



FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

“Impacto del Sistema EcoFavBen en la Predicción de Contaminación por CO en la Ciudad de Cajamarca”

Tesis para optar el título profesional de:

Ingeniero En Sistemas Computacionales

Autores:

Noe Elias Briones Chilon

Erick Omar Hernandez Valencia

Asesora:

Dra. Patricia Janet Uceda Martos

Cajamarca - Perú

2020

AGRADECIMIENTO

A Dios por habernos dado la vida y por haber puesto en nuestro camino a aquellas
personas que nos apoyaron en toda la carrera.

Agradecemos a nuestra asesora Dra. Patricia Janet Uceda Martos, por el acompañamiento
y ayuda en todo el proceso de la elaboración de tesis.

TABLA DE CONTENIDOS

AGRADECIMIENTO	2
ÍNDICE DE TABLAS	5
ÍNDICE DE FIGURAS	7
RESUMEN	9
CAPÍTULO I. INTRODUCCIÓN	11
1.1. Realidad problemática	11
1.2. Formulación del problema	18
1.3. Objetivos	18
1.4. Hipótesis	18
CAPÍTULO II. METODOLOGÍA	19
2.1. Tipo de Investigación	19
2.2. Materiales, instrumentos y métodos	19
2.2.1. <i>Población</i>	19
2.2.2. <i>Muestra</i>	19
2.2.3. <i>Técnicas</i>	21
2.2.4. <i>Instrumentos</i>	21
2.3. Procedimiento	22
CAPÍTULO III. RESULTADOS	23
Objetivo 1: Diseñar e implementar el sistema EcoFavBen para el registro y procesamiento de datos de niveles de monóxido de carbono en la ciudad de Cajamarca.	23
1.1. Sprint 01	26
1.1.1. <i>Desarrollo del sprint 01</i>	26
1.2. Sprint 02	31
1.2.1. <i>Desarrollo del sprint 02</i>	32
1.3. Sprint 03	36
1.3.1. <i>Desarrollo del sprint 03</i>	37
1.4. Sprint 04	39
1.4.1. <i>Desarrollo del sprint 04</i>	40
Objetivo 2: Identificar y clasificar las zonas con mayores niveles de CO en la ciudad de Cajamarca.	41
1.5. Primera Recolección de datos	44
1.5.1. <i>Punto de Control: 2 de mayo – Mario Urteaga</i>	45
1.5.2. <i>Punto de Control: Cruce de Jesús</i>	46

1.5.3.	<i>Punto de Control: Qhapac Ñan</i>	47
1.5.4.	<i>Punto de Control: Recoleta</i>	48
1.5.5.	<i>Punto de Control: Universidad privada del norte</i>	49
1.5.6.	<i>Punto de Control: Puente amarillo</i>	50
1.5.7.	<i>Punto de Control: Fonavi II</i>	51
1.5.8.	<i>Punto de Control: Plazuela Bolognesi</i>	52
1.5.9.	<i>Punto de Control: Urrelo – M. Urteaga</i>	53
1.5.10.	<i>Punto de Control: Ovalo Musical</i>	54
1.5.11.	<i>Punto de Control: Plaza de Armas</i>	55
1.5.12.	<i>Punto de Control: El Quinde</i>	56
1.5.13.	<i>Punto de Control: Sucre</i>	57
1.5.14.	<i>Punto de Control: Ovalo el Inca</i>	58
1.5.15.	<i>Punto de Control: Real Plaza</i>	59
1.5.16.	<i>Punto de Control: M. Seoane – H. Rubio</i>	60
1.6.	Interpretación de datos	61
	Objetivo 3: Realizar predicción de niveles de monóxido de carbono en la ciudad de Cajamarca, en función de los datos obtenidos por el sistema EcoFavBen	63
1.7.	Predicción de Datos	63
1.7.1.	<i>Implementación del Algoritmo FastTreeTweedie</i>	65
1.7.2.	<i>Datos Obtenidos con el algoritmo FastTreeTweedie</i>	68
1.8.	Segunda Recolección de datos	69
1.9.	Comparación de Datos de predicción con datos reales	72
1.10.	Contrastación de la Hipótesis	76
	CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES	78
4.1.	Discusión	78
4.2.	Conclusiones	80
	REFERENCIAS	82
	ANEXOS	87
	ANEXO N° 1. Definición de Monóxido de Carbono (CO)	87
	ANEXO N° 2. Instrumento: Ficha de Registro	93
	ANEXO N° 3. Metodología Scrum	94
	ANEXO N° 4. Materiales para el prototipo del circuito sensorial	100
	ANEXO N° 5. Decreto Supremo N° 003-2017-MINAM	112
	ANEXO N° 6. Algoritmo Árbol de Decisión	116
	ANEXO N° 7. Product Backlog del Sistema EcoFavBen	133
	ANEXO N° 8. Matriz de Consistencia	135
	ANEXO N° 9. Operacionalización de Variables	137
	ANEXO N° 10. Validaciones del Instrumento	139

ÍNDICE DE TABLAS

Tabla 1 <i>Requisitos funcionales del sistema EcoFavBen</i>	23
Tabla 2 <i>Requisitos no funcionales del sistema EcoFavBen</i>	24
Tabla 3 <i>Sprint backlog del sprint 1</i>	26
Tabla 4 <i>Materiales para la implementación del circuito electrónico</i>	26
Tabla 5 <i>Sprint backlog del sprint 2</i>	31
Tabla 6 <i>Sprint backlog del sprint 3</i>	36
Tabla 7 <i>Sprint backlog del sprint 4</i>	39
Tabla 8 <i>Zonas en la ciudad de Cajamarca</i>	41
Tabla 9 <i>Puntos seleccionados, y la zona a la que pertenecen</i>	42
Tabla 10 <i>Días de recolección de datos 2 de mayo – Mario Urteaga</i>	45
Tabla 11 <i>Días de recolección de datos Cruce de Jesús</i>	46
Tabla 12 <i>Días de recolección de datos Qhapac Ñan</i>	47
Tabla 13 <i>Días de recolección de datos Recoleta</i>	48
Tabla 14 <i>Días de recolección de datos Universidad privada del norte</i>	49
Tabla 15 <i>Días de recolección de datos Puente amarillo</i>	50
Tabla 16 <i>Días de recolección de datos Fonavi II</i>	51
Tabla 17 <i>Días de recolección de datos Plazuela Bolognesi</i>	52
Tabla 18 <i>Días de recolección de datos Urrelo – M. Urteaga</i>	53
Tabla 19 <i>Días de recolección de datos Ovalo Musical</i>	54
Tabla 20 <i>Días de recolección de datos Plaza de Armas</i>	55
Tabla 21 <i>Días de recolección de datos El Quinde</i>	56
Tabla 22 <i>Días de recolección de datos Sucre</i>	57
Tabla 23 <i>Días de recolección de datos Ovalo el Inca</i>	58
Tabla 24 <i>Días de recolección de datos Real Plaza</i>	59
Tabla 25 <i>Días de recolección de datos M. Seoane – H. Rubio</i>	60
Tabla 26 <i>Promedio de contaminación por cada punto de control</i>	61
Tabla 27 <i>Datos ingresados al entrenamiento automático</i>	64
Tabla 28 <i>Resultado del entrenamiento automático</i>	65
Tabla 29 <i>Predicción de niveles de CO</i>	68

Tabla 30 <i>Resultados Obtenidos día 1</i>	69
Tabla 31 <i>Resultados Obtenidos día 2</i>	70
Tabla 32 <i>Resultados Obtenidos día 3</i>	71
Tabla 33 <i>Error relativo de la predicción del día 1</i>	73
Tabla 34 <i>Error relativo de la predicción del día 2</i>	74
Tabla 35 <i>Error relativo de la predicción del día 3</i>	75
Tabla 36 <i>Partes y características de Arduino Uno R3</i>	101
Tabla 37 <i>Sprint Backlog Del Sprint 1</i>	108
Tabla 38 <i>Tipo de predicciones que puede hacer con ML.NET</i>	117
Tabla 39 <i>Patrones arquitectónicos de ML.NET</i>	123
Tabla 40 <i>Instructores para árbol de decisión</i>	132
Tabla 41 <i>Cuadro de Sprint Backlog del Sistema EcoFavBen</i>	133
Tabla 42 <i>Matriz de consistencia</i>	135
Tabla 43 <i>Operacionalización de variables</i>	137

ÍNDICE DE FIGURAS

<i>Figura 1.</i> Mapa de vista de tráfico de la ciudad de Cajamarca	21
<i>Figura 2.</i> Arquitectura funcional.....	25
<i>Figura 3.</i> Diagrama de despliegue	25
<i>Figura 4.</i> Pruebas utilizando el software Proteus profesional.....	27
<i>Figura 5.</i> Acoplamiento de los módulos en el base protoboard (Sensor MQ-9, Arduino Uno, Modulo Bluetooth)	28
<i>Figura 6.</i> Código fuente que permite obtener lecturas	29
<i>Figura 7.</i> Primeros resultados de lectura de monóxido de carbono (CO).....	30
<i>Figura 8.</i> Base de datos, utilizando una sola tabla de referencia	32
<i>Figura 9.</i> Interface principal de la aplicación móvil	33
<i>Figura 10.</i> Ventana de iniciar monitoreo	34
<i>Figura 11.</i> Configuración de la dirección del servidor	35
<i>Figura 12.</i> Esquema de base de datos	37
<i>Figura 13.</i> Pruebas de conexión utilizando la herramienta PostMan.....	38
<i>Figura 14.</i> Pruebas de conexión utilizando la herramienta PostMan.....	38
<i>Figura 15.</i> Primeros resultados de contaminación por zonas.....	40
<i>Figura 16.</i> Puntos de control de una zona registrada en el sistema.....	40
<i>Figura 17.</i> Puntos dentro del mapa de la ciudad de Cajamarca	43
<i>Figura 18.</i> Promedio de contaminación por cada punto de control	62
<i>Figura 19.</i> Código donde se inicializa el objeto de entrenamiento	66
<i>Figura 20.</i> Creación del contexto utilizando el modelo de entrenamiento	66
<i>Figura 21.</i> Validación del contexto.....	66
<i>Figura 22.</i> Entrenamiento del modelo.....	66
<i>Figura 23.</i> Guardar el modelo	67
<i>Figura 24.</i> Fragmento de código fuente donde hizo uso del modelo	67
<i>Figura 25.</i> Dato predicho	67
<i>Figura 26.</i> Pruebas de normalidad de los datos reales y la predicción a través del sistema EcoFavBen	76
<i>Figura 27.</i> Resultado de prueba T-Student	77
<i>Figura 28.</i> Diagrama Scrum.....	99
<i>Figura 29.</i> Diagrama ciclo de vida de Scrum.....	100

<i>Figura 30.</i> Partes de Arduino Uno R3.....	102
<i>Figura 31.</i> Descripción de los pines de Arduino Uno.....	102
<i>Figura 32.</i> Sensor MQ-9	103
<i>Figura 33.</i> Sensibilidad del sensor MQ-9	104
<i>Figura 34.</i> Estructura y configuración del sensor MQ-9.....	105
<i>Figura 35.</i> Modulo Bluetooth HC-05.....	106
<i>Figura 36.</i> Data sheet del módulo Bluetooth HC-05.....	107
<i>Figura 37.</i> Circuito sensorial armado.....	109
<i>Figura 38.</i> Código Arduino	109
<i>Figura 39.</i> Comprobando comunicación de Arduino con sensor.....	110
<i>Figura 40.</i> Formula de Calibración	110
<i>Figura 41.</i> Datos de Sensor calibrado	111
<i>Figura 42.</i> Flujo De Trabajo Del Código	118
<i>Figura 43.</i> El modelo más básico es una regresión lineal	119
<i>Figura 44.</i> Descripción a las del conjunto de entrenamiento	120
<i>Figura 45.</i> Catálogos que muestran de IntelliSense	124
<i>Figura 46.</i> Esquema y modelos de datos.....	127

RESUMEN

En la actualidad la contaminación ambiental ha aumentado de manera drástica convirtiéndose en una alerta mundial, gran parte de la contaminación mundial se debe a la emisión de diversos gases, generados por las actividades humanas, siendo el más perjudicial para la salud humana el monóxido de carbono (CO), el cual viene afectando a un 90% de la población mundial, además es responsable del deceso prematuro de siete millones de personas cada año a nivel mundial.

La iniciativa de la presente investigación corresponde a la necesidad de implementar un prototipo sensorial y desarrollar un sistema web y aplicación móvil para la detección de los niveles de contaminación por CO presentes en diferentes zonas de la ciudad de Cajamarca. Al detectar y presentar los resultados, causa un efecto imprescindible para contrastar el impacto que genera las grandes concentraciones de CO y de esta manera poder sensibilizar a los ciudadanos: planteándose una solución práctica a dicha dificultad mediante la aplicación del sistema EcoFavBen.

