

Department of Computer Science and Engineering



POLITECNICO
MILANO 1863

Design and Implementation of Mobile Applications
Design Document (DD)



WaterAPProval -v1.0-

Author

Laura Schiatti 904738
laura.schiatti@mail.polimi.it

September 7th, 2020

Contents

[1. Introduction](#)

[1.1. Context](#)

[1.2. Purpose](#)

[1.3. Definitions, Acronyms, Abbreviations](#)

[1.3.1. Definitions](#)

[1.3.2. Acronyms](#)

[1.4. Document structure](#)

[2. Architectural design](#)

[2.1. Overview](#)

[2.2. Selected architectural styles and patterns](#)

[2.2.1. Model View Controller architecture \(MVC\)](#)

[State management](#)

[2.3. Other design decisions](#)

[3. External web-service APIs](#)

[3.1. Aquapp API 2.0: Online REST API for Water Quality Monitoring](#)

[3.1.1. Useful endpoints](#)

[3.1.2. Local data model](#)

[3.2. Google Maps API \(v2\)](#)

[3.3. Invemar ICAM web service](#)

[4. User characteristics](#)

[5. User interface design](#)

[6. Requirements](#)

[4.1. Functional requirements](#)

[4.5. Device exploited features](#)

[7. Test plan](#)

[7.1. Test scenarios](#)

[7.2. Test cases](#)

[8. References](#)

1. Introduction

1.1. Context

Water is one of the most important elements on Earth, indeed water covers around 70% of the Earth's surface. World wide efforts have been made to monitor the state of sea water bodies, since the lack of awareness, industrialization and the dynamics of the planet itself are generating critical changes in it. Sea level is expected to rise alarmingly in the coming years. Fauna and flora are affected by pollution. Oil spills are killing life at sea.

Any physical, chemical, or biological change in water quality that has a harmful effect on living organisms or makes water unsuitable for desired uses. Pollutants may be biodegradable, non-biodegradable, or slowly degradable.

It is important to understand water quality issues to be aware of how we can impact, as highlighted by the UNESCO.:

... Water quality and wastewater monitoring are fundamental tools in the management of freshwater resources and they provide essential information characterizing the physical, chemical and/or biological status of water resources, determining trends and changes over time, and identifying emerging water quality issues (UNESCO-IHP International Initiative on Water Quality (IIWQ) Symposium on Scientific, 2015).

Monitoring the water bodies that surround a city will establish the current state of the sea. It is possible to determine the degree of contamination, the water level and the environmental conditions of the marine fauna and flora through a system of sensors. At the same time, this sensor system can help detect pollutants such as oil, which can help to take action quickly, reducing damage to the ecosystem.

That is how the idea of *AquApp* came about. *AquApp* is an information system for a collaborative sensor network developed at Universidad Tecnológica de Bolívar in Cartagena de Indias, Colombia. The services provided by *Aquapp* allow the collection, processing and presentation of data of a pilot program for the monitoring and follow-up of

1.2. Purpose

WaterAPProval is a mobile application that provides reliable and freely available information on physicochemical characteristics and the *Marine and Estuarine Water Quality Index for Fauna and Flora Preservation* (ICAMpff, by its initials in Spanish) of the water bodies of the city of Cartagena de Indias, Colombia. The data to be displayed is obtained from API provided by AquApp platform, which supports the water quality monitoring program of the bayous and lagoons system of the city that started in 2015.

The app plays an unquestionable role at determining the chemical properties of flora and fauna that may or may not favor the survival of permanent and temporary species in the region. The ecosystemic services, which are the many and varied benefits to humans gifted by the natural environment and from healthy ecosystems, may be also compromised. For instance, the water it plays a fundamental involvement from an industrial point of view, as it becomes one of the main requirements for an optimal functioning of all industrial activities that affect public health, this is, the proximity to a water body that supplies clean and safe drinking water continuously.

