incidents                            | incidents.index                      |
| POST      | api/incidents                            | incidents.store                      |
| GET|HEAD  | api/incidents/{incident}                 | incidents.show                       |
| DELETE    | api/incidents/{incident}                 | incidents.destroy                    |
| PUT|PATCH | api/incidents/{incident}                 | incidents.update                     |
| GET|HEAD  | api/infractions                          | infractions.index                    |
| POST      | api/infractions                          | infractions.store                    |
| GET|HEAD  | api/infractions/{infraction}             | infractions.show                     |
| PUT|PATCH | api/infractions/{infraction}             | infractions.update                   |
| DELETE    | api/infractions/{infraction}             | infractions.destroy                  |
| POST      | api/logout/{id}                          |                                      |
| POST      | api/register                             |                                      |
| POST      | api/samples                              | samples.store                        |
| GET|HEAD  | api/samples                              | samples.index                        |
| GET|HEAD  | api/samples/{sample}                     | samples.show                         |
| DELETE    | api/samples/{sample}                     | samples.destroy                      |
| PUT|PATCH | api/samples/{sample}                     | samples.update                       |
| GET|HEAD  | api/strs                                 |                                      |
| GET|HEAD  | api/user                                 |                                      |
| POST      | api/users                                | users.store                          |
| GET|HEAD  | api/users                                | users.index                          |
| DELETE    | api/users/{user}                         | users.destroy                        |
| PUT|PATCH | api/users/{user}                         | users.update                         |
| GET|HEAD  | api/users/{user}                         | users.show                           |
| GET|HEAD  | home                                     | home                                 |
| POST      | login                                    |                                      |
| GET|HEAD  | login                                    | login                                |
| POST      | logout                                   | logout                               |
| POST      | oauth/authorize                          | passport.authorizations.approve      |
| DELETE    | oauth/authorize                          | passport.authorizations.deny         |
| GET|HEAD  | oauth/authorize                          | passport.authorizations.authorize    |
| GET|HEAD  | oauth/clients                            | passport.clients.index               |
| POST      | oauth/clients                            | passport.clients.store               |
| PUT       | oauth/clients/{client id}                | passport.clients.update              |
| DELETE    | oauth/clients/{client id}                | passport.clients.destroy             |
| POST      | oauth/personal-access-tokens             | passport.personal.tokens.store       |
| GET|HEAD  | oauth/personal-access-tokens             | passport.personal.tokens.index       |
| DELETE    | oauth/personal-access-tokens/{token id}  | passport.personal.tokens.destroy     |
```



```
| GET|HEAD  | oauth/scopes                             | passport.scopes.index                |
| POST      | oauth/token                              | passport.token                       |
| POST      | oauth/token/refresh                      | passport.token.refresh               |
| GET|HEAD  | oauth/tokens                             | passport.tokens.index                |
| DELETE    | oauth/tokens/{token id}                  | passport.tokens.destroy              |
| POST      | password/email                           | password.email                       |
| GET|HEAD  | password/reset                           | password.request                     |
| POST      | password/reset                           | password.update                      |
| GET|HEAD  | password/reset/{token}                   | password.reset                       |
| GET|HEAD  | register                                 | register                             |
| POST      | register                                 |                                      |
+-----------+------------------------------------------+--------------------------------------+
```