Se diseñó e implementó un prototipo sensorial y sistema EcoFavBen para la medición del nivel de CO en la ciudad de Cajamarca, que permitió obtener datos de los niveles de CO, utilizando dispositivos electrónicos para procesar la información como lo es el “Arduino UNO”, que proporciona una fácil conexión con los sensores de CO (MQ-9), y envían los datos obtenidos a un equipo móvil a través del Módulo Bluetooth HC-05, generando información importante que puede ser utilizada para diferentes individuos que requieran tomar decisiones de prevención.

En la presente investigación se demuestra que el sistema EcoFavBen tiene un impacto positivo en la predicción de contaminación por CO en la ciudad de Cajamarca, ya que se han podido identificar los niveles de CO por zonas en la ciudad de Cajamarca utilizando un intervalo de 1 segundo para obtener datos más precisos.

Al finalizar demostramos que el sistema EcoFavBen ha impactado de manera positiva en la predicción de contaminación por CO en la ciudad de Cajamarca, ya que se obtuvieron datos aceptables; no obstante, aún no se los puede considerar confiables, puesto que según el análisis de error estadístico relativo que se aplicó, se tuvo un margen máximo del 92%, pero en la mayoría de casos, el margen de error relativo es menor al 30%, siendo estos datos aceptables.

Palabras clave: Monóxido de Carbono (CO), Niveles de Contaminación, Algoritmo Árbol de Decisión, Predicción, Detección, Monitoreo.

CAPÍTULO I. INTRODUCCIÓN

1.1. Realidad problemática

La contaminación atmosférica produce graves problemas en la salud de los seres humanos, los cuales vienen afectando a un 90% de la población mundial, además es responsable del deceso prematuro de siete millones de personas cada año, entre los cuales 600.000 son niños (ONU, 2019).

La contaminación por CO proveniente del parque automotor es actualmente uno de los problemas más graves que se presenta en las ciudades con un crecimiento exponencial.

En los últimos 10 años Cajamarca ha crecido de manera exponencial en cuanto a población y parque automotor. Según el Instituto Nacional de Estadística e Informática, Cajamarca es el quinto departamento más poblado del país con un total de 1 341 012 habitantes, concentrando su mayor población en la provincia de Cajamarca con 348 433 hasta el año 2017 (INEI, 2018).

Según SUNARP (2018), el año 2017 en Cajamarca se inscribieron 10419 automóviles.

El CO, es considerado uno de los mayores contaminantes de la atmósfera terrestre. Sus principales fuentes productoras responsables de aproximadamente 80% de las emisiones, son los vehículos automotores que utilizan como combustible gasolina o diésel y los procesos industriales que utilizan compuestos del carbono, también se produce por la combustión incompleta de hidrocarburos (Téllez, 2019).

El CO, es un gas muy tóxico tanto para las personas como para los animales, carece de propiedades organolépticas, es incoloro, inodoro. Sus potenciales efectos adversos a largo plazo son poco conocidos por la población (Brown, 2004).

Las características descritas anteriormente, hacen que el CO sea definido como un "asesino silencioso" por muchos autores. Este gas es parcialmente soluble en agua, alcohol y benceno,

resultado de la oxidación incompleta del carbono durante el proceso de combustión. Consta de un átomo de carbono unido mediante enlace covalente (con una longitud de 0,1128 nm) a un átomo de oxígeno (CO).

Los niveles permisibles para el CO, según el Decreto Supremo 003-2017-MINAM aprobado el 7 de junio del 2017, es de 10000 $\mu\text{g}/\text{m}^3$ en un periodo de 8 horas (Ministerio del Ambiente - Perú, 2017). Esto convertido en partes por millón (PPM) con la ecuación obtenida del Instituto Nacional de Ecología y Cambio Climático de México, resulta 9 PPM (INECC, 2019).

En los últimos años la revolución digital está produciendo cambios profundos en los hábitos de consumo de los seres humanos, pues estos, día a día hacen uso de enormes cantidades de información (datos) que les ayudan en distintas actividades que realizan.

En las últimas dos décadas la disponibilidad de datos digitales se ha incrementado casi 1000 veces mientras que con relación a la algoritmia se ha registrado mejoras de 10 veces en su eficiencia al tiempo que la velocidad de computación ha aumentado 100 veces su desempeño.

Las técnicas de aprendizaje automático o Machine learning están experimentando un auge sin precedentes en diversos ámbitos, tanto en el mundo académico como en el empresarial y constituyen una importante palanca de transformación. Estas pueden definirse como un conjunto de métodos capaces de detectar automáticamente patrones de datos. Bajo esta definición, el concepto de aprendizaje automático lleva existiendo al menos desde los años 50. Este concepto de aprendizaje automático incluye desde entonces el uso de los patrones detectados para realizar predicciones, o para tomar otros tipos de decisiones en entornos de incertidumbre.

Frente a las técnicas de estadísticas clásicas, la introducción de técnicas de Machine Learning permite mejorar el proceso de estimación de modelos, no solo con relación al aumento del poder predictivo a través de las nuevas tecnologías y técnicas de selección de variables, sino también en la mejora de la eficiencia de los procesos a través de la automatización (Managements Solutions, 2018).

Los dispositivos móviles están de continuo en el día a día de los jóvenes acaparando su atención e interés por las nuevas tendencias tecnológicas puesto que el número de dispositivos móviles ha superado por primera vez el número de personas en el mundo: ya hay más de 7.300 millones de teléfonos frente a 7.200 millones de personas. La mayor cantidad de dispositivos móviles en proporción a la población la encontramos en Europa del Este con una penetración del 139%, en segundo lugar, se encuentra América del Sur con un 126% y Europa Occidental con un 125%. La penetración más baja se encuentra en el Sur de Asia con un 77% seguido de África con un 79%. Los dispositivos móviles con sistema operativo ANDROID son los que dominan el mercado a nivel mundial con 76.6%, 3 de cada 4 personas usan este sistema operativo, entonces al analizar estos datos podemos deducir que las personas en Latinoamérica poseen en su mayoría al menos un dispositivo móvil (Rivero, 2015).

En la actualidad, 9 de cada 10 Latinoamericanos utilizan un dispositivo móvil regularmente. Las aplicaciones a nivel de Latinoamérica son la forma favorita para conectarse desde los dispositivos móviles puesto que se descarga un promedio de 18 aplicaciones en sus dispositivos y según una encuesta realizada casi el 90% del tiempo de conexión se destina al uso de aplicaciones. Y las personas que más utilizan dispositivos de escritorio en Latinoamérica son los jóvenes de 15 a 24 años y esto se ve reflejado en el Perú con un

porcentaje de 29%, esto deja en claro que los jóvenes se sienten atraídos más por las cosas novedosas y la nueva tecnología emergente. (COMSCORE, 2015)

Los antecedentes considerados en la investigación son:

Estrada y Reinoso (2009), en España, desarrollaron la investigación *Diseño e implementación de un prototipo de monitoreo remoto de contaminación ambiental utilizando tecnología GSM*, donde elaboraron un prototipo que permitió realizar mediciones de forma automática mediante sensores de CO, CO₂, temperatura y humedad. Los datos adquiridos los enviaron en forma de SMS, a través de un teléfono celular a una aplicación de monitoreo, teniendo menos de 5 minutos como intervalo de retardo del envío y recepción de los mensajes en la etapa de pruebas, ya que en este periodo permitía obtener data necesaria para la toma de decisiones.

Salazar Soria (2009), en su investigación *Análisis y medición de contaminación acústica en sectores de alta densidad vehicular de la ciudad de Quito*, donde tuvieron como objetivo realizar un mapa de contaminación acústica, afirmaron que los datos obtenidos fueron totalmente aleatorios debido a las condiciones en las que realizaron las mediciones, así como los sectores no tenían características similares y las circunstancias varían en el transcurso de la medición cada segundo.

En España también, Valencia (2012), desarrolló la investigación *Infraestructura Polivalente Basada en Sensores Inalámbricos Aplicada a la Monitorización Medioambiental Georreferenciada*, donde elaboró una infraestructura tecnológica polivalente basada en sensores inalámbricos, que aplicaron a la monitorización georreferenciada de parámetros

medioambientales. En ella, se lograron analizar los componentes básicos de un Sistema de Información Geográfica que permiten gestionar los datos que llegan al centro de control desde la red sensorial de forma georreferenciada.

Puerta (2015), en su investigación *Desarrollo de una API para la descripción y gestión de Servicios Web*, en este proyecto se desarrolló dos API, una para la descripción de servicios y otra para su monitorización y gestión. La API para la descripción de servicios tiene como finalidad recopilar la información necesaria para que los servicios se puedan ejecutar. La API para la monitorización y la gestión estuvo orientada a controlar el estado en el que se encontraron los servicios, de los cuales se extrajo información relevante para la toma de decisiones y por otra parte, para ejecutar y parar servicios locales de forma cómoda. Además, implementaron una interfaz web dedicada a utilizar la API de monitorización y gestión de servicios de forma gráfica.

Huanca (2015), en su investigación *Revisión sistemática de la calidad del software en prácticas ágiles*, concluye que el proceso Scrum reportó significativas mejoras de la calidad en los estudios de caso y el mayor incremento de productividad en comparación a todas las otras prácticas ágiles. También se detectó una disminución en la densidad de defectos, disminución del esfuerzo, además de un incremento en la satisfacción del cliente. Otras prácticas ágiles como el *planning game*, refactorización y pruebas unitarias fueron también algunas de las prácticas que contribuyeron activamente a la mejora de calidad. El proceso SCRUM reportó mejoras de hasta un 30% en la calidad del producto, y se percibió un incremento de la productividad desde un 16% hasta un 250% en los casos de estudio.

Haylinos y Gonzales (2017), en su investigación *Metodologías ágiles en la implementación de una aplicación móvil para la gestión de citas en la Clínica Dental “Perio Dent” – Huancayo*, utilizaron metodologías de desarrollo de software ágiles, que permitieron a los pequeños grupos de desarrollo concentrarse en la tarea de construcción de software fomentando prácticas de fácil adopción y un entorno ordenado que ayude a que las personas trabajen mejor y permita que los proyectos finalicen exitosamente. La metodología propuesta en esta investigación, avanza en el conocimiento teórico de estos procesos analizando estos principios mencionados y reuniendo prácticas y patrones que contribuyen a la implementación y posterior adaptación del proceso a la realidad de cada organización. Finalmente se observó que el diseño y la implementación de la aplicación móvil utilizando la metodología ágil SCRUM fue de un 29% de significancia, ya que permitió medir los tiempos de consultas y los tiempos en gestionar una cita online en tiempo real y haciendo estas reservas de citas en menor tiempo que el normal.

León, (2018), en su investigación *Desarrollo de un modelo algorítmico basado en árboles de decisión para la predicción de la permanencia de un paciente en un proceso psicoterapéutico*, concluyó que el flujo algorítmico establecido para la predicción en base a los resultados previos, permitió ser más específico para determinar la permanencia del paciente en el proceso psicoterapéutico. También se verificó que la precisión de la predicción de las clases depende del tamaño del conjunto de datos al cual pertenece la clase, en consecuencia, se obtuvo bajos niveles de precisión en las clases minoritarias. Por lo que se observó que los niveles de precisión por el modelo del algoritmo Árbol de Decisiones, superan las pruebas en un 70%, por lo que se puede definir como aceptable, sin embargo,

disto de la precisión por modelo de los análisis realizados con los datos de entrenamiento que se tuvo.

Espinoza y Gutiérrez, (2015), en su investigación *Sistema de información para la toma de decisiones, usando técnicas de análisis predictivo para la empresa IASACORP international S.A.*, concluyó que en cuanto a la mejora del proceso de presupuestos de ventas, la herramienta de predicción, ha permitido que genere una proyección del cálculo de probabilidades de ventas de las líneas y familias de productos con una aproximación del 95% de confiabilidad, y de cómo resultado un proceso de presupuesto de ventas mejor del que se venía usando. Esto principalmente por que el uso de la información estadística en la herramienta de predicción, nos da un respaldo en base al algoritmo usado (árbol de decisión). En cuanto a la mejora en los tiempos de preparar los reportes de estimación de venta y plan de compras en base a los resultados del pronóstico, la herramienta de predicción ha permitido que este proceso se realice en 1 día, versus el proceso anterior que tomaba entre 3 a 4 días.

Conociendo algunos conceptos:

El CO, es un gas incoloro e inodoro que se forma por la combustión incompleta de material orgánico, y la falta de oxígeno. Es considerado uno de los mayores contaminantes de la atmósfera terrestre, y uno de los mayores problemas ambientales de América Latina. Las principales fuentes productoras de este contaminante son los vehículos automotores; los procesos industriales; los incendios forestales y urbanos y la incineración de materia orgánica. Los vehículos automotores y los procesos industriales son responsables de aproximadamente 80 % de las emisiones de monóxido de carbono a la atmósfera, para mayor detalle (Agency, 2020) (ver Anexo 1).