1.3. Definitions, Acronyms, Abbreviations

1.3.1. Definitions

- **Nodes:** monitoring points with several sensors collecting data from the water bodies.
- **Water bodies:** any significant accumulation of water, generally on a planet's surface.
- **Water quality indexes:** Tool used to evaluate the physicochemical characteristics of a water body according to its use.
- **ICAMpff:** The *Marine and Estuarine Water Quality Index for Fauna and Flora Preservation* was developed by the Institute of Marine and Coastal Research (INVEMAR, due to its initials in Spanish). This index is meant to preserve the marine biota. It considers seven parameters: dissolved oxygen, nitrates and nitrites, total phosphates, total suspended solids, biological oxygen demand, fecal coliforms and pH, to study the impact of domestic contamination in marine waters.

1.3.2. Acronyms

- DD: Design Document
- GUI: Graphical User Interface
- MVC: Model View Controller

1.4. Document structure

The purpose of this document is to discuss more technical aspects regarding architectural and design choices that must be made, so as to follow well-oriented implementation and testing processes.

More precisely, the document is divided in seven parts, each one devoted to approach each one of the steps required to apply requirements engineering techniques.

- Chapter 1 gives an introduction of the design document. It contains the purpose and the scope of the document, as well as some abbreviation in order to provide a better understanding of the document to the reader.
- Chapter 2 deals with the architectural design of the application.
- Chapter 3 gives all details with respect to external APIs consumed by the application.
- Chapter 4 contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.
- Chapter 5 presents the test plan.
- Chapter 6 refers to the graphical user interface.
- Chapter 7 includes the reference documents.

2. Design pattern

2.1. Overview

Flutter provides a lot of flexibility in deciding how to organize and architect applications. But, even if this freedom is valuable, it can also lead to apps with large classes, inconsistent schemes, etc. These types of issues can make testing, maintaining and extending your apps difficult.

The ideal situation is to select a design pattern that helps us to build the application and that is easier to test and maintain. Allowing the application to be clear, reusable, scalable testable, performant and maintainable.

2.2. Selected design patterns

According to the principles of clean architecture, the goal is to keep core business logic separate from the UI, database, network and third-party packages. The main reason behind this is that the core business logic doesn't change frequently, while all the others often do. Besides, the single responsibility principle states that each component in your app should do one thing only.

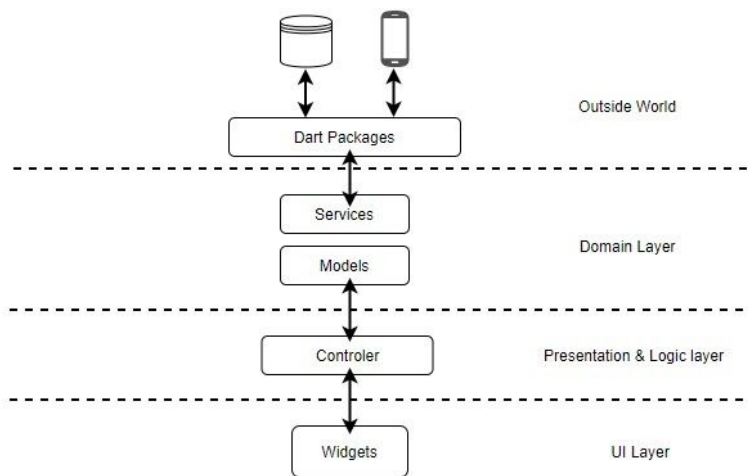
Regardless of the project we are trying to build it is important to break up our requirements into smaller and more manageable building blocks (or components), the main idea is to build the app by composing blocks together. Among the possible patterns to create scalable and maintainable apps there are MVC, MVP and MVVM.

2.2.1. Model View Controller (MVC)

The Model View Controller is a design pattern helps to build applications that are easier to test and maintain. The main goal behind it is to to separate three aspects of a software application. In this case, Model-View-Controller describes the separation of the app's Data (Model) from the app's interface (View) from the app's Logic (Controller).

