

Universidad Siglo 21



Trabajo Final de Graduación

Licenciatura en Informática

Proyecto de Aplicación Profesional (PAP)

Optimización de recolección de residuos con IoT

Alejandro Ricatti

VINF01892

Tabla de contenidos

Tabla de ilustraciones.....	4
Optimización de Recolección de Residuos con IoT.....	6
Resumen.....	6
Abstract.....	6
Introducción - Marco de referencia institucional.....	7
Antecedentes.....	7
Descripción del problema.....	7
Justificación.....	8
Objetivo general del proyecto.....	8
Objetivos específicos del proyecto.....	8
Objetivos específicos del sistema.....	9
Límite.....	9
Alcance.....	9
No Contempla.....	9
Marco Teórico.....	10
Marco teórico del negocio.....	10
Actividad del cliente.....	10
Situación actual del mercado.....	10
Tamaño de la industria de residuos.....	10
Tipos de contenedores y basureros.....	11
Residuos sólidos municipales.....	11
Componentes de la administración de residuos municipales.....	12
Inconvenientes con los métodos de recolección actuales.....	12
Competencia.....	12
Marco teórico del Sistema.....	13

IoT (Internet of Things).....	13
Node.js.....	14
V8.....	15
MongoDB.....	15
Web Sockets.....	16
Diseño Metodológico.....	16
Planificación del proyecto.....	17
Procesos de Negocio.....	17
Metodología de desarrollo.....	17
Historias de usuario.....	18
Épicas.....	18
Sprints.....	19
Story Points.....	19
Burn Down Charts.....	19
Relative Mass Valuation.....	20
Control de Versiones.....	20
Modelado.....	20
Programación.....	20
Implementación.....	21
Relevamiento.....	21
Relevamiento Estructural.....	21
Relevamiento Funcional.....	22
Organigrama.....	22
Funciones de las Áreas.....	23
Procesos de negocios.....	23
Diagnóstico.....	25
Proceso: Asignación de rutas.....	25

Proceso: Generación de rutas.....	26
Propuesta.....	26
Sensores.....	26
Aplicación Web.....	26
Aplicación Móvil.....	27
Listado de Requerimientos funcionales.....	27
Listado de Requerimientos no funcionales.....	27
Listado de Requerimientos Candidatos.....	28
Diagrama de Gantt.....	28
Desarrollo del Producto/servicio.....	29
Análisis y Diseño.....	29
Entidades.....	29
Colecciones.....	29
Planificación.....	30
Estimaciones.....	30
Backlog.....	30
Épica: Desarrollo del sensor.....	30
Épica: Desarrollo de la plataforma web.....	31
Épica: Desarrollo de la aplicación móvil.....	32
Sprints.....	33
Sprint 1.....	33
Sprint 2.....	34
Sprint 3.....	35
Sprint 4.....	36
Sprint 5.....	38
Tickets finalizados por sprint.....	39
Riesgos del proyecto.....	39

Matriz FODA.....	40
Plan de negocios.....	40
Interfaces de usuario.....	41
Selección de componentes de hardware para el sensor.....	45
Sensor ultrasónico.....	46
Energía.....	47
Microcontrolador.....	48
Selección de tecnología para la aplicación web y API.....	48
Selección de tecnología para la aplicación móvil.....	49
Selección de plataforma.....	49
Roles.....	49
Presupuesto.....	50
Costos individuales de componentes utilizados en los prototipos.....	50
Entrevista.....	51
Conclusiones.....	53
Bibliografía.....	54
Anexo I.....	56
Modelos de Entrevista.....	56
Para empresas de recolección:.....	56
Para entes gubernamentales:.....	56

Tabla de ilustraciones

Ilustración 1: Ruta de recolección sin optimizar.....	23
Ilustración 2: Estructura organizacional de la sección de Waste management del Dublin City Council.....	24
Ilustración 3: Proceso de asignación de rutas.....	25
Ilustración 5: Diagrama de Gantt.....	29
Ilustración 6: Modelos de entidades.....	30
Ilustración 7: Burndown chart Sprint 1.....	35

Ilustración 8: Burndown chart Sprint 2.....	36
Ilustración 9: Burndown chart Sprint 3.....	37
Ilustración 10: Burndown chart Sprint 4.....	38
Ilustración 11: Burndown chart Sprint 5.....	39
Ilustración 12: Tickets finalizados por sprint.....	40
Ilustración 13: Login.....	43
Ilustración 14: Vista general (mapa).....	43
Ilustración 15: Vista de basurero (reporte de basurero).....	44
Ilustración 16: Diálogo Agregar/Editar basureros.....	45
Ilustración 17: Login y Mapa – App Móvil.....	46

Optimización de Recolección de Residuos con IoT

Resumen

El objetivo de este proyecto es determinar la viabilidad de optimizar la recolección de basura de contenedores de la vía pública utilizando Internet of Things.

La recolección de residuos de los cestos de basura en la vía pública se realiza de una forma poco eficiente hoy en día, la causa principal es la falta de eficiencia derivada de la poca información con la que se cuenta en el proceso, por eso, el foco de este trabajo es implementar un prototipo de sensores de capacidad (llenado) conectados que permitirán optimizar las rutas de recolección y la planificación de instalación de nuevos basureros.

Partiendo de la especificación del problema y su justificación, este trabajo propone una solución y documenta la implementación de un prototipo de sensor conectado para determinar la viabilidad de la utilización de IoT en la recolección de residuos. Todas las tecnologías y metodologías utilizadas son establecidas y descriptas para proveer una perspectiva completa del proyecto.

Abstract

The objective of the following project is to establish how feasible it is to optimize trash collection in city bins with the help of the Internet of Things.

Waste collection in public bins is done in a very inefficient way nowadays, the main cause is the lack of information available for the process. This is why the focus of the project is to implement a prototype of connected fill sensors that will allow for collection route optimizations and better planning.

Parting from the problem statement and it's justification, this work proposes a solution and documents the implementation of a prototype to determine the feasibility of using IoT for trash collection. All the technologies and methodologies used are established and described in order to provide a full perspective of the project.

Introducción - Marco de referencia institucional

El proyecto elegido para la aplicación profesional apunta a resolver problemas en la recolección de residuos de basureros de la vía pública mediante la utilización de nuevas tecnologías.

Actualmente las ciudades de todo el mundo enfrentan grandes desafíos a la hora de organizar la recolección de residuos, esto se debe a que deben planificarse rutas a recorrer y se necesita de mucha mano de obra. La falta de información sobre las condiciones de los basureros públicos hace que estas rutas sean muchas veces recorridas sin necesidad y que se descuiden lugares donde se necesita más recolección.

Se tomará como caso de estudio al City Council de Dublin, en Irlanda.

Antecedentes

La recolección de residuos de la vía pública se realiza de una manera ineficiente en la actualidad. Se utilizan rutas predeterminadas y estáticas, lo que implica muchas veces recorrer zonas en las que no hay necesidad de realizar recolecciones. Esto es obviamente un problema mayor, puesto que se desperdician recursos como combustible y mano de obra.

A esto se puede sumar que el crecimiento de los centros urbanos en muchos casos excede la capacidad de adaptación de los gobiernos locales, causando grandes problemas logísticos y de falta de personal.

Descripción del problema

Para obtener esta información se entrevistó personalmente a Brian Hanney, director del área de residuos de Dublin City Council, en la República de Irlanda.

Los siguientes puntos pueden extraerse de la situación actual:

- Costos de operación en incremento por el crecimiento de las ciudades y una operatoria poco eficiente: La forma actual de recolectar basura no ha cambiado en mucho tiempo, y la forma de realizar las tareas ya no escala a ciudades de gran tamaño
- Excesiva cantidad de mano de obra para tareas rutinarias y repetitivas: Este es probablemente el mayor problema. El costo de mantener un gran número de operarios y la dificultad para coordinarlos representan los mayores problemas que tiene el City Council hoy en día

-Desperdicio de combustible y horas hombre debido a la manera estática en la que las rutas están planeadas: Las rutas de recolección cambian con muy poca frecuencia, las ciudades sin embargo tienen actividades y eventos diferentes todas las semanas del año, lo que lleva a que muchas veces se envíen camiones de recolección a lugares donde no ha habido mucha actividad y se descuiden otros lugares, que deberían ser atendidos con mayor frecuencia.

El problema entonces se puede resumir en la siguiente pregunta: ¿Cómo puede aprovecharse la infraestructura de comunicaciones en las ciudades para optimizar el proceso de recolección de residuos mediante una solución del tipo Internet of Things?

Justificación

Según Brian Hanney, la ciudad cuenta con 500 empleados que realizan las tareas de recolección diariamente, este número ha disminuido constantemente desde 2008, cuando el total era de 800 a causa de la crisis económica en la que se vio afectada Europa (Eichengreen, 2015). Sin embargo las áreas de la ciudad cubiertas por los servicios son las mismas y se han instalado más basureros, por lo que el trabajo ha aumentado y el personal ha disminuido por cuestiones presupuestarias.

Es necesario incorporar tecnología para disminuir la longitud de las rutas de recolección y aportar nueva información que permita planificar mejor las operaciones. La alternativa es simplemente contratar mayor cantidad de personal, pero esto tiene un impacto masivo en el presupuesto.

Objetivo general del proyecto

Desarrollar un sistema web que permita visualizar rutas de recolección de basura optimizadas y reportes de contenedores individuales con información obtenida mediante sensores conectados.

Objetivos específicos del proyecto

Validar la factibilidad y necesidad de la utilización de sensores y rutas dinámicas para mejorar el proceso de recolección.

Desarrollar prototipos de las aplicaciones web y móvil que permitan el trazado de rutas y la visualización de los datos recolectados por los sensores.

Desarrollar un prototipo del sensor WiFi.

Objetivos específicos del sistema

El objetivo del sistema es planificar las rutas de recolección de residuos de basureros en la vía pública mediante el uso de sensores y aplicaciones web y móviles. El objetivo secundario es proveer información necesaria para la toma de decisiones en el proceso de decisión de instalación de nuevos basureros.

Límite

El sistema abarca desde la provisión de información del estado de los basureros hasta la producción de rutas optimizadas, datos estadísticos y reportes sobre el uso y recolección de residuos en los basureros públicos.

En cuanto al componente de hardware, los sensores, se abarca todo el proceso desde el diseño y prototipado.

Alcance

Los procesos a ser relevados son el de generación de rutas y el de asignación de rutas.

No Contempla

La solución propuesta no contempla la utilización de la plataforma y sensores para residuos domiciliarios, ya que la recolección de estos sigue un proceso completamente diferente.

Los sensores no funcionarán en condiciones extremas, como ser temperaturas bajo cero, demasiado elevadas, inundaciones, y otras situaciones climáticas fuera de lo normal.

Marco Teórico

Marco teórico del negocio

Actividad del cliente

El cliente se dedica a la planificación, instalación, mantenimiento y servicio de los basureros públicos en la ciudad de Dublin, República de Irlanda. Es un ente público que depende directamente del gobierno local.

Situación actual del mercado

La demanda de soluciones inteligentes para el área de residuos viene incrementando a una tasa anual del 43% (ABI Research, 2014).

Ciudades como Londres están invirtiendo en basureros inteligentes y gastando hasta US\$45.000 por basurero (Independent, 2013), lo que indica que existe mucho lugar para nuevas soluciones y también que alternativas económicas serán bienvenidas. Asimismo los problemas de recolección de basura son globales, afectando tanto al sector público como al privado.

Existen ciudades como Helsinki en Finlandia, y Universidades como Tufts en los Estados Unidos que han implementado soluciones similares a la propuesta en este trabajo, ahorrando hasta un 45% en costos de recolección (Tufts University & Save That Stuff, 2014).

Tamaño de la industria de residuos

Si se toman en cuenta todas las actividades que involucra la recolección y el procesamiento de residuos, se está hablando de una industria de 410.000 millones de dólares al año globalmente (First Research, 2016).

En este sector, solo en Estados Unidos se encuentran más de veinticuatro mil empresas, de las cuales las principales cincuenta representan más de la mitad de los ingresos anuales de la industria.

La actividad que más dinero genera es la de recolección, que se lleva el 55% del total, seguida por el tratamiento y volcado (20%), y tareas de limpieza de materiales tóxicos o contaminantes (15%). La mayor parte del negocio involucra el la recolección y volcado de residuos domiciliarios y municipales no peligrosos. (First Research, 2016)

El total de toneladas de residuos recolectados anualmente es de 11.000 millones a nivel global.

Tipos de contenedores y basureros

En las ciudades de hoy en día se utilizan múltiples tipos de basureros, muchas veces diseñados particularmente para cada ciudad, y esto presenta un desafío a la hora de crear un sensor que pueda ser instalado en cualquiera de ellos. También se utilizan contenedores grandes para sectores en los que la recolección se realiza con menor frecuencia.

La capacidad promedio de un basurero va entre los 150 y 250 litros. La de los contenedores en cambio suele ser de 7.600 a 30.000 litros (ex-cell Kaiser, 2017).

En cuanto a los materiales, los basureros pueden ser de metal o plásticos, ambos afectan en mayor o menor medida la transmisión y recepción de datos de los sensores que se puedan instalar en su interior.

Residuos sólidos municipales

Los residuos que se recolectan de la vía pública se conocen como Municipal solid waste, o residuos sólidos municipales. Su composición varía según el país, la época, y múltiples otros factores. Estos residuos no incluyen desechos industriales, de agricultura, o médicos. La siguiente es una clasificación típica:

Biodegradables: Comida, papel, desechos verdes (pasto, ramas de plantas, desechos de jardinería)

Materiales reciclables: Papel, cartón, vidrio (botellas, frascos, ...), latas de diferentes metales, plásticos, telas, neumáticos, etc.

Materiales inertes: Residuos de construcción y demolición, tierra, piedras y rocas.

Desechos compuestos: Tetra packs, ropas de acrílicos y telas mezcladas, juguetes.

Desechos peligrosos: Pinturas, químicos, neumáticos, baterías, focos, lamparas fluorescentes, aerosoles, fertilizantes

Desechos eléctricos y electrónicos: productos electrónicos, lavadoras, heladeras, televisiones, computadoras, teléfonos.

