Software Requirements of TidySantCugat phase I

This document aims to describe the required software and system architecture to support this project.

Backend server

The backend server have to serve the different apps. The functionalities are:

- Ingest georeferences from different sources:
  - Garbage collection database
  - Retail customers
- Compute the closest garbage containers given a location
- Storage and serve the notifications from users about the status of containers
- Provide a CRUD data access for entities behind these functionalities

Components:

- PostgreSQL database with postGIS to store georeferences and compute distances
- HTTP restfull API server with OAuth2
- Batch process to pull data from Garbage collection service provider's database
- Depending on the volume of notifications per minute a queue like Kafka/RabbitMQ would be required

Technologies suggested:

- Java Spring web-mvc to develop the backend API application
- Java Spring batch to develop the batch process

Mobile app for retail customers

This application is intended to provide these functionalities:

- User login, registry, user profile and profile update
- Storage of user favorite places, that is places of interest
- Show closest garbage containers to
  - a favorite place
  - the current location of the user
- Show the alarms of closest garbage containers to:
  - a favorite place
  - the current location of the user
- Let users push notifications about issues of containers (broken, fully, …)
- Notify user when the status of a container near a favorite place is updated (deactivatable)
- Map of status of containers with CartoDB

The prototype has been developed with ionic, hence is available for ios and android. The application should be available at least for these two platforms, but could be possible for any with geolocation sensors available. All these functionalities are supported by the backend API and those that are notification-like (user is notified about an external event) should pull the data from the same API continuously with some interval for simplicity.

Web app for administration

This application is intended to provide these functionalities:

- User login, registry, user profile and profile update
- Administration of all users
- Visualization of garbage collectors routes in a time-lapse for some time-window (yesterday, last week, …) via CartoDB maps
- Visualization of current status of containers in the zone via CartoDB maps
- List of issues in routes and schedules
- List of issues notified by users

Public web for data trasparency

This web is intended to provide these functionalities:

- Visualization of garbage collectors routes in a time-lapse for some time-window (yesterday, last week, …) via CartoDB maps
- Visualization of current status of containers in the zone via CartoDB maps
- Access to raw data.