

**Universidad Siglo 21**



**INGENIERÍA EN SOFTWARE**

**Sistema de Gestión de Residuos con Contenedores Inteligentes**

**García Calzada Francisco**

**SOF01039**

**Año 2020**

## Resumen

Dentro del área céntrica de la ciudad de Córdoba, la recolección de residuos se realiza a través de contenedores. Sin embargo, podemos observar que la capacidad de estos no siempre es suficiente para contener toda la basura que se genera diariamente, esto lleva a que la basura sea depositada alrededor del contenedor provocando contaminación, olores desagradables y suciedad. Por otro lado, que tanta basura se encuentre acumulada dentro de un espacio reducido puede provocar posibles incendios debido a que entre tantos desperdicios se encuentren brasas que no fueron del todo apagadas.

En este trabajo tomamos estas problemáticas mencionadas y diseñamos un prototipo de software que brinda una posible solución. Esta consiste en instalar en la tapa de cada contenedor un dispositivo que haciendo uso del internet de las cosas los conecta a cada uno de ellos a una central y que además permite medir el nivel de basura, temperatura, humedad y presencia de fuego. Luego nuestro software toma toda esta información y la muestra en una interfaz web, permitiendo de esta manera poder diagramar mejores rutas para la recolección, reforzar contenedores que se ven saturados de basura e incluso poder alertar de aquellos que se están prendiendo fuego.

Palabras clave: contenedor inteligente, basura, internet de las cosas

## **Abstract**

In the downtown area of the city of Cordoba, waste collection is done through containers. However, we can observe that the capacity of these is not always enough to contain all the garbage that is generated daily, this leads to the garbage being deposited around the container causing contamination, unpleasant smells and dirt. On the other hand, the fact that so much garbage is accumulated within a reduced space can cause possible fires due to the fact that among so much garbage there are embers that were not completely extinguished.

In this work we took these problems mentioned and designed a software prototype that provides a possible solution. This consists of installing a device on the lid of each container that, using the Internet of things, connects each one of them to a central unit and also allows us to measure the level of garbage, temperature, humidity and the presence of fire. Then our software takes all this information and shows it in a web interface, allowing this way to diagram better routes for collection, reinforce containers that are saturated with garbage and even be able to alert those who are setting fire.

Keywords: smart bin, garbage, internet of things

# Índice

<b>Título</b> .....	<b>8</b>
<b>Introducción</b> .....	<b>8</b>
Antecedentes .....	8
Descripción del área problemática .....	10
<b>Justificación</b> .....	<b>11</b>
<b>Objetivo general</b> .....	<b>12</b>
<b>Objetivos Específicos</b> .....	<b>12</b>
<b>Marco Referencial</b> .....	<b>12</b>
Dominio del Problema .....	12
Actividad del cliente.....	15
TICs.....	15
Conceptos básicos .....	15
Plataforma .....	16
Hardware .....	17
Entorno de desarrollo integrado.....	18
Controlador de placa Arduino .....	18
Librerías.....	18
Herramientas .....	19
Base de datos.....	19
Web server.....	20
Lenguajes de programación .....	20

Competencias .....	21
<b>Diseño metodológico .....</b>	<b>22</b>
Elección de la metodología .....	22
Herramientas para el desarrollo .....	23
Recolección de datos .....	23
Planificación de actividades .....	24
<b>Relevamiento.....</b>	<b>25</b>
Relevamiento estructural .....	25
Relevamiento funcional.....	27
<b>Proceso de negocios .....</b>	<b>29</b>
<b>Diagnóstico y propuesta.....</b>	<b>30</b>
<b>Objetivos, Límites y Alcances del prototipo.....</b>	<b>32</b>
Objetivos del prototipo.....	32
Límites.....	32
Alcance.....	32
<b>Descripción del sistema.....</b>	<b>33</b>
Product Backlog.....	33
Historias de usuario.....	34
Sprint Backlog.....	40
Diagrama de clase.....	42
Diagrama de entidad-relación.....	43
Prototipos de interfaces de pantalla.....	44
Diagrama de arquitectura.....	47
<b>Seguridad .....</b>	<b>49</b>
<b>Análisis de costos .....</b>	<b>50</b>

<b>Análisis de riesgos .....</b>	<b>53</b>
<b>Conclusiones .....</b>	<b>54</b>
<b>Demo .....</b>	<b>55</b>
<b>Referencias .....</b>	<b>56</b>
<b>Anexo I: Modelo de entrevista para el relevamiento de datos .....</b>	<b>59</b>

## Índice de ilustraciones

Ilustración 1. Diagrama de Gantt de la planificación de actividades.....	24
Ilustración 2. Estructura interna para organización modelada.....	25
Ilustración 3. Diagrama BPMN del proceso de negocios.....	29
Ilustración 4. Diagrama de Clases.....	42
Ilustración 5. Diagrama de Entidad Relación.....	43
Ilustración 6. Prototipo de interfaz gráfica.....	44
Ilustración 7. Prototipo de interfaz gráfica.....	45
Ilustración 8. Prototipo de interfaz gráfica.....	46
Ilustración 9. Diagrama de arquitectura.....	48

## Índice de tablas

Tabla 1. Comparación de funcionalidades entre competencias.....	21
Tabla 2, 3. Relevamiento funcional.....	27, 28
Tabla 4,5. Diagnóstico.....	30
Tabla 6. Product Backlog.....	33
Tabla 7-15. Historia de usuario.....	34-40
Tabla 16-19. Sprint Backlog.....	40, 41
Tabla 20. Costos de desarrollo.....	50
Tabla 21. Costos operativos.....	51, 52
Tabla 22. Análisis de riesgo.....	53, 54

# **Título**

Sistema de Gestión de Residuos con Contenedores Inteligentes.

## **Introducción**

Desde hace más de 10 años, los barrios céntricos de la Ciudad de Córdoba disponen de contenedores de basura para facilitar el trabajo de la recolección, sin embargo, hasta el momento no se cuenta con un sistema que permita optimizar el trabajo.

Los contenedores en determinadas zonas o momentos del día no dan abasto con su capacidad para almacenar toda la basura que se genera y la única solución hasta el momento es esperar a que el camión de la basura realice su recorrido habitual para recogerla. Por otro lado los incendios de las cajas compactadoras de los camiones suele ocurrir debido a que en la basura que se tira se encuentran restos de brasas que no fueron correctamente apagadas poniendo en riesgo al personal a cargo de la tarea y al estado del camión.

Este trabajo propone un sistema, que haciendo uso del Internet de las Cosas, conecte a todos los contenedores y de esta manera monitorear los niveles de basura de cada uno de ellos en todo momento, como así también notificar la presencia de fuego y de esta manera evitar posibles incendios.

### *Antecedentes:*

El primer antecedente que se tiene sobre la recolección de residuos en la ciudad fue el 12 de febrero de 1785, en ese entonces el intendente Rafael de Sobremonte prohibió a los vecinos que arrojaran la basura a las calles y propuso un sistema en el cual los vecinos tenía que arrojar la basura en tachos ubicados en la vereda y estos eran vaciados por empleados municipales en carros tirados por caballos o burros y la basura era depositada en los cauces del arroyo La Cañada.



Recién en la década de 1970 se replanteó el servicio. La recolección pasó a ser llevada a cabo con camiones y a través de una empresa privada llamada Cooperativa de Servicios y Provisión de Camiones Córdoba Limitada. El servicio era prestado por unos 150 operarios y trabajó hasta el 28 de febrero de 1978. Un día después tomó el servicio la empresa Venturino que estaba a cargo de la recolección de residuos en la ciudad de La Plata pero fue por un período corto, hasta el 30 de Noviembre de 1981, cuando la sucedió Aseo.

Aseo estaba conformada por una empresa norteamericana Waste Management Internacional y Sideco, del grupo Macri, contaba con 1.100 empleados y más de 50 camiones y brindó un servicio que es recordado por muchos por su eficiencia y calidad pero para el momento ese presupuesto significaba el 50% de la ciudad y provocó que en 1984, el intendente Ramón Mestre rescindiera su contrato. El 25 de Julio de ese mismo año se contrató de forma directa a Guambaré, una empresa que recolectaba residuos en Villa Carlos Paz y en ciudades del conurbano bonaerense pero por deficiencia en la prestación del servicio se contrató otra empresa, Chietti, para reforzar el servicio.

En 1986, hubo una nueva licitación y la ganó Clima, empresa conformada por el grupo Roggio y por Hormas, esa empresa operó 12 años en Córdoba. El grupo Roggio ganó nuevamente la licitación en 1998, esta vez con Cliba, facturando en los primeros años 3 millones de dólares por mes. En el año 2009, luego de una licitación fallida, se retira Cliba y se traspasa el servicio a la empresa municipal Crese con un presupuesto de 3.7 millones de dólares por mes, según la cotización de ese año y se empieza aplicar el uso de contenedores de basura dispuesto en las zonas céntricas de la ciudad para optimizar la recolección y la recolección diferenciada para poder reciclar.

En el año 2012 Crese tercerizó el servicio a Lusa y Cotreco, pasando el presupuesto a valer 6 millones de dólares por mes pero solo para la recolección ya que para el barrido se creó el Esop municipal.

A partir del 1 de diciembre de 2018 y hasta la actualidad la recolección de residuos en la ciudad pasó a estar en manos de tres empresas, Lusa, Urbacor y LAM cubriendo la zona norte, sur y céntrica de la ciudad. (La Voz del Interior, 2018)

*Descripción del área problemática:*

Como expresa la Municipalidad de Córdoba, el manejo de los residuos sólidos urbanos representa uno de los retos más importantes para la gestión urbana.

La ciudad de Córdoba produce 35.500 toneladas de residuos domiciliarios por mes y través del Ente de Servicios y Obras Públicas (ESyOP), procesa alrededor de 70.000 toneladas por mes, generados en la capital y en las localidades del Área Metropolitana. (Córdoba, Higiene Urbana, 2018)

Según la ordenanza municipal 12.647 que establece el Marco Regulatorio de Residuos y sus reglamentaciones:

El área del microcentro de Córdoba, junto a los barrios cerrados y otras entidades como hospitales, escuelas, centros de participación comunal, etc. Disponen de contenedores de basura, evitando de esta manera la recolección casa por casa y haciendo un uso eficiente de la recolección de residuos dentro de las áreas con mayor densidad poblacional. (Ordenanza 12649, 2017)

Ocurre que los contenedores en más de una ocasión se encuentran vacíos o con muy poco nivel de basura, lo que provoca que el recorrido que hace el camión de recolección sea ineficiente y esto se traduce en pérdidas tanto económicas como de tiempo. Cuando se habla de pérdidas económica se hace referencia a ciertos aspectos como un mal uso del combustible para el camión, del pago al personal para llevar a cabo dicha tarea e incluso del mantenimiento que conlleva el uso de la flota para sus respectivas tareas. Cuando se habla de pérdidas de tiempo se hace referencia a la cantidad de horas que son necesarias para llevar a cabo dicha tarea y que tiene una relación directa con lo económico.

Por otro lado, ocurre que entre toda la basura que se acumula en los contenedores, no es extraño que se encuentren brasas que no están del todo apagadas y que sin darnos cuenta producen calor y cuando el contenedor es recolectado por el camión de basura en algún momento este se prende fuego y es muy difícil poder apagarlo, produciendo grandes pérdidas económicas y poniendo en riesgo a todo el personal involucrado.

## **Justificación**

Las empresas que están a cargo de la recolección de residuos en la ciudad de Córdoba, carecen de un sistema eficiente que permita detectar los niveles de basura en los contenedores y posibles emisiones de humo que indiquen que los residuos depositados se están prendiendo fuego.

El desarrollo de este trabajo, permitió crear una herramienta que satisface la necesidad de poder controlar en tiempo real los niveles de basura en los contenedores y de esta manera poder diagramar eficientemente los recorridos para su recolección y por otro lado poder evitar posibles incendios detectando las emisiones de humo dentro de estos contenedores antes de que estos sean recolectados.

También se incorpora como elemento innovador el concepto del internet de las cosas, ya que lo que se hizo fue conectar los contenedores de basura que están distribuidos en diferentes partes de la ciudad con internet y de esta manera se obtienen datos como, por ejemplo, cuáles son los sectores que más basura producen, si las personas cumplen con los horarios para sacar la basura o si alcanzan la cantidad de contenedores para hacer frente a la cantidad de basura que se descarta. Se obtienen datos los cuales permiten generar políticas al respecto.