Desechos tóxicos: Herbicidas, pesticidas, fungicidas.

Desechos biomédicos: Restos humanos resultantes de amputaciones y operaciones, medicamentos vencidos.

Componentes de la administración de residuos municipales

La industria de residuos municipales tiene 4 componentes: reciclado, composting (procesamiento de materiales orgánicos para reutilización), desechado, y conversión de residuos en energía a través de incineración.

Los procesos son los siguientes:

Recolección

Procesamiento y separación: Puede ser en la fuente, por medio de basureros exclusivos para cada tipo de residuos, o en una planta de procesamiento.

Transporte

Desechado/Reciclado/Composting

Generación de energía

Inconvenientes con los métodos de recolección actuales

Las ciudades crecen constantemente, por lo que la cantidad de residuos que deben ser recolectados también, esto hace que planificar y ejecutar las tareas de recolección sea un desafío muy grande. Las tareas de recolección consumen 55% del presupuesto total de manejo de residuos de las ciudades, y en países del tercer mundo, esta cifra puede llegar hasta el 90% (Surya Prakash Chandak, 2010).

Competencia

Existen varios competidores a nivel global, pero los principales son los siguientes:

Big Belly Solar: Provee basureros enteros, equipados con paneles solares y compactadores de basura. Su principal fuerte es que los compactadores reducen la necesidad de recolectar la basura frecuentemente, su mayor desventaja está en el costo.

Enevo: Es una empresa que provee sensores similares, la mayor diferencia es el modelo de negocios, en vez de vender los sensores, ellos cobran una tarifa mensual.

SmartBin y Farsite Communications: Proveen sensores similares, pero su foco es la industria alimenticia (su mayor base de instalación es en contenedores de aceite y basureros de gran porte fuera de restaurantes).

Competidor	Costo	Modelo de negocio	Industria de foco
Big Belly	Muy Alto	Venta del dispositivo + Costo mensual de acceso a la plataforma	Residuos
Enevo	Alto	Costo mensual por acceso a la plataforma, largos contratos	Residuos
Smart Bin & FarSite	Alto	Venta del dispositivo + Costo mensual de acceso a la plataforma	Industria Alimenticia
Bintelligence	-	Venta del dispositivo + Plataforma gratuita + Costo por servicios adicionales	Sensores de propósito general, con foco en residuos inicialmente

Marco teórico del Sistema

Para poder comprender mejor las elecciones realizadas en cuanto a las tecnologías a utilizar en el sistema, se exponen los conceptos necesarios.

Para implementar el prototipo, se necesita tener conocimiento tanto de componentes de hardware, como de herramientas de software y tecnologías IT.

IoT (Internet of Things)

Con el avance de internet en velocidad y expansión geográfica, cada vez más dispositivos están conectados a la red. Internet of Things se basa en conectar distintos objetos a internet para obtener información y también para proveer funcionalidad adicional. La capacidad de obtener información y controlar una gran cantidad de dispositivos de manera remota brinda enormes ventajas en todos los aspectos, desde el productivo, con la optimización de procesos, la salud, con dispositivos que permiten hacer seguimientos muy precisos de los indicadores de salud de las personas, confort, con la posibilidad de controlar dispositivos de una casa de manera remota, y muchas otras utilidades.

El concepto de dispositivos conectados se discutió por primera vez en 1982 (Wikipedia, 2017), cuando se modificó una dispensadora de Coca Cola para permitirle reportar su inventario y la temperatura de los refrescos de manera remota. Si bien en los 90 se realizaron múltiples investigaciones académicas sobre el concepto, no fue hasta 1999 que el campo tomó popularidad, con un framework D2D (Device to Device) diseñado por Bill Joy, y avances en la tecnología RFID hechos en MIT por Kevin Ashton, quien consecuentemente fue el primero en utilizar el término Internet of Things.

Se estima que para el 2020 existirán entre veinte mil millones y treinta mil millones de dispositivos conectados (Gartner, 2015; ABI Research, 2013). La percepción general es que este avance es en su mayoría un gran beneficio para la sociedad.

Algunas de las aplicaciones que se le pueden dar a IoT son las siguientes:

Medios de comunicación: La utilización de dispositivos que permitan conocer a los consumidores pueden ayudar a optimizar los esfuerzos de marketing e información. La captura de datos también puede ser utilizada para informar con mayor certeza sobre acontecimientos.

Monitoreo ambiental: Controlar el estado de fuentes de agua, suelos, o ecosistemas completos a través de dispositivos conectados puede servir para prevenir o alertar sobre desastres naturales, alteraciones en el ambiente o diversos problemas.

Infraestructura: El monitoreo de rutas, puentes, basureros, plantas de energía, etc., permite optimizar costos, prevenir desperfectos o averías, y mejorar la calidad de infraestructura en general.

Manufactura: El control de procesos y maquinarias a través de dispositivos conectados permite la reducción de costos y el incremento en la productividad.

Salud y medicina: El monitoreo de pacientes permite obtener información con mayor frecuencia y detectar problemas potenciales. También pueden implementarse sistemas de notificación de emergencias para ayudar a prevenir accidentes y también reducir el impacto de los mismos.

Domótica: Significa automatización de viviendas. Conectar diferentes dispositivos, como luces, termostatos, sistemas de riego y otros, permite reducir el consumo de energía, el control remoto de instalaciones, y provee información útil para administrar edificios.

Transporte: Los dispositivos conectados pueden utilizarse para recolectar y distribuir información de diferentes medios de transporte, como rutas, vías, y aeropuertos.

Node.js

Es un entorno de ejecución open source y multiplataforma, que utiliza Javascript como lenguaje principal y se ejecuta sobre el motor de Javascript V8, desarrollado por Google.

La principal ventaja de este lenguaje es su modelo dirigido por eventos, que le permite hacer uso de I/O Asíncrono. Esto incrementa considerablemente la escalabilidad de aplicaciones web que reciben elevados números de peticiones por segundo, algo fundamental en el caso de soluciones IoT donde se tienen miles de sensores enviando información con mucha frecuencia.

La forma en la que Node.js consigue esto, es a través de un modelo de un solo hilo y llamadas de entrada y salida no bloqueantes, que permite manejar miles de conexiones concurrentes sin tener que realizar cambios de contexto (Nodejs.org, 2017).

Para el manejo de paquetes (instalación y administración de bibliotecas y utilidades), se utiliza Node Package Manager, o npm. Esta es una herramienta de suma utilidad para mantener el proyecto con sus

dependencias actualizadas y disponibles desde cualquier lugar.

Node.js fue creado por Ryan Dahl en 2009.

V8

El motor de ejecución utilizado por Node.js es V8, que fue desarrollado por Google, inicialmente para ser parte del navegador Chrome. Es uno de los motores más rápidos de la actualidad gracias a que compila Javascript a código nativo en vez de interpretarlo (Google, 2017).

MongoDB

Es un motor de bases de datos orientadas a documentos. Cabe destacar que es un modelo no relacional de bases de datos (NoSQL). Esto quiere decir que en vez de trabajar con tablas y relaciones, se trabaja con documentos en formato JSON (javascript object notation), que permiten mayor flexibilidad, ya que no hace falta definir las estructuras de datos a priori. Debido a esto, la integración con aplicaciones desarrolladas en lenguajes orientados a objetos se vuelve mucho más simple y directa.

Principales características:

Consultas Ad-hoc: Soporta búsquedas por campos, rangos, y expresiones regulares. Los valores de retorno pueden ser documentos, o campos específicos de los resultados.

Indexación: Cualquiera de los campos de un documento puede ser indexado para permitir consultas más rápidas.

Replicación: En escenarios donde se requiere alta disponibilidad, la replicación permite mantener dos copias o más de los datos. Cuando el set primario falla, se puede utilizar una de sus réplicas para evitar problemas de disponibilidad.

Balance de cargas: El concepto de balance de cargas se refiere a la distribución del tráfico y las consultas en múltiples servidores para permitir un mejor tiempo de respuesta. En el caso de MongoDB, esto se logra con una técnica llamada *sharding* (particionamiento horizontal), que consiste en dividir los sets (equivalentes a tablas en modelos relacionales) en diferentes servidores.

Almacenamiento de archivos: Al ser una base de datos NoSQL, existe una gran flexibilidad respecto de los datos que pueden guardarse, esto incluye archivos, ya sean de texto o binarios. La ventaja de utilizar esto a comparación de métodos tradicionales de administración de archivos es que se pueden utilizar todas las características anteriormente indicadas para manejar los archivos de una aplicación.

Aggregation/Map Reduce: Es un modelo de programación que facilita la generación y procesamiento de grandes volúmenes de datos. Las operaciones de map reduce permiten filtrar y organizar los resultados de consultas de una forma simple y eficiente.

MongoDB fue desarrollado por Mongo Inc, y su versión inicial salió al público en 2009 (MongoDB.com, 2017).

Web Sockets

El protocolo de web sockets permite la comunicación full-duplex (ambas partes pueden enviar y recibir datos al unísono) entre un navegador web y un servidor. Está definida en el RFC 6455 (IETF, 2011) y está en proceso de estandarización por la W3C (World Wide Web Consortium), la principal organización de estándares web del mundo.

A pesar de trabajar en navegadores web, no utiliza el protocolo HTTP luego del mensaje inicial (handshake) para establecer la conexión, el resto de las operaciones se realizan en un protocolo propio basado en TCP.

La ventaja que tiene sobre otras formas de comunicación entre cliente y servidor como REST, es que funciona de la misma manera que los sockets del sistema operativo, o sea, permite lectura y escritura sin necesidad de esperar, y una vez establecida la conexión, el socket permanece abierto.

Este protocolo será aplicado en el desarrollo de la plataforma web, para proveer acceso inmediato a los datos que ingresan sin tener que refrescar las páginas en el navegador.

Diseño Metodológico

La problemática se estudiará a través de entrevistas a autoridades de empresas de recolección de residuos y de la división de residuos del Dublin City Council. Se entrevistará repetidamente a las mismas personas, puesto que una vez planteado el problema y desarrollado un prototipo de solución, existe la necesidad de corroborar que lo que se tiene es de utilidad y vale la pena llevar al mercado.

El análisis de la información será fundamental a la hora de tomar decisiones tanto de diseño como de negocio, puesto que el producto final se debe ajustar a las necesidades de los usuarios.

La validación inicial de si la información a recolectar sirve se realizará en forma manual, recorriendo calles y anotando cuánto espacio hay en los basureros utilizando una cinta métrica. Los datos se cargarán manualmente en una plataforma web mínima, que permita visualizar la información en forma de mapas y reportes básicos.

Planificación del proyecto

Para la planificación del proyecto se utilizará Asana, una herramienta que permite crear, administrar y

seguir las tareas requeridas para completar el trabajo (Asana, 2017).

Procesos de Negocio

Se utilizarán diagramas BPM (Business Process Model) generados usando Dia, una herramienta open source para generar este tipo de gráficos (Ko, 2008).

Metodología de desarrollo

Se utilizará la metodología RAD (Rapid Application Development), para poder generar y descartar prototipos rápidamente y en forma iterativa. Los sprints serán de una semana, y al final se presentará un burndown chart para mostrar la progresión de las tareas. Para los fines prácticos, se asignarán puntos a cada tarea según la cantidad de horas en vez de utilizar sistemas basados en Fibonacci para especificar la complejidad. En cuanto a la información diaria, se utilizará Scrum para informar el estado y progreso.

RAD surgió como respuesta a los procesos de desarrollo de las décadas 70 y 80, que se asemejaban a los modelos tradicionales de ingeniería aplicados en la construcción y la industria, donde luego de las etapas de diseño y planificación, realizar cambios se volvía significativamente más costoso. El software es un producto completamente distinto, y tiene la particularidad de que muchas veces sus requerimientos tienden a cambiar durante el ciclo de vida y desarrollo, es por esto que las técnicas rígidas como Cascada tienden a ser ineficientes y costosas.

Las ventajas de utilizar esta metodología son las siguientes:

Reducción de riesgos: Probar las partes complejas de un sistema utilizando prototipos permite descartar soluciones muy costosas o complicadas de antemano, ahorrando tiempo y dinero (CaseMaker Inc, 2000).

Los usuarios son mejores usando y reaccionando a prototipos que creando especificaciones: RAD permite validar las necesidades reales de los usuarios a medida que se avanza. En la metodología de Cascada el usuario debía especificar los requerimientos de antemano, solo para darse cuenta una vez finalizado el proceso de desarrollo que el producto final no se ajusta a las necesidades reales.

Los prototipos son utilizables y pueden evolucionar en el producto final: Esto provee la ventaja adicional de obtener funcionalidad del sistema mucho más rápido que con métodos tradicionales.

Son 4 las fases principales del proceso de desarrollo propuestas por James Martin, uno de los precursores de RAD (Martin, 1991):

Planificación: Se establecen las necesidades del negocio, el alcance del proyecto, y los requerimientos base del sistema. Termina cuando el equipo de desarrollo y los stakeholders se ponen de acuerdo con los resultados de planificación obtenidos.

Diseño: Se desarrollan prototipos para cada proceso del sistema, y en base a esto se realizan cambios, hasta obtener una versión que satisface los requerimientos. Este proceso es continuo e interactivo.

Construcción: Al igual que la fase de diseño, es continua y se basa en la participación de los usuarios para obtener sugerencias de cambios. Las tareas que cubre son la programación y las pruebas.

Fase de corte: Es el equivalente a la etapa de implantación de los procesos tradicionales. Se despliega el nuevo sistema, se entrena a los usuarios y se provee soporte.

Historias de usuario

Las historias de usuario son una descripción concisa de los requerimientos que deben cumplirse para una tarea de desarrollo. Indica el 'quién, cuándo, y por qué' de un requerimiento. Tienen un formato particular que aporta claridad:

Como un <ROL>, cuando <EVENTO>, quiero que <OBJETIVO> para que <BENEFICIO>.

Ejemplo: Como un usuario común, cuando accedo al sitio web, quiero que se muestre el video de presentación para conocer de qué se trata la empresa.