- **Model:** helps to represent the real-time data or information that will be used in the Android app development. <SERVICES>
- **View:** is the interface offered to the users. This element utilizes a set of libraries to present the data more accurately and appropriately. <UI>
- **Controller:** this layer typically contains the business logic of your application, and acts as a mediator between model and view components <LOGIC>

In other words, the model is the part of the application that is responsible for the logic needed for the treatment of data. Normally model objects retrieve data (and store data) from a database. The view is the part of the application that is responsible for the visualization of the data. Usually views are created from the model data. The controller is the part of the application that is in charge of the interaction with the user. Normally, drivers read data from a view.



2.3. Other design decisions

In the following section a list of suggested development frameworks and technologies are listed. Today, with several options available for startups to develop mobile apps in 2020, the selected framework depends on the singular needs of the project, and even on the decision among considering cross-platform mobile apps or native apps.

To begin with, for the development, considering that WaterAPProval is a startup, it is suggested to consider a framework that allows cross-platform mobile development, and at the same time achieves native performance, that keeps compatibility with other technologies. Also to have minimal configuration and with a short learning curve. The selected framework was Flutter powered by Google, the stable version v1.17.5.

Flutter has the following advantages over other cross-platform mobile app development frameworks.

- **Fast development cycle:** with stateful hot reload, you can repaint the app live without building it again.
- **Single code base:** use a single code base to build applications in iOS and Android.
- **Flexible UIs and widgets:** a wide range of widgets and tools are available for building flexible apps.
- **Native performance:** The Flutter framework provides complete native support, even for scrolling, navigation, icons, and fonts.
- **Open source:** Flutter is open-source and available to use or study at free of cost.

Furthermore, it is suggested to use Google Maps platform for displaying the map with the water bodies and nodes considering the compatibility with Google products. The main recommendation is to choose official libs due to guaranteed maintenance.

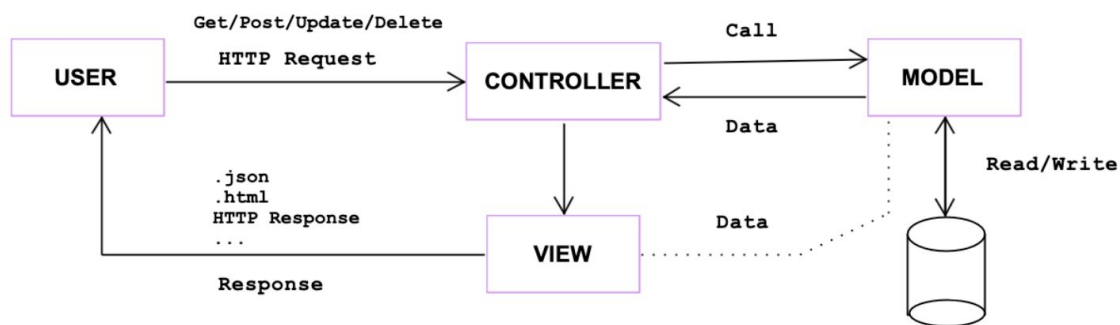
With respect to the data presentation in a visual chart format, Apache Echarts library was selected. Apache ECharts (incubating)™ is an incubation project at The Apache Software Foundation (ASF).

Finally, the only external APIs to whom WaterAPP environment depends on, are the services provided by AquApp which is very stable, Google Maps and Echarts. All of them do not depend on any external services.

3. External web-service APIs

Fetching data from the internet is necessary for most apps. In the case of WaterApproval, there are 3 consumed APIs. In order to interact with external services provided by the APIs, the flutter *Future-based library for making HTTP requests* was used.

As said before, the communication between WaterAPProval and their users is done via HTTP requests following REST principles. REST (Representation State Transfer) is an architectural style for communication based on strict use of HTTP request types.



The RESTful HTTP requests are categorized according to method types as the following:

- **GET:** used to retrieve resource representation/information only - and not to modify it in any way.
- **POST:** used to create a new resource into the collection of resources.
- **PUT:** used primarily to update existing resource (if the resource does not exist then API may decide to create a new resource or not).
- **DELETE:** used to delete resources (identified by the Request-URI).