1.2. Formulación del problema

¿Cuál es el impacto del sistema EcoFavBen en la predicción de contaminación por CO en la ciudad de Cajamarca?

1.3. Objetivos

1.3.1. Objetivo general

Determinar el impacto del sistema EcoFavBen en la predicción de contaminación por CO en la ciudad de Cajamarca.

1.3.2. Objetivos específicos

- Diseñar e implementar sistema EcoFavBen para el registro y procesamiento de datos de niveles de monóxido de carbono en la ciudad de Cajamarca.
- Identificar y clasificar las zonas con mayores niveles de monóxido de carbono en la ciudad de Cajamarca.
- Realizar predicción de niveles de monóxido de carbono en la ciudad de Cajamarca, en función de los datos obtenidos por el sistema EcoFavBen.

1.4. Hipótesis

1.4.1. Hipótesis general

El sistema EcoFavBen tiene un impacto positivo en la predicción de contaminación por CO en la ciudad de Cajamarca.

CAPÍTULO II. METODOLOGÍA

2.1. Tipo de Investigación

2.1.1. Según el propósito

Aplicada.

2.1.2. Según el diseño de investigación

Se considera el diseño de investigación experimental, por el motivo que se busca establecer el impacto del sistema EcoFavBen aplicando como estímulo una ficha de lectura y registro de datos.

El tipo de diseño de investigación es cuasi experimental, porque para el estudio de problemas en los cuales no se puede tener control absoluto de las situaciones, pero se pretende tener el mayor control posible, aun cuando se estén usando grupos ya formados. Es decir, el cuasi experimento se utiliza cuando no es posible realizar la selección aleatoria de los sujetos participantes en dichos estudios (Hernández, 2014).

2.2. Materiales, instrumentos y métodos

2.2.1. Población

Según Lepkowski (2008), describe que una población es un conjunto de todos los casos que concuerdan con una serie de especificaciones, en este caso, la población son 22 barrios de la ciudad de Cajamarca.

2.2.2. Muestra

Según Ochoa (2020), hablamos de muestreo no probabilístico cuando no tenemos acceso a una lista completa de los individuos que forman la población, por lo que

el tipo de muestra que se utilizará en esta investigación será el muestro no probabilístico por conveniencia, para lo cual se tomó 12 zonas teniendo como referencia el registro del volumen vehicular tomado del Plan de Desarrollo Urbano de la municipalidad Provincial de Cajamarca en el *capítulo IV: Diagnostico Urbano*, muestra los puntos con mayor congestión vehicular de la ciudad en las intersecciones de la Vía de evitamiento - hoyos rubio, Vía de evitamiento - jr. El inca, Vía de evitamiento - jr. Soane, Ovalo musical, Av. San Martín - av. Atahualpa, Av. Sucre - av. Atahualpa Av. De los héroes - av. El maestro.

Otras zonas de congestión se producen en el eje de la Av. Los Héroes incluyendo su empalme con las Avenidas Atahualpa y Sucre y vías aledañas. Igualmente, la Av. El Maestro y el Jr. Amalia Puga, por su intenso tráfico vehicular y peatonal son focos de congestión a lo largo del día.

la Av. Hoyos Rubio desde su intersección con la Vía de Evitamiento Norte, la zona del Centro Comercial El Quinde y su ingreso al Centro por el Jr. Revilla Pérez es una arteria normalmente congestionada (Municipalidad de Cajamarca, 2020).

Lo mencionado anteriormente es confirmado por Google Maps en su mapa de vista de tráfico en la ciudad de Cajamarca (ver Figura 1).

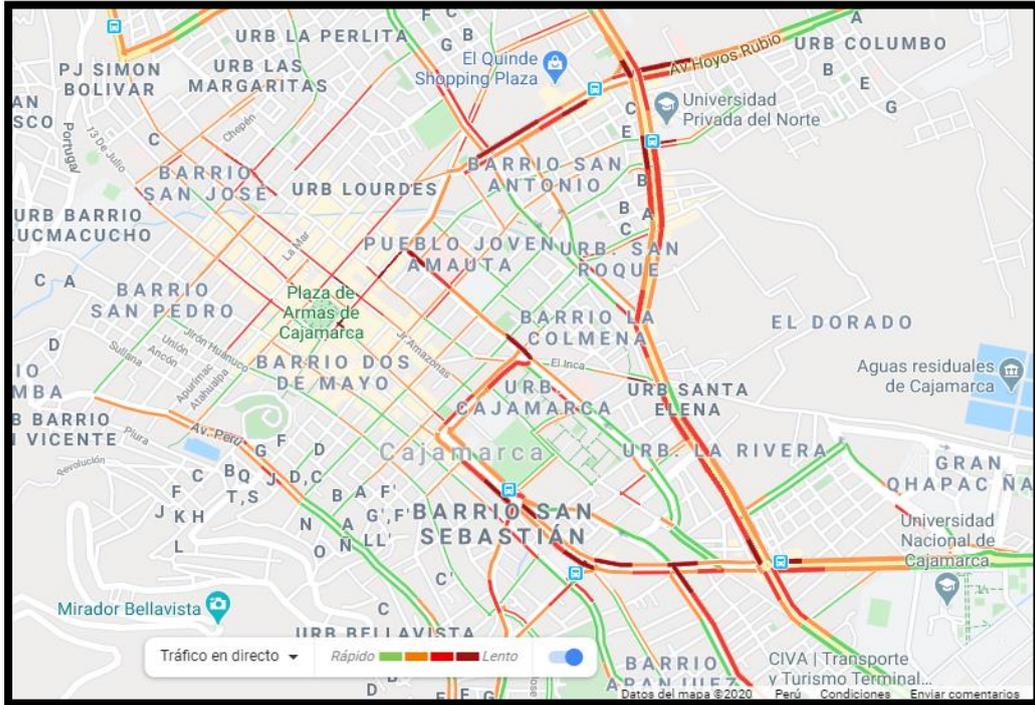


Figura 1. Mapa de vista de tráfico de la ciudad de Cajamarca

Fuente: (google, 2020)

2.2.3. Técnicas

Para la presente investigación se ha tomado como técnica la lectura y el registro automatizado de datos a través del Sistema EcoFavBen.

2.2.4. Instrumentos

Se utilizó como instrumento:

- ❖ Ficha de registro (Ver Anexo 2).

2.3. Procedimiento

Para el desarrollo de la investigación se efectuó los siguientes pasos:

- Determinación de los módulos electrónicos necesarios para crear un prototipo sensorial que detecte y mida el nivel de CO.
- Implementación en el protoboard del prototipo sensorial con cada una de sus conexiones.
- Seleccionar la metodología ágil para el desarrollo del sistema EcoFavBen.
- Desarrollar la aplicación móvil.
- Implementar la solución web del sistema, el cual consumirá los datos obtenidos, para poder visualizarlos en el mapa.
- Para realizar la recolección de datos se hizo recorridos en 12 zonas de la ciudad de Cajamarca, con la ayuda del prototipo del circuito sensorial el cual tiene la función de captar el CO a través del sensor MQ-9, y enviar los datos obtenidos a la aplicación móvil a través del módulo bluetooth HC-05. La toma de datos se hizo en dos instancias, la primera para tener datos históricos con los cuales realizar la predicción, y la segunda para poder corroborar la predicción del sistema EcoFavBen.
- Clasificar midiendo los niveles de CO por zona de mayor nivel de concentración de CO.
- Elegir el algoritmo de predicción y clasificación para el sistema EcoFavBen.
- Visualizar datos de predicción en el sistema EcoFavBen.
- Elaboración y presentación del informe final.

CAPÍTULO III. RESULTADOS

Objetivo 1: Diseñar e implementar el sistema EcoFavBen para el registro y procesamiento de datos de niveles de monóxido de carbono en la ciudad de Cajamarca.

Para el desarrollo del sistema EcoFavBen se utilizó la metodología SCRUM, ya que, como lo menciona Muñoz (2017), la metodología SCRUM resalta como la más apropiada para el desarrollo de un producto de software (ver Anexo 3).

Product Backlog

A continuación, se determinan los requerimientos funcionales y no funcionales del sistema EcoFavBen.

Requerimientos funcionales

En la tabla se muestran los requisitos funcionales del sistema EcoFavBen (ver Tabla 1).

Tabla 1 *Requisitos funcionales del sistema EcoFavBen*

DESCRIPCIÓN	PRIORIDAD
Lectura de nivel de monóxido de carbono en tiempo real	ALTA
Establecer zonas de análisis de niveles de monóxido de carbono	ALTA
Establecer puntos de control de referencia para cada zona establecida	ALTA
Mostrar datos estadísticos (gráficos) de la información almacenada	ALTA
Predicción de niveles de contaminación por medio de algoritmos de machine learning	ALTA
Guardar los datos que el módulo de lectura obtenga	ALTA
Guardar el punto de ubicación para cada dato leído	ALTA

Requerimientos no funcionales

En la tabla se muestran los requisitos no funcionales del sistema EcoFavBen (ver Tabla 2)

Tabla 2 *Requisitos no funcionales del sistema EcoFavBen*

DESCRIPCIÓN	PRIORIDAD
El sistema funcione de forma remota (Web)	ALTA
El equipo encargado de la lectura de datos sea portable	ALTA

Diseño del sistema

- En base a los descrito anteriormente, se optó lo siguiente:
- Implementar un módulo de lectura de niveles de monóxido de carbono, utilizando Arduino y el sensor MQ-9 (ver Anexo 4).
- Implementar un módulo de recolección de datos móvil (Android), módulo que tendrá como función principal, obtener datos del módulo de lectura y enviarlos al servidor.
- Implementar una Web API, quien se encargará de recibir y procesar los datos obtenidos por el módulo de recolección de datos, según sea conveniente.
- Implementar un módulo Web para poder administrar la información guardada por los módulos anteriores.

Teniendo en cuenta lo mencionado, se plantea la arquitectura funcional mostrada (ver Figura 2).

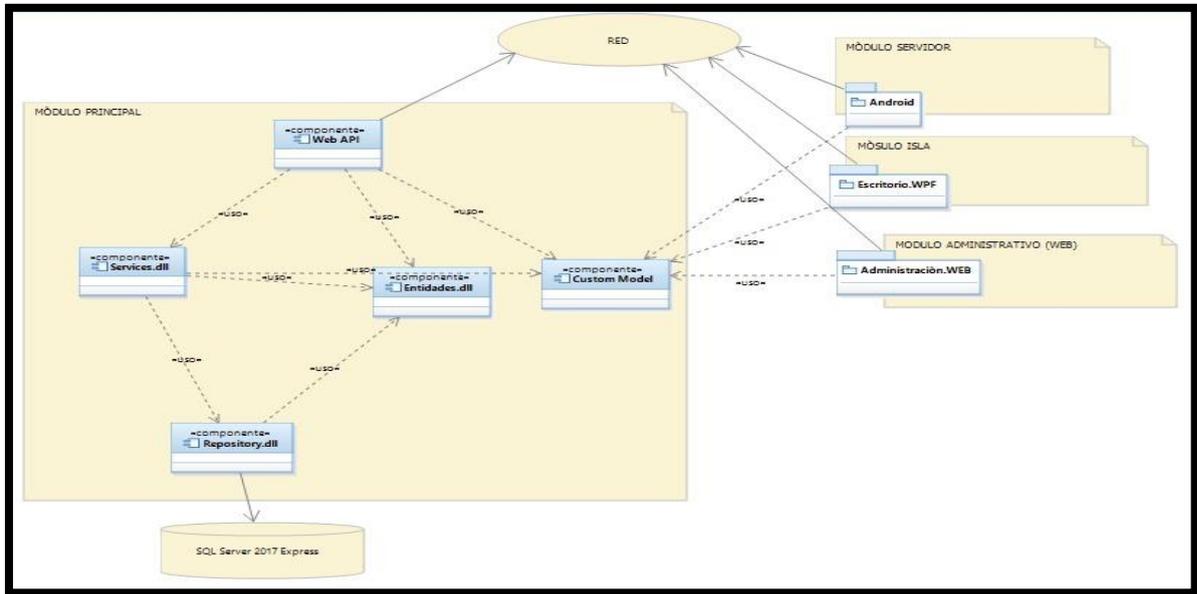


Figura 2. Arquitectura funcional

En base a los requerimientos establecidos se diseñó un diagrama de despliegue, mostrado (ver Figura 3).