## **Objetivo general del trabajo**

Analizar, diseñar y desarrollar un prototipo de sistema de información que permita visualizar los niveles de basura de cada contenedor como así también poder detectar la presencia de fuego que indique un posible incendio en cada uno de ellos.

## **Objetivos específicos del trabajo**

- Identificar cada contenedor y su ubicación dentro de la ciudad.
- Interpretar los distintos niveles de basura dentro del contenedor.
- Detectar y notificar las emisiones de fuego dentro del contenedor.

## **Marco Teórico Referencial**

*Dominio del problema:*

*Residuo:*

Según la Ordenanza Municipal N° 9612 que tiene como objeto regular, en el ámbito del Municipio de la Ciudad de Córdoba, la generación, manipulación, operación, transporte, tratamiento y disposición final de las distintas categorías de residuos, desechos o desperdicios, en su segunda sección lo define o entiende como:

A) Al material que se desecha tras cualquier proceso de tratamiento convencional, lo que queda del metabolismo de los organismos vivos y de la utilización o descomposición de materias vivas o inertes y de la transformación de energía. B) Todas aquellas materias resultantes de actividades que no constituyan el producto a elaborar u obtener y que el poseedor no utilizará nuevamente en el proceso productivo. C) Todo objeto, energía o elemento en estado sólido, líquido o gaseoso rechazado, descartado o desechado remanente de cualquier proceso,

actividad, uso descomposición, transformación, tratamiento o destrucción de materia o energía, que carece de utilidad o valor para el poseedor o dueño. D) Todo tipo de sobras, restos, sedimentos, desperdicios, basuras, cenizas, virutas, detritus o heces. E) Cualquier otro material descrito como residuo por la legislación nacional o provincial vigente”. (Ordenanza 9612, 1997)

#### *Tipos de residuos:*

Los residuos se clasifican en:

- RESIDUOS CONVENCIONALES: Los originados en los procesos o actividades de consumo en el desarrollo de las actividades humanas, que, por su composición, calidad, volumen o peso, se determinen como tales.
- RESIDUOS NO CONVENCIONALES: Los que pueden causar daño a la salud humana o al Medio Ambiente, por si mismos o al entrar en contacto con otros residuos.
- RESIDUOS DOMICILIARIOS: los desechos sólidos generados en el desarrollo de las actividades humanas en sus hogares, incluidos los inorgánicos reciclables y los residuos sólidos que se encuentren en la vía pública o inmuebles del dominio público del Municipio, tales como calles, áreas peatonales, plazas, paseos y lugares de esparcimiento público.
- RESIDUOS INORGÁNICOS RECICLABLES: Se denomina así a todo residuo no degradable o medianamente degradable susceptible de ser recuperado, acondicionado o reutilizado después de ser sometido a un determinado proceso.
- RESIDUOS ORGÁNICOS: Aquellos desechos susceptibles de degradación a través de la actividad de organismos biológicos.
- RESIDUOS PATÓGENOS: Todos aquellos desechos en estado sólido, semisólido, líquido o gaseoso, que presentan características de toxicidad o actividad biológica y que por sus condiciones puedan ser vehículo de enfermedades, y sean generados en cualquier lugar. Asimismo, los definidos como tales por la legislación provincial, nacional o internacional aplicable.

- RESIDUOS INDUSTRIALES: Todos los elementos sólidos, líquidos, barros, orgánicos e inorgánicos, en tambores o granel, provenientes de las actividades industriales o de servicios público o privado.
- RESIDUOS PELIGROSOS: Todo material desechado que pueda causar daño, efectivo o potencial, directa o indirectamente a seres vivos, poner en peligro su vida o salud o contaminar el aire, agua, suelo o el ambiente en general, por que presenta características peligrosas y biológicamente perjudiciales.
- RESIDUOS DOMICILIARIOS ESPECIALES: Aquellos desechos domiciliarios sólidos, líquidos o gaseosos, susceptibles de provocar daños a la salud de las personas o al medio ambiente. (Ordenanza 9612, 1997)

*Contenedor de basura:*

Según la Real Academia Española, es un embalaje metálico grande y recuperable, de tipos y dimensiones normalizados internacionalmente y con dispositivos para facilitar su manejo. Recipiente amplio para depositar residuos diversos. (Real Academia Española, s.f.)

*Características que debe cumplir un contenedor de basura:*

Los contenedores de basura deben tener una capacidad máxima de hasta 1100 litros, ser aptos para su elevación por sistema de carga, tener muñones sobre placa fijada a soporte interno, ser fabricado de acuerdo a las normas internacionales fijadas por su fabricante, ser de polietileno de alta densidad, con protección contra los rayos ultra violetas, resistente a golpes y temperaturas extremas, ser construido sin aristas peligrosas, tener oculto tornillos, pernos, bulones, etc., para evitar accidentes o actos de vandalismo, tener paredes lisas y redondas para evitar acumulación de residuos y debe contar con válvulas de desagote, tener ruedas con banda de caucho de 200 mm de diámetro como mínimo para amortiguar el ruido y facilitar su traslado y contar con un sistema de bloqueo de las ruedas. (Ordenanza 12649, 2017)

### *Actividad del cliente:*

Para este trabajo se definió como cliente una organización modelada basada en empresas de gestión de residuos. Cuando hablamos de gestión de residuos lo entendemos como:

El proceso de planificación, desarrollo, evaluación y regulación de las actividades que se producen desde su generación hasta su disposición final, considerando las particularidades de las distintas etapas y las diferentes categorías de residuos, disminuyendo el impacto ambiental inherente a ellos, cuyo control está a cargo de la autoridad de aplicación". (Ordenanza 9612, 1997) .

### *Tecnologías de la información y comunicación*

#### **Conceptos básicos:**

#### *Internet de las cosas (IoT):*

Es una red de objetos, cada uno de ellos integrados con sensores, que están conectados a internet. (Institute of Electrical and Electronics Engineers)

Una red que interconecta objetos físicos y virtuales valiéndose de Internet. Tales dispositivos utilizan software embebido, que le permite no solo la conectividad a Internet, sino que además brindan servicios en función de acciones dictadas remotamente las cuales pueden ser resultado de eventos específicos o del aprendizaje de la información recibida. En resumen, se trata de la interconexión digital de los objetos. (Ministerio de Modernización)

#### *Web server*

Un servidor web es un software o hardware que utiliza HTTP (Protocolo de transferencia de hipertexto) y otros protocolos para responder a las solicitudes de los clientes realizadas a través de la World Wide Web (WWW). El software del servidor web controla cómo un usuario accede a los archivos alojados. Se accede a través de los nombres de dominio de los sitios web y garantiza la entrega del contenido del sitio al usuario solicitante. Como hardware, un servidor web es una computadora que contiene software de servidor web y otros archivos relacionados

con un sitio web, como documentos HTML, imágenes y archivos JavaScript. El hardware del servidor web está conectado a Internet y permite el intercambio de datos con otros dispositivos conectados. (Techtarget, 2019)

### *Arduino:*

Arduino es una plataforma electrónica de código abierto basada en hardware y software fáciles de usar.

Arduino nació en el Instituto de Diseño de Interacción Ivrea como una herramienta fácil para la creación rápida de prototipos, dirigida a estudiantes sin experiencia en electrónica y programación. (Arduino, Arduino, s.f.)

### **Plataforma**

#### *Sigfox*

Sigfox fue fundado por Ludovic Le Moan y Christophe Fournet en 2010 con la visión de conectar cada objeto de nuestro mundo físico al universo digital. Los dos fundadores franceses construyeron una red global dedicada a Internet de las cosas basada en datos de baja potencia y largo alcance que ofrece un servicio de conectividad de extremo a extremo. (Sigfox, s.f.)

Sigfox ofrece una solución de comunicaciones basada en software, donde toda la complejidad de la red y la informática se gestiona en la nube, en lugar de en los dispositivos. Todo eso junto, reduce drásticamente el consumo de energía y los costos de los dispositivos conectados. (SigFox, s.f.)



## **Hardware**

Arduino MKR Fox 1200: es una placa potente que combina la funcionalidad de la conectividad Zero y SigFox. Es la solución ideal para los fabricantes que desean diseñar proyectos de IoT con una experiencia previa mínima en redes con un dispositivo de baja potencia. Puede funcionar durante más de seis meses con 2 baterías AA de 1.5V con un uso típico. El diseño incluye la capacidad de alimentar la placa usando dos baterías AA o AAA de 1.5V o 5V externos. El cambio de una fuente a otra se realiza automáticamente. (Arduino, Arduino MKR FOX 1200, s.f.)

Mini microswitch: son interruptores que necesitan muy poca cantidad de fuerza (presión) para funcionar y a una velocidad rápida. (Components 101, s.f.)

HC-SR04 sensor ultrasónico: es un sensor que usa un sonar para determinar la distancia a un objeto como lo hacen los murciélagos o los delfines. Ofrece una excelente detección de rango sin contacto con alta precisión y lecturas estables en un paquete fácil de usar, de 2 cm a 400 cm y su funcionamiento no se ve afectado por la luz solar o material negro. (Cytron Technology)

DHT11 sensor de temperatura y humedad: es un sensor de temperatura y humedad con una salida de señal digital calibrada, que garantiza una alta confiabilidad y una excelente capacidad de resistencia a largo plazo. (AOSONG)

KY-026 sensor de fuego: es un sensor de llama para Arduino que detecta la luz infrarroja emitida por el fuego, este módulo tiene salidas digitales y analógicas y un potenciómetro para ajustar la sensibilidad. Es de uso general en sistemas de detección de fuego. (Arduino Modules, s.f.)

Raspberry Pi 3 modelo B: Raspberry Pi es la tercera marca de computadoras más vendida en el mundo. Es una computadora del tamaño de una tarjeta de crédito que se conecta con un

televisor o pantalla, y un teclado y mouse. Puede ser usada para aprender codificación y construir proyectos electrónicos, y para muchas de las cosas que hace la PC de escritorio, como hojas de cálculo, procesamiento de textos, navegación en Internet y juegos. (Raspberry Pi, Raspberry Pi Documents, s.f.)

### ***Entorno de desarrollo integrado***

Arduino IDE: es un editor de código abierto para Arduino que hace que sea fácil de escribir código y subirlo a cualquier tipo de placa. Esta desarrollado en Java y se encuentra disponible para Windows, Mac OS y Linux. (Arduino, Arduino Software, s.f.)

### ***Controlador de placa Arduino***

Arduino SAMD boards: es un controlador que debe ser instalado dentro del Arduino IDE y que permite leer entradas (luz en un sensor, un dedo en un botón, un mensaje de twitter) y convertirlo en una salida, como por ejemplo activar un motor, encender un led, publicar algo en línea. Permite decirle a la placa qué hacer enviando un conjunto de instrucciones al micro controlador de la placa.

### ***Librerías***

Arduino Low Power: es una biblioteca que permite utilizar las funciones de baja potencia que vienen con las placas Arduino. (Arduino, Arduino Low Power, s.f.)

Arduino Sigfox para MKR Fox 1200: esta librería permite conectar la placa MKR Fox 1200 con la plataforma SigFox y realizar operaciones de alto nivel, facilitando la integración. (Arduino, Arduino SigFox, s.f.)

DHT sensor library: esta librería permite poder trabajar con el sensor DHT 11 de temperatura y humedad. (Arduino, Arduino DHT, s.f.)

Adafruit Unified Sensor Driver: es una librería que permite crear una capa de abstracción de sensor unificada utilizada por muchas bibliotecas de sensores, permitiendo de esta manera que si alguna pieza deja de ser fabricada o los requisitos de un sistema cambian y haya que

cambiar algún sensor, este pueda ser integrado fácilmente al sistema existente. (Arduino, Arduino AdaFruit, s.f.)

RTCZero: es una librería que nos permite utilizar el reloj de tiempo real en nuestra placa Arduino MKR Fox 1200 y de esta manera realizar un seguimiento de la hora actual para programar acciones en un momento determinado. (Arduino, Arduino RTC, s.f.)