Épicas

Las épicas son historias de usuario muy grandes, que deben subdividirse para poder trabajarse. Por lo que se convierten en grupos de historias de usuario más pequeñas.

Sprints

El trabajo se dividirá en varias iteraciones, denominadas sprints. Por lo general son de 2 semanas, pero pueden ser más cortos o largos según el tipo y la necesidad del proyecto.

Un sprint comienza con la planificación, donde los desarrolladores, los project managers, y los stakeholders, coordinan las historias de usuario que serán incluidas en la iteración. Luego, durante el sprint se tienen reuniones diarias para reportar el progreso y evaluar bloqueos. Al final del período, se tiene una reunión de repaso para evaluar la performance y planificar adaptaciones necesarias.

Story Points

Un *story point* es una estimación del esfuerzo que llevará completar una historia, está directamente relacionado con su complejidad. Al no ser más que estimaciones, pueden tomar más o menos tiempo del estimado al realizar el trabajo, por lo que no se los debe tomar como compromisos de finalización (Zaheerabbas Shaुकatali, 2016).

Burn Down Charts

Un *burn down chart* es un gráfico que permite representar el trabajo restante respecto del tiempo que queda. Se utiliza para predecir cuando el trabajo de un sprint estará completo, y también para poder visualizar la distribución del esfuerzo realizado en un sprint. Las tareas pueden representarse tanto en tiempo como en *story points*.

El eje Y del gráfico contiene el trabajo, y el eje X indica el tiempo. También se traza una línea ideal de trabajo que conecta el inicio (N puntos) con el final (0 puntos). La línea que indica el trabajo real no suele ser una recta, sino que fluctúa alrededor de la línea ideal.

Si la línea de trabajo está por encima de la ideal al final del sprint, significa que faltaron horas y quedaron tareas sin completar. Por el contrario, si la línea de trabajo toca 0 antes que la ideal, quiere decir que se terminaron las tareas antes del tiempo planeado (James, Walter, 2010).

Relative Mass Valuation

La técnica seleccionada para realizar las estimaciones de trabajo en este proyecto se conoce como Relative Mass Valuation, y sirve para evaluar el backlog entero al inicio del proyecto, cuando no se tienen referencias debido a que no se ha comenzado a desarrollar. Para utilizar este método, se deben especificar todas las historias de usuario, y luego ordenarlas según su tamaño y complejidad. El paso siguiente consiste en asignar puntos, para esto se comienza con la historia más simple, que se toma como referencia para determinar los puntos de las demás (Zouggari, 2016).

Control de Versiones

El término control de versiones se refiere a la utilización de herramientas para administrar y registrar diferentes versiones del código y otros recursos (imágenes, textos, etc) en un proyecto de software (Loeliger, 2012).

Git es la herramienta que se utilizará en este caso. Al ser una aplicación simple, no se utilizarán

distintas branches para diferentes releases, sino que master siempre estará al día con los últimos cambios completados. En cuando a las tareas/stories, estas sí se separarán individualmente en branches, nombradas según el número de ticket, y se hará merge a master cuando estén completas (Loeliger, 2012).

Modelado

UModel es la herramienta que se utilizará para modelar el sistema (Altova, 2017).

Programación

La aplicación web se programará con Node.js, se utilizará MongoDB como motor de base de datos. Para las aplicaciones móviles se utilizará Ionic (IonicFramework, 2017)

Implementación

El sistema será implementado sobre la plataforma Amazon Web Services, corriendo en instancias EC2 Linux.

Relevamiento

Relevamiento Estructural

En la actualidad, la recolección de residuos en la ciudad de Dublin, Irlanda, se realiza utilizando rutas pre definidas, el personal debe chequear visualmente los basureros uno por uno, vaciándolos de ser necesario. Esto representa una gran pérdida de tiempo y lleva a que se requiere un gran número de personal para mantener la ciudad limpia.

El City Council de Dublin cuenta con 500 empleados tiempo completo para llevar a cabo las tareas en 5.000 basureros instalados en toda la ciudad.

Algunos sectores, como las zonas céntricas, requieren más personal que otros, ya que la actividad comercial y la densidad de población hacen que los basureros se llenen con mayor rapidez.

Existen 3 tipos de basureros en la vía pública, todos con la parte de arriba cubierta.

Como no se cuenta con información sobre el estado de los basureros. Se utilizan rutas estáticas que se recorren varias veces al día.

La ilustración 1 presenta una ruta estática, en la que solo los basureros rojos están llenos y necesitan recolección, pero por falta de información se deben controlar todos.



Ilustración 1: Ruta de recolección sin optimizar

Cabe destacar que la ciudad tiene 1.800.000 habitantes, siendo la más grande de Irlanda.

Hardware: Actualmente no se utiliza hardware para la recolección de datos sobre el estado de los basureros. Las rutas son especificadas en papel y generalmente no cambian, por lo que los empleados las conocen de memoria.

Software: No se utiliza un software particular para la planificación de rutas ni la administración de basureros.

Relevamiento Funcional

Organigrama

La ilustración 2 muestra el organigrama del área de recolección de residuos del Dublin City Council. Se construyó a base de la información provista por el personal.



Ilustración 2: Estructura organizacional de la sección de Waste management del Dublin City Council

Funciones de las Áreas

- Dirección: Se encarga de las tareas de gerencia general, y responde directamente al Dublin City Council. Las decisiones aquí tomadas alcanzan a toda la organización. La planificación de rutas e instalación de basureros se realizan en conjunto entre la dirección y el área de operaciones.
- Área de depósito y taller: Tiene como objetivo el almacenamiento y reparación de los activos de la organización (camiones, basureros, elementos de limpieza y herramientas, etc.).
- Área de administración: Como su nombre lo indica, se encarga de la parte administrativa, tanto contable como de recursos humanos.
- Área de operaciones: Es el área operativa de la organización, y se divide en dos: Residuos y Barrido y Limpieza. Los operarios de residuos son los que se encargan de la recolección en camiones, tanto de contenedores como basureros pequeños. El área de barrido y limpieza se encarga de mantener limpias las veredas y calles, cuentan con camionetas pequeñas para cargar menores cantidades de basura y sus elementos de limpieza.

Procesos de negocios

En esta sección se presentan los procesos que son relevantes al proyecto, con diagramas BPM y descomposición en pasos.

Proceso: Asignación de rutas.

Roles: Gerente de operaciones, Chofer de camión de recolección.

Pasos:

- El chofer solicita una ruta al iniciar el turno.
- El gerente de operaciones de turno busca una ruta no asignada.
- El gerente de operaciones asigna la ruta a ser recorrida a la hora de iniciar el turno.
- El chofer recorre la ruta especificada durante el turno, vaciando los basureros cuando es necesario. Este paso no termina al final de la ruta, sino al finalizar el turno, cuando el chofer lleva la basura a descargar y vuelve al depósito con el camión.

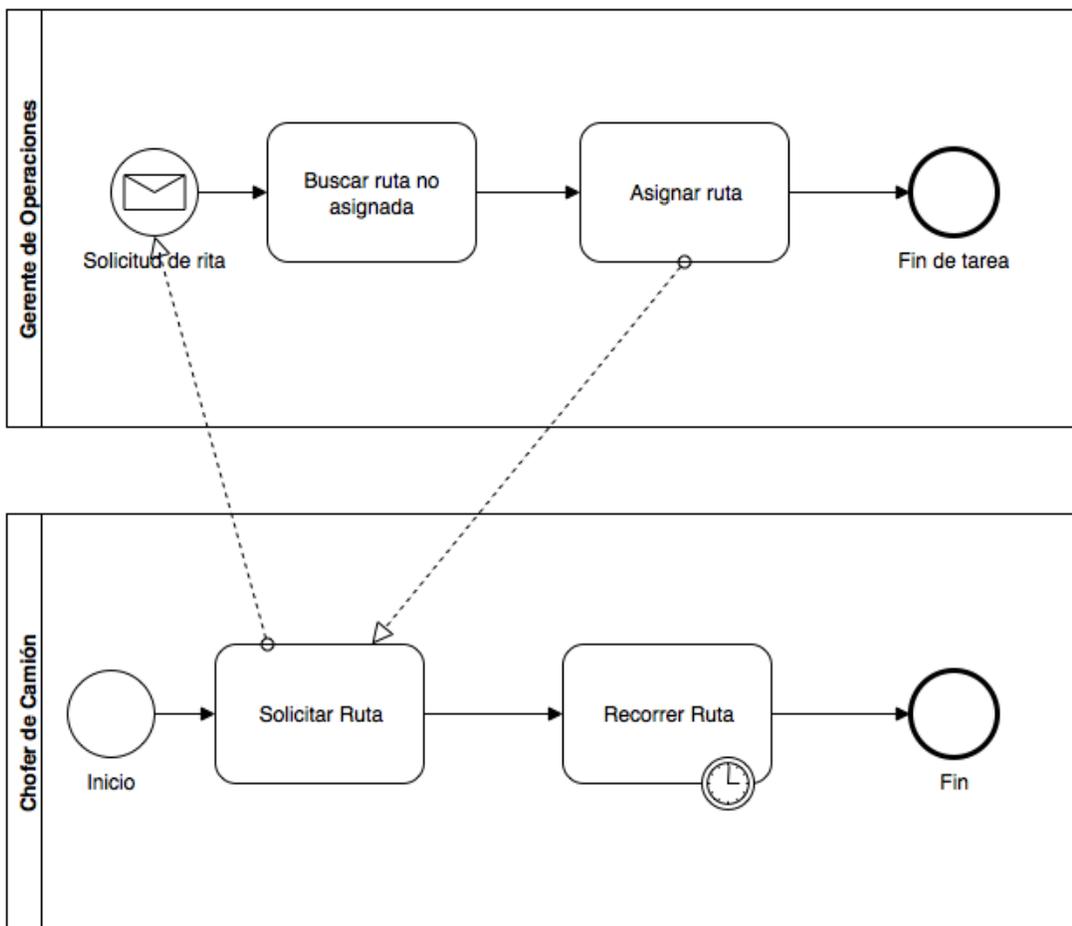


Ilustración 3: Proceso de asignación de rutas

Proceso: Generación de rutas.

Roles: Gerente de operaciones.

Pasos:

-El gerente de operaciones especifica las rutas a recorrer utilizando un mapa de los basureros de la ciudad.

-Una vez generadas las rutas, solo se cambian cuando se instalan nuevos basureros en la misma o en las cercanías.

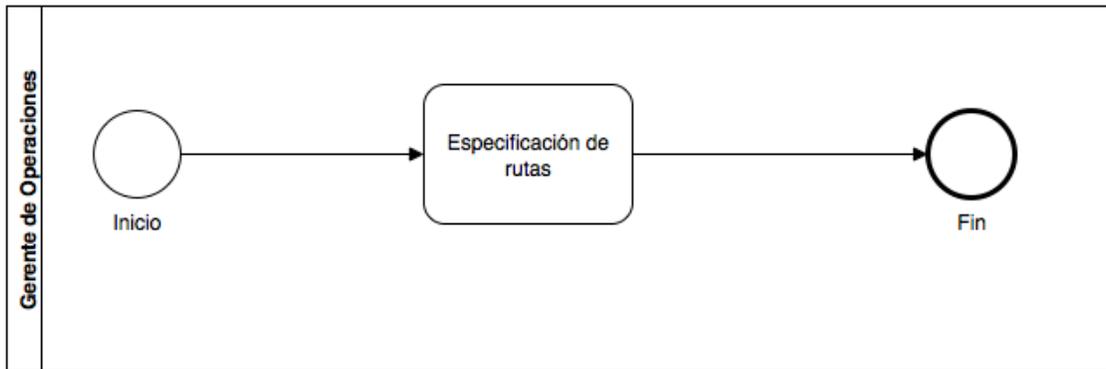


Ilustración 4: Proceso de generación de rutas

Diagnóstico

Con los procesos identificados y la información recolectada, se puede elaborar un diagnóstico que nos permitirá formular una propuesta de solución. En esta sección se detallan los problemas de los procesos y sus causas. También se elabora una propuesta de solución acorde.

Proceso: Asignación de rutas.

Problemas: Las rutas asignadas en forma estática son ineficientes, puesto que se recorren lugares que no necesitan recolección. Esto lleva a una pérdida de tiempo, combustible, y un mal aprovechamiento del personal.

Causas: Los niveles de actividad en las distintas zonas de las ciudades cambian constantemente, y esto lleva a que los basureros no tengan siempre los mismos niveles de uso. La falta de información actualizada genera estos problemas.

Proceso: Generación de rutas.

Problemas: No se cuenta con información suficiente, como niveles de actividad y frecuencia de recolección, y las rutas se basan simplemente en proximidad.

Causas: En la actualidad no se recolecta información de uso de los basureros, y si bien se ha intentado hacer que los empleados registren los niveles de uso en planillas, no se ha tenido un buen resultado.

Propuesta

La solución propuesta comprende 3 componentes: Los sensores, la aplicación web, y la aplicación móvil. Los sensores proveerán datos a través de una API, que se guardarán en la base de datos de la que la aplicación web y móvil se servirán.

La elección de esta propuesta está alineada con la necesidad del Dublin City Council de obtener una solución que no sea privativa en términos de costos y les provea la información necesaria para optimizar rutas y tomar decisiones de planificación.