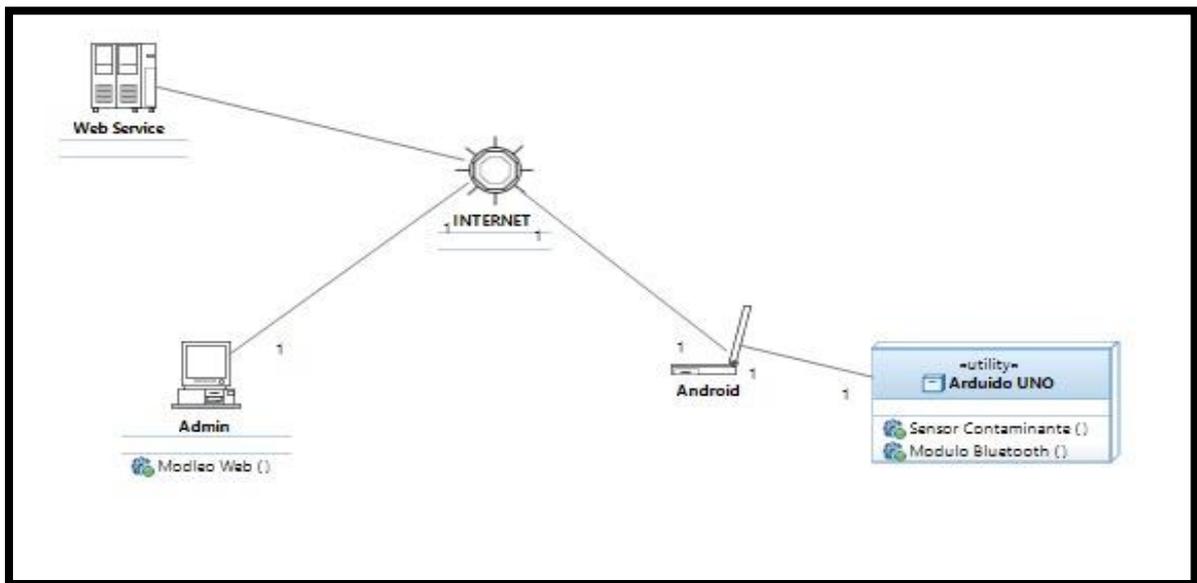


Figura 3. Diagrama de despliegue

1.1. Sprint 01

En el primer sprint, se tuvo como objetivo diseñar e implementar el módulo de lectura de datos. En la tabla se muestra el sprint backlog del srpint 1 (ver Tabla 3).

Tabla 3 *Sprint backlog del sprint 1*

Id	Sprint	Tarea	Historial de Usuario	Responsable	Duración
ID1	1	Implementar circuito electrónico	Obtener dato de CO del ambiente	Noe	5

1.1.1. Desarrollo del sprint 01

En la tabla se muestra los materiales (ver Tabla 4), que se utilizó para la implementación del circuito electrónico que permitirá leer los niveles de CO en el medio ambiente.

Tabla 4 *Materiales para la implementación del circuito electrónico*

Nro.	MATERIAL	CANTIDAD	COSTO UNITARIO (S/.)	COSTO TOTAL (S/.)
1	Sensor MQ-9	1	15	15
2	Arduino Uno R3	1	45	45
3	Modulo Bluetooth	1	29	29
4	Batería de 9 Volteos	1	12	12
5	Protoboard	1	10	10
6	Elementos Electrónicos	1	5	5
Subtotal				116
Improvisto (10%)				11.6
TOTAL				127.6

Para el funcionamiento y acoplamiento de los materiales antes mencionados, se realizó pruebas utilizando el software Proteus profesional, versión 8 (ver Figura 4).

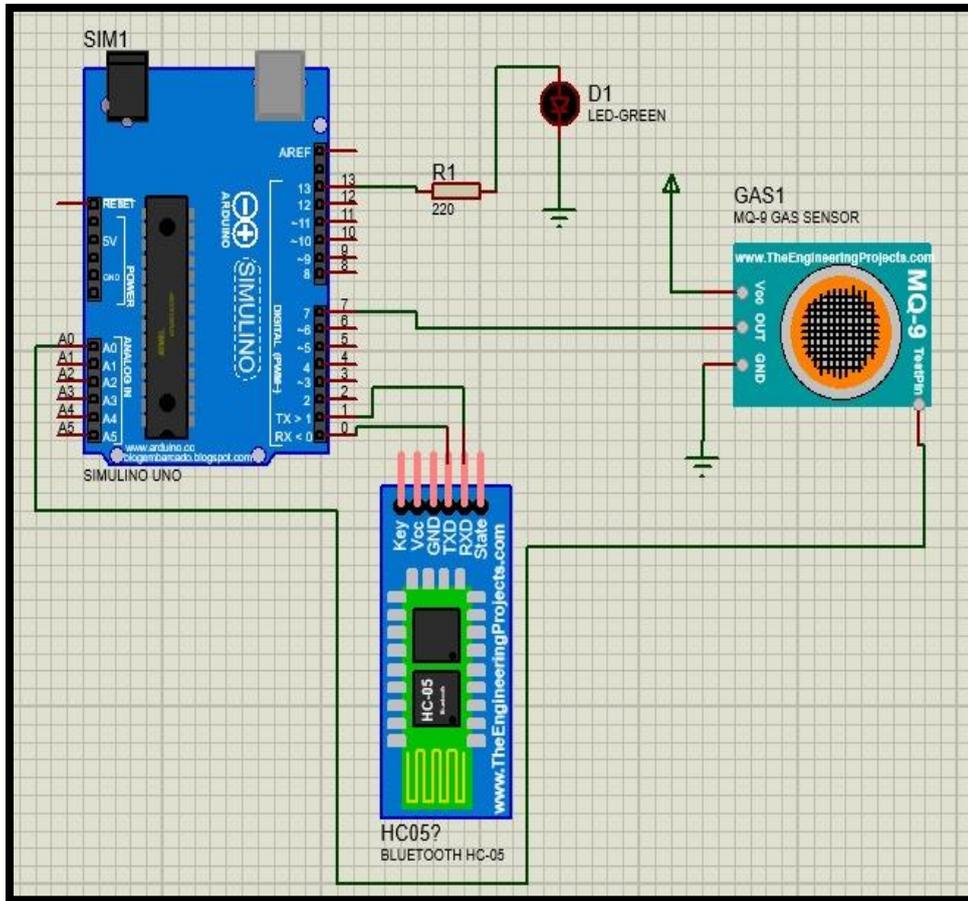


Figura 4. Pruebas utilizando el software Proteus profesional

Después de las pruebas realizadas en el software Proteus, y obtener un diseño eficiente del circuito, se optó por la implementación con los materiales descritos (ver Figura 5).

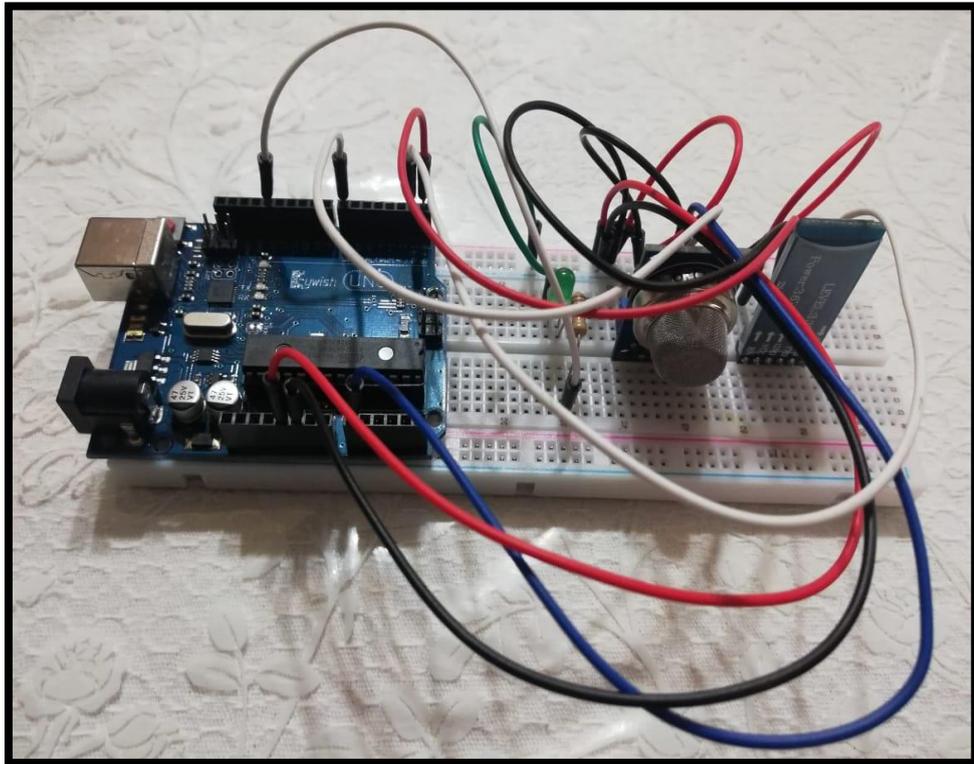


Figura 5. Acoplamiento de los módulos en el base protoboard (Sensor MQ-9, Arduino Uno, Modulo Bluetooth)

Utilizando el IDE de Arduino se implementó un código fuente que permite obtener lecturas en partes por millón (PPM) de monóxido de carbono en su entorno (ver Figura 6).

```
final_final_final.txt
#include <SoftwareSerial.h> // Incluimos la libreria SoftwareSerial
SoftwareSerial BT(10,11); // Definimos los pines RX y TX del Arduino conectados al Bluetooth

/*****Hardware Related Macros*****/
const int calibrationLed = 13; //when the calibration start , LED pin 13 will light up , off when finish calibrating
const int MQ_PIN = A0; //define which analog input channel you are going to use
int RL_VALUE = 1; //define the load resistance on the board, in kilo ohms
float RO_CLEAN_AIR_FACTOR = 9.86; //RO_CLEAN_AIR_FACTOR=(Sensor resistance in clean air)/RO,
//which is derived from the chart in datasheet

/*****Software Related Macros*****/
int CALIBRATION_SAMPLE_TIMES=50; //define how many samples you are going to take in the calibration phase
int CALIBRATION_SAMPLE_INTERVAL=500; //define the time interval(in milisecond) between each samples in the
//calibration phase
int READ_SAMPLE_INTERVAL=50; //define how many samples you are going to take in normal operation
int READ_SAMPLE_TIMES=5; //define the time interval(in milisecond) between each samples in
//normal operation

/*****Application Related Macros*****/
#define GAS_LPG 0
#define GAS_CO 1
#define GAS_SMOKE 2

/*****Globals*****/
float LPGCurve[3] = {2.3,0.3,-0.43}; //two points are taken from the curve.
//with these two points, a line is formed which is "approximately equivalent"
//to the original curve.
//data format: { x, y, slope}; point1: (lg200, 0.21), point2: (lg10000, -0.59)
float COCurve[3] = {2.3,0.21,-0.43}; //two points are taken from the curve.
```

Figura 6. Código fuente que permite obtener lecturas

Después de cargar la compilación del código fuente en el Arduino y utilizando el monitor de puerto serial, se obtuvo los primeros resultados de lectura de monóxido de carbono (CO) (ver Figura 7).

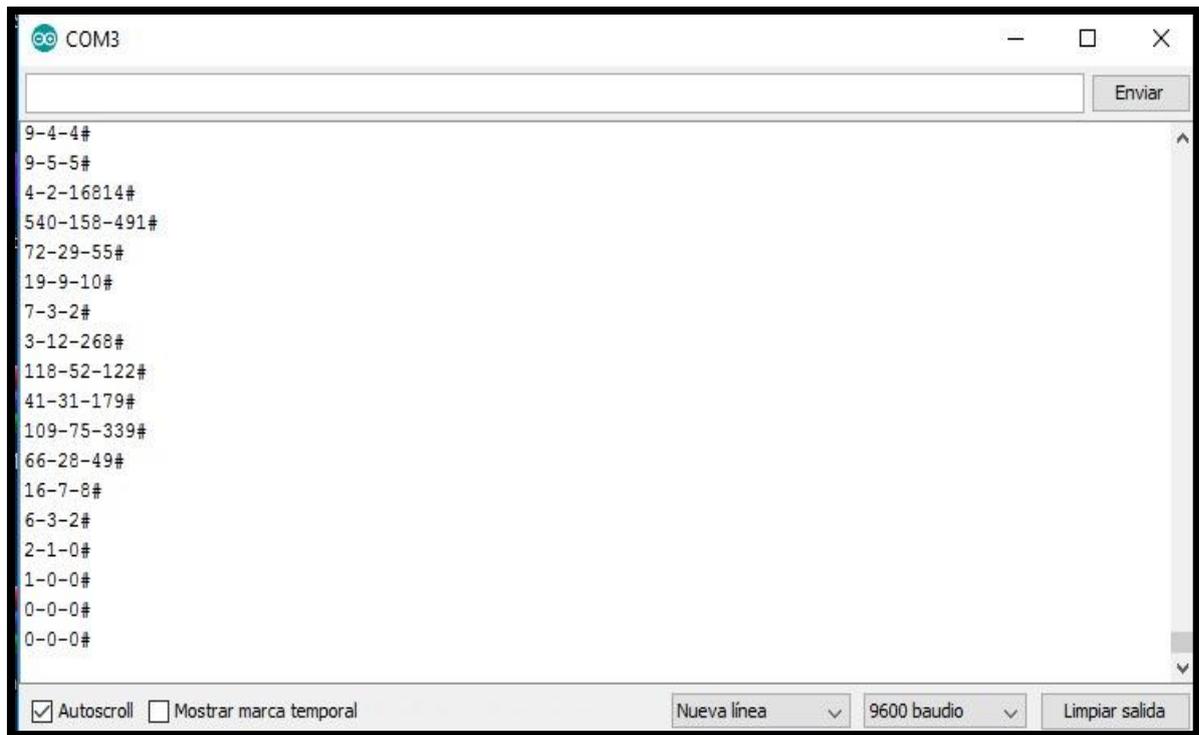


Figura 7. Primeros resultados de lectura de monóxido de carbono (CO)

1.2. Sprint 02

El objetivo del segundo sprint es el de diseñar e implementar aplicativo móvil que permita obtener los datos que el módulo de lectura (Arduino) pueda obtener por medio del sensor MQ-9. En la tabla se muestra el sprint backlog del sprint 2 (ver Tabla 5).