### ***Herramientas***

Node-Red: es una herramienta de programación para conectar dispositivos de hardware, API y servicios en línea. Principalmente, es una herramienta visual diseñada para Internet de las cosas, pero también se puede usar para otras aplicaciones para ensamblar muy rápidamente flujos de varios servicios. Es de código abierto y fue creado originalmente por la organización IBM Emerging Technology. Permite a los usuarios unir servicios web y hardware mediante la sustitución de tareas comunes de codificación de bajo nivel (como un servicio simple que se comunica con un puerto serie), y esto se puede hacer con una interfaz visual de arrastrar y soltar. En este caso la herramienta se utiliza para obtener los datos de SigFox y enviárselos al servidor que se va a encontrar alojado en la Raspberry. (Open Source, 2017)

### ***Base de datos***

MariaDB: es uno de los servidores de bases de datos más populares del mundo y está hecho por los desarrolladores originales de MySQL. Convierte los datos en información estructurada en una amplia gama de aplicaciones, desde banca hasta sitios web. Es un reemplazo mejorado y directo para MySQL. MariaDB se utiliza porque es rápido, escalable y robusto, con un rico ecosistema de motores de almacenamiento, complementos y muchas otras herramientas que lo hacen muy versátil para una amplia variedad de casos de uso.

MariaDB está desarrollado como software de código abierto y como una base de datos relacional que proporciona una interfaz SQL para acceder a los datos. (MariaDB, s.f.)

## ***Web server***

Apache: es un software de creación, implementación y administración de servidores web de código abierto. Desarrollado inicialmente por un grupo de programadores de software y ahora es mantenido por la Apache Software Foundation. Está diseñado para crear servidores web que tienen la capacidad de alojar uno o más sitios web basados en HTTP. Las características notables incluyen la capacidad de soportar múltiples lenguajes de programación, secuencias de comandos del lado del servidor, un mecanismo de autenticación y soporte de bases de datos. (TechoPedia, 2017)

## ***Lenguajes de programación***

PHP: es un lenguaje de secuencias de comandos web incrustado en HTML. Esto significa que el código PHP se puede insertar en el HTML de una página web. Cuando se accede a una página PHP, el servidor PHP en el que reside la página lee o "analiza" el código PHP. El resultado de las funciones de PHP en la página generalmente se devuelve como código HTML, que el navegador puede leer. Debido a que el código PHP se transforma en HTML antes de cargar la página, los usuarios no pueden ver el código PHP en una página. Esto hace que las páginas PHP sean lo suficientemente seguras como para acceder a bases de datos y otra información segura. El objetivo del lenguaje es permitir a los desarrolladores web escribir páginas generadas dinámicamente de forma rápida y sencilla. PHP también es excelente para crear sitios web basados en bases de datos. (TechTerms, s.f.)

JavaScript: es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. JavaScript no debe ser confundido con el lenguaje de programación Java. Ambos "Java" y "Javascript" son marcas registradas de Oracle en Estados Unidos y otros países. Sin embargo, los dos lenguajes de programación tienen muchas diferencias en las sintaxis, semántica y usos. (MDN, 2020)

## Competencias

A continuación, a través de una tabla comparativa se presentan dos empresas que brindan servicios a la recolección de residuos y que utilizan el Internet de las Cosas como medio para ofrecer un valor agregado, por lo que pueden ser tomados como competidores del sistema que se propone en este trabajo.

Luego se detalla una breve descripción sobre cada una de ellas, informando quiénes son, a qué se dedican y qué productos ofrecen.

	Puede acoplarse a cualquier tipo de contenedor	Detecta materiales sólidos y líquidos	Se comunica de manera inalámbrica	Soporta ambientes extremos	Detecta presencia de fuego	Fuente
Ecube Labs	Si	Si	Si	Si	No	(EcubeLabs, s.f.)
Eco-bits	No	No	SI	Si	No	(EcoBits, 2018)

Tabla 1. Comparación de funcionalidades entre las competencias. Elaboración propia

- Ecube Labs: es un fabricante de contenedores de basura inteligentes que hace uso del Internet de las Cosas para poder conectarlos y obtener información sobre cada uno de ellos. También fabrican compactadores de desechos portátiles, alimentados a través de energía solar. (EcubeLabs, s.f.)
- Eco-bits: es un fabricante de dispositivos IoT que son instalados en los contenedores de basura y que utilizando la tecnología cognitiva propuesta por IBM Watson permite conectar a cada contenedor con su aplicación Mobile o web y desde allí monitorear todos los datos brindados por estos. (EcoBits, 2018)

## Diseño Metodológico

### *Metodología*

Como punto de partida, para el desarrollo de este trabajo, se utilizaron las metodologías ágiles y Scrum como marco de trabajo.

Las metodologías ágiles o el desarrollo de software ágil es un término general para un conjunto de marcos y prácticas basadas en los valores y principios expresados en el Manifiesto para el desarrollo de software ágil y los 12 principios detrás de él (Agile Alliance, s.f.).

El marco de trabajo de Scrum es heurístico. Se basa en el aprendizaje continuo y en la adaptación a los factores fluctuantes. Reconoce que el equipo no lo sabe todo al inicio de un proyecto y evolucionará a través de la experiencia. Scrum, está compuesto por tres artefactos que son un backlog del producto, un backlog de sprint y un incremento con la definición de "finalizado".

- **El backlog del producto** es la lista de trabajo maestra que se tiene que realizar y que debe mantener el propietario del producto o el gestor del producto. Es una lista dinámica de funciones, requisitos, mejoras y correcciones que actúa como entrada para el backlog del sprint. Se trata, básicamente, de la lista de tareas del equipo.
- **El backlog del sprint** es la lista de elementos, historias de usuario o resolución de errores seleccionados por el equipo de desarrollo para implementarlos en el ciclo de sprint actual. Antes de cada sprint, en la reunión de planificación del sprint, el equipo elige en qué elementos del backlog del producto va a trabajar durante el sprint.
- **El incremento** (u objetivo del sprint) es el producto final que se puede usar y que se obtiene de un sprint.

(Atlassian, s.f.)

## *Herramientas para el desarrollo*

### ***Visual Studio Code***

Visual Studio Code es un editor de código optimizado con soporte para operaciones de desarrollo como depuración, ejecución de tareas y control de versiones. Su objetivo es proporcionar solo las herramientas que un desarrollador necesita para un ciclo rápido de creación de código-depuración y deja flujos de trabajo más complejos a IDE más completos, como Visual Studio IDE. (Visual Studio Code, 2020)

### *Recolección de datos*

Para la recolección de datos, en este trabajo se utilizaron dos técnicas. La primera fue a través de una entrevista y la segunda a partir de observaciones de campo personal.

La entrevista fue realizada a un ingeniero ambiental, que estuvo a cargo, entre otras funciones, de gestionar el Pliego de condiciones generales y particulares de la basura y que se encuentra detallado en la ordenanza municipal número 12649 de Córdoba Capital. Además, fue presidente de la empresa Crese por un breve período y actualmente sigue brindando servicios a las empresas que hoy están a cargo de la recolección de residuos de la ciudad. Gracias a estas dos técnicas se pudo recolectar información acerca de los contenedores de basura y de las empresas que hoy están a cargo del servicio, aunque para cierto tipo de información, como, por ejemplo, cómo están compuestas las empresas, cuántos empleados tienen, qué funciones cumple cada rol, entre otros., no se pudo acceder y es por eso que este trabajo es realizado para una organización modelada.

## Planificación de actividades

La planificación de actividades se puede visualizar en el siguiente diagrama de Gantt

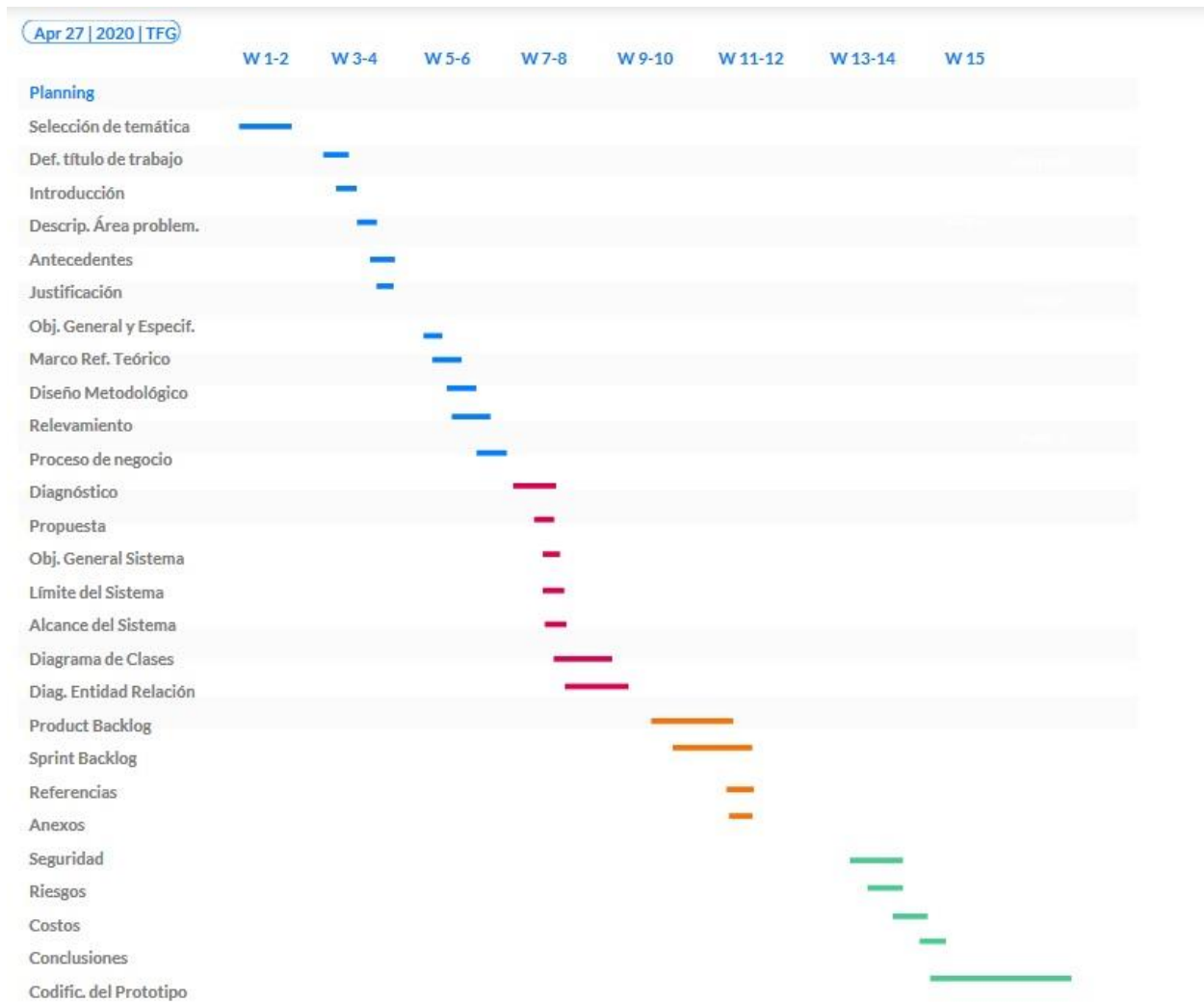


Ilustración 1. Diagrama de Gantt de la planificación de actividades. Elaboración propia.



## Relevamiento

Los siguientes relevamientos fueron desarrollados y aplicados para una organización modelada, basado sobre información obtenida en la entrevista y observaciones de campo personal, por lo tanto, cada uno de los siguientes puntos funcionará como un modelo genérico que se podrá aplicar en cualquier organización que cumpla con las características expuestas.

### *Relevamiento estructural*

La infraestructura necesaria para implementar el sistema debe cumplir las siguientes características:

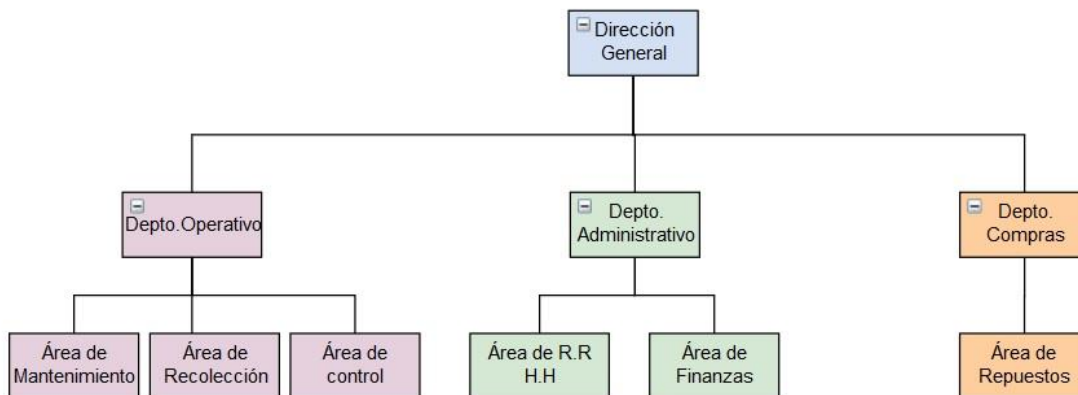


Ilustración 2. Estructura interna para organización modelada. Fuente: Elaboración propia

- **Dirección General:** es el nivel más alto dentro de la organización, aquí se encuentra el presidente y vicepresidente, desde aquí surgen las políticas que la organización debe seguir y se controlan los departamentos que se encuentran en un nivel inferior.
- **Depto. Operativo:** su función es controlar las áreas que dependen de él. Desde aquí se dictan las normas y tareas que deben seguir y respetar las demás áreas.
- **Área de Mantenimiento:** se encarga de mantener la flota de camiones operativa para cumplir con los servicios de recolección de residuos.
- **Área de Recolección:** se encarga de realizar la recolección de residuos.