Sensores

- Utilizan buscadores de rango ultrasónicos para determinar el nivel de los basureros
- Disponibles en versión WiFi o 3G
- Deben poder ser instalados en cualquier basurero o contenedor existente
- La frecuencia de escaneo debe poder ser especificada desde el servidor, sin tener que visitar la ubicación física del sensor. Esto permitirá ajustar los sensores individualmente y optimizar la vida de la batería

Aplicación Web

- Debe permitir visualizar mapas con el estado de todos los basureros que cuenten con sensores instalados
- Debe proporcionar reportes de recolección y uso: Individuales de cada basurero y por regiones
- Los mapas se actualizarán en vivo, al igual que los gráficos que indican la situación actual
- La configuración de los sensores se realizará a través de la aplicación web, por lo que deberá contar con mecanismos e interfaces que lo permitan

- Los niveles mínimos para recolección se podrán configurar a través de esta aplicación
- Contará con diferentes niveles de acceso y privilegios para restringir el uso no autorizado de funciones
- Proveerá una API que permitirá a las aplicaciones móviles consumir los datos y a los sensores enviarlos

Aplicación Móvil

- Proporcionará mapas en vivo con el estado de los basureros, indicando cuales están listos para recolección
- Permitirá la generación de rutas óptimas para la recolección basada en datos actuales
- Los datos serán consumidos de la plataforma web a través de APIs

Listado de Requerimientos funcionales

- Trazado de rutas óptimas basado en ubicación actual y estado de los basureros.
- Visualización de basureros llenos y no llenos en un mapa
- Reportes de estado individuales para cada basurero, con historial.
- Reportes generales de recolección y actividad
- Administración de configuración de sensores a través de la plataforma web

Listado de Requerimientos no funcionales

- La aplicación web utilizará *responsive design*
- Los sensores deben tener una duración de batería mínima de 1 año
- Los sensores deben soportar condiciones climáticas adversas, siempre y cuando no sean extremas
- Las aplicaciones móviles deben ser nativas
- El costo de producción de los sensores no debe superar los €40
- No es aceptable tener que reemplazar las baterías de los sensores con mucha frecuencia, por lo que la

solución deberá utilizar baterías de larga duración para soportar como mínimo un año sin necesidad de reemplazos o hacer uso de paneles solares para recargar baterías LiPo.

- Los sensores deben soportar golpes y condiciones extremas de temperatura y humedad.

Listado de Requerimientos Candidatos

- Predicción de estados basado en historial de cada basurero
- Calculador de ubicación óptima de basureros nuevos basado en la actividad por zona
- Heat Map de actividad

Diagrama de Gantt

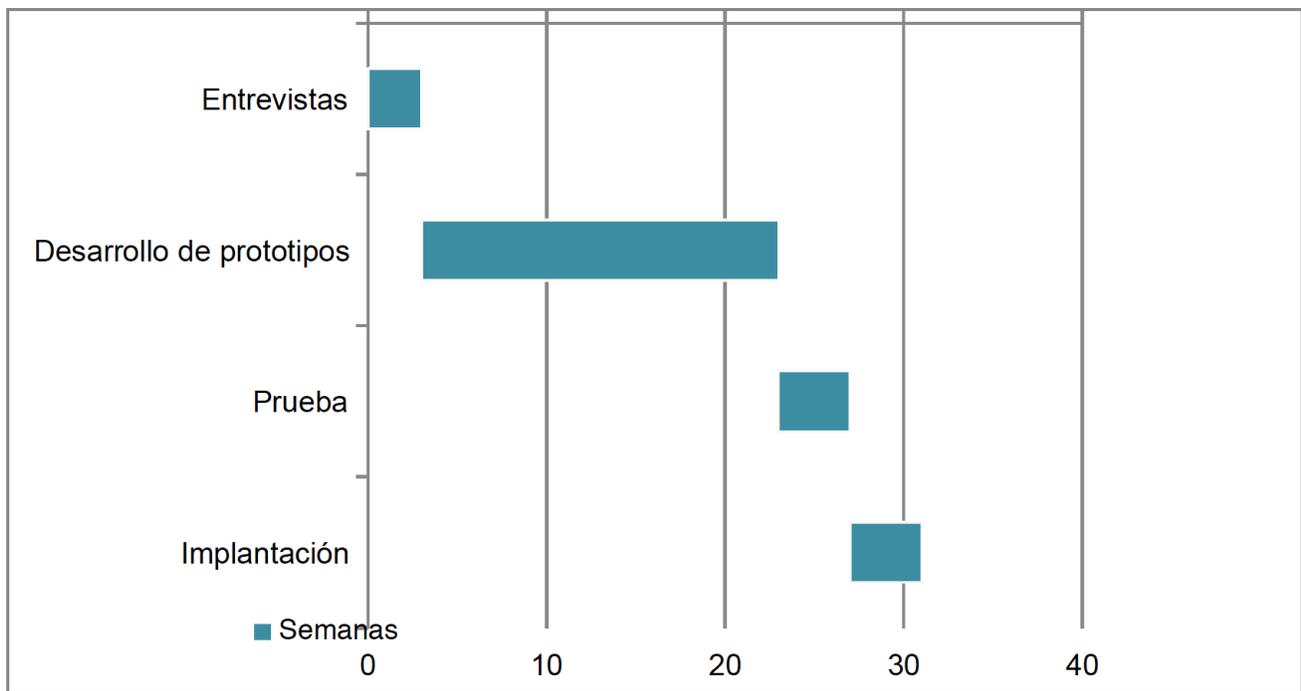


Ilustración 5: Diagrama de Gantt

Desarrollo del Producto/servicio

Análisis y Diseño

Entidades

Las siguientes son las entidades necesarias para la aplicación.

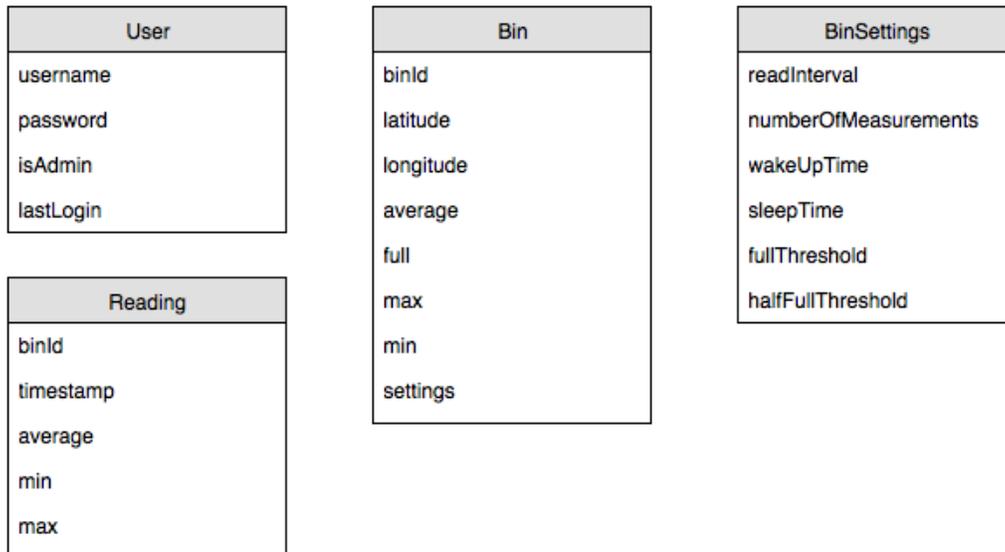


Ilustración 6: Modelos de entidades

Colecciones

Como el motor de base de datos es MongoDB, se crearán las siguientes colecciones:

- Bins: Colección de Bin, todos los basureros. Cada Bin contiene BinSettings.
- Users: Colección de User.
- Today: Todas las lecturas (Reading) del día, se limpia cada medianoche.
- ThisMonth: Todas las lecturas del mes.
- History: Colección histórica de lecturas, no se borran.

Planificación

Estimaciones

Una de las tareas más difíciles a la hora de planear un proyecto ágil, es estimar el esfuerzo para cada historia de usuario planteada. Si bien no deben tomarse las estimaciones como compromisos (puesto que son solamente lo que el equipo piensa van a tomar ciertas tareas), es importante acercarse lo más

posible a valores reales, puesto que la planificación del proyecto se ve afectada.

Como se indicó en el marco teórico, la técnica utilizada en este proyecto se conoce como *Relative Mass Valuation*. Las tareas e historias de usuario se identificarán con BIN-# seguido de un nombre indicativo.

Backlog

Épica: Desarrollo del sensor

BIN-1 Investigar sobre componentes necesarios y precios

Se deben conseguir componentes de bajo consumo que permitan leer y transmitir los datos. También debe investigarse cual es la mejor forma de suministrar energía al dispositivo.

Estimación: 20

BIN-2 Desarrollar un prototipo de hardware básico

Debe poder medir el espacio restante utilizando un sensor ultrasónico y enviar los datos a través de WiFi.

Estimación: 40

BIN-3 Desarrollar el algoritmo de lecturas

El sensor debe proveer lecturas confiables, por lo que deben tomarse varias y luego calcularse el máximo, el mínimo, y el promedio.

Estimación: 8

BIN-4 Desarrollar los algoritmos de conexión y reconexión

La conexión debe adquirirse automáticamente, y debe suspenderse luego de enviar los datos para ahorrar energía. En el caso de que la red no esté disponible al momento de realizar una lectura, deben guardarse los datos en memoria, y reenviarlos cuando la conexión pueda establecerse.

Estimación: 4

BIN-5 Desarrollar el algoritmo de configuración

Debe activarse cada vez que se envían datos, y chequear si existe una nueva configuración para el sensor en el servidor, de ser así, deben actualizarse los valores locales.

Estimación 2

BIN-6 Integrar el software para el sensor

Combinar los diferentes componentes de software e instalarlos en el dispositivo de hardware.

Estimación: 12

BIN-7 Modelar una carcasa para impresión 3D

El sensor necesita una carcasa que sea pequeña, resistente a golpes y humedad, y que facilite la instalación en diferentes tipos de contenedores.

Estimación: 4

Total de puntos en épica: 90

Épica: Desarrollo de la plataforma web

BIN-8 Crear los modelos de entidades

Los modelos necesarios son Users, Bins, BinSettings, y Readings.

Estimación: 5

BIN-9 Desarrollar la API

Se necesitará una API a través de la cual se pueda hacer el CRUD de las entidades. La API será utilizada tanto por la aplicación web como la móvil.

Estimación: 10

BIN-10 Login

Como un usuario sin sesión iniciada, cuando accedo a la plataforma, debo ver la pantalla de login para

poder ingresar mis credenciales de usuario.

Estimación: 4

BIN-11 Vista principal

Como un usuario con sesión iniciada, cuando accedo a la plataforma, debo poder ver el mapa principal con basureros llenos, y las acciones disponibles.

Estimación: 6

BIN-12 Vista de basurero

Como un usuario con sesión iniciada, cuando hago click en un basurero, debo poder ver la pantalla de estado, con los gráficos correspondientes.

Estimación: 6

BIN-13 Agregar/Eliminar basurero

Como un usuario con sesión iniciada y permisos de administrador, cuando elijo la opción de agregar/eliminar basurero en el menú, debo poder ver el diálogo correspondiente.

Estimación: 6

Total de puntos en épica: 37

Épica: Desarrollo de la aplicación móvil

BIN-14 Login

Como un usuario sin sesión iniciada, cuando abro la app debo poder ver la pantalla de login para ingresar mis credenciales

Estimación: 2

BIN-15 Mapa

Como un usuario con sesión iniciada, cuando abro la app debo poder ver el mapa con los basureros y la ruta a seguir basada en mi posición y los basureros llenos.

Estimación: 15

BIN-16 Conectar la app a la API

La app debe leer los datos del server a través de la API.

Estimación: 3

Total de puntos en épica: 20

Sprints

El total de puntos planificados es de 147 en 16 historias y tareas, por lo que se dividió el trabajo en 5 sprints de 30 puntos aproximadamente. La duración de los sprints es de 1 semana.

Sprint 1

El primer sprint tiene dos objetivos, por una parte se necesita determinar, qué componentes serán necesarios para el prototipo (para realizar los pedidos necesarios) y por otra parte, se deben desarrollar los modelos básicos del sistema, así como también los elementos de la API que nos permitan realizar el ABM básico.

BIN-1 Investigar sobre componentes necesarios y precios

BIN-8 Crear los modelos de entidades

BIN-9 Desarrollar la API

Completados: BIN-1, BIN-8 (25 puntos)

En progreso: BIN-9 (10 puntos)

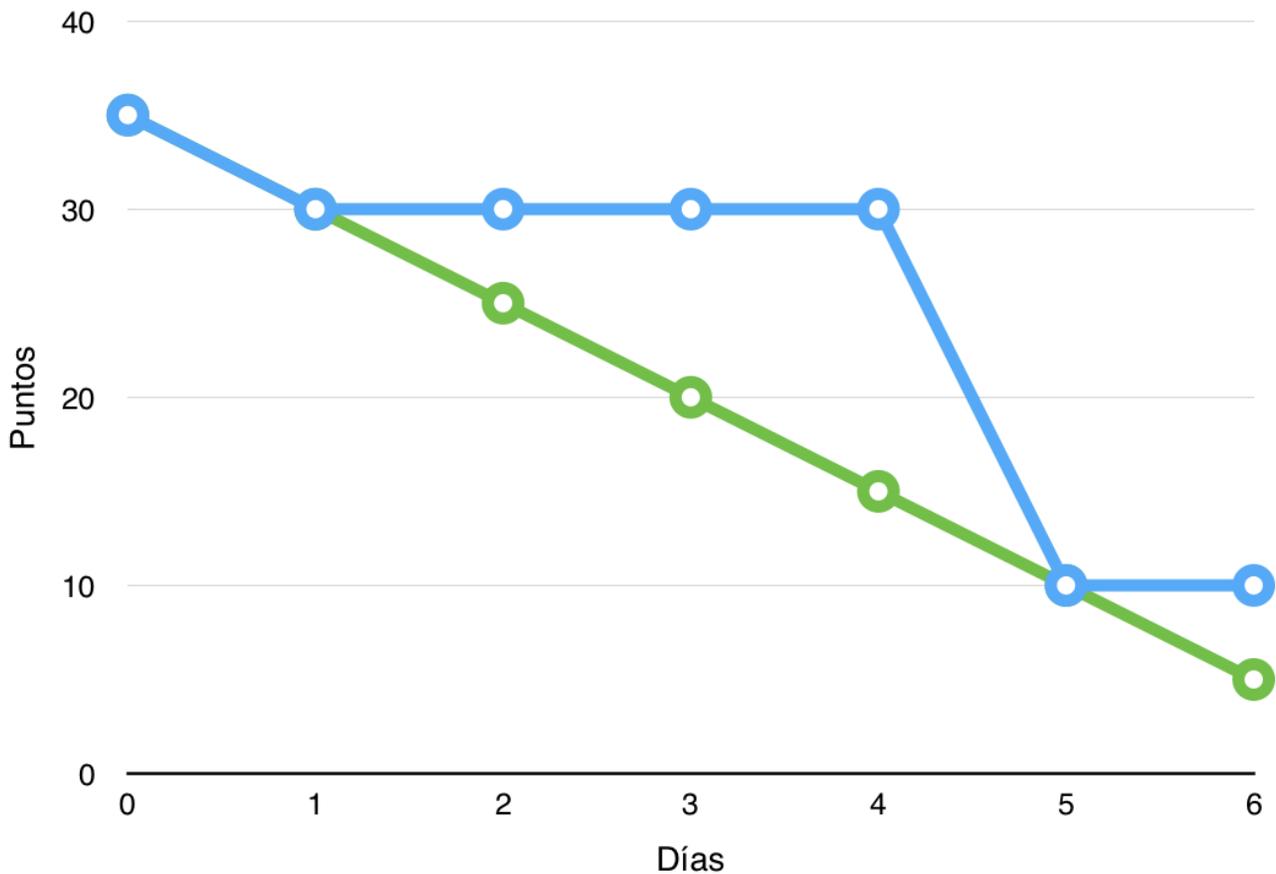


Ilustración 7: Burndown chart Sprint 1

Sprint 2

Con los modelos ya establecidos y los componentes comprados, se debe continuar con el desarrollo de la API, ya que es el componente central del sistema. También se comienza con el desarrollo de un prototipo de hardware que nos permita realizar las lecturas y enviarlas al servidor.