Tabla 5 *Sprint backlog del sprint 2*

Id	Sprint	Tarea	Historial de Usuario	Responsable	Duración
ID2	2	Leer datos de Arduino	visualizar datos provenientes del módulo de lectura (Arduino)	Erick	3
		Base de datos en SQLite	Guardar datos obtenidos	Noe	1
		Enviar datos a Web Api	Sincronizar datos con Web Api	Erick	1

1.2.1. Desarrollo del sprint 02

Para el diseño del aplicativo móvil que permita leer datos del módulo de lectura (Arduino), se utilizó el IDE Android Studio y el lenguaje de programación Java, teniendo las funcionalidades específicas descritas a continuación:

- Obtener dato obtenido por el módulo de lectura por cada segundo.
- Guardar dato de forma local o en servidor
- Sincronizar información de registros guardados localmente con servidor.

Con la finalidad de que el aplicativo móvil guarde los datos de lectura de manera interna, se utiliza el motor de base de datos base de datos SQLite.

En la aplicación se ha creado una base de datos, utilizando una sola tabla de referencia, utilizando los siguientes campos (ver Figura 8):

- Id: identificador del registro.
- Dato: dato obtenido desde el módulo de lectura (Arduino)
- Latitud: coordenada de latitud obtenida desde el GPS del móvil.
- Longitud: coordenada de longitud obtenida desde el GPS del móvil.
- Fecha: Fecha y hora del registro.



Column Name	Data Type
id	INTEGER
dato	NUMERIC
latitud	TEXT
longitud	TEXT
fecha	BLOB

Figura 8. Base de datos, utilizando una sola tabla de referencia

Como paso siguiente se implementó la aplicación y sus funcionalidades específicas. En la figura se muestra interface principal de la aplicación móvil (ver Figura 9).



Figura 9. Interface principal de la aplicación móvil

En la figura se muestra la ventana de iniciar monitoreo donde se configura la App EcoFavBen, ingresando la dirección del servidor y el intervalo de tiempo para la toma de datos (ver Figura 10).

CONFIGURACIÓN

Ip: Servidor:
http://192.168.10.12:8080

Tiempo Act. (Seg.):
1

Seleccione Un Dispositivo

Figura 10. Ventana de iniciar monitoreo

En la figura se muestra el apartado donde se configura la dirección del servidor, con el fin de que se pueda sincronizar los datos guardados de manera interna (ver Figura 11).

The screenshot displays a web interface for synchronizing data with a server. The title is "Sincronizar Datos Con Servidor". Below the title, there is a label "Ip: Servidor:" followed by a text input field containing the URL "http://192.168.10.12:8080". A grey button labeled "PROBAR CONECCIÓN" is positioned below the input field. Underneath, there is a checkbox labeled "Servidor Conectado" which is currently unchecked. Below the checkbox, the text "Datos Por Sincronizar:" is followed by the number "0". At the bottom of the interface, there is a prominent blue button labeled "SINCRONIZAR DATOS".

Figura 11. Configuración de la dirección del servidor

1.3. Sprint 03

El tercer sprint tiene como objetivo implementar la Web API, la cual tendrá como objetivo principal recibir y procesar datos desde el aplicativo móvil. En la tabla se muestra el sprint backlog del sprint 3 (ver Tabla 6).

Tabla 6 *Sprint backlog del sprint 3*

Id	Sprint	Tarea	Historial de Usuario	Responsable	Duración
ID3	3	Guardar datos de aplicativo móvil en base de datos	Guardar datos de aplicativo móvil	Erick	3
		Asignar cada dato a un punto de referencia más cercano guardado en el sistema	Asignar dato a un punto de referencia	Noé	1

1.3.1. Desarrollo del sprint 03

La Web API se ha desarrollado utilizando lenguaje de programación C#, y el IDE Visual Studio 2019 Community, y tiene los siguientes requerimientos específicos:

- Guardar datos de la aplicación móvil.
- Asignar a cada dato un punto de referencia en el mapa de Cajamarca.

El motor de base de datos que se utiliza en el sistema EcoFavBen es SQL Server 2017 Express Edition, con el esquema de base de datos mostrada (ver Figura 12).

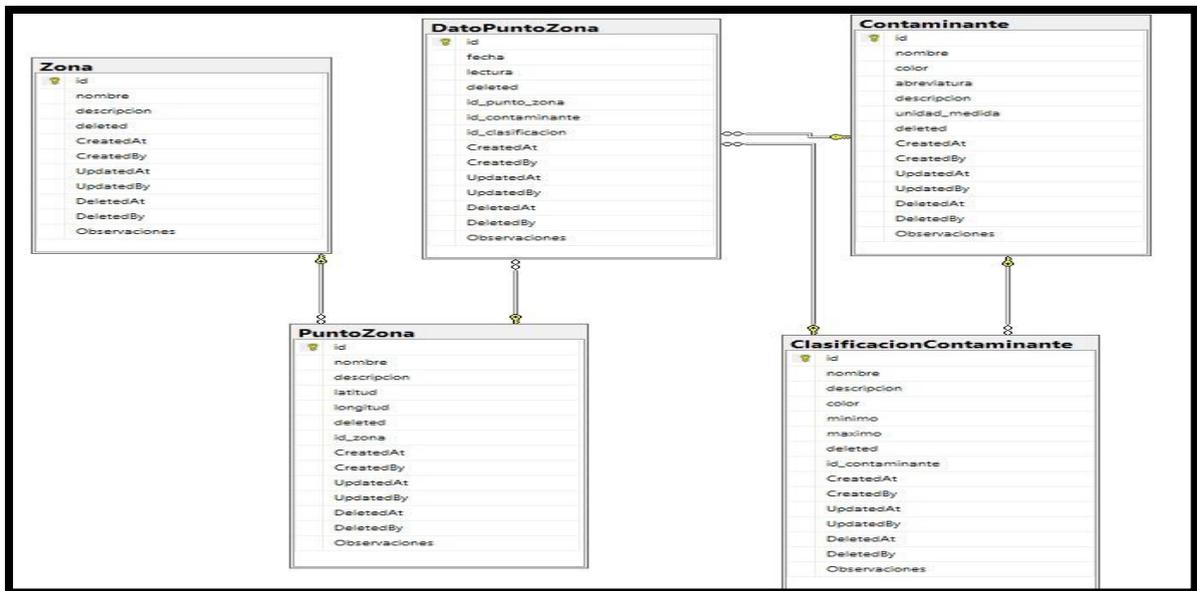


Figura 12. Esquema de base de datos

Una vez realizados la implementación y el despliegue de la Web API, se realizan las primeras pruebas de conexión utilizando la herramienta PostMan (ver Figura 13 y Figura 14).

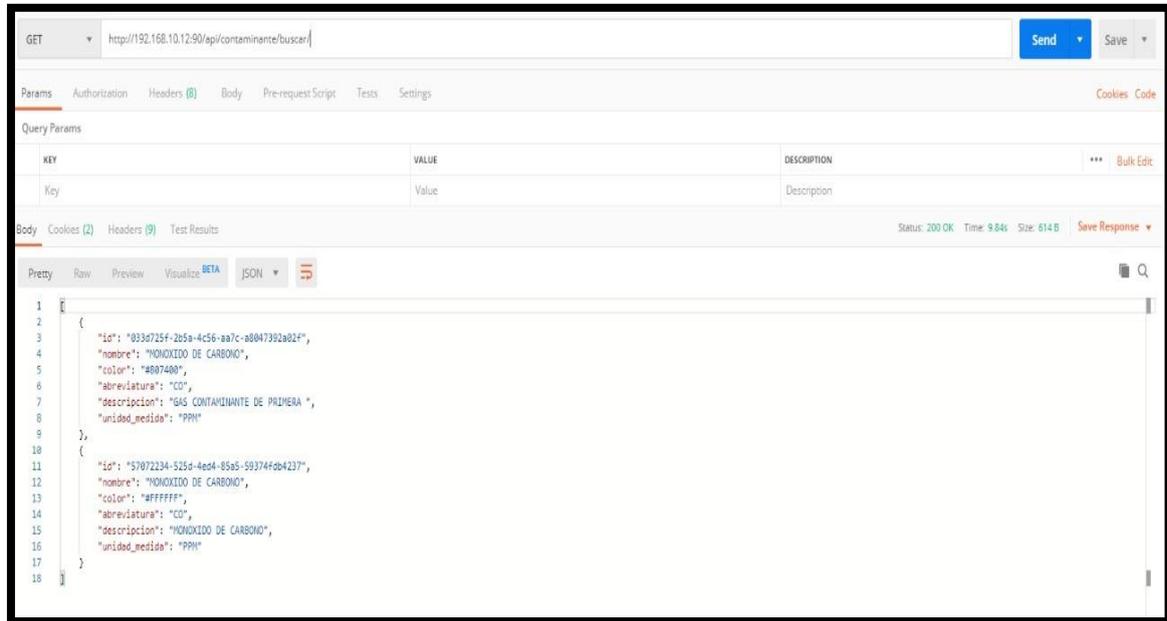


Figura 13. Pruebas de conexión utilizando la herramienta PostMan

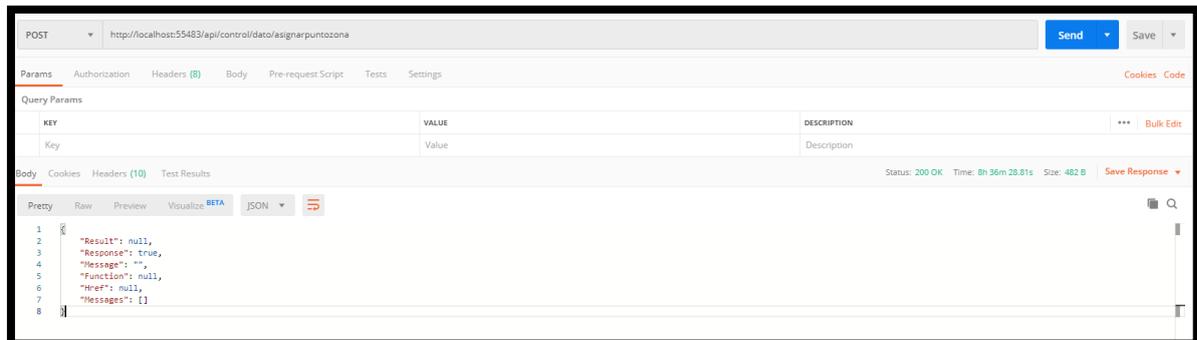


Figura 14. Pruebas de conexión utilizando la herramienta PostMan

1.4. Sprint 04

El cuarto sprint tiene como objetivo el diseño e implementación del aplicativo web, cuya función principal es mostrar información procesada de los datos guardados desde el aplicativo móvil, que a su vez ha recibido información del módulo de lectura (Arduino). En la tabla se muestra el sprint backlog del sprint 4 (ver Tabla 7).

Tabla 7 *Sprint backlog del sprint 4*

Id	Sprint	Tarea	Historial de Usuario	Responsable	Duración
ID3	3	Gestionar zona	Guardar zona	Erick	1
			Editar zona	Erick	1
			Eliminar zona	Erick	1
			Buscar zona	Erick	1
			Lista zonas	Erick	1
		Gestionar punto de control	Guardar punto de control	Noe	1
			Editar punto de control	Noe	1
			Eliminar punto de control	Noe	1
			Buscar punto de control	Noe	1
			Listar punto de control	Noe	1
		Datos por zona	ver datos por zona	Erick	1
		Datos por punto de control	Ver datos por punto de control	Noe	1
		Predecir datos utilizando algoritmo de Machine Learning	Ver Predicción de datos	Erick	2

1.4.1. Desarrollo del sprint 04

La Web API se ha desarrollado utilizando lenguaje de programación C#, y el IDE Visual Studio 2019 Community. También se ha utilizado el patrón de desarrollo MVC.

En la figura se muestran los primeros resultados de contaminación por zonas (ver Figura 15).