- **Área de control:** se encarga de monitorear los contenedores de basura y diagramar las rutas de recolección.
- **Depto. Administrativo:** su función es controlar las áreas que dependen de él. Desde aquí se dictan las normas y tareas que deben seguir y respetar las demás áreas.
- **Área de R.R H.H:** se encarga de controlar los contratos de todos los empleados de la organización, como también de la incorporación o suspensión de personal.
- **Área de Finanzas:** se encarga de liquidar los haberes de todos los empleados como también de controlar y presentar los informes financieros de la empresa.
- **Depto. Compras:** su función es controlar las áreas que dependen de él. Desde aquí se dictan las normas y tareas que deben seguir y respetar las demás áreas.
- **Área de Repuestos:** se encarga de gestionar la compra de insumos necesarios de la empresa.

## Relevamiento Funcional

A continuación, se detalla el relevamiento funcional con roles y procesos dentro de una organización genérica.

### Roles

- Responsable de control (RC).
- Responsable de políticas y diagramar rutas (RPDR).
- Jefe del área de control (JAC).
- Operario de recolección (OR).

### Procesos

Nombre	Determinar niveles de basura y generar políticas al respecto.
Roles implicados	<ul style="list-style-type: none"><li>• Responsable de Control (RC).</li><li>• Responsable de políticas y diagramar rutas (RPDR).</li><li>• Jefe del área de control (JAC).</li><li>• Operario de recolección (OR).</li></ul>
Pasos	<ol style="list-style-type: none"><li>1. El OR registra en una grilla los niveles de basura de los contenedores que va recogiendo.</li><li>2. El OR al final su turno le entrega al RC la grilla.</li><li>3. El RC controla que la grilla tenga todos los datos necesarios completados y se lo entrega al RPDR.</li><li>4. El RPDR genera una copia de la grilla para archivar en su repertorio y posteriormente generar políticas al respecto y diagramar futuras rutas.</li><li>5. El RPDR le entrega la grilla original al JAC y este registra la recolección del día.</li></ol>

Tabla 2. Relevamiento funcional. Elaboración propia

Nombre	Detección de incendios en los contenedores
Roles implicados	<ul style="list-style-type: none"> <li>• Responsable de Control (RC).</li> <li>• Jefe del área de control (JAC).</li> <li>• Operario de recolección (OR).</li> </ul>
Pasos	<ol style="list-style-type: none"> <li>1. El OR detecta emisión de humo dentro del camión y verifica el estado del fuego (controlable/incontrolable) y notifica al RC.</li> <li>2. El RC le ordena al OR que descienda del camión junto con el resto del personal que se encuentra en el momento.</li> <li>3. El RC le notifica inmediatamente de la situación al JAC.</li> <li>4. Si el fuego es controlable dentro del camión el JAC le ordena al OR que se dirija al Área de Mantenimiento de la empresa para poder apagar el incendio y controlar el estado del vehículo. Si el fuego es incontrolable, llama a bomberos para que acuda al lugar donde se presenta el hecho.</li> </ol>

Tabla 3. Relevamiento funcional. Elaboración propia

## Proceso de negocio

A continuación, a través de un diagrama BPMN, se detalla los procesos que fueron relevados en el apartado anterior.

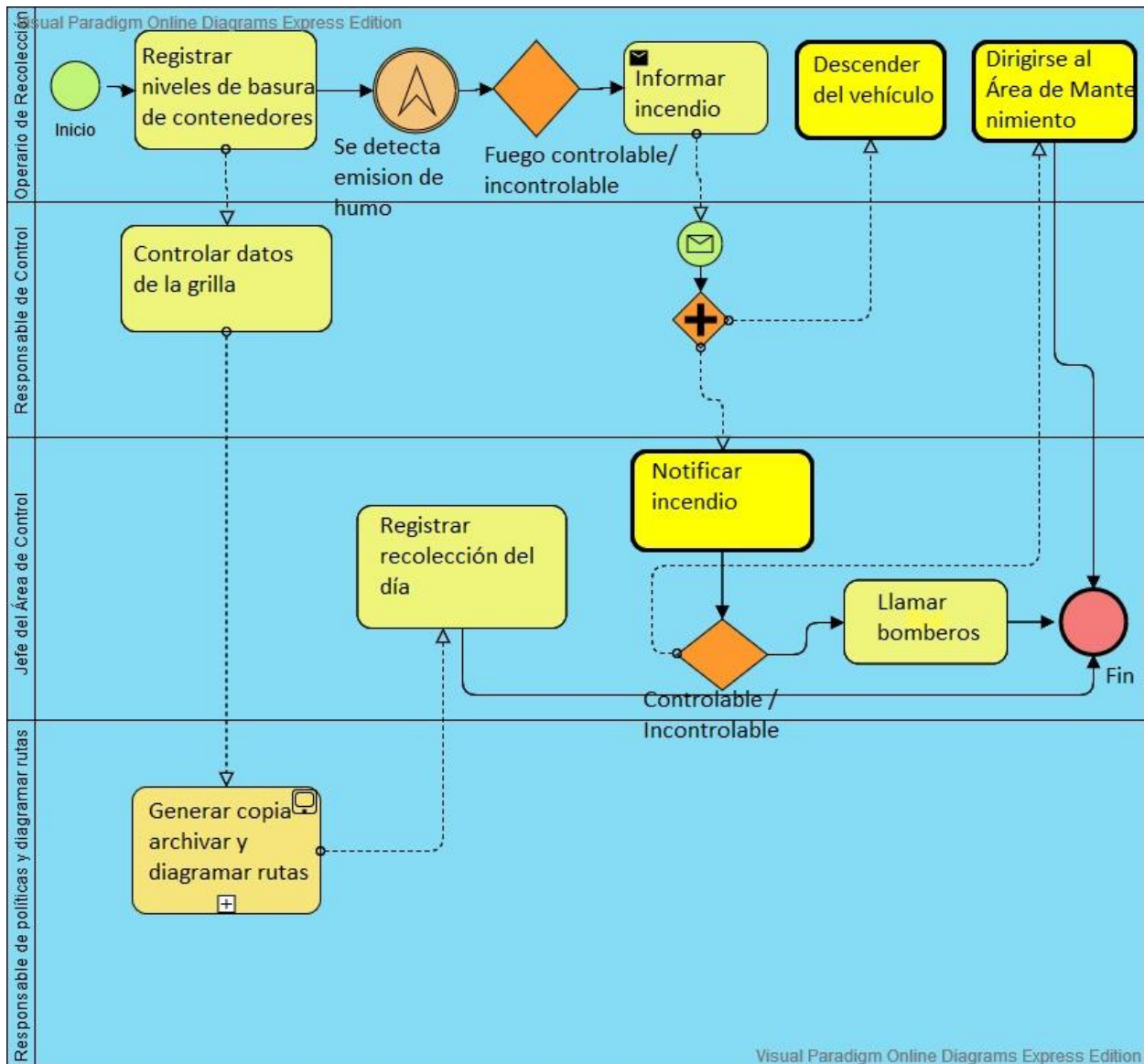


Ilustración 3. Diagrama BPMN del proceso de negocios. Elaboración propia.

## Diagnóstico y propuesta

### Diagnóstico

A partir del relevamiento realizado se detectaron los siguientes problemas.

Nombre	Determinar niveles de basura y generar políticas al respecto
Problemas	<ul style="list-style-type: none"><li>• Las rutas no siempre se completan o se realizan.</li><li>• Para obtener grilla es necesario esperar que se termine por completo la ruta y esto hace que proceso sea muy lento.</li><li>• La grilla se pierde, rompe o ensucia.</li></ul>
Causas	<ul style="list-style-type: none"><li>• Las condiciones climáticas, como las lluvias hacen que sea imposible terminar o empezar una ruta. El estado de las rutas impide que se pueda completar el recorrido. El vandalismo obliga al camión retirarse del lugar.</li><li>• La comunicación entre los diversos actores y el tiempo que consume cada recorrido hace el proceso sea demasiado lento.</li></ul>

Tabla 4. Diagnóstico. Elaboración propia

Nombre	Detección de incendios en los contenedores
Problemas	<ul style="list-style-type: none"><li>• La detección de fuego es realizada por el mismo personal que se encuentra trabajando en el camión de basura.</li></ul>
Causas	<ul style="list-style-type: none"><li>• No se cuenta con un detector de fuego dentro de los contenedores que notifiquen el riesgo.</li></ul>

Tabla 5. Diagnóstico. Elaboración propia

## *Propuesta*

A partir de los problemas detectados en el diagnóstico se propuso el desarrollo de un sistema informático que permite automatizar la medición de los distintos niveles de basura en cada contenedor como así también detectar y notificar la presencia de fuego, evitando posibles incendios.

Para que el sistema pueda realizar de manera correcta estas tareas es necesario que cada contenedor disponga un dispositivo instalado en su parte superior, en la tapa, la cual debe permanecer siempre cerrada. Este dispositivo está compuesto por un sensor de ultrasonido que permite detectar los distintos niveles de basura, por un sensor de temperatura para monitorear la misma, por un sensor de fuego que permiten la detección de posibles brasas prendidas y por una placa Arduino que haciendo uso el Internet de las Cosas permite comunicarse con el sistema a través de la plataforma Sigfox y enviar toda la información recabada por los sensores.

Luego toda esa información que almacena la placa Arduino viaja hacia la plataforma Sigfox y de esta hacia el servidor permaneciendo disponible para ser consultada por una computadora. De esta manera se evita que el operario tenga que encargarse del llenado de una grilla y solo se tenga que enfocar en su trabajo de recolección ya que el monitoreo sobre los distintos niveles de basura en cada contenedor es llevado a cabo desde la central por el responsable de control. Al mismo tiempo esta misma persona puede observar la presencia de fuego en los contenedores mientras se hacen las rutas y en caso de que algún contenedor suponga un riesgo para los operarios y el camión poder avisarles por vía sms o llamada telefónica.

## Objetivos, límites y alcances del prototipo

### *Objetivos del prototipo*

A través del uso del Internet de las Cosas conectar todos los contenedores de residuos a un sistema que detecte automáticamente y en tiempo real los distintos niveles de basura en cada uno de ellos como así también emisiones de fuego evitando posibles incendios dentro del camión de basura.

### *Límite*

El sistema comprende desde determinar los distintos niveles de basura en cada contenedor hasta la detección de posibles incendios de manera automática y en tiempo real junto con su posición geográfica.

### *Alcance*

- Detección y visualización del nivel de basura de cada contenedor.
- Detección y visualización del porcentaje de humedad en cada contenedor.
- Detección y aviso de presencia de fuego en cada contenedor.
- Ubicación geográfica de cada contenedor.

### *No contempla*

- Cálculo de la ruta más rápida para llegar a cada contenedor.
- Detección de los distintos tipos de residuos.



## Descripción del sistema

Ya con el objetivo, límite y alcance definido se procedió a crear el Product Backlog, el cual contiene una lista de las historias de usuario requeridas para la implementación del prototipo. A cada historia de usuario se la identificó con un número único, se le dio un título, descripción, criterios de aceptación, nivel de prioridad (alta, media, baja) y puntos según el nivel de complejidad.