3.1. Aquapp API 2.0: Online REST API for Water Quality Monitoring

- Aquapp 2.0 is an online public REST API to carry out the monitoring and subsequent follow-up of some physical-chemical and biological parameters of the water bodies of Cartagena de Indias.
- API official URL: <https://aquapp.utb.edu.co/api/v1/>
- *No authentication required.*

3.1.1. Useful endpoints

Nodes and Water bodies:

Get an array of Element model instances. Elements can be water bodies or nodes depending on **form** attribute value by means of endpoint:

```
GET /elements/open/jsonata
```

To get nodes the request parameters are

Parameters:

```
query: ([ $ ])
```

```
additionalFilters:
```

```
[{"category": "tracked-objects"}, {"form": "5dacb798a52adb394573ca70"}]
```

To get water bodies the request parameters are

Parameters:

```
query: ([ $ ])
```

```
additionalFilters:
```

```
[{"category": "tracked-objects"}, {"form": "5dac93e7e67d5a13c95a99ed"}]
```

Data on water bodies

Get data records by means of endpoint:

```
GET /data/open/vm2
```

Parameters:

```
query=this.data
```

To Filter data records by a specific set of nodes [node_IDs], the `additionalFilters` parameter is required.

Parameters:

```
query=this.data
```

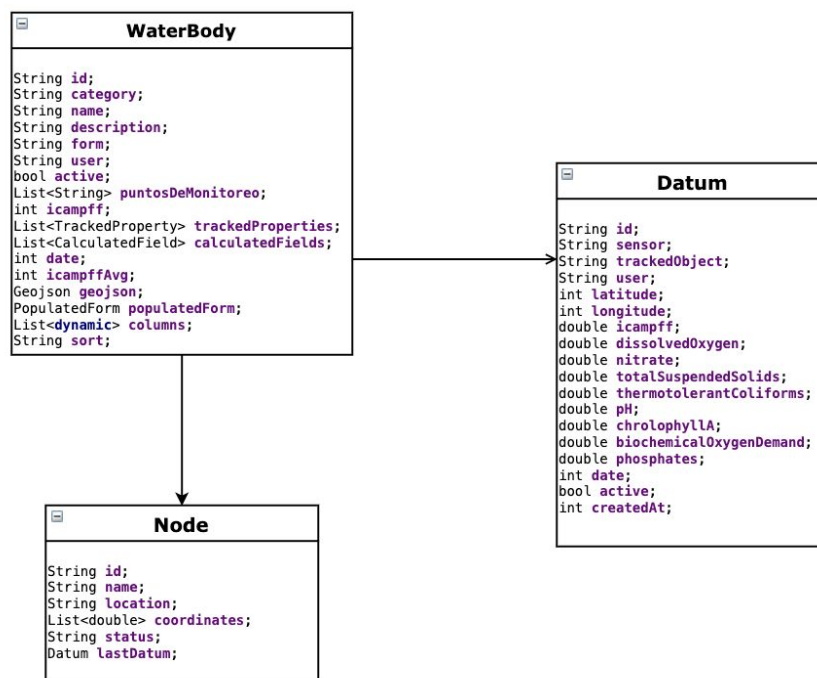
```
additionalFilters:
```

```
[{"trackedObject": {"inq": [node_IDs]}}]
```

3.1.2. Local data model

To be able to manipulate the JSON objects obtained from AquApp API it was necessary to have a local data model. Both water bodies and nodes are considered elements in the Aquapp API, but locally are treated separately.

- **WaterBody**: main attributes of the water bodies in the system.
- **Node**: main attributes of the nodes (monitoring points) in the system. The possible status for a node are UNKNOWN, OFF, REAL TIME and NON REAL TIME.
- **Datum**: physicochemical characteristics measured from the water bodies.



3.2. Google Maps API (v2)

To add the map based on Google Maps to the application it was required to use the Google Maps SDK for Android (and iOS). Mainly, it is required to *generate an API Key* to make requests and use the *google_maps_flutter* plugin.

3.3. Invemar ICAM web service

- It is an online free service provided by Invemar to compute the ICAMPff index according to the request parameters.