Figura 15. Primeros resultados de contaminación por zonas

Dentro de una zona se puede tener muchos puntos de control, por ello podemos ver los puntos de control que estén asociados a una determinada zona. En la figura se muestra un ejemplo de los puntos de control de una zona registrada en el sistema (ver Figura 16).



Figura 16. Puntos de control de una zona registrada en el sistema

Objetivo 2: Identificar y clasificar las zonas con mayores niveles de CO en la ciudad de Cajamarca.

Los niveles establecidos en la norma del Ministerio del Ambiente (MINAM), menciona que el nivel máximo de concentración de CO es de $10000 \mu\text{g}/\text{m}^3$ lo que equivale a 9 ppm, con una exposición máxima de 8 horas (Ministerio del Ambiente - Perú, 2017) (ver Anexo 5).

Con el fin de recolectar información de forma concisa, se ha establecido 12 zonas en la ciudad de Cajamarca (ver Tabla 8).

Tabla 8 *Zonas en la ciudad de Cajamarca*

Nombre	Descripción
La colmena	Barrio la colmena
Fonavi II	Fonavi II, barrio San Antonio
San Martín	Barrio San Martín
Víctor Raúl	Barrio Víctor Raúl
San roque	Barrio San roque
Pueblo nuevo	Barrio pueblo nuevo
San Sebastián	Barrio San Sebastián
Qhapac ñan	Gran Qhapac ñan
Joven amauta	Barrio joven amauta
San Antonio	Barrio San Antonio
Aranjuez	Barrio Aranjuez
Dos de mayo	Barrio dos de mayo

De la misma manera se han determinado puntos de control, dentro de cada zona. Estos puntos han sido seleccionados estratégicamente, teniendo en cuenta el nivel de tránsito automotor.

En la tabla se pueden ver los puntos seleccionados, y la zona a la que pertenecen (ver Tabla 9).

Tabla 9 *Puntos seleccionados, y la zona a la que pertenecen*

Punto Nombre	Descripción	Zona
Upn	Frente A La Universidad Privada Del Norte	San Antonio
Recoleta	Frente A La Iglesia La Recoleta	San Sebastian
Puente Amarillo	Puente Amarillo	La Colmena
Fonavi II	Intersección Entre Vía De Evitamiento Y Av, Hoyos Rubio	Fonavi II
Plazuela Bolognesi	Plazuela Bolognesi	San Sebastian
Urrelo - M. Urteaga	Interseccion Jr. Urrle Y Av. Mario Urteaga	Joven Amauta
Ovalo Musical	Intersección Av. Atahualpa Con Via De Evitamiento	Victor Raul
Plaza De Armas	Pileta	Dos De Mayo
Quinde	Quinde	Pueblo Nuevo
Sucre	Intersección Entre Av. Atahualpa Y Jr Sucre	San Sebastian
Ovalo El Inca	Ovalo El Inca	La Colmena
2 De Mayo - M. Urteaga	Interseccion Jr. Dos De Mayo Y Jr. Mario Urteaga	Joven Amauta
Real Plaza	Real Plaza	San Roque
Cruce De Jesús	Intersección Entre Av San Martín De Porres Y Av Atahualpa	San Martín
M. Seoane - H. Rubio	Intersección Manuel Seoane Y Hoyos Rubio	Fonavi II
Qhapac Ñan	Gran Qhapac Ñan	Qhapac Ñan

En la figura se puede observar la distribución de los puntos dentro del mapa de la ciudad de Cajamarca (ver Figura 16).

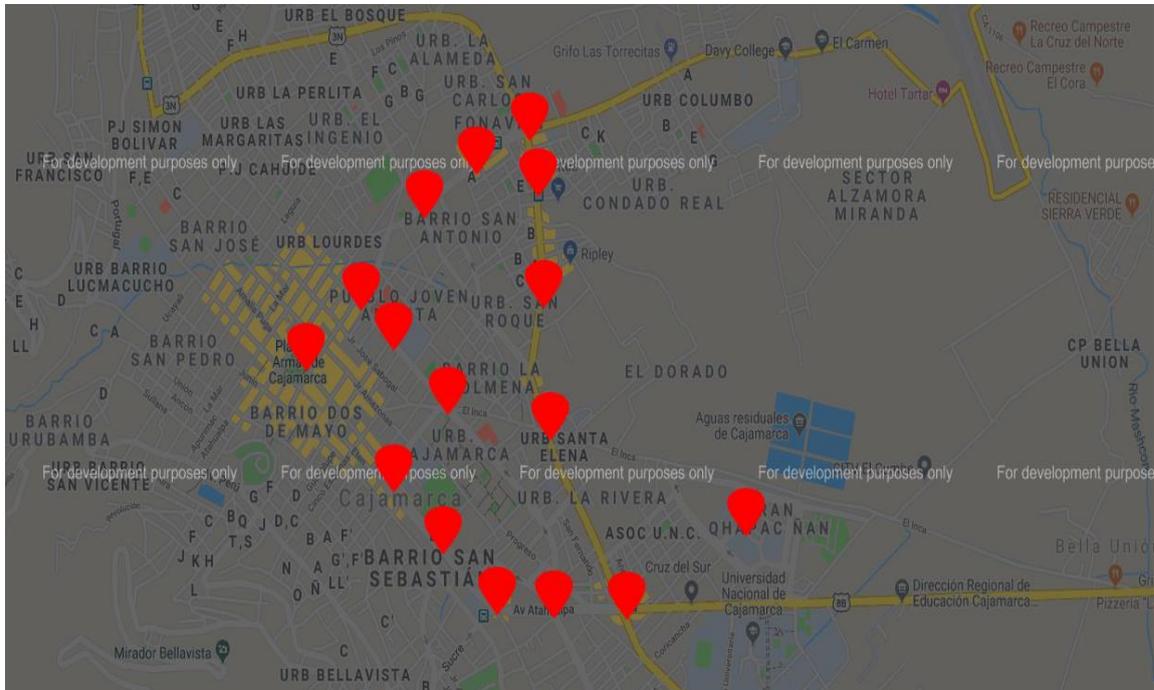


Figura 17. Puntos dentro del mapa de la ciudad de Cajamarca

En base a lo referido se ha optado por hacer una recolección de datos teniendo en cuenta lo siguiente:

- Los datos de nivel de CO serán tomados y guardados cada segundo.
- Se hará un recorrido portando el módulo de lectura (Arduino) por la ciudad de Cajamarca recolectando datos. Estos datos serán guardados de forma local en primera instancia para luego ser migrados al servidor para ser procesados.
- Los datos recolectados estarán asociados a coordenadas de ubicación (longitud y latitud), lo que permitirá asociarlos a un punto de control de referencia, y por ende a una zona en específico.

1.5. Primera Recolección de datos

La primera recolección de datos se realizó en un periodo de 12 días, ya que los periodos (días) en los que se realizó la toma de datos, fueron muy pocos, porque se tuvo un tiempo limitado para realizar la investigación, haciendo recorridos por la ciudad de Cajamarca, teniendo en cuenta las horas de mayor concentración vehicular, teniendo como referencia el registro del volumen vehicular tomado del Plan de Desarrollo Urbano de la municipalidad Provincial de Cajamarca en el *capítulo IV: Diagnostico Urbano*, muestra los puntos con mayor congestión vehicular de la ciudad en las intersecciones de la Vía de evitamiento - hoyos rubio, Vía de evitamiento - jr. El inca, Vía de evitamiento - jr. Soane, Ovalo musical, Av. San Martín - av. Atahualpa, Av. Sucre - av. Atahualpa Av. De los héroes - av. El maestro (Municipalidad de Cajamarca, 2020).

1.5.1. Punto de Control: 2 de mayo – Mario Urteaga

En el punto de “INTERSECCION JR. DOS DE MAYO Y JR. MARIO URTEAGA”, en los 12 días de recolección de datos se obtuvo un promedio de 30.33 partes por millón (PPM). Información que se detalla (ver Tabla 10).

Tabla 10 *Días de recolección de datos 2 de mayo – Mario Urteaga*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	33055	7,66	5379
2	1	14461	34,80	6534
3	0	1783	37,72	6463
4	1	46966	27,39	5206
5	0	20928	41,60	3050
6	1	27621	39,74	4964
7	0	10311	46,96	6070
8	0	10066	19,12	5361
9	1	12025	36,78	3550
10	0	41403	45,01	3766
11	0	29600	14,74	6507
12	0	16544	12,44	7272
Total			30,33	64122

1.5.2. Punto de Control: Cruce de Jesús

En el punto de “INTERSECCIÓN ENTRE AV SAN MARTÍN DE PORRES Y AV ATAHUALPA”, en los 12 días de recolección de datos se obtuvo un promedio de 27.65 partes por millón (PPM). Información que se detalla (ver Tabla 11).

Tabla 11 *Días de recolección de datos Cruce de Jesús*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	31291	39,68	3695
2	1	48556	32,08	7384
3	0	44889	37,84	7914
4	1	6355	14,77	6552
5	0	24579	13,08	6123
6	1	23681	21,02	4549
7	0	20470	42,73	5041
8	0	31176	37,74	4297
9	1	31897	21,17	5583
10	0	16220	14,92	5464
11	0	8456	30,64	6953
12	0	14725	26,13	6216
Total			27,65	69771

1.5.3. Punto de Control: Qhapac Ñan

En el punto de "GRAN QHAPAC ÑAN", en los 12 días de recolección de datos se obtuvo un promedio de 0.66 partes por millón (PPM). Información que se detalla (ver Tabla 12).

Tabla 12 *Días de recolección de datos Qhapac Ñan*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	0	1	0,50	2219
2	0	2	0,47	1334
3	0	1	0,50	2439
4	0	3	0,51	1543
5	0	1	0,51	1079
6	0	4	0,51	2243
7	0	3	0,51	2258
8	0	2	0,52	1098
9	0	3	1,00	6509
10	0	2	1,52	2978
11	0	2	0,50	3117
12	0	203	0,93	2038
Total			0,66	28855

1.5.4. Punto de Control: Recoleta

En el punto de “RECOLETA”, en los 12 días de recolección de datos se obtuvo un promedio de 17.63 partes por millón (PPM). Información que se detalla (ver Tabla 13).

Tabla 13 *Días de recolección de datos Recoleta*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	2569	3,29	4437
2	1	5569	3,77	6505
3	0	1268	2,60	6355
4	1	3596	3,86	6321
5	0	126598	3,86	2988
6	1	4698	60,95	4531
7	0	29884	6,99	5962
8	0	21549	12,16	4816
9	1	26465	20,92	6176
10	0	1654	14,13	2545
11	0	458496	73,53	6843
12	0	16845	5,50	4844
Total			17,63	62323

1.5.5. Punto de Control: Universidad privada del norte

En el punto de “UPN”, en los 12 días de recolección de datos se obtuvo un promedio de 19.36 partes por millón (PPM). Información que se detalla (ver Tabla 14).

Tabla 14 *Días de recolección de datos Universidad privada del norte*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	3696	3,27	5902
2	1	30888	11,31	6917
3	0	29601	10,28	5907
4	1	24181	24,18	6784
5	0	31193	15,81	4770
6	1	17295	9,23	7148
7	0	34345	13,48	7609
8	0	37015	39,69	4209
9	1	37214	33,48	4516
10	0	5382	23,01	3625
11	0	31813	17,34	3528
12	0	11647	31,20	7699
Total			19,36	68614

1.5.6. Punto de Control: Puente amarillo

En el punto de “PUENTE AMARILLO”, en los 12 días de recolección de datos se obtuvo un promedio de 24.46 partes por millón (PPM). Información que se detalla (ver Tabla 15).

Tabla 15 *Días de recolección de datos Puente amarillo*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	40107	17,85	7093
2	1	34076	32,14	4468
3	0	16895	28,34	7154
4	1	23908	27,21	6762
5	0	21881	38,80	6884
6	1	33191	11,60	3970
7	0	31189	11,60	6994
8	0	45243	6,92	4253
9	1	43479	30,98	3198
10	0	12227	8,65	4250
11	0	23251	32,99	3983
12	0	29162	46,39	4982
Total			24,46	63991

1.5.7. Punto de Control: Fonavi II

En el punto de “FONAVI II”, en los 12 días de recolección de datos se obtuvo un promedio de 25.44 partes por millón (PPM). Información que se detalla (ver Tabla 16).

Tabla 16 *Días de recolección de datos Fonavi II*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	3780	30,41	7610
2	1	33059	48,15	7261
3	0	18998	15,69	5165
4	1	8964	9,81	7896
5	0	2026	33,75	6692
6	1	6963	9,40	7394
7	0	18149	23,07	6134
8	0	37704	48,25	5018
9	1	33528	32,58	6431
10	0	4427	27,78	3575
11	0	17892	13,51	4353
12	0	15223	12,88	4282
Total			25,44	71811

1.5.8. Punto de Control: Plazuela Bolognesi

En el punto de "PLAZUELA BOLOGNESI", en los 12 días de recolección de datos se obtuvo un promedio de 32.38 partes por millón (PPM). Información que se detalla (ver Tabla 17).

Tabla 17 *Días de recolección de datos Plazuela Bolognesi*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	30614	41,65	5649
2	1	37647	22,81	6354
3	0	35252	48,68	7718
4	1	16138	35,17	3334
5	0	15043	42,74	7317
6	1	6590	38,09	4282
7	0	21854	4,31	7747
8	0	22408	37,18	3911
9	1	46245	33,53	5226
10	0	4648	37,31	5326
11	0	15849	20,24	7495
12	0	3903	26,85	7503
Total			32,38	71862

1.5.9. Punto de Control: Urrelo – M. Urteaga

En el punto de “INTERSECCION JR. URRELO Y AV. MARIO URTEAGA”, en los 12 días de recolección de datos se obtuvo un promedio de 26.54 partes por millón (PPM). Información que se detalla (ver Tabla 18).

Tabla 18 *Días de recolección de datos Urrelo – M. Urteaga*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	24860	39,10	4164
2	1	26165	33,64	3094
3	0	2490	5,54	4566
4	1	32979	36,41	3393
5	0	41518	23,72	7841
6	1	7129	41,98	6782
7	0	34473	31,51	3945
8	0	14559	12,60	4866
9	1	2092	16,30	7695
10	0	3956	24,64	3853
11	0	25589	7,78	3013
12	0	49110	45,22	5736
Total			26,54	58948

1.5.10. Punto de Control: Ovalo Musical

En el punto de "INTERSECCIÓN AV. ATAHUALPA CON VIA DE EVITAMIENTO", en los 12 días de recolección de datos se obtuvo un promedio de 31.57 partes por millón (PPM). Información que se detalla en la (ver Tabla 19).

Tabla 19 *Días de recolección de datos Ovalo Musical*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	44609	29,30	6958
2	1	46815	38,06	7502
3	0	47075	27,62	3142
4	1	48175	48,37	3060
5	0	48153	16,78	4503
6	1	29614	23,78	3967
7	0	11913	46,21	4854
8	0	27939	40,76	3462
9	1	14697	32,82	6388
10	0	18460	6,14	3466
11	0	15583	48,90	3308
12	0	36825	20,14	3678
Total			31,57	54288

1.5.11. Punto de Control: Plaza de Armas

En el punto de "PLAZA DE ARMAS", en los 12 días de recolección de datos se obtuvo un promedio de 5.56 partes por millón (PPM). Información que se detalla (ver Tabla 20).

Tabla 20 *Días de recolección de datos Plaza de Armas*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	6650	3,14	6309
2	1	4760	1,19	5513
3	0	8758	8,28	7647
4	1	2061	3,36	5558
5	0	9549	4,90	7860
6	1	2819	9,61	7480
7	0	1707	8,75	5588
8	0	4101	4,92	7592
9	1	6276	3,60	5470
10	0	1846	8,13	7095
11	0	8167	7,97	3147
12	0	7128	2,85	5200
Total			5,56	74459

1.5.12. Punto de Control: El Quinde

En el punto de “CENTRO COMERCIAL EL QUINDE”, en los 12 días de recolección de datos se obtuvo un promedio de 29.44 partes por millón (PPM). Información que se detalla (ver Tabla 21).

Tabla 21 *Días de recolección de datos El Quinde*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	10896	32,25	6140
2	1	1016	49,83	3379
3	0	48539	25,07	5141
4	1	33071	35,89	3020
5	0	37290	9,90	6971
6	1	2576	45,22	7013
7	0	18257	22,45	3355
8	0	47003	21,79	4181
9	1	15124	46,83	4656
10	0	836	5,92	7353
11	0	5475	19,88	7750
12	0	10644	38,26	6097
Total			29,44	65056

1.5.13. Punto de Control: Sucre

En el punto de "INTERSECCIÓN ENTRE AV. ATAHUALPA Y JR SUCRE", en los 12 días de recolección de datos se obtuvo un promedio de 20.09 partes por millón (PPM). Información que se detalla (ver Tabla 22).

Tabla 22 *Días de recolección de datos Sucre*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	19623	26,38	6224
2	1	9148	2,05	6968
3	0	2571	12,71	5626
4	1	39912	6,07	6483
5	0	12938	22,74	6927
6	1	47236	6,20	5857
7	0	24600	42,22	3455
8	0	1264	13,46	3218
9	1	16218	10,97	7535
10	0	47398	49,99	3495
11	0	27536	37,78	5097
12	0	41580	10,46	3068
Total			20,09	63953

1.5.14. Punto de Control: Ovalo el Inca

En el punto de "OVALO EL INCA", en los 12 días de recolección de datos se obtuvo un promedio de 30.56 partes por millón (PPM). Información que se detalla (ver Tabla 23).

Tabla 23 *Días de recolección de datos Ovalo el Inca*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	18143	15,94	6840
2	1	30780	3,52	5544
3	0	28568	38,04	7328
4	1	987	22,22	4652
5	0	1934	3,54	7516
6	1	4324	44,50	7566
7	0	8263	27,55	7883
8	0	35008	43,46	5017
9	1	21855	27,06	4861
10	0	17332	47,42	3985
11	0	22339	46,73	7724
12	0	6625	46,74	6388
Total			30,56	75304

1.5.15. Punto de Control: Real Plaza

En el punto de “CENTRO COMERCIAL REAL PLAZA”, en los 12 días de recolección de datos se obtuvo un promedio de 18.32 partes por millón (PPM). Información que se detalla (ver Tabla 24).

Tabla 24 *Días de recolección de datos Real Plaza*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	36033	23,94	6678
2	1	9078	6,05	5140
3	0	32870	13,84	3079
4	1	45492	3,02	7333
5	0	37385	27,76	7983
6	1	525	14,53	4219
7	0	23967	9,46	3240
8	0	2054	5,79	6409
9	1	25892	2,20	3253
10	0	28425	49,38	5810
11	0	22317	41,91	4134
12	0	46228	21,98	3971
Total			18,32	61249

1.5.16. Punto de Control: M. Seoane – H. Rubio

En el punto de “INTERSECCIÓN MANUEL SEOANE Y HOYOS RUBIO”, en los 12 días de recolección de datos se obtuvo un promedio de 26.73 partes por millón (PPM). Información que se detalla (ver Tabla 25).

Tabla 25 *Días de recolección de datos M. Seoane – H. Rubio*

DIA	MINIMO	MAXIMO	PROMEDIO	CANTIDAD
1	1	31536	16,40	5737
2	1	38071	38,96	7651
3	0	2590	8,56	6925
4	1	45835	49,33	7157
5	0	4642	26,29	4371
6	1	1261	6,91	7183
7	0	37344	29,75	5389
8	0	35243	4,97	6712
9	1	30558	39,01	4474
10	0	39369	36,93	3360
11	0	33733	24,72	7979
12	0	23367	38,90	3230
Total			26,73	70168

1.6. Interpretación de datos

Después de haber hecho una toma de datos de manera aleatoria se ha conseguido obtener un resumen del promedio de contaminación por cada punto de control (ver Tabla 26).

Tabla 26 *Promedio de contaminación por cada punto de control*

Punto de Control	Promedio
UPN	19,36
Recoleta	17,63
Puente Amarillo	24,64
Fonavi Ii	25,44
Plazuela Bolognesi	32,38
Urrelo - M. Urteaga	26,54
Ovalo Musical	31,57
Plaza De Armas	5,56
Quinde	29,44
Sucre	20,09
Ovalo El Inca	30,56
2 De Mayo - M. Urteaga	30,33
Real Plaza	18,32
Cruce De Jesús	27,65
M. Seoane - H. Rubio	26,63
Qhapac Ñan	0,66
PROMEDIO GENERAL	22,93

Se observó que el promedio más alto se corresponde al punto de control “PLAZUELA BOLOGNESI”, con un promedio de 32.38 partículas por millón (PPM), y el promedio más bajo le corresponde al punto de control “QHAPCA ÑAN”, con un puntaje de 0.66 partes por millón (PPM). Esto se puede observar de forma gráfica (ver Figura 18).

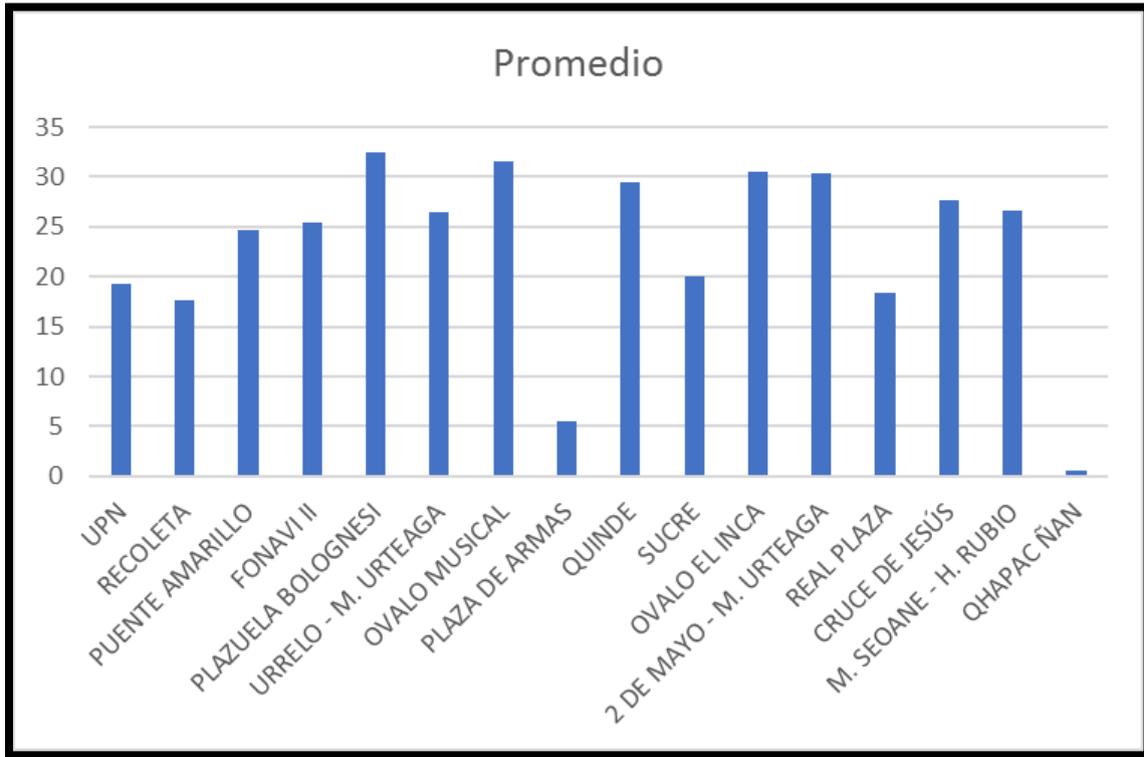


Figura 18. Promedio de contaminación por cada punto de control

Objetivo 3: Realizar predicción de niveles de monóxido de carbono en la ciudad de Cajamarca, en función de los datos obtenidos por el sistema EcoFavBen.

1.7. Predicción de Datos

Para la predicción de datos se hará uso de una librería denominada “ML.Net”, cuyo creador es la empresa de software Microsoft, y tiene funciones específicas para trabajar con machine learning.

ML.Net tiene un módulo de entrenamiento automático en base a datos específicos. Con ayuda de esta herramienta y los datos obtenidos por el sistema EcoFavBen se ha hecho el procedimiento de entrenamiento para poder elegir el mejor algoritmo que se adapte al tipo de datos que se ha recopilado.

En la tabla se observan los datos ingresados al entrenamiento automático (ver Tabla 27).