### *Product Backlog*

N°	Título	Prioridad	Puntos	Dependencias
1	Registro de usuarios	Media	5	
2	Autenticación de usuarios	Alta	7	1
3	Configuración de sensor para detectar niveles de basura dentro del contenedor	Alta	8	
4	Configuración de sensor para detectar presencia de fuego	Alta	8	
5	Emitir señal de nivel de basura del contenedor	Alta	6	3
6	Emitir señal de presencia de fuego	Alta	6	4
7	Capturar datos desde los sensores y enviarlos a plataforma lot.	Alta	8	5, 6
8	Comunicar servidor con plataforma lot	Alta	9	7
9	Visualizar datos en tiempo real	Alta	7	2

Tabla 6. Product Backlog. Elaboración propia

## Historias de usuario

A continuación, se desarrollan todas las historias de usuario mencionadas en el Product Backlog.

<b>ID</b>	1	<b>Puntos de historia</b>	5	<b>Prioridad</b>	Media
<b>Título</b>	Registro de usuarios				
<b>Descripción</b>	Como usuario quiero registrarme para utilizar el sistema.				
<b>Criterios de aceptación</b>	<p>Escenario: Registro exitoso</p> <p>DADO un usuario que aún no se encuentra registrado en el sistema CUANDO ingresa sus datos personales, email y contraseña validados ENTONCES el sistema lo registra como nuevo usuario</p> <p>Escenario: Usuario existente</p> <p>DADO un usuario que ya se encuentra registrado en el sistema CUANDO ingresa sus datos personales, email y contraseña validados ENTONCES el sistema verifica que el usuario ya se encuentra registrado y muestra un mensaje de error</p>				

Tabla 7. Historia de usuario. Elaboración propia

<b>ID</b>	2	<b>Puntos de historia</b>	7	<b>Prioridad</b>	Alta
<b>Título</b>	Autenticación de usuarios				
<b>Descripción</b>	Como usuario quiero autenticarme en el sistema con mi usuario y contraseña para utilizarla.				

<b>Criterios de aceptación</b>	<p>Escenario: Autenticación exitosa</p> <p>DADO un usuario registrado en el sistema</p> <p>CUANDO ingresa su email y contraseña correctamente</p> <p>ENTONCES el sistema le permite acceder a sus funciones y al contenido propio del usuario</p> <p>Escenario: Autenticación fallida</p> <p>DADO un usuario registrado en la aplicación</p> <p>CUANDO ingresa su email y contraseña</p> <p>Y alguno de sus datos son incorrectos</p> <p>ENTONCES se muestra un mensaje de error</p>
--------------------------------	--

Tabla 8. Historia de usuario. Elaboración propia

<b>ID</b>	3	<b>Puntos de historia</b>	8	<b>Prioridad</b>	Alta
<b>Título</b>	Configuración de sensor para detectar niveles de basura dentro del contenedor				
<b>Descripción</b>	Como usuario quiero que el sensor HC-SR04 sea capaz de detectar los niveles de basura dentro del contenedor				
<b>Criterios de aceptación</b>	<p>Escenario: El sensor detecta correctamente el nivel de basura</p> <p>DADO un sensor ultrasónico instalado en la tapa del contenedor</p> <p>CUANDO se ingresa basura dentro del contenedor y se cierra la tapa</p> <p>ENTONCES el sensor determina el nivel de basura dentro del contenedor</p> <p>Escenario: El sensor no detecta el nivel de basura</p> <p>DADO un sensor ultrasónico instalado en la tapa del contenedor</p>				

	<p>CUANDO se ingresa basura dentro del contenedor y no se cierra la tapa</p> <p>ENTONCES el sensor no puede determinar el nivel de basura dentro del contenedor</p>
--	---

Tabla 9. Historia de usuario. Elaboración propia

<b>ID</b>	4	<b>Puntos de historia</b>	8	<b>Prioridad</b>	Alta
<b>Título</b>	Configuración de sensor para detectar presencia de fuego				
<b>Descripción</b>	Como usuario quiero que el sensor KY-026 sea capaz de detectar la presencia de fuego dentro del contenedor				
<b>Criterios de aceptación</b>	<p>Escenario: El sensor detecta correctamente presencia de fuego</p> <p>DADO un sensor de fuego instalado en la tapa del contenedor</p> <p>CUANDO se ingresa basura prendida fuego dentro del contenedor y se cierra la tapa</p> <p>ENTONCES los sensores determinan la presencia de fuego</p> <p>Escenario: El sensor no detecta la presencia de fuego</p> <p>DADO un sensor de fuego instalado en la tapa del contenedor</p> <p>CUANDO se ingresa basura prendida fuego dentro del contenedor y no se cierra la tapa</p> <p>ENTONCES no se puede determinar la presencia de fuego dentro del contenedor</p>				

Tabla 10. Historia de usuario. Elaboración propia

<b>ID</b>	5	<b>Puntos de historia</b>	6	<b>Prioridad</b>	Alta
<b>Título</b>	Emitir señal de nivel de basura del contenedor				
<b>Descripción</b>	Como usuario quiero que el sensor HC-SR04 sea capaz de emitir una señal a la placa Arduino MKR Fox 1200.				
<b>Criterios de aceptación</b>	<p>Escenario: El sensor detecta correctamente los niveles de basura</p> <p>DADO un sensor ultrasónico ya configurado e instalado</p> <p>CUANDO se ingresa basura dentro del contenedor y se cierra la tapa</p> <p>ENTONCES el sensor determina el nivel de basura dentro del contenedor y se lo comunica a la placa Arduino MKR Fox 1200</p>				

Tabla 11. Historia de usuario. Elaboración propia

<b>ID</b>	6	<b>Puntos de historia</b>	6	<b>Prioridad</b>	Alta
<b>Título</b>	Emitir señal de presencia de fuego				
<b>Descripción</b>	Como usuario quiero que el sensor KY-026 sea capaz de emitir una señal a la placa Arduino MKR Fox 1200.				
<b>Criterios de aceptación</b>	<p>Escenario: El sensor detecta correctamente la presencia de fuego</p> <p>DADO un sensor de fuego ya configurado e instalado</p> <p>CUANDO se ingresa basura prendida fuego dentro del contenedor y se cierra la tapa</p> <p>ENTONCES el sensor determina el nivel la presencia de fuego y se lo comunica a la placa Arduino MKR Fox 1200</p>				

Tabla 12. Historia de usuario. Elaboración propia

<b>ID</b>	7	<b>Puntos de historia</b>	8	<b>Prioridad</b>	Alta
<b>Título</b>	Capturar datos desde los sensores y enviarlos a plataforma lot.				
<b>Descripción</b>	Como usuario quiero que la placa Arduino MKR Fox 1200 tome todos los datos de los sensores y se comunice con la plataforma Sigfox.				
<b>Criterios de aceptación</b>	<p>Escenario: La placa Arduino MKR Fox 1200 se comunica correctamente con Sigfox</p> <p>DADO un conjunto de datos recolectados por la placa Arduino MKR Fox 1200</p> <p>CUANDO pasan 10 segundos y la placa se identifica correctamente con su dispositivo en la plataforma</p> <p>ENTONCES la placa Arduino MKR Fox 1200 envía esos datos a la plataforma</p> <p>Escenario: La placa Arduino MKR Fox 1200 no se comunica correctamente con Sigfox</p> <p>DADO un conjunto de datos recolectados por la placa Arduino MKR Fox 1200</p> <p>CUANDO pasan 10 segundos y la placa no se pudo identificar correctamente con su dispositivo en la plataforma por una mala configuración</p> <p>ENTONCES la placa Arduino MKR Fox 1200 no puede enviar datos</p>				

Tabla 13. Historia de usuario. Elaboración propia

<b>ID</b>	8	<b>Puntos de historia</b>	9	<b>Prioridad</b>	Alta
<b>Título</b>	Comunicar servidor con plataforma lot				
<b>Descripción</b>	Como usuario quiero que el servidor se comuniquen con la plataforma Sigfox				
<b>Criterios de aceptación</b>	<p>Escenario: El servidor se conecta con la plataforma Sigfox</p> <p>DADO una solicitud desde el servidor</p> <p>CUANDO la solicitud llega a la plataforma</p> <p>ENTONCES la plataforma envía los datos solicitados al servidor</p> <p>Escenario: El servidor no se conecta correctamente con la plataforma</p> <p>DADO una solicitud desde el servidor</p> <p>CUANDO la solicitud no llega a la plataforma por una mala configuración</p> <p>ENTONCES el servidor notifica con un mensaje de error.</p>				

Tabla 14. Historia de usuario. Elaboración propia

<b>ID</b>	9	<b>Puntos de historia</b>	7	<b>Prioridad</b>	Alta
<b>Título</b>	Visualizar datos en tiempo real				
<b>Descripción</b>	Como usuario quiero poder ver todos los datos recolectados en una plataforma web				
<b>Criterios de aceptación</b>	Escenario: el usuario puede ver correctamente los datos recolectados en tiempo real				

	<p>DADO un usuario que se identificó correctamente en el sistema</p> <p>CUANDO quiere acceder a los datos de los contenedores</p> <p>ENTONCES el usuario debe seleccionar el contenedor que desea ver</p>
--	---

Tabla 15. Historia de usuario. Elaboración propia

### *Sprint Backlog*

Para el primer sprint se seleccionó aquellas historias de usuario que fueron consideradas primordiales para el desarrollo del sistema.

<b>ID</b>	1	<b>Puntos de historia</b>	5	<b>Prioridad</b>	Media
<b>Título</b>	Registro de usuarios				
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Diseñar interfaz de registro de usuario</li> <li>• Diseñar base de datos de usuario</li> <li>• Diseñar y aplicar test unitario</li> </ul>				

Tabla 16. Sprint Backlog. Elaboración propia

<b>ID</b>	2	<b>Puntos de historia</b>	7	<b>Prioridad</b>	Alta
<b>Título</b>	Autenticación de usuarios				
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Diseñar interfaz de acceso de usuario</li> <li>• Diseñar y aplicar test unitario</li> </ul>				

Tabla 17. Sprint Backlog. Elaboración propia



<b>ID</b>	7	<b>Puntos de historia</b>	8	<b>Prioridad</b>	Alta
<b>Título</b>	Capturar datos desde los sensores y enviarlos a plataforma lot.				
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Programar placa Arduino MKR Fox 1200 para que tomen todos los datos de los sensores</li> <li>• Activar e instalar placa en la plataforma Sigfox</li> <li>• Diseñar y aplicar test unitario</li> </ul>				

Tabla 18. Sprint Backlog. Elaboración propia

<b>ID</b>	8	<b>Puntos de historia</b>	9	<b>Prioridad</b>	Alta
<b>Título</b>	Comunicar servidor con plataforma lot				
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Diseñar flujo node-red</li> <li>• Diseñar base de datos</li> <li>• Configurar servidor web</li> </ul>				

Tabla 19. Sprint Backlog. Elaboración propia

## Diagrama de clases

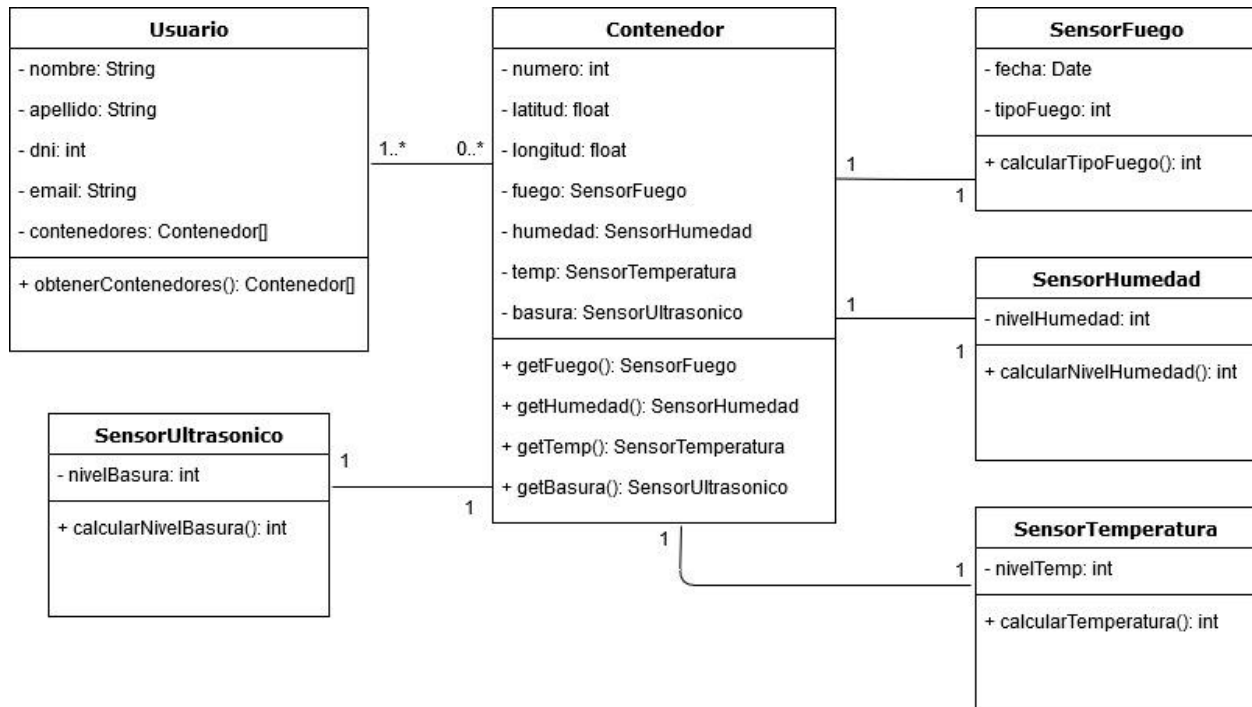


Ilustración 4. Diagrama de Clases. Elaboración propia.