BIN-9 Desarrollar la API

BIN-2 Desarrollar un prototipo de hardware básico

Completados: BIN-9 (10 puntos)

En progreso: BIN-2 (40 puntos)

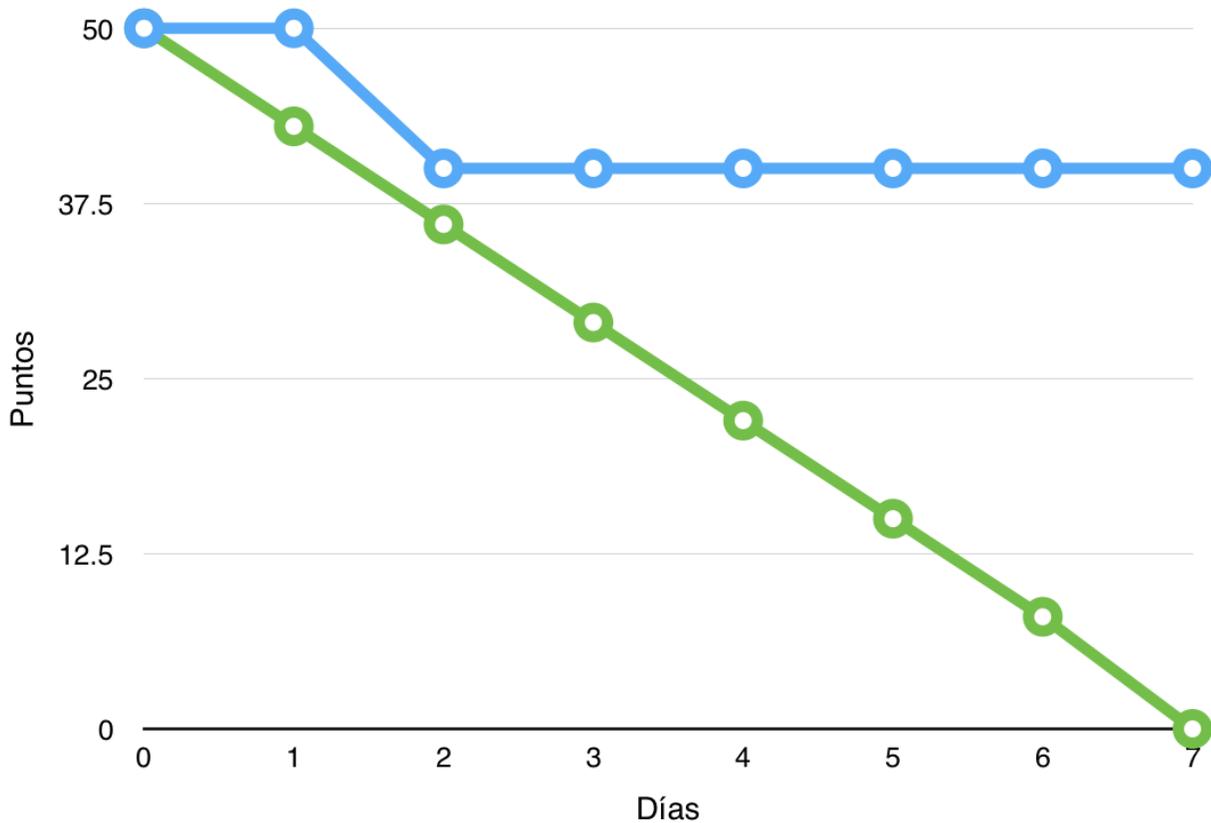


Ilustración 8: Burndown chart Sprint 2

Sprint 3

El tercer sprint fue íntegramente dedicado al prototipo de hardware, completando el desarrollo del componente físico y creando los algoritmos de conexión y lecturas. Esto nos permite realizar lecturas reales y probar el concepto básico.

BIN-2 Desarrollar un prototipo de hardware básico

BIN-3 Desarrollar el algoritmo de lecturas

BIN-4 Desarrollar los algoritmos de conexión y reconexión

Completados: BIN-2, BIN-3, BIN-4 (52 puntos)

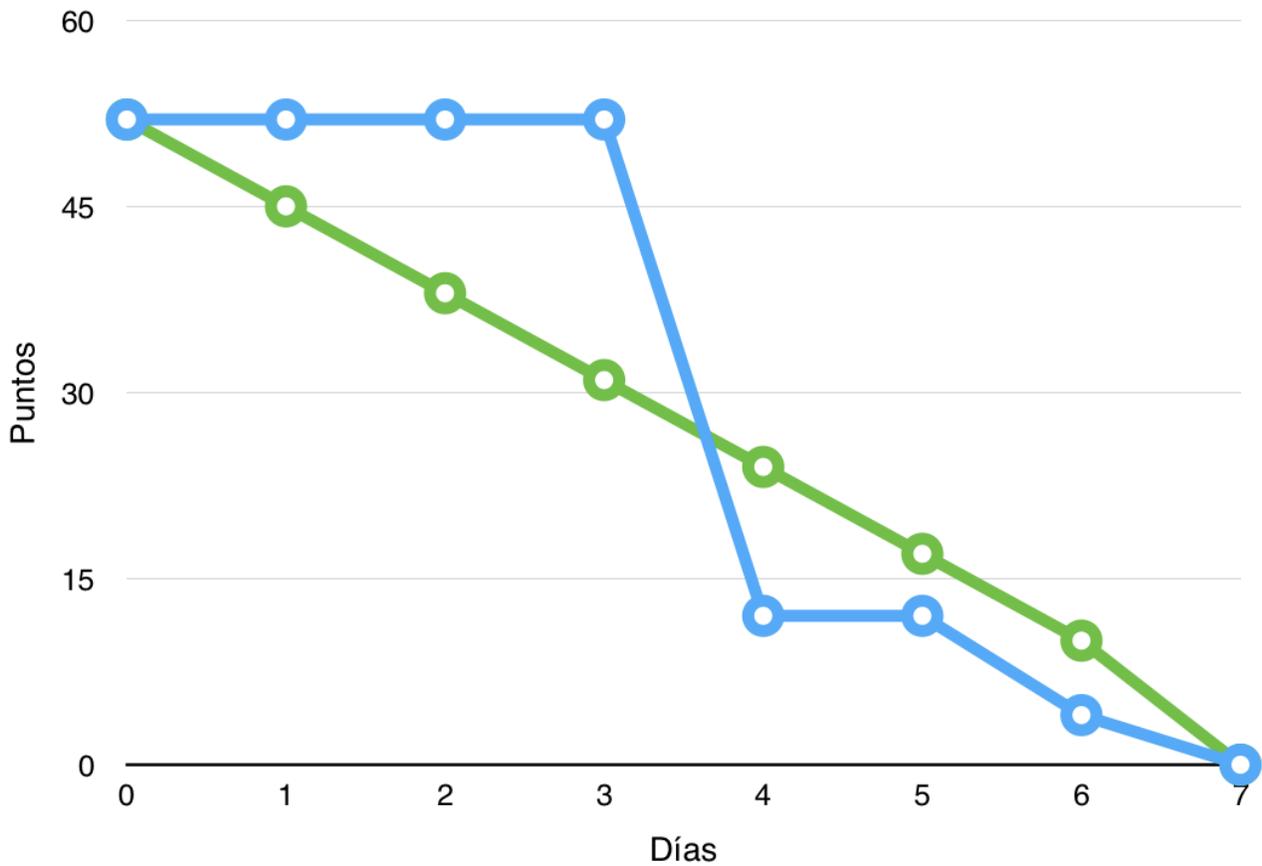


Ilustración 9: Burndown chart Sprint 3

Sprint 4

Con el prototipo de hardware desarrollado y funcionando, se necesita darle la flexibilidad de administrar la cantidad de lecturas desde el servidor, también se deben integrar todos los algoritmos en un solo programa que corra sobre el microcontrolador. Si bien no es de tanta importancia, se completan los puntos libres del sprint con un modelo 3d de la carcasa. Se comienza el desarrollo de la aplicación web con el login y la vista principal (mapa de basureros con sus estados).

BIN-5 Desarrollar el algoritmo de configuración

BIN-6 Integrar el software para el sensor

BIN-7 Modelar una carcasa para impresión 3D

BIN-10 Login

BIN-11 Vista principal

Completados: BIN-5, BIN-6, BIN-7, BIN-10, BIN-11 (28 puntos)

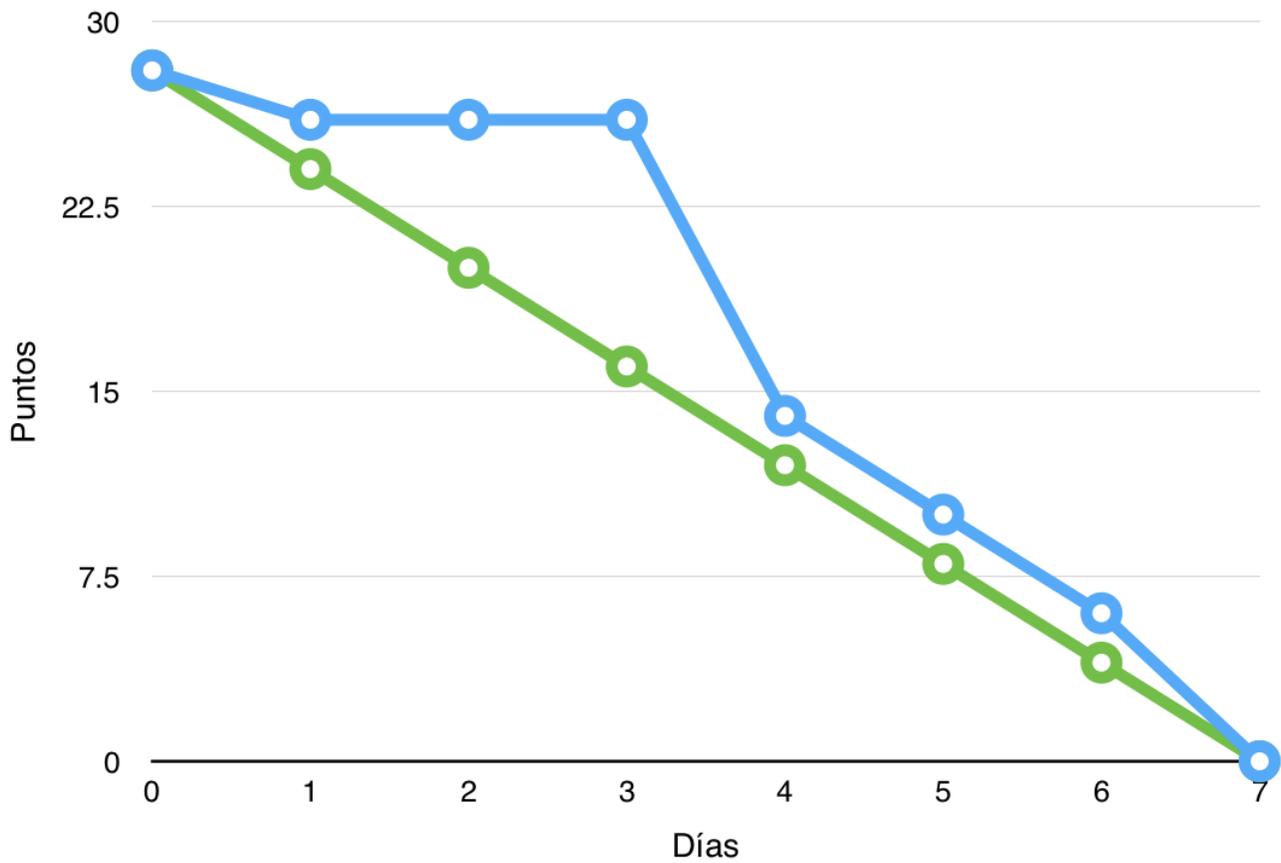


Ilustración 10: Burndown chart Sprint 4

Sprint 5

Para el último sprint se dejaron las tareas relacionadas con la aplicación móvil y las remanentes de la aplicación web.