Tabla 27 Datos ingresados al entrenamiento automático

	1	2	3	4	5	6	7	8	9	10	11	12
UPN	3,2 7	11, 31	10, 28	24, 18	15, 81	9,2 3	13, 48	39, 69	33, 48	23, 01	17, 34	31, 2
RECOLET A	3,2 9	3,7 7	2,6	3,8 6	3,8 6	60, 95	6,9 9	12, 16	20, 92	14, 13	73, 53	5,5
PUENTE AMARILL O	17, 85	32, 14	28, 34	27, 21	38, 8	11, 6	11, 6	6,9 2	30, 98	8,6 5	32, 99	46, 39
FONAVI II	30, 41	48, 15	15, 69	9,8 1	33, 75	9,4	23, 07	48, 25	32, 58	27, 78	13, 51	12, 88
PLAZUEL A BOLOGNE SI	41, 65	22, 81	48, 68	35, 17	42, 74	38, 09	4,3 1	37, 18	33, 53	37, 31	20, 24	26, 85
URRELO - M. URTEAGA	39, 1	33, 64	5,5 4	36, 41	23, 72	41, 98	31, 51	12, 6	16, 3	24, 64	7,7 8	45, 22
OVALO MUSICAL	29, 3	38, 06	27, 62	48, 37	16, 78	23, 78	46, 21	40, 76	32, 82	6,1 4	48, 9	20, 14
PLAZA DE ARMAS	3,1 4	1,1 9	8,2 8	3,3 6	4,9	9,6	8,7 5	4,9 2	3,6	8,1 3	7,9 7	2,8 5
QUINDE	32, 25	49, 83	25, 07	35, 89	9,9	45, 22	22, 45	21, 79	46, 83	5,9 2	19, 88	38, 26
SUCRE	26, 38	2,0 5	12, 71	6,0 7	22, 74	6,2	42, 22	13, 46	10, 97	49, 99	37, 78	10, 46
OVALO EL INCA	15, 94	3,5 2	38, 04	22, 22	3,5 4	44, 5	27, 55	43, 46	27, 06	47, 42	46, 73	46, 74
2 DE MAYO - M. URTEAGA	7,6 6	34, 8	37, 72	27, 39	41, 6	39, 74	46, 96	19, 12	36, 78	45, 01	14, 74	12, 44
REAL PLAZA	23, 94	6,0 5	13, 84	3,0 2	27, 76	14, 53	9,4 6	5,7 9	2,2	49, 38	41, 91	21, 98
CRUCE DE JESÚS M.	39, 68	32, 08	37, 84	14, 77	13, 08	21, 02	42, 73	37, 74	21, 17	14, 92	30, 64	26, 13
SEOANE - H. RUBIO	16, 4	38, 96	8,5 6	49, 33	26, 29	6,9 1	29, 75	4,9 7	39, 01	36, 93	24, 72	38, 9
QHAPAC ÑAN	0,5	0,4 7	0,5	0,5 1	0,5 1	0,5 1	0,5 1	0,5 2	1	1,5 2	0,5	0,9 3

Como resultado del entrenamiento automático obtenemos (ver tabla 28).

Tabla 28 *Resultado del entrenamiento automático*

Algoritmo	R-Squared	Absolute Loss	Squared Loss	RMS Loss	Duración
Sdca	0,9002	0,84	9,48	3,08	5,2
LightGbm	0,9339	0,42	6,28	2,51	4,2
FastTree	0,9411	0,42	5,59	2,36	4,2
FastTreeTweedie	0,9424	0,41	5,47	2,34	2,5
FastForest	0,8768	1,37	11,7	3,42	5,1
Ols	0,9032	0,73	9,19	3,03	1,8

Fuente: (Microsoft, 2020)

Lo que se puede observar es que el algoritmo con menos margen de error es el FastTreeTweedie, con un R-Square de 9.424. Lo que lo convierte en el algoritmo seleccionado para esta investigación (ver Anexo 6).

1.7.1. Implementación del Algoritmo FastTreeTweedie

Gracias a ML.Net se puede tener acceso a algoritmos de predicción de una manera sencilla, reduciendo el tamaño de código fuente.

Como primer paso, se generó un modelo de entrenamiento en base a los datos establecidos y al algoritmo seleccionado, en este caso el FastTreeTweedie. ML.Net crea un modelo de entrenamiento con extensión zip, el cual nos va a servir para generar datos de predicción.

En la figura se observa un fragmento de código donde se inicializa el objeto de entrenamiento del algoritmo FastTreeTweedie (ver Figura 19).

```
var trainer = mlContext.Regression.Trainers.FastTreeTweedie(labelColumnName: "Label", featureColumnName: "Features", minimumExampleCountPerLeaf: 5);
```

Figura 19. Código donde se inicializa el objeto de entrenamiento

Una vez creado el objeto de entrenamiento se crea un contexto utilizando el modelo de entrenamiento y los datos con los que el modelo va a ser entrenado (ver Figura 20).

```
var trainingPipeline = mlContext.Transforms.Concatenate(outputColumnName: "NumFeatures",  
    nameof(DatosCo.year), nameof(DatosCo.month), nameof(DatosCo.units),  
    nameof(DatosCo.avg), nameof(DatosCo.count),  
    nameof(DatosCo.max), nameof(DatosCo.min), nameof(DatosCo.prev))  
.Append(mlContext.Transforms.Categorical.OneHotEncoding(outputColumnName: "CatFeatures",  
    inputColumnName: nameof(DatosCo.datotId)))  
.Append(mlContext.Transforms.Concatenate(outputColumnName: "Features", "NumFeatures", "CatFeatures"))  
.Append(mlContext.Transforms.CopyColumns(outputColumnName: "Label",  
    inputColumnName: nameof(DatosCo.next)))  
.Append(trainer);
```

Figura 20. Creación del contexto utilizando el modelo de entrenamiento

Ahora se valida el contexto (ver Figura 21).

```
var crossValidationResults = mlContext.Regression.CrossValidate(data: trainingDataView,  
    estimator: trainingPipeline, numberOfFolds: 6, labelColumnName: "Label");
```

Figura 21. Validación del contexto

Posteriormente se entrena el modelo con el algoritmo FastTreeTweedie (ver Figura 22).

```
var model = trainingPipeline.Fit(trainingDataView);
```

Figura 22. Entrenamiento del modelo

Y por último se guarda el modelo para utilizarlo posteriormente (ver Figura 23).

```
mlContext.Model.Save(model, trainingDataView.Schema, outputModelPath);
```

Figura 23. Guardar el modelo

Una vez generado el modelo entrenado, en la figura se observa un fragmento de código fuente donde hacemos uso de dicho modelo e ingresando como referencia el último dato real que se haya obtenido (ver Figura 24).

```
ITransformer trainedModel;  
using (var stream = File.OpenRead(outputModelPath))  
{  
    trainedModel = mlContext.Model.Load(stream, out var modelInputSchema);  
}
```

Figura 24. Fragmento de código fuente donde hizo uso del modelo

Después de haber cargado el modelo, se procede a obtener el dato de predicción (ver Figura 25).

```
var predictionEngine = mlContext.Model.CreatePredictionEngine<DatosCo, PreccionProgresion>(trainedModel);  
PreccionProgresion prediction = predictionEngine.Predict(dato);
```

Figura 25. Dato predicho

1.7.2. Datos Obtenidos con el algoritmo FastTreeTweedie

Una vez implementado el algoritmo de predicción en el sistema EcoFavBen y teniendo el modelo de entrenamiento, se realizó la predicción de 3 días, en base a lo ingresado al sistema anteriormente, solo se pudo predecir estos 3 días porque el algoritmo Árbol de decisión necesita periodos de entrenamiento considerables a razón de la cantidad de datos que se tenga, ya que según, León (2018), verifico que la precisión de la predicción de las clases depende del tamaño del conjunto de datos al cual pertenece la clase, por lo cual haciendo un cálculo simple de comparación con los datos captados en los 12 días de recolección, solo se pudo tener como límite máximo la predicción de 3 días para que tenga una mayor precisión. Obteniendo los datos descritos (ver Tabla 29).

Tabla 29 *Predicción de niveles de CO*

Punto de Control	P01	P02	P03
UPN	21,15	25,13	22,17
RECOLETA	17,63	14,09	16,40
PUENTE AMARILLO	21,96	24,02	24,04
FONAVI II	23,85	20,78	25,12
PLAZUELA BOLOGNESI	33,75	41,22	33,81
URRELO - M. URTEAGA	28,60	28,54	30,56
OVALO MUSICAL	28,50	22,19	26,37
PLAZA DE ARMAS	6,13	5,90	6,32
QUINDE	28,21	27,98	22,87
SUCRE	20,77	26,60	20,44
OVALO EL INCA	28,71	20,21	25,77
2 DE MAYO - M. URTEAGA	30,67	35,93	34,20
REAL PLAZA	17,26	18,20	14,96
CRUCE DE JESÚS	29,79	28,16	32,50
M. SEOANE - H. RUBIO	28,15	26,83	31,82
QHAPAC ÑAN	0,65	0,53	0,58

1.8. Segunda Recolección de datos

En esta etapa de la investigación se buscó comparar los datos de predicción por parte del algoritmo, con datos del entorno real.

Basado en el primer día obtenemos los resultados que se describen en la tabla. Se observa que la diferencia es significativa, teniendo un error relativo de 44% (ver Tabla 30).

Tabla 30 *Resultados Obtenidos día 1*

Punto De Control	P01	Dato Real	Error Absoluto	Error Relativo (%)
UPN	21,15	23,37	-2,22	9%
RECOLETA	17,63	19,04	-1,41	7%
PUENTE AMARILLO	21,96	21,75	0,21	1%
FONAVI II	23,85	21,4	2,45	11%
PLAZUELA BOLOGNESI	33,75	37,28	-3,53	9%
URRELO - M. URTEAGA	28,6	32,91	-4,31	13%
OVALO MUSICAL	28,5	32,52	-4,02	12%
PLAZA DE ARMAS	6,13	11	-4,87	44%
QUINDE	28,21	32,79	-4,58	14%
SUCRE	20,77	19,67	1,1	6%
OVALO EL INCA	28,71	31,18	-2,47	8%
2 DE MAYO - M. URTEAGA	30,67	25,95	4,72	18%
REAL PLAZA	17,26	20,26	-3	15%
CRUCE DE JESÚS	29,79	28,21	1,58	6%
M. SEOANE - H. RUBIO	28,15	27,16	0,99	4%
QHAPAC ÑAN	0,65	0,58	-0,07	12%

En el caso del día 2, los datos se detallan en la tabla. Se observa que la diferencia es significativa, teniendo un error relativo de 92% (ver Tabla 31).

Tabla 31 *Resultados Obtenidos día 2*

Punto de Control	P02	Dato Real	Error Absoluto	Error Relativo (%)
UPN	25,13	19,65	5,48	28%
RECOLETA	14,09	11,51	2,58	22%
PUENTE AMARILLO	24,02	24,41	-0,39	2%
FONAVI II	20,78	17,32	3,46	20%
PLAZUELA BOLOGNESI	41,22	37,02	4,2	11%
URRELO - M. URTEAGA	28,54	33,36	-4,82	14%
OVALO MUSICAL	22,19	21,27	0,92	4%
PLAZA DE ARMAS	5,9	3,07	2,83	92%
QUINDE	27,98	24,67	3,31	13%
SUCRE	26,6	23,32	3,28	14%
OVALO EL INCA	20,21	22,89	-2,68	12%
2 DE MAYO - M. URTEAGA	35,93	40,87	-4,94	12%
REAL PLAZA	18,2	21,13	-2,93	14%
CRUCE DE JESÚS	28,16	32,52	-4,36	13%
M. SEOANE - H. RUBIO	26,83	28,89	-2,06	7%
QHAPAC ÑAN	0,53	0,48	-0,05	10%

En el caso del día 3, los detalles se aprecian en la tabla. Se observa que la diferencia es significativa, teniendo un error relativo de 33% (ver Tabla 32).

Tabla 32 *Resultados Obtenidos día 3*

Punto de Control	P03	Dato Real	Error Absoluto	Error Relativo (%)
UPN	22,17	24,23	-2,06	9%
RECOLETA	16,4	13,84	2,56	18%
PUENTE AMARILLO	24,04	29,02	-4,98	17%
FONAVI II	25,12	31,44	-6,32	20%
PLAZUELA BOLOGNESI	33,81	40,55	-6,74	17%
URRELO - M. URTEAGA	30,56	35,63	-5,07	14%
OVALO MUSICAL	26,37	29,02	-2,65	9%
PLAZA DE ARMAS	6,32	9,41	-3,09	33%
QUINDE	22,87	22,7	0,17	1%
SUCRE	20,44	23,77	-3,33	14%
OVALO EL INCA	25,77	21,67	4,1	19%
2 DE MAYO - M. URTEAGA	34,2	28,18	6,02	21%
REAL PLAZA	14,96	20,99	-6,03	29%
CRUCE DE JESÚS	32,5	36,17	-3,67	10%
M. SEOANE - H. RUBIO	31,82	34,49	-2,67	8%
QHAPAC ÑAN	0,58	0,55	-0,03	5%

1.9. Comparación de Datos de predicción con datos reales

En los siguientes resultados se muestra la comparación entre los datos de predicción realizado por el sistema con los datos reales tomados en campo.

Error absoluto: es la diferencia entre el valor de la medida y el valor tomado como exacto. Puede ser positivo o negativo, según si la medida es superior al valor real o inferior (la resta sale positiva o negativa). Tiene unidades, las mismas que las de la medida (Vivanco, 2005).

Sin embargo, para facilitar el manejo y el análisis se emplea el error absoluto definido como:

$$EA = VV - VE$$

Error relativo: es el cociente (la división) entre el error absoluto y el valor exacto. Si se multiplica por 100 se obtiene el tanto por ciento (%) de error. Al igual que el error absoluto puede ser positivo o negativo (según lo sea el error absoluto) porque puede ser por exceso o por defecto. no tiene unidades (Vivanco, 2005).

$$ER = EA / VV$$

Donde:

EA: Error Absoluto

ER: Error Relativo

VV: Valor Verdadero

VE: Valor Experimental

Para nuestro caso tendremos las siguientes equivalencias:

Valor Verdadero	Dato Real
Valor Experimental