## Diagrama de entidad-relación

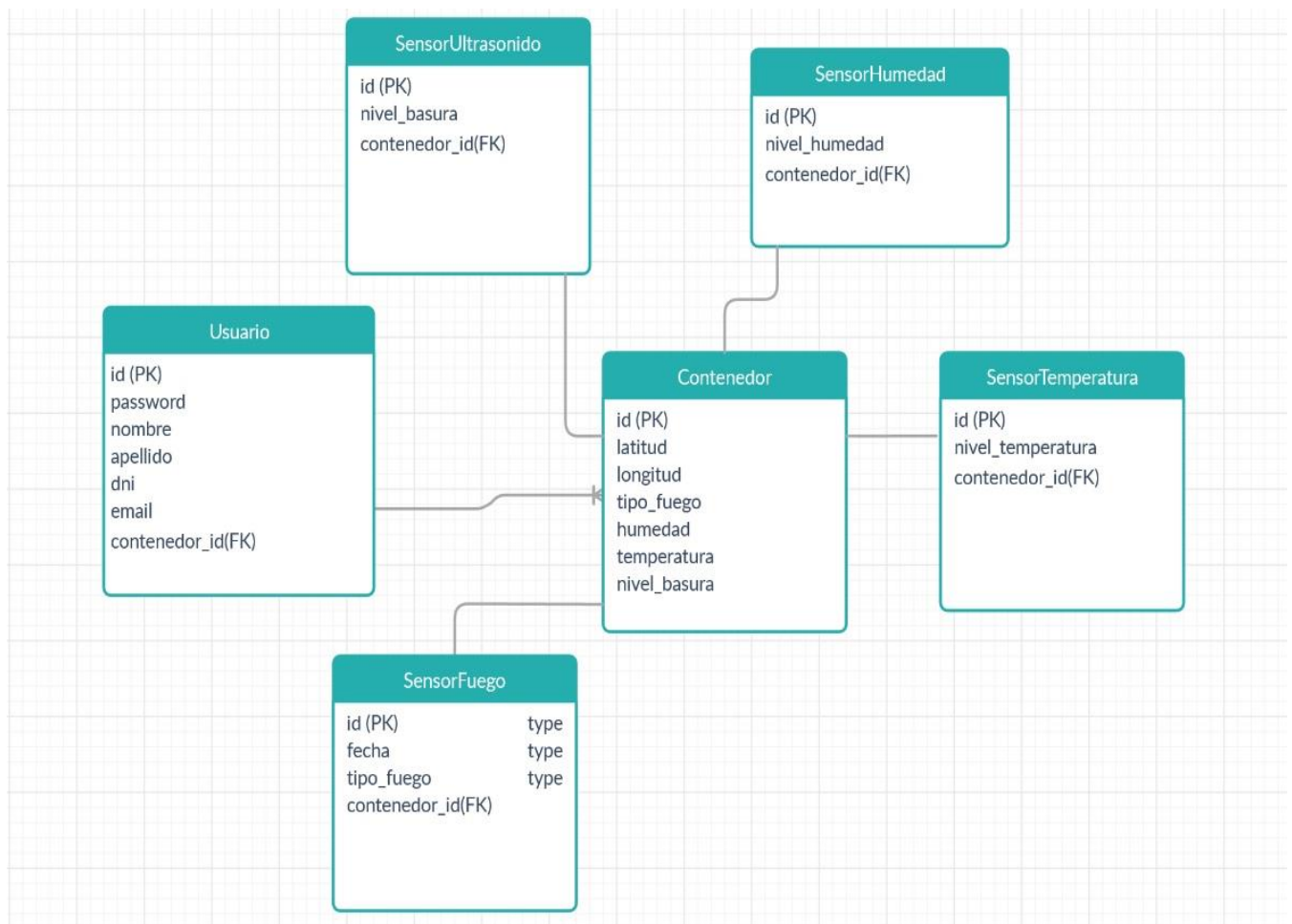


Ilustración 5. Diagrama de Entidad Relación. Elaboración propia.

## *Prototipos de Interfaces de pantalla*

A continuación, se describen una serie de escenarios junto a su prototipo de interfaz de pantalla. Cabe aclarar que dichos prototipos pueden verse sujetos a cambios durante el desarrollo o mantención del sistema pero que de ninguna manera afectan al funcionamiento del mismo.

El primer escenario, Ilustración n° 6, corresponde a la interfaz gráfica del inicio de sesión o login. Lo que se busca en este caso es respetar los patrones de UX<sup>1</sup> modernos y presentar una vista sencilla y amigable al usuario. Esta es la primera pantalla que el usuario ve al ingresar al sistema.

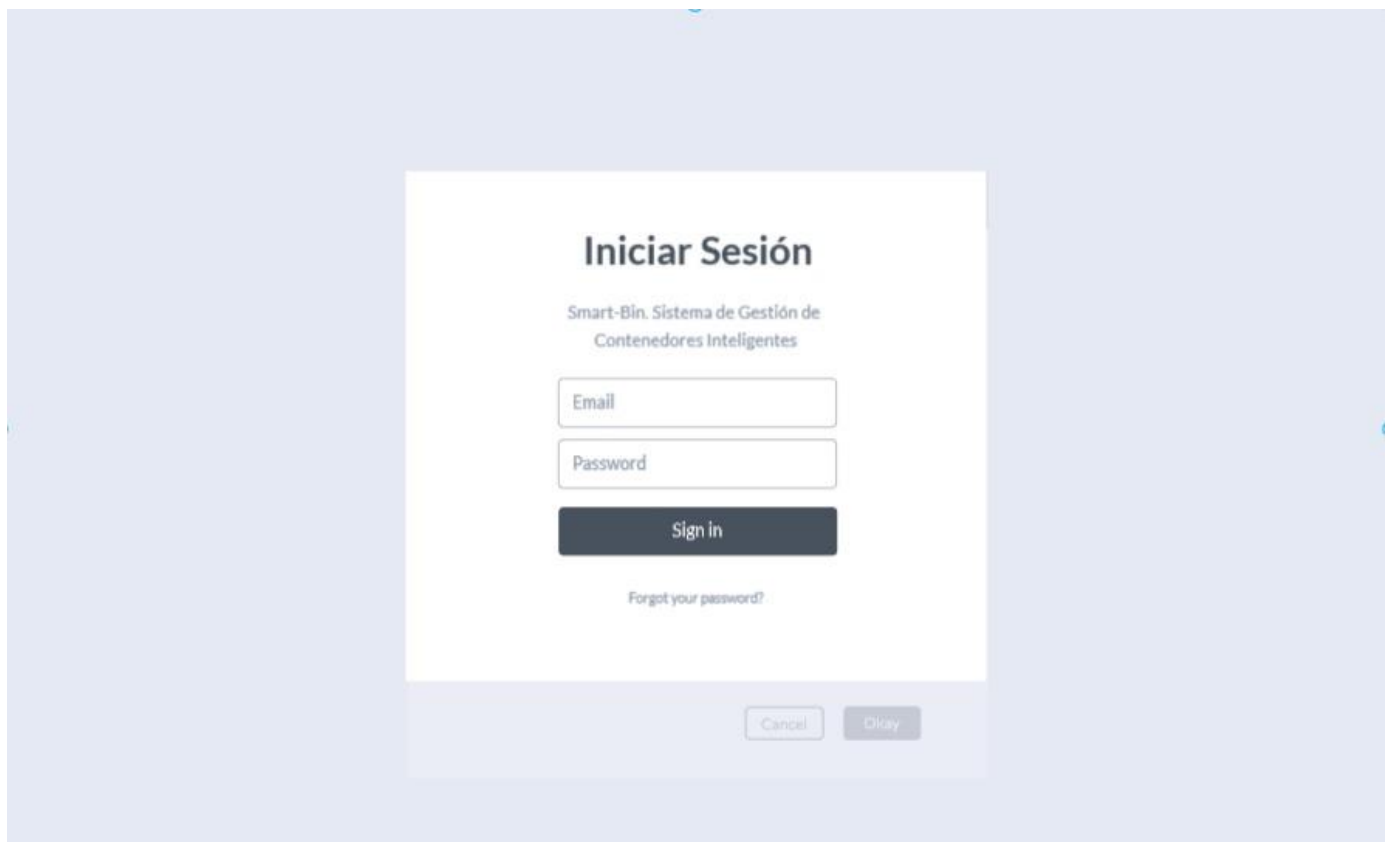


Ilustración 6. Prototipo de interfaz gráfica: login. Elaboración propia.

---

<sup>1</sup> UX: hace referencia a cómo una persona se siente mientras usa cualquier producto o servicio digital.

El segundo escenario, Ilustración n° 7, trata sobre cómo se ve la información más relevante del sistema y por lo tanto es considerado como la pantalla principal. Aquí se pueden observar los distintos contenedores que se encuentran registrados, junto a los datos recolectados en tiempo real de cada uno de ellos y son ordenados a través de una grilla. Por otro lado, también se presentan las opciones que el usuario dispone para realizar e interactuar con el sistema.

Aquí es donde más cambios se pueden presentar a nivel diseño pero que de ninguna manera afectan al curso normal del proceso.

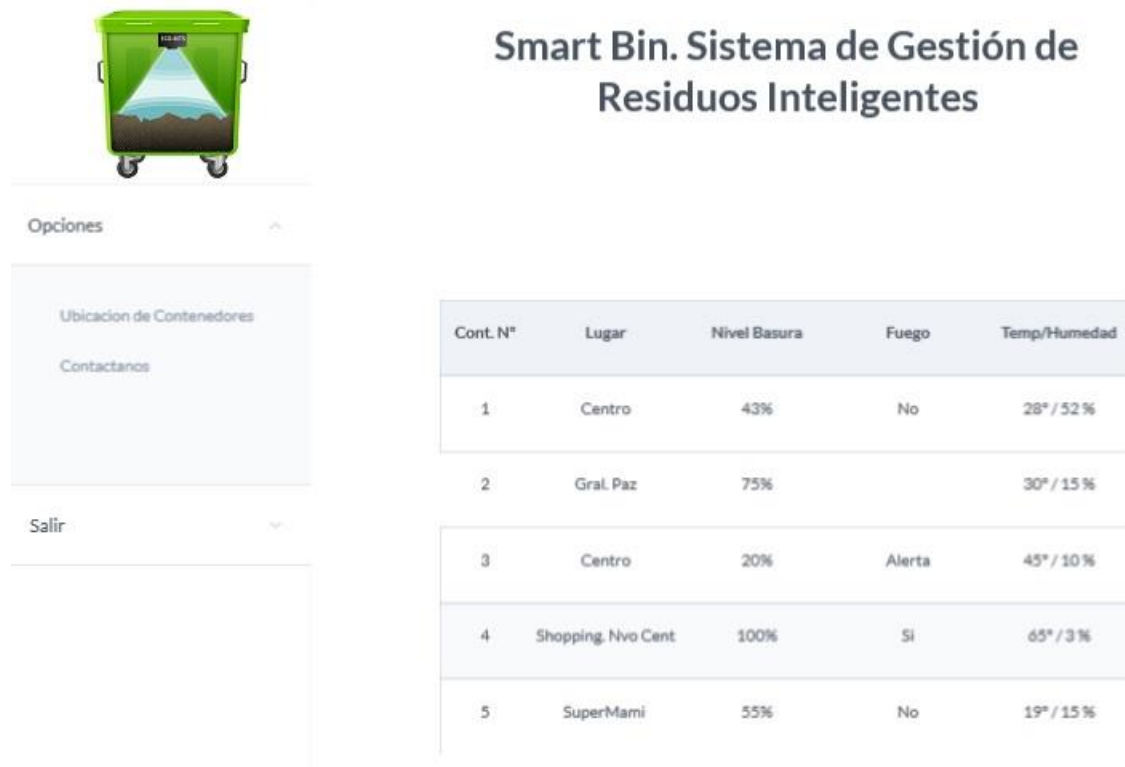


Ilustración 7. Prototipo de interfaz gráfica: pantalla principal. Elaboración propia.