BIN-12 Vista de basurero

BIN-13 Agregar basurero

BIN-14 Login (Móvil)

BIN-15 Mapa (Móvil)

BIN-16 Conectar la aplicación móvil a la API

Completados: BIN-12, BIN-13, BIN-14, BIN-15, BIN-16 (32 puntos)

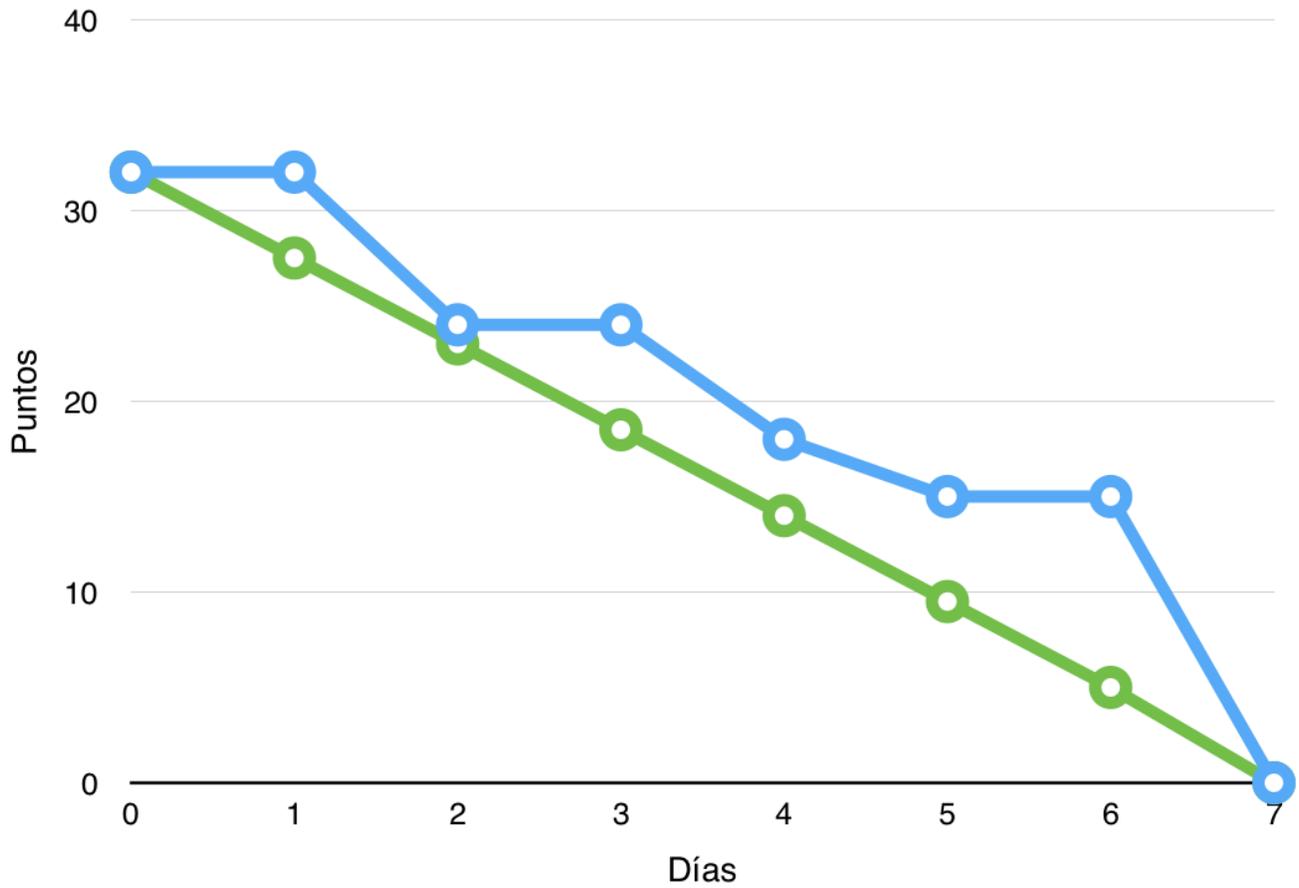


Ilustración 11: Burndown chart Sprint 5

Tickets finalizados por sprint

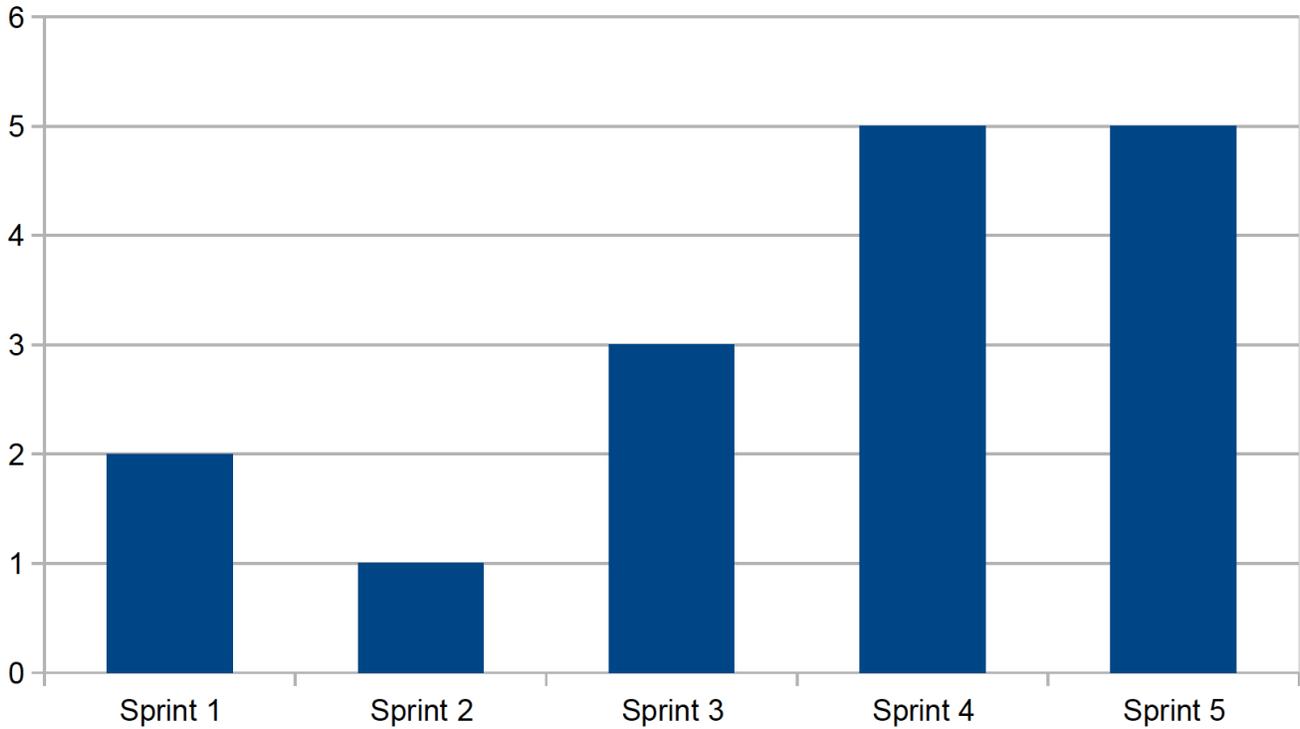


Ilustración 12: Tickets finalizados por sprint

Riesgos del proyecto

Al ser un proyecto que incluye hardware, existen riesgos considerables. Uno de los principales inconvenientes suele ser encontrar proveedores de componentes confiables y que puedan realizar envíos rápidamente. También es un riesgo importante que el hardware sea destruido por vándalos o empleados mal intencionados, pero estos problemas son del tipo operacional.

Existen también riesgos en el proyecto por el lado del software, el mayor siendo que la aplicación tenga problemas de escalabilidad una vez alcanzado cierto número de sensores instalados.

Matriz FODA

Fortalezas	Oportunidades
<ul style="list-style-type: none"> Bajos costos de sensores y de instalación 	<ul style="list-style-type: none"> Poca competencia en el mercado Demanda no satisfecha, los

<ul style="list-style-type: none"> • El reemplazo de unidades puede realizarse con simplicidad • Se elimina la necesidad de planear rutas manualmente • Se evitan recorridos donde la recolección no es necesaria 	<p>competidores no ofrecen precios competitivos</p> <ul style="list-style-type: none"> • El foco de la competencia es el sector privado, esto crea una oportunidad en el sector público • Tecnología nueva, con potencial disruptivo en el mercado de recolección de residuos
<p>Debilidades</p> <ul style="list-style-type: none"> • Los dispositivos no pueden operar bajo condiciones climáticas extremas • El sector público se mueve lentamente, por lo que venderles el producto se vuelve más complicado • Los sensores solo pueden ser instalados en sectores con buena conectividad 	<p>Amenazas</p> <ul style="list-style-type: none"> • El personal de recolección se siente amenazado por la presencia de sensores, piensan que su fuente de trabajo está en riesgo • En zonas de mayor criminalidad, los costos de reemplazo de dispositivos por robos o averías son más altos • Al ser un proyecto de hardware, se depende mucho de la capacidad de producción de los fabricantes

Plan de negocios

Existen múltiples modelos de negocios que pueden ser aplicados. A continuación se presentan los más interesantes:

Venta de sensores y distribución gratuita del software: Tiene la ventaja de permitir que los clientes se encarguen del manejo de sus propios servidores para correr la aplicación, lo que significa menor responsabilidad por parte de Bintelligence. La desventaja es que una vez vendida cierta cantidad de sensores para un cliente, solo necesitarán más cuando deban reemplazar algunos por destrucción o cuando quieran expandir su red. La escalabilidad de este modelo no es muy buena en términos monetarios y los niveles de ingresos pueden variar significativamente de un mes a otro.

Venta de sensores y cobro mensual por acceso a la plataforma: El modelo con mayor potencial de ingresos, pero con el inconveniente de que existen competidores muy grandes utilizando esta misma estrategia y se vuelve muy difícil ofrecer una ventaja competitiva sobre ellos.

Instalación de sensores gratuita y cobro mensual por acceso a la plataforma: Puede resultar muy atractivo para los gobiernos municipales puesto que les permitiría simplificar el presupuesto que tienen disponible para tecnología. El problema es que requiere una inversión inicial bastante grande por parte de Bintelligence y además los costos de reemplazo de sensores por averías pueden volverse un problema.

Instalación de sensores gratuita y cobro por acceso a los datos y plataforma a terceros: Este modelo solo es aplicable a ciudades que tienen tercerizado el servicio de recolección de basura. En este caso el gobierno solo debe permitir la instalación de los sensores, y el cobro se realizaría a las empresas privadas de recolección que necesiten el acceso a los datos. El inconveniente del costo inicial de instalación sigue presente en este caso, pero la ventaja principal es que al no existir un gasto para los gobiernos municipales no hace falta un proceso de licitación. Otra limitación es que muchos gobiernos tienen su propia recolección o trabajan con una sola empresa, lo que hace que el presupuesto de la empresa sea altamente dependiente del gobierno.

Venta de sensores al costo y cobro mensual según la cantidad de dispositivos instalados: En este caso la ganancia por la venta de sensores es nula, pero como de todas formas no es una fuente de ingreso estable, se cuenta con el cobro mensual por acceso a la plataforma según la cantidad de dispositivos instalados. Esta estrategia permite obtener una ventaja competitiva y resulta en un flujo de ingresos estable.

Interfaces de usuario

En esta sección se presentan las diferentes pantallas que formarán parte de la aplicación con el objetivo de proveer una guía visual tanto para el desarrollo como para documentación.

Login



Username
ale

Password
.....

SIGN IN

Stay signed in

[Need help?](#)

Ilustración 13: Login

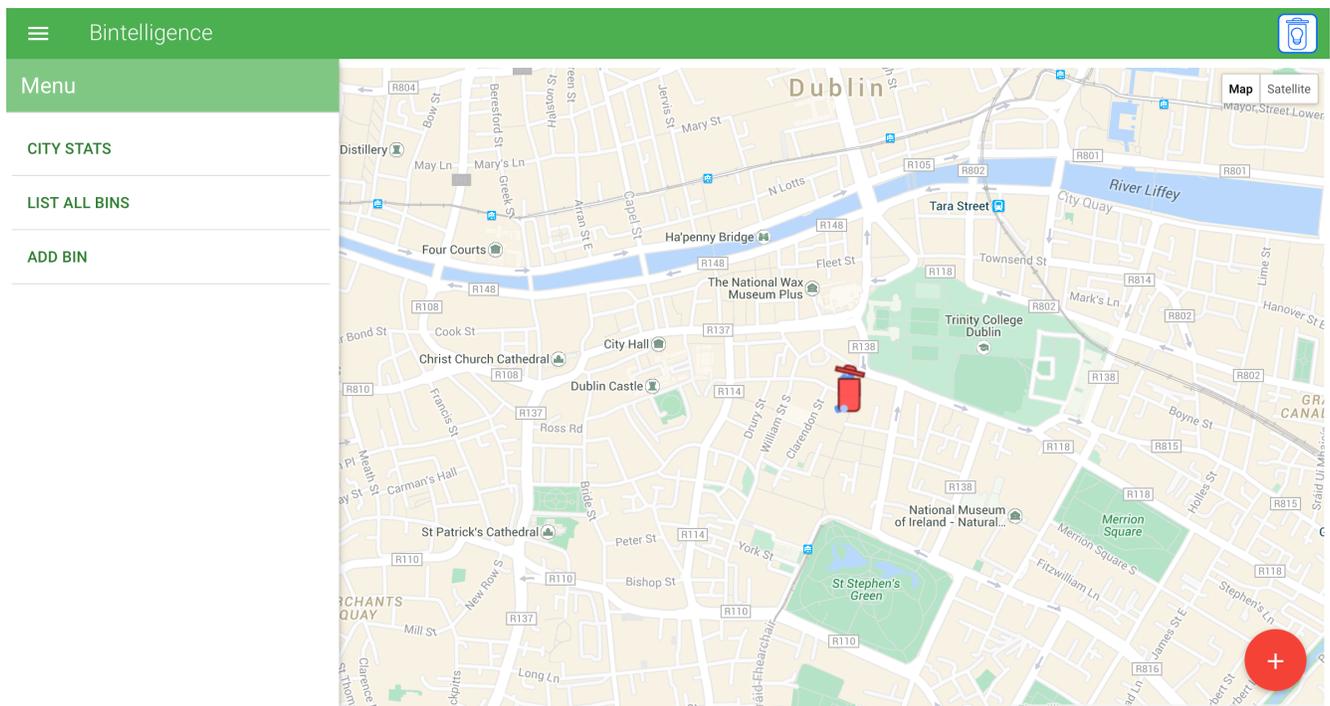


Ilustración 14: Vista general (mapa)

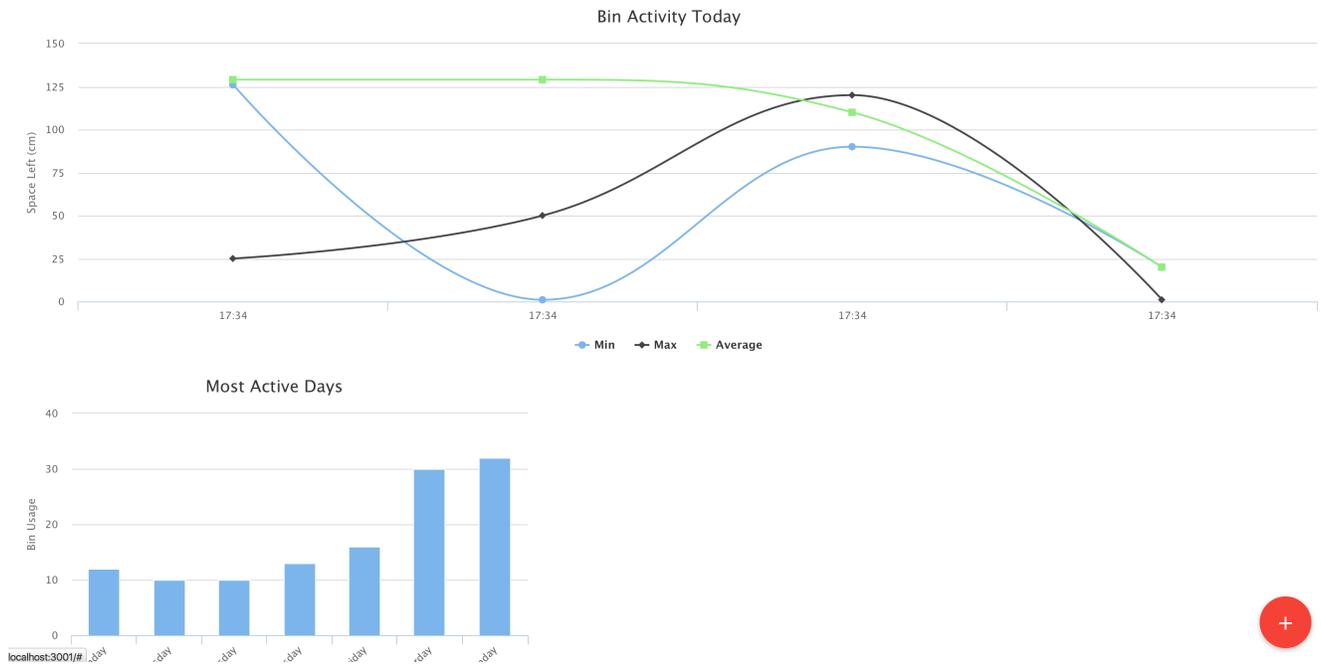


Ilustración 15: Vista de basurero (reporte de basurero)

Add a new bin ✕

Bin ID:

Latitude:

Longitude:

Read interval:

Measurements per reading:

Wake up time:

Sleep time:

Full threshold (cm):

Half full threshold (cm):

CANCEL **SAVE**

Ilustración 16: Diálogo Agregar/Editar basureros

Login



Username
ale

Password
.....

SIGN IN

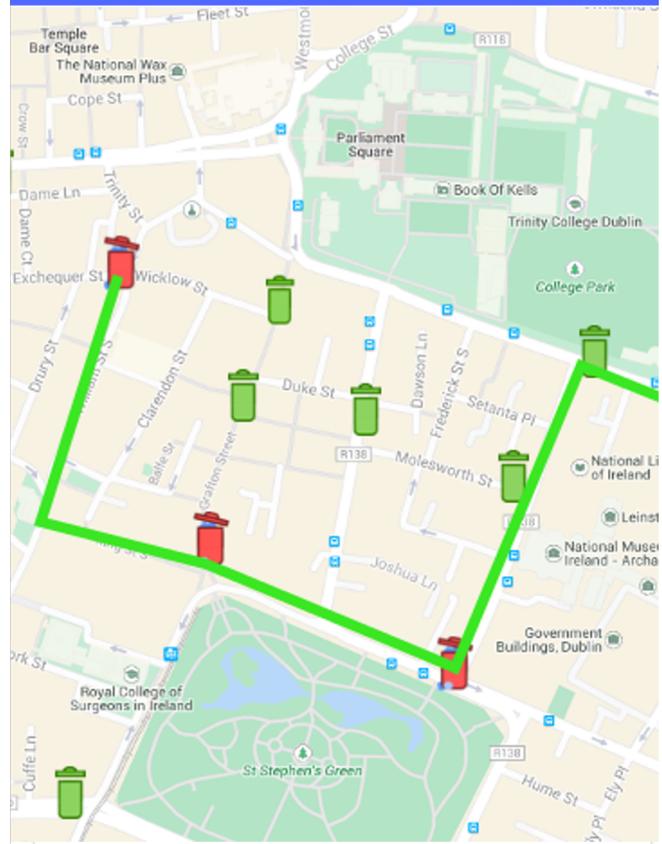


Ilustración 17: Login y Mapa – App Móvil

Selección de componentes de hardware para el sensor

Luego de investigar sobre la disponibilidad de sensores de este tipo en el mercado, se llegó a la conclusión de que no existen empresas que ofrezcan un dispositivo que se ajuste a las necesidades del proyecto y lo vendan por unidad, por lo que fue necesario desarrollar un prototipo propio.

Los requerimientos básicos del sensor son los siguientes:

Debe permitir medir el nivel de los contenedores en los que se instala

No es viable conectarlo a la red eléctrica, por lo que se necesitarán baterías

El costo por unidad debe ser inferior a los 50 euros, para que la instalación masiva sea económica, y también para que el costo de reemplazo de unidades averiadas no sea significativo.

No es necesario que opere en condiciones extremas (temperaturas bajo cero, inundaciones), por lo que los componentes no necesitan ser especiales.

Se desarrollaron prototipos con dos tipos de sensores de profundidad: infrarrojos y ultrasónicos.

Los sensores infrarrojos fueron descartados con el primer prototipo, puesto que no son aptos para ser utilizados cuando hay mucha luz solar en el ambiente (las lecturas eran nulas o muy imprecisas). Fue por esto que se decidió continuar solamente con sensores ultrasónicos.

Sensor ultrasónico

Las lecturas con este tipo de sensores se realizan mediante la emisión de pulsos ultrasónicos (con frecuencias mayores a 20khz, 40khz para los utilizados en el prototipo). Basado en en el tiempo que tarda en rebotar la onda, se puede calcular la distancia del objeto más cercano. Para esto, los sensores cuentan con un emisor y un receptor.

Una vez recibido el pulso, se puede calcular la distancia al objeto contra el cual rebotó, dividiendo la velocidad del sonido (343,2 metros por segundo) por el tiempo que tomó el rebote. Como el sonido viaja a diferentes velocidades dependiendo de las condiciones del ambiente, esta medida no es muy precisa, pero se acerca bastante a los valores exactos, y es más que suficiente para el caso de medición de nivel de contenedores.

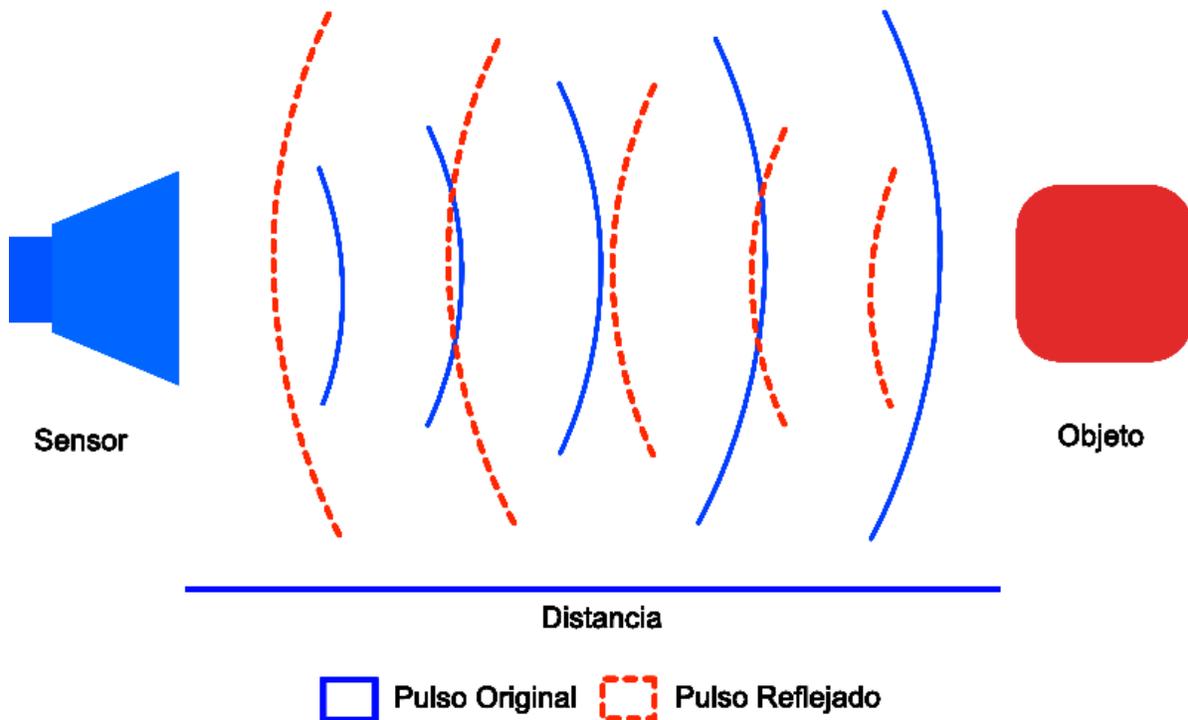


Ilustración 18: Medición de distancia con un sensor ultrasónico

Energía

Uno de los desafíos más grandes al desarrollar sensores que deben ser instalados en la vía pública es el de suministrar energía.

Primeramente se intentó utilizar pequeños paneles solares con baterías recargables de 2000mah para alimentar los dispositivos, pero no resultó ser la elección adecuada, por diversas razones:

Los sensores se instalan por dentro de los contenedores, y los paneles solares deben ir por fuera, por lo que debían taladrarse huecos y pasar cables.

Instalar paneles que estén expuestos en la vía pública es muy mala idea, reciben mucho daño y son fáciles de vandalizar.

Los paneles solares representaban más del 50% del costo total de los dispositivos.

Hay regiones geográficas en las que la energía recolectada con paneles es muy poca durante el invierno. Por ejemplo, en el mes de diciembre, la cantidad de horas de luz se reduce drásticamente en los países al norte de Europa (Paul Burgess, 2009).

Además de todo esto, al utilizar paneles solares se necesitan baterías recargables, pues la corriente provista no es continua. El uso de baterías de este tipo en la vía pública trae a su vez más problemas:

Son costosas y tienen una cantidad de ciclos de recarga limitados, por lo que deben ser reemplazadas pasado cierto tiempo.

Las baterías de litio, las mismas que se utilizan en teléfonos celulares y la mayoría de los dispositivos electrónicos de hoy en día, tienen la desventaja de no poder ser recargadas en temperaturas muy bajas o elevadas (Suthar, Sonawane, Braatz, Subramanian, 2015). Por otra parte, las baterías basadas en níquel no tienen el mismo problema, y pueden ser cargadas hasta a 18 grados bajo cero, pero son más costosas aún, más contaminantes, y no pueden cargarse al 100%, por lo que se debe agregar un controlador de carga que tome en cuenta la temperatura, aumentando la complejidad del dispositivo y también su costo.

Debido a todas las razones anteriores, y luego de consultar con el Dublin City Council, se decidió optar por baterías no recargables, que tienen una duración de entre uno y tres años dependiendo de la configuración del sensor (frecuencia de lecturas y envío de datos). Esto permite prescindir del uso de paneles solares, facilita la instalación y disminuye significativamente el costo del dispositivo. Además las condiciones climáticas del lugar no afectan su funcionamiento.

Microcontrolador

Afortunadamente, no se necesita mucho poder de procesamiento ni memoria para poder controlar el sensor ultrasónico y enviar los datos al servidor, por lo que se hizo foco en utilizar microcontroladores

económicos y con muy bajo consumo de energía.

En la etapa principal se utilizó un Intel Galileo (Intel, 2015) para conectar todo y demostrar la viabilidad del hardware, el problema con esto era que Galileo es una plataforma para realizar prototipos, pero son de un tamaño muy grande, costosos, y consumen mucha energía por su gran capacidad de procesamiento (400mhz). El problema del tamaño se solucionó cambiando a Intel Edison (Intel, 2015), que se parece físicamente a una tarjeta SD, y al igual que Galileo, tiene gran poder de procesamiento (500mhz), 2gb de memoria, y un sistema operativo Linux, lo que facilitaba muchísimo la programación.

Los últimos prototipos fueron creados utilizando microcontroladores ESP8266 (Espressif, 2017), que además de ser de bajo consumo y costo, proveen conectividad WiFi en el mismo chip.

Características relevantes:

Procesador RISC de 32bits Tensilica Xtensa, con velocidad de reloj de hasta 80mhz

64kib de RAM de instrucciones, 96kib RAM de datos

Memoria flash de 512kib

WiFi b/g/n, con soporte de autenticación WEP, WPA/WPA2 y redes abiertas

16 pins de propósito general

Costo de US\$2.5

Selección de tecnología para la aplicación web y API

La solución IoT que se aborda en este trabajo tiene la característica de utilizar una cantidad masiva de dispositivos sensores, que envían con bastante frecuencia datos al servidor, esto produce la necesidad de elegir una tecnología que permita atender una gran cantidad de conexiones y solicitudes en paralelo. Además, cabe destacar que el procesamiento realizado es mínimo y con una cantidad pequeña de datos.

Teniendo estos dos puntos en cuenta, se decidió utilizar Node.js como lenguaje para el *backend* de la aplicación web y API. La razón de la decisión fue la arquitectura manejada por eventos que posee, la cual permite entrada y salida asíncrona.