- API official URL: <http://buritaca.invemmar.org.co/ICAMWebService>
- *No authentication required*

Data on water bodies

Get data records by means of endpoint:

GET /calculate-icam-ae

Parameters:

```
od => dissolvedOxygen
no3 => nitrate
sst => totalSuspendedSolids
cct => thermotolerantColiforms
ph => pH
po4 => phosphates
dbo => biochemicalOxygenDemand
cla => chlorophyllA
```

The response structure of the endpoint is a simple JSON is as follows:

```
{
  "calification": "INADECUADA",
  "value": 47.09,
  "acounterParameter": 8,
  "trustfulness": 100,
  "procedimiento": "sin error"
}
```

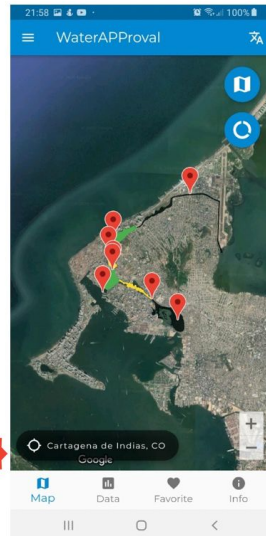
4. User characteristics

- WaterAPProval is freely available for all users, anyone willing to access the application. Mainly, anyone can access all the functionalities of the app without the need of registering and/or signing. Are mainly anonymous users. Therefore, no constraints derived from the characteristics need to be satisfied to be able to use the application.

5. User interface design



1

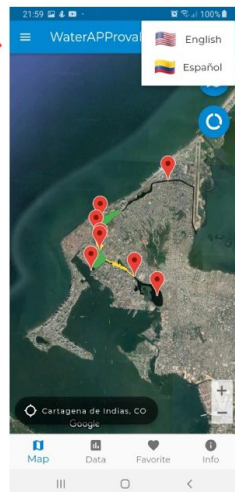


2

3

The home page of the application is the map view with ICAMpff on the water bodies and markers on the nodes. There are several functionalities that are clearly depicted on it.

2

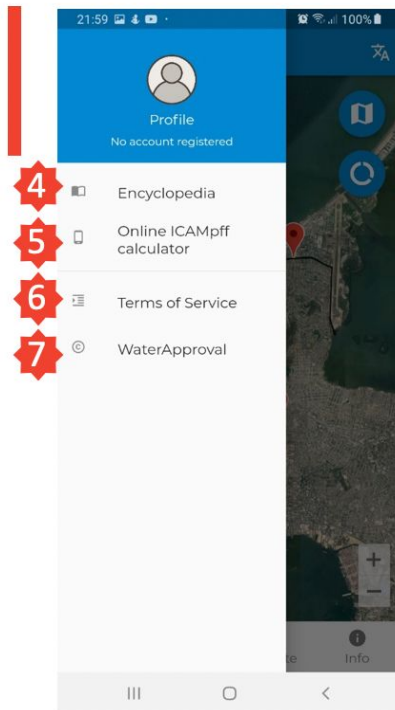


3

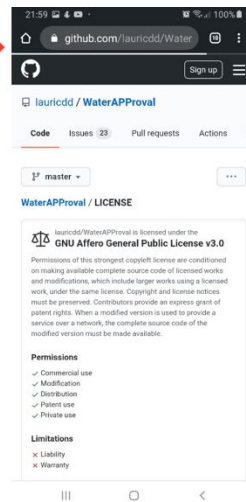
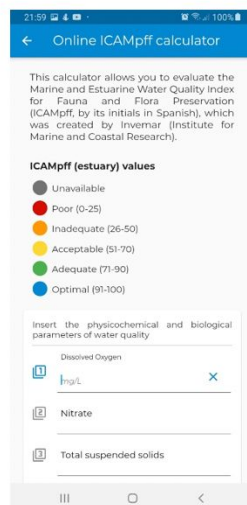
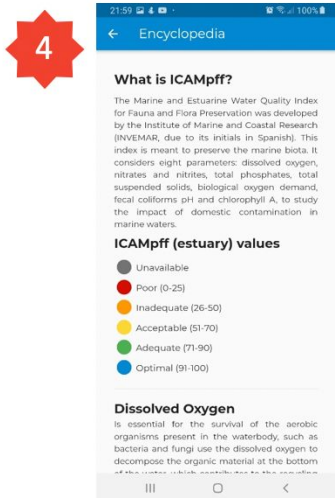
From point (1), a user can read about the city of Cartagena de Indias; from point (2), can see the multi-language feature; and from point (3), users are able to check the ICAMpff values convention.

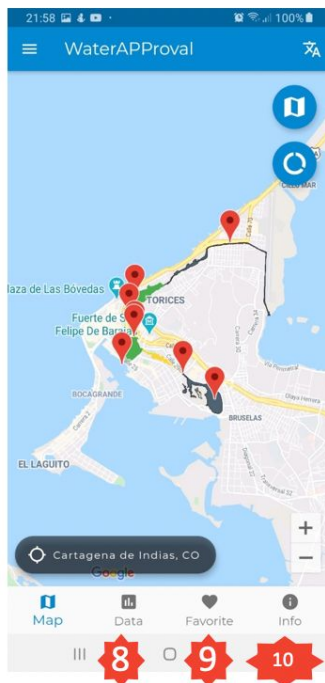


1

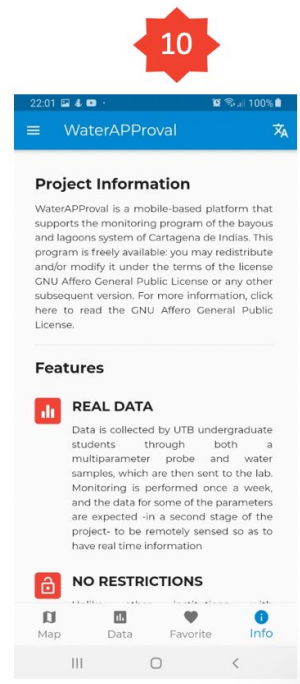
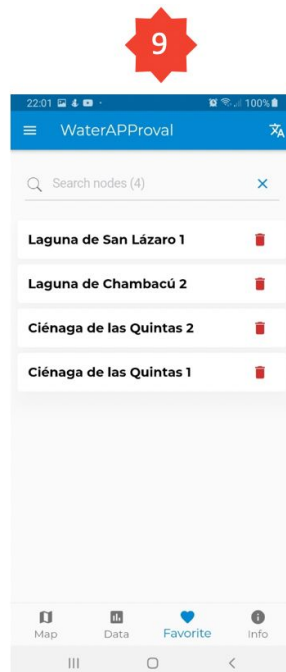
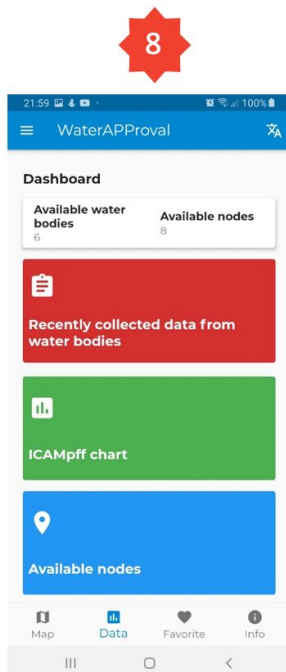


Now, by displaying the drawer new functionalities are available. From point (4), the user has access to the encyclopedia; from point (5) the online ICAMpff calculator form; from (6) the application LICENSE; and from (7) the application versioning info.



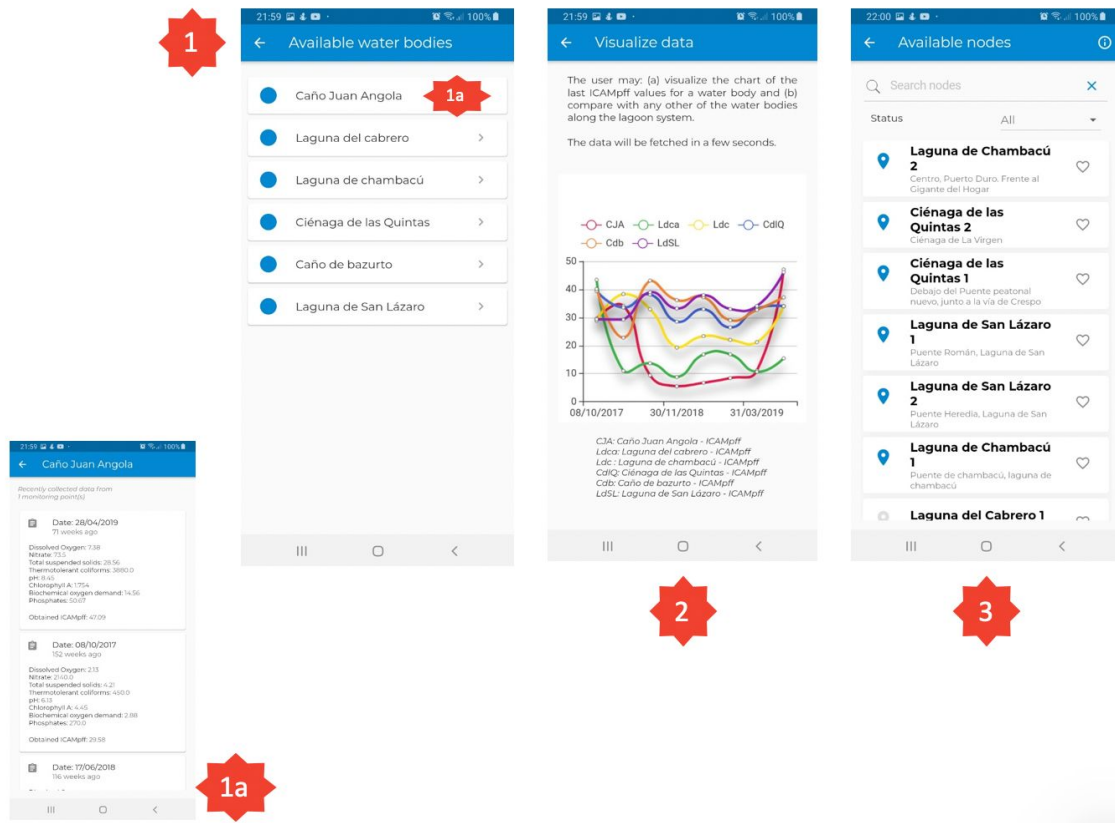
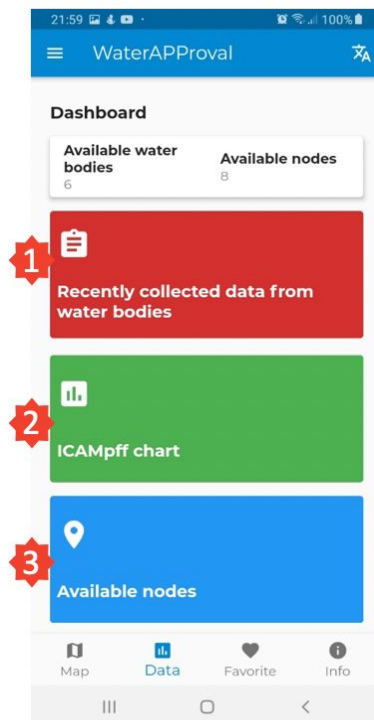


By selecting the data tab (8) it is possible to access the dashboard in which the user can have access to the different data visualization options; tapping on (9) the user can have access to the favorites list and on (10) there is the project explanation.



The dashboard contains most of the core functionalities of the application since it provides the data by means of several visualization formats:

from (1) and (3) the user can check the raw data, mainly recently collected data by water body and by node ; and from (2) <chart> the user can visualize a chart with the last computed ICAMpff values.



6. Requirements

4.1. Functional requirements

The functional requirements are those which are the fundamental actions of the system. This section shows how the goals specified during the requirements collection:

1. The application must allow a user to visualize two types of maps (SATELLITE and ROADMAP) of the city of Cartagena de Indias (CO) in real time.
 - a. The map must contain each of the nodes located along the water bodies system.
 - b. The map must contain a polygon on each of the water bodies and the color should correspond to the value of the last ICAMpff value computed for the water body.
2. The application must allow the to user to tap on the markers of the map to get node information.
3. The application must allow a user to check information of any nodes, including the current status (UNKNOWN, OFF, REAL TIME and NON REAL TIME) and the coordinates. Also must be able to filter the list by node name and status.
4. The application must allow a user to obtain the data collected from each node in two ways: as raw data grouped by dates and water body or as a chart.
5. The application must allow a user to compare the ICAMpff value obtained from different water bodies for research purposes.
6. The application must allow a user to compute online the ICAMpff index providing the corresponding parameters by means of an online calculator.
7. The application must allow a user to check the meaning of each measured parameter by searching in the encyclopedia.
8. The application must allow the user to choose the language of interest among spanish and english.



9. The application must allow the user to have a custom favorite list containing all the nodes of interest for future research. Also must be able to filter the list by node name.

4.5. Device exploited features

- **Internationalizing feature:** allow to have several locale options for the user to select from.
- **Data persistence:** cache data locally using SharedPreferences and SQLite database.
- **Gesture detector:** allow widgets to detect features and respond to them. For instance to hide the keyboard after editing text.
- **Keyboard type:** determine the type of keyboard to use for editing the text.
- **URL launcher:** allow launching a URL in the mobile platform.
- **Fetch data from external services:** allow fetching data from the internet and use that data inside the application.
- **Device orientation:** allow to update the UI based on orientation (portrait or landscape). Prevent device orientation changes while rotating.
- **Data visualization widgets:** update the charts reactively when data change.
- **Internet Connectivity:** check whether there is an Internet connection available on the app and behave accordingly.

7. Test plan

First of all, the external services and their APIs that are going to be used in the application should be available and ready. This applies to the already mentioned services, to the DBMS and the server on which it will be running on.

7.1. Test scenarios

In the following section a list of test scenarios is described. The test scenarios were categorized by its importance, based on how critical they are for the system.

ID	Description	Importance
TC01	Validate data fetching from external APIs	High
TC02	Validate if the user is able to visualize the map	High
TC03	Validate if the user is able to visualize the data on the dashboard	High
TC04	Validate icampff computation done by the calculator	Medium
TC05	Validate multi-language feature	Medium
TC06	Validate data persistence of favorite nodes and selected language	Medium
TC07	Validate if the user is able to tap on the markers of the map to get node information.	Low

7.2. Test cases

The tests cases were done manually, following the steps described below and based on the importance described in the Test Scenarios section. Not all test cases were executed, but all of them *completed with a positive output*.

8. References

- INVEMAR. (2014) Protocolo Indicador Calidad Ambiental de Agua ICAMpff. Retrieved from: <http://www.invemar.org.co/documents/10182/14479/04+ProtocoloIndicadorCalidaddeAguadigital.pdf/c16bd915-0b24-446b-9fe0-dbbc239111dd>
- IEEE. (2016) AquApp, an information system for a collaborative sensor network
- UNESCO. (2015) Innovations for Improved water quality monitoring in the SDGs Framework. International Scientific Symposium on Scientific, Technological and Policy, Kyoto. Retrieved from: <https://en.unesco.org/waterquality-IIWQ/activities-projects/monitoring-water-quality-wastewater>
- IEEE. (1993). IEEE Recommended Practice for Software Requirements, Specifications (IEEE 830-1993). Retrieved from: <https://standards.ieee.org/standard/830-1993.html>
- Flutter framework. A cross-platform UI toolkit conceived by Google. Retrieved from: <https://flutter.dev/docs>
- List of state management approaches. Retrieved from: <https://flutter.dev/docs/development/data-and-backend/state-mgmt/options>
- Dart general purpose programming language official documentation. Retrieved from: <https://dart.dev/>
- Google Maps Platform - <https://cloud.google.com/maps-platform>
- Repository: the source code has been published on GitHub. It can be accessed using <https://github.com/lauricdd/WaterAPPProval>.