Por último, tenemos el escenario en donde el usuario desea ver la ubicación geográfica del contenedor que le interesa. Si bien para este caso puede que ocurra algún cambio de diseño siempre se respeta el diseño del mapa, ya que este es un servicio brindado por otra empresa tecnológica y que por lo tanto no permite modificaciones sobre ella.

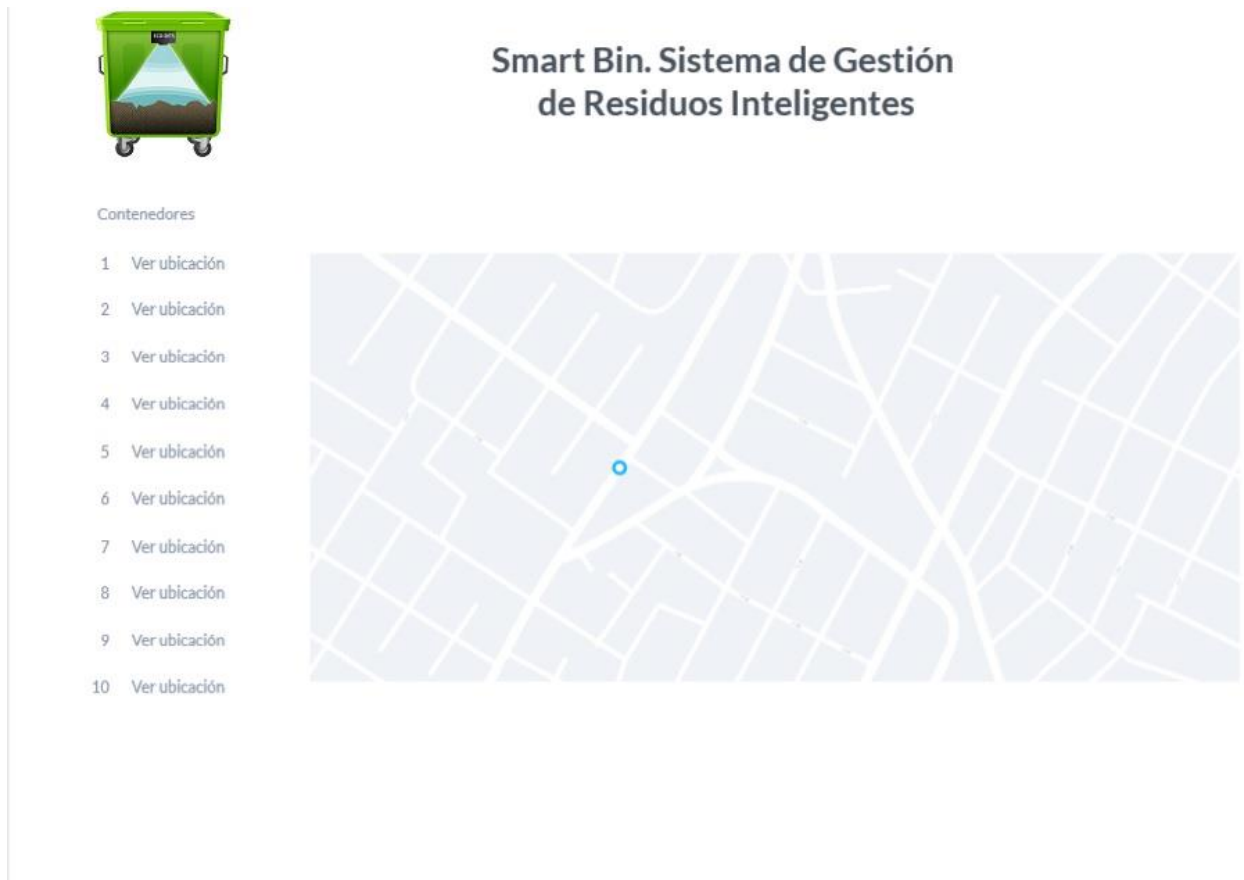


Ilustración 8. Prototipo de interfaz gráfica: ubicación de contenedores. Elaboración propia.

## *Diagrama de arquitectura*

A continuación, se muestra y se detalla a través de una ilustración el diagrama de arquitectura del sistema.

Todo comienza desde el contenedor de basura, aquí se encuentra el dispositivo que se encarga de recolectar toda la información que nos interesa, está compuesto por:

- Placa Arduino MKR Fox 1200: cuya función será recolectar, procesar y almacenar toda la información proveniente de los demás sensores.
- Mini microswitch: que nos ayuda a saber cuándo la tapa del contenedor está cerrada y que es condición para que todo lo demás funcione.
- HC-SR04: sensor ultrasónico que nos permite saber el nivel de basura dentro del contenedor.
- DHT11: sensor de temperatura y humedad que nos ayuda a interpretar si existe un posible riesgo de incendio dentro del contenedor.
- KY-026: sensor de fuego que junto con el DHT11 son los que nos determina la presencia o no de fuego.

Una vez que la placa Arduino MKR Fox 1200 obtiene todos los datos de los demás sensores, que conforman el dispositivo que va instalado en la tapa del contenedor, este envía todos los datos de manera inalámbrica a la plataforma Sigfox. Dentro de esta plataforma tenemos registrados cada contenedor junto a sus datos, esto lo logramos gracias a que cada placa Arduino es tomada y registrada como un solo objeto y que por lo tanto almacena solo sus datos. Luego desde la plataforma Sigfox podemos conectarnos a nuestro servidor alojado en la Raspberry Pi 3, esto lo logramos a través de la herramienta Node-Red, que nos facilita la conexión de nuestros dispositivos IoT con nuestro servidor Apache.

Desde el lado del servidor, utilizamos Apache para poder llevar a cabo la correcta instalación y configuración del mismo. Una vez que tenemos eso resuelto, instalamos nuestra base de datos, donde vamos a almacenar toda la información proveniente del paso anterior. Para

ello utilizamos María DB. Luego procedemos a conectar nuestro sitio web con nuestra base de datos y crear un dominio ya sea público o privado para su acceso.

Por último, una vez que ya tenemos nuestro servidor instalado y configurado, nuestra base de datos funcionando y nuestro sitio ya en línea, podemos lograr que el usuario pueda registrarse, conectarse e interactuar con cada uno de los contenedores.



Ilustración 9. Diagrama de arquitectura. Elaboración propia.



# Seguridad

## *Acceso a la aplicación*

Para ingresar al sistema se requiere el inicio de sesión con email y contraseña y es responsabilidad de cada usuario conservar la confidencialidad de la clave que le fue otorgada.

Todas las contraseñas elegidas deberán cumplir con los siguientes requisitos mínimos para ser validos:

- Longitud mínima de 8 caracteres alfanuméricos.
- Contener al menos una letra minúscula.
- Contener al menos una letra mayúscula.
- Contener al menos un número.
- No contener caracteres especiales.

Dentro de la base de datos, para la contraseña se utilizará un cifrado MD5. Esta es una función de cifrado tipo hash que toma una cadena de texto como entrada y devuelve un número de 128 bits. Como este cifrado es irreversible, al momento de validar la contraseña ingresada por el usuario, lo que se realiza es el cifrado de la misma y luego comparar ambos hashes.

Con respecto al manejo de perfiles, en este caso se definieron dos:

- Responsable de control: perfil que puede ver toda la información que brinda cada dispositivo dentro de cada contenedor, en cualquier momento, pero no realizar ninguna alta, baja o modificación de un dispositivo.
- Administrador: perfil que puede ver toda la información que brinda cada dispositivo dentro de cada contenedor y puede realizar altas, bajas o modificaciones de algún dispositivo.

### *Política de respaldo de información*

Se optó por la siguiente política de respaldo de información:

- Periodicidad: 3 veces por día. La primera a la mañana, antes que el servicio de recolección comience. La segunda a la tarde, una vez que el servicio de recolección terminó y la última a la noche, luego de que la actividad comercial haya terminado.
- Tiempo de retención: 7 días.
- Lugar de almacenamiento: Google Drive.
- Forma de iniciación del proceso: automática, a través de script PHP<sup>2</sup>.

## **Análisis de Costos**

En el presente apartado, se identificarán los costos de desarrollo y operativos necesarios para llevar a cabo este proyecto. A continuación, se analiza cada uno de ellos considerando un cliente modelado que tiene 30 contenedores y una sola base de control.

Por otro lado, debido a que algunos componentes o servicios se encuentran en el mercado con precio en moneda extranjera serán expresados con esos valores y luego convertidos a pesos tomando la cotización que establece en Banco Nación de aquella moneda en el momento, por lo que puede estar sujeto a variaciones.

---

<sup>2</sup> Script PHP: es un código de programación que contiene comandos que se van ejecutando de forma secuencial y se lo utiliza para controlar el comportamiento de un programa en específico.

### Costos de desarrollo

Cargo	Fuente	Sueldo Mensual	Cantidad Meses	Costo Total
Programador Full Stack	(GlassDoor, GlassDoor, 2020)	\$ 65.000	3	\$ 195.000
Líder de proyecto	(GlassDoor, GlassDoor, 2020)	\$ 64.000	3	\$192.000
Programador Arduino	(GlassDoor, GlassDoor, 2020)	\$58.000	1	\$58.000
Técnico Instalador	(GlassDoor, GlassDoor, 2020)	\$45.000	1	\$45.000
Total				\$ 490.000

Tabla 20. Costos de desarrollo. Elaboración propia

### Costos operativos

Dentro de esta sección, se encuentran aquellos que deben ser abonados solo una vez, llamados costos operativos iniciales, que se detallan a continuación y luego están los costos operativos mensuales, que son aquellos que deben ser abonados todos los meses para llevar a cabo el servicio.

Costos Operativos Iniciales	Fuente	Cantidad	Fuente USD	Costo unitario USD	Costo unitario pesos	Total
Arduino MKR Fox 1200	(Kit, 2020)	30	(Nacion, 2020)	\$ 47.88	\$ 3.448	\$ 103.420
Mini microswitch	(Sparkfun, 2020)	30	(Nacion, 2020)	\$ 2.49	\$ 179.28	\$ 5348

HC-SR04	(UniBot, UniBot, 2020)	30	-	-	\$ 200	\$ 6000
DHT11	(UniBot, UniBot, 2020)	30	-	-	\$ 150	\$ 4500
KY-026	(MonarcaElectronica, 2020)	30	-	-	\$ 113	\$ 3390
Raspberry Pi 3 B	(TodoMicro, 2020)	3	-	-	\$ 4733.98	\$ 14201.94
Laptop Monitoreo	(Venex, 2020)	2	-	-	\$ 94.990	\$ 189.980
Total						\$ 326,839.94

Costos Operativos Mensuales	Fuente	Cantidad	Fuente USD	Costo unitario USD	Costo unitario pesos	Total
Internet	(Fibertel, 2020)	50 Mb	-	-	\$ 1908	\$ 1908
SigFox	(SigFox, 2020)	30	(Nacion, 2020)	\$ 8	\$ 576	\$ 17.280
Total						\$19.188

Tabla 21. Costos operativos. Elaboración propia

En resumen, el costo total del proyecto es de \$ 836,027.94 pesos, de los cuales \$ 490.000 pertenecen a costos de desarrollo, \$ 326,839.94 a costos operativos iniciales y \$ 19.188 a costos operativos mensuales.

## Análisis de Riesgos

A continuación, se presenta una matriz de riesgo donde se enumeran aquellos detectados para el desarrollo del proyecto, junto con su probabilidad de ocurrencia, que puede ser baja, media o alta y su efecto o impacto en el proyecto en el caso que sucedan que va desde 1 a 5, de bajo a alto. Además, se detalla sus acciones de contingencia.