Para los mapas, se utilizó Google Maps, que fue elegido principalmente por la disponibilidad de mapas completos y actualizados de prácticamente cualquier ciudad del mundo, pero también por contar con una API simple y gratuita.

El *frontend* de la aplicación utiliza Bootstrap, sin ningún framework adicional. Bootstrap cuenta con jQuery, plantillas CSS3 que funcionan en prácticamente cualquier navegador, y simplifica la creación de sitios *responsive* (que se adaptan a diferentes tamaños y tipos de pantalla).

Como motor de base de datos se escogió MongoDB, al ser una solución NoSQL se pueden manejar los datos con mayor facilidad, como objetos, la mayor desventaja es que no se cuenta con tanta flexibilidad al crear reportes dinámicamente, pero no será necesario hacer esto por ahora.

Selección de tecnología para la aplicación móvil

Inicialmente la idea era crear aplicaciones nativas, tanto para iOS como Android, pero luego de un análisis sobre la necesidad de esto, se resolvió utilizar la misma tecnología que en el servidor para crear un cliente liviano. La razón principal es que al ser una aplicación que no hace uso de componentes específicos de las plataformas, no tiene sentido utilizar lenguajes exclusivos. Además al utilizar lo mismo que en la aplicación web, se mantienen consistentes el estilo y la experiencia de usuario.

Selección de plataforma

La escalabilidad es el criterio más importante a la hora de elegir una plataforma. No es aceptable tener que migrar con frecuencia y se debe asegurar la continuidad del servicio. Es por esto que se eligió Amazon Web Services como plataforma, ya que permite modificar la cantidad de memoria, CPU, y capacidad de almacenamiento de manera flexible. Además se cuenta con diversas distribuciones de Linux para instalar en los servidores virtuales, de las cuales EC2 es la más conveniente, por facilitar la conexión al resto de los servicios de AWS a través de diversas aplicaciones pre-instaladas.

Roles

Se implementarán solamente 2 roles para el prototipo:

Administrador: Con acceso a los reportes, mapas, ruteo, y con la capacidad de agregar, editar y remover sensores de la plataforma a través de la aplicación web.

Usuario no administrador: Podrán acceder a lo mismo que los administradores, pero sin la capacidad de agregar, editar y remover sensores.

La razón para esta simplificación de roles es que no hay una necesidad de mayores restricciones para el prototipo, y además la información disponible no es sensible para ningún tipo de usuario (no hay problemas en permitir a los recolectores ver datos adicionales sobre el estado de los basureros).

Presupuesto

El siguiente presupuesto contempla solamente los costos de hardware e instalación de dispositivos, la razón es que al tratarse de prototipos para validar el proyecto y no una implantación comercial, no se necesita un presupuesto adicional. Ninguna fase de este proyecto será facturada a algún tercero.

Etapa	Estimación	Razón
Validación	US\$2500	Costos básicos para insumos requeridos por la validación
Prototipo	US\$5000	Desarrollo de prototipos de hardware, con diferentes componentes, impresión 3D de carcasas, y otros costos
Pruebas Iniciales	US\$3000	Producción e instalación de varios prototipos en lugares públicos
Refinamiento	US\$3000	Producción e instalación de varios prototipos en lugares públicos.

Costos individuales de componentes utilizados en los prototipos

Durante la etapa de desarrollo de dispositivos fueron utilizados diferentes tipos de componentes de hardware, variando en costo, potencia y otros factores. Los costos del hardware por unidad se indican en la tabla de esta sección. Adicionalmente, se hicieron gastos para cosas como impresión 3d de carcasas, equipos de soldar, multímetros y otras herramientas necesarias.

Componente	Descripción	Costo
Intel Galileo	Plataforma de prototipado	US\$70
Intel Edison	System on a Chip para prototipado	US\$38
ESP2866	Transceiver wifi con microcontrolador	US\$6
NodeMCU	Microcontrolador con wifi integrado	US\$9
Paneles solares flexibles	Paneles de 1Watt 6Volts	US\$20
HC-SR04	Sensores ultrasónicos	US\$4
Seed Studio Ultrasonic Sensor	Sensores ultrasónicos	US\$15
Baterías de litio 2000mAh 3.7v	Baterías recargables	US\$15
Baterías de litio 200 mAh 3.7v	Baterías recargables	US\$6
Baterías tipo E 9v	Baterías no recargables	US\$3
Conector para baterías 9v	Caja conectora	US\$7
Componentes electrónicos variados	Resistencias, capacitores, level shifters, transistores, etc.	US\$40

Entrevista

La obtención de la información necesaria para definir los requerimientos del proyecto se realizó a través de entrevistas a Brian Hanney, el director del área de residuos en Dublin City Council. De las respuestas, se obtuvo la siguiente información:

¿Cómo se establecen las rutas de recolección?

Las rutas se diagraman en base a los niveles de actividad de las diferentes zonas. En la principal calle de comercios y sus alrededores se tiene un equipo de 20 personas haciendo recorridos constantes. En las zonas residenciales y menos frecuentadas por otra parte, se planean rutas más largas y con menor frecuencia. Una vez que las rutas se establecen, no se modifican a no ser que se agreguen o remuevan basureros en la zona, por lo que se los empleados las conocen de memoria.

¿Cuál es el costo de recolección por contenedor?

Este cálculo no está hecho, y los costos varían mucho según la zona.

¿Han intentado utilizar sensores para obtener mejor información?

Hasta ahora no, pero se ha intentado utilizar tags RFID en combinación con una aplicación móvil para hacer que los empleados carguen datos del estado de los basureros a medida que los recorren. El resultado no fue muy bueno, muchos empleados decidían ignorar la carga de datos, y también se perdían muchos tags.

¿Cuántos empleados y cuántos basureros tienen?

Actualmente se cuenta con un total de 500 empleados, 300 menos que en 2008, esto hace que recorrer los 5000 basureros que están instalados en la ciudad sea una tarea difícil.

¿Qué tipo de información necesitan para poder planificar mejor las tareas?

El interés principal es conocer los niveles de actividad con mayor detalle, la información con la que se cuenta en la actualidad viene de los propios empleados que recorren la ciudad. Contar con un mayor grado de detalle podría permitir una mejor planificación de rutas y también de ubicación de basureros.

Conclusiones

Para la realización del presente trabajo, se aplicaron conocimientos adquiridos a lo largo de la carrera,

abarcando gran parte de las materias cursadas. Sin estos conocimientos fundamentales, la planificación, desarrollo, y ejecución hubieran sido imposibles.

El proyecto resultante permite apreciar la factibilidad de una red de sensores instalados en contenedores de basura para optimizar la recolección. Queda demostrado también que la Internet of Things servirá en el futuro para optimizar procesos de gran escala y colaborar con la gestión de recursos en todo tipo de operaciones. También se puede apreciar que el mayor desafío no está en la tecnología ni los costos, sino en factores humanos, como la adaptación del personal de recolección a los nuevos métodos y su resistencia al cambio.

Por supuesto que existe una gran distancia por recorrer entre un prototipo validado y un producto comercialmente disponible, pero teniendo la seguridad de que existe un mercado y de que los beneficios de la tecnología son reales, es posible continuar con el esfuerzo para convertir esto en un producto real.

Bibliografia

ABI Research. (2014) Smart Bins and Refuse Collection. Obtenido de <https://www.abiresearch.com/market-research/product/1021028-smart-bins-and-refuse-collection-technolog/>

Eichengreen Barry. (2015) The Irish crisis and the EU from a distance. Obtenido de https://www.imf.org/external/np/seminars/eng/2014/ireland/pdf/Eichengreen_IrishCrisisEU.pdf

Enevo Group. (2014) Tufts University & Save That Stuff. Obtenido de <http://webcache.googleusercontent.com/search?q=cache:JtObkzhLM7cJ:www.enevo.com/wp/wp-content/uploads/2014/09/Enevo-Case-Study-Tufts-University-Save-That-Stuff.pdf>

Node.js. (2017) Obtenido de <https://nodejs.org>

Google Inc. (2017) V8. Obtenido de <https://developers.google.com/v8/>

MongoDB. (2017) Obtenido de <http://mongodb.com>

IETF. (2011) The WebSocket protocol. Obtenido de <https://tools.ietf.org/html/rfc6455>

Asana. (2017) Obtenido de <https://asana.com/>

Ionic Framework. (2017) Obtenido de <http://ionicframework.com/>

Ko Ryan. (2008) Business process management standards. Obtenido de <https://ryanko.files.wordpress.com/2008/03/bpm-journal-koleelee-bpms-survey.pdf>

Altova. (2017) Umodel. Obtenido de <https://www.altova.com/umodel.html>

Burgess Paul. (2009) Variation in light intensity at different latitudes and seasons, effects of cloud cover, and the amount of direct and diffused light. Obtenido de http://www.ccf.org.uk/conferences/downloads/P_Burgess.pdf

Suthar, Sonawane, Braatz, Subramanian. (2015) Optimal low temperature charging of lithium ion batteries. Obtenido de https://www.researchgate.net/publication/282407632_Optimal_Low_Temperature_Charging_of_Lithium-ion_Batteries

CASEMaker Inc. (2000) Rapid Application Development. Obtenido de http://www.eng.auburn.edu/~sealscd/COMP6620/project/SoftwareDevelopmentProcess/rad_wp.pdf

Loeliger Jon. (2012) Version control with Git. Obtenido de <http://www.foo.be/cours/dess-20122013/b/OReilly%20Version%20Control%20with%20GIT.pdf>

Walter James. (2013) Scrum Reference Card. Obtenido de <http://scrumreferencecard.com/ScrumReferenceCard.pdf>

Shaukatali Zaheerabbas. (2016) Demystifying Story Points. Obtenido de <https://agilealliance.org/wp-content/uploads/2016/01/Demystifying-Story-Points1-0.pdf>

Zouggari Slimane. (2016) Relavite Mass Valuation. Obtenido de <https://www.flowless.eu/relative-mass-valuation/>

ex-cell Kaiser. (2017) Product Catalog. Obtenido de https://www.ex-cell.com/media/wysiwyg/customer-service/Ex-Cell-Kaiser-Catalog-2017-SS_Web_.pdf

Independent UK. (2013) Smartbins use devices' Wi-Fi connections to log their MAC address and track customer habits. Obtenido de <http://www.independent.co.uk/life-style/gadgets-and-tech/news/updated-londons-bins-are-tracking-your-smartphone-8754924.html>

Wikipedia. (2017) Internet of Things. Obtenido de https://en.wikipedia.org/wiki/Internet_of_things

Intel Corporation. (2015) Intel Galileo Board. Obtenido de <http://ark.intel.com/products/78919/Intel-Galileo-Board>

Intel Corporation. (2015) Intel Edison Compute Module. Obtenido de <http://ark.intel.com/products/84572/Intel-Edison-Compute-Module-IoT>

Espressif. (2017) ESP8266 Programming Guide. Obtenido de http://www.espressif.com/sites/default/files/documentation/20a-esp8266_rtos_sdk_programming_guide_en.pdf

Anexo I

Modelos de Entrevista

Las siguientes son preguntas de guía para las entrevistas.

Para empresas de recolección:

- ¿Cómo se establecen las rutas de recolección?
- ¿Cuál es el costo de recolección por contenedor?
- ¿Cuál es el margen de ganancia por cada contenedor recolectado?
- ¿Han intentado utilizar sensores para obtener mejor información? Cuáles?
- ¿Les interesaría trabajar con los basureros públicos de la ciudad?
- ¿Cuántos empleados y cuantos contenedores tienen?

Para entes gubernamentales:

- ¿Cómo se establecen las rutas de recolección?
- ¿Cuál es el costo de recolección por contenedor?
- ¿Han intentado utilizar sensores para obtener mejor información? Cuáles?
- ¿Cuántos empleados y cuantos basureros tienen?
- ¿Qué tipo de información necesitan para poder planificar mejor las operaciones?
- ¿Creen que existirá resistencia por parte del personal cuando se instalen sensores?
- ¿Les interesaría proveer información de los basureros a empresas privadas para tercerizar parte de la recolección?
- ¿Cómo planifican la instalación de nuevos basureros en la actualidad?

ANEXO E – FORMULARIO DESCRIPTIVO DEL TRABAJO FINAL DE GRADUACION

AUTORIZACIÓN PARA PUBLICAR Y DIFUNDIR TESIS DE POSGRADO O GRADO A LA UNIVERSIDAD SIGLO 21

Por la presente, autorizo a la Universidad Siglo21 a difundir en su página web o bien a través de su campus virtual mi trabajo de Tesis según los datos que detallo a continuación, a los fines que la misma pueda ser leída por los visitantes de dicha página web y/o el cuerpo docente y/o alumnos de la Institución:

Autor-tesista	Ricatti Alejandro
DNI (del autor-tesista)	32516275
Título y subtítulo <i>(completos de la Tesis)</i>	Optimización de recolección de residuos con IoT
Correo electrónico	alericatti@gmail.com
Unidad Académica <i>(donde se presentó la obra)</i>	Universidad Siglo 21
Datos de edición: <i>Lugar, editor, fecha e ISBN (para el caso de tesis ya publicadas), depósito en el Registro Nacional de Propiedad Intelectual y autorización de la Editorial.</i>	

Otorgo expreso consentimiento para que la copia electrónica de mi Tesis sea publicada en la página web y/o el campus virtual de la Universidad Siglo 21 según el siguiente detalle:

Texto completo de la Tesis	SI
Publicación parcial <i>(Informar que capítulos se publicarán)</i>	

Otorgo expreso consentimiento para que la versión electrónica de este libro sea publicada en la en la página web y/o el campus virtual de la Universidad Siglo 21.

Lugar Fecha:

Firma autor-tesista		Aclaración autor-tesista
----------------------------	--	---------------------------------

Esta Secretaría/Departamento de Grado/Posgrado de la Unidad Académica:

_____certifica que la tesis adjunta es la aprobada y registrada en esta dependencia.

Firma Autoridad

Aclaración Autoridad

Sello de la Secretaría/Departamento de Posgrado