Id	Tipo	Riesgo	Probabilidad Ocurrencia	Efecto/ Impacto	Acciones Contingencia
1	Técnico	Rotura de alguno de los componentes instalados en el contenedor de basura	Alta	5	Tener personal capacitado y repuestos para atender de inmediato el contenedor
2	Técnico	Robo del detector instalado en el contenedor de basura	Media	5	Recubrir el dispositivo con una caja de seguridad anti vandalismo preparada para no interferir en los detectores del mismo.
3	Técnico	Pérdida de conexión con el dispositivo	Media	5	Gestionar un canal privado de comunicación para tener contacto con la empresa WND quien es la encargada de llevar el servicio Sigfox en el país.
4	Proyecto	Subestimar la cantidad de tiempo y personal para llevar a cabo el proyecto	Media	5	Reformular y renegociar alcance, costo de proyecto y tiempo

5	Proyecto	Problemas de financiamiento	Media	4	Exigir primero pago de costos operativos iniciales para comenzar proyecto y luego negociar nueva forma de pago para el resto de los costos.
---	----------	-----------------------------	-------	---	---

Tabla 22. Análisis de riesgos. Elaboración propia

## Conclusiones

Con el trabajo ya finalizado, podemos concluir que pudimos crear un sistema de gestión de residuos con contenedores inteligentes. Esto lo pudimos lograr instalando en la tapa de cada contenedor un dispositivo que mide en tiempo real los niveles de basura, humedad, temperatura y hasta la presencia de fuego y que haciendo uso del internet de las cosas conecta a cada uno de ellos con una central encargada de llevar a cabo el control necesario. Por otro lado, pudimos diseñar un sistema que toma estos valores de cada contenedor y los muestra sobre una interfaz web.

Las razones que nos llevaron a elegir esta temática fueron la de poner unir un concepto innovador como lo es el internet de las cosas con los conocimientos adquiridos a lo largo de la carrera y por otro lado brindar una solución a una problemática que hoy tiene la ciudad de Córdoba.

Finalizando desde una mirada profesional y personal me siento complemente a gusto con todos los conocimientos que pude adquirir a lo largo de mi carrera, con todas las oportunidades que la Universidad me dio para poder aplicarlas y de estar seguro que esta es la profesión que quiero para mi vida.

### *Demo*

En el siguiente enlace se podrá encontrar el código fuente del prototipo:

<https://github.com/franciscogarciacalzada/smart-bin>

## Referencias

- Agile Alliance. (s.f.). *What is Agile Software Development*. Obtenido de <https://www.agilealliance.org/agile101/>
- Agile Manifesto. (s.f.). *Manifiesto por el desarrollo ágil*. Obtenido de <https://agilemanifesto.org/iso/es/manifiesto.html>
- AOSONG. (s.f.). *DHTC 11 Manual*. Obtenido de <https://akizukidenshi.com/download/ds/aosong/DHT11.pdf>
- Arduino. (s.f.). *Arduino*. Obtenido de <https://www.arduino.cc/en/Guide/Introduction>
- Arduino. (s.f.). *Arduino AdaFruit*. Obtenido de <https://www.arduinolibraries.info/libraries/adafruit-unified-sensor>
- Arduino. (s.f.). *Arduino DHT*. Obtenido de <https://www.arduinolibraries.info/libraries/dht-sensor-library>
- Arduino. (s.f.). *Arduino Low Power*. Obtenido de <https://www.arduino.cc/en/Reference/ArduinoLowPower>
- Arduino. (s.f.). *Arduino MKR FOX 1200*. Obtenido de <https://www.arduino.cc/en/Main.ArduinoBoardMKRFox1200>
- Arduino. (s.f.). *Arduino RTC*. Obtenido de <https://www.arduino.cc/en/Reference/RTC>
- Arduino. (s.f.). *Arduino SigFox*. Obtenido de <https://www.arduinolibraries.info/libraries/arduino-sig-fox-for-mkr-fox1200>
- Arduino. (s.f.). *Arduino Software*. Obtenido de <https://www.arduino.cc/en/main/software>
- Arduino Modules. (s.f.). *KY-026 Sensor*. Obtenido de <https://arduinomodules.info/ky-026-flame-sensor-module/>
- Atlassian. (s.f.). *Scrum qué es*. Obtenido de <https://www.atlassian.com/es/agile/scrum>
- Components 101. (s.f.). *Micro Switch*. Obtenido de <https://components101.com/switches/micro-switch-or-snap-action-switch>
- Córdoba, M. d. (2018). *Higiene Urbana*. Obtenido de <https://www.cordoba.gob.ar/servicios/vecino/higiene-urbana/>
- Córdoba, M. d. (s.f.). <https://www.cordoba.gob.ar/servicios/vecino/higiene-urbana/>.
- Córdoba, M. d. (s.f.). *Ordenanza Municipal*. Obtenido de <https://www.cordoba.gob.ar/wp-content/uploads/downloads/2014/10/Ord.-9.612-Residuos.pdf>
- Cytron Technology. (s.f.). *HC-SR04 Manual*. Obtenido de [https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL\\_pfa39RsB-x2qR4vP8saG73rE/edit](https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit)



EcoBits. (28 de noviembre de 2018). *extrainfo*. Obtenido de <http://www.extrainfo.com.ar/2017/11/28/eco-bits-el-contenedor-de-residuos-con-tecnologia-iot-que-permite-optimizar-la-recoleccion/>

EcubeLabs. (s.f.). *Ecubelabs*. Obtenido de <https://www.ecubelabs.com/es/>

Fibertel. (Junio de 2020). *Fibertel*. Obtenido de <https://www.cablevisionfibertel.com.ar/internet/fibertel-50-megas>

GlassDoor. (2020). *Glass Door*. Obtenido de [https://www.glassdoor.com.ar/Sueldos/c%C3%B3rdoba-team-leader-sueldo-SRCH\\_IL.0,7\\_IM959\\_KO8,19.htm](https://www.glassdoor.com.ar/Sueldos/c%C3%B3rdoba-team-leader-sueldo-SRCH_IL.0,7_IM959_KO8,19.htm)

GlassDoor. (2020). *Glass Door*. Obtenido de [https://www.glassdoor.com.ar/Sueldos/buenos-aires-programador-arduino-sueldo-SRCH\\_IL.0,12\\_IM963\\_KO13,24\\_KE25,32.htm](https://www.glassdoor.com.ar/Sueldos/buenos-aires-programador-arduino-sueldo-SRCH_IL.0,12_IM963_KO13,24_KE25,32.htm)

GlassDoor. (2020). *Glass Door*. Obtenido de [https://www.glassdoor.com.ar/Sueldos/buenos-aires-instalador-tecnico-sueldo-SRCH\\_IL.0,12\\_IM963\\_KO13,23\\_KE24,31.htm](https://www.glassdoor.com.ar/Sueldos/buenos-aires-instalador-tecnico-sueldo-SRCH_IL.0,12_IM963_KO13,23_KE24,31.htm)

GlassDoor. (2020). *GlassDoor*. Obtenido de [https://www.glassdoor.com.ar/Sueldos/c%C3%B3rdoba-programador-full-stack-sueldo-SRCH\\_IL.0,7\\_IM959\\_KO8,19\\_KE20,30.htm](https://www.glassdoor.com.ar/Sueldos/c%C3%B3rdoba-programador-full-stack-sueldo-SRCH_IL.0,7_IM959_KO8,19_KE20,30.htm)

Institute of Electrical and Electronics Engineers. (s.f.). *IEEE*. Obtenido de [https://iot.ieee.org/images/files/pdf/IEEE\\_IoT\\_Towards\\_Definition\\_Internet\\_of\\_Things\\_Revisio\\_n1\\_27MAY15.pdf](https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revisio_n1_27MAY15.pdf)

Interior, L. V. (2018). la recoleccion en cordoba desde sus-inicios.

Interior, L. V. (2018). *La Voz del Interior*. Obtenido de <https://www.lavoz.com.ar/politica/la-recoleccion-en-cordoba-desde-sus-inicios>

Kit, E. (2020). *Electro Kit*. Obtenido de <https://www.electrokit.com/en/product/arduino-mkr-fox-1200-2/>

La Voz del Interior. (2018). *La Voz del Interior*. Obtenido de <https://www.lavoz.com.ar/politica/la-recoleccion-en-cordoba-desde-sus-inicios>

MariaDB. (s.f.). *MariaDB*. Obtenido de <https://mariadb.org/about/>

MDN. (2020). *Mozilla MDN*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>

Ministerio de Modernizacion. (s.f.). *Ministerio de Modernizacion*. Obtenido de <https://www.argentina.gob.ar/sites/default/files/paperbenchmarkinternacional-iot.pdf>

MonarcaElectronica. (2020). *MonarcaElectronica*. Obtenido de <https://monarcaelectronica.com.ar/productos/ky-026-sensor-infrarrojo-flama-llama-fuego-arduino-mona/>

Nacion, B. (9 de Junio de 2020). *Banco Nacion*. Obtenido de <https://www.bna.com.ar/Personas>

Open Source. (2017). *Everything you need to know about node-red*. Obtenido de <https://opensourceforu.com/2017/09/node-red/>

Ordenanza 12649, 12649 (Municipalidad de Cordoba 2017). Obtenido de [https://servicios2.cordoba.gov.ar/docs/licitaciones/2017/024514/ordenanza\\_12649\\_2.pdf](https://servicios2.cordoba.gov.ar/docs/licitaciones/2017/024514/ordenanza_12649_2.pdf)

Ordenanza 9612, 9612 (Municipalidad de Cordoba 27 de Diciembre de 1997). Obtenido de <https://www.cordoba.gov.ar/wp-content/uploads/downloads/2014/10/Ord.-9.612-Residuos.pdf>

Ordenanza Municipal, 12649 (2017).

Raspberry Pi. (s.f.). *Raspberry Pi*. Obtenido de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Raspberry Pi. (s.f.). *Raspberry Pi Documents*. Obtenido de <https://www.raspberrypi.org/documentation/faqs/>

Real Academia Española. (s.f.). *Rae*. Obtenido de <https://dle.rae.es/contenedor>

SigFox. (2020). *SigFox*. Obtenido de <https://buy.sigfox.com/buy>

Sigfox. (s.f.). *SigFox Our Story*. Obtenido de <https://www.sigfox.com/en/sigfox-story>

SigFox. (s.f.). *SigFox Technology*. Obtenido de <https://www.sigfox.com/en/what-sigfox/technology>

Sparkfun. (2020). *Sparkfun*. Obtenido de <https://www.sparkfun.com/products/13119>

Techopedia. (2017). *What is Apache Web server*. Obtenido de <https://www.techopedia.com/definition/4851/apache-web-server>

Techtarget. (2019). *Techtarget*. Obtenido de <https://whatis.techtarget.com/definition/Web-server>

TechTerms. (s.f.). *PHP*. Obtenido de <https://techterms.com/definition/php>

TodoMicro. (2020). *TodoMicro*. Obtenido de <https://www.todomicro.com.ar/raspberry-pi/118-raspberry-pi-3-b.html>

UniBot. (2020). *UniBot*. Obtenido de <https://www.unibot.com.ar/productos/modulo-sensor-ultrasonico-hc-sr04/>

UniBot. (2020). *UniBot*. Obtenido de <https://www.unibot.com.ar/productos/sensor-termohigrometro-dht11/>

Venex. (2020). *Venex*. Obtenido de <https://www.venex.com.ar/notebooks/notebook-lenovo-v330-i5-8250u-4g-1tb-156.html>

Visual Studio Code. (2020). *Visual Studio Code FAQ*. Obtenido de <https://code.visualstudio.com/docs/supporting/faq>

## **Anexo I: Modelo de entrevista para el relevamiento de datos**

1. ¿Qué cargo ocupaste y qué cargo ocupas actualmente dentro de las empresas de recolección de residuos?
2. ¿Dónde puedo conseguir información sobre el pliego de la basura y de las empresas de recolección de residuos de la ciudad?
3. ¿Sabes si existe la propuesta por parte de las empresas de recolección para instalar contenedores inteligentes?
4. ¿Qué sector de la ciudad tiene a cargo cada una de las empresas de recolección de residuos?
5. ¿Se puede acceder a información más sensible de las empresas, como, por ejemplo, cómo está conformada, qué presupuesto manejan, qué funciones cumple cada rol, cuáles son los procesos para llevar a cabo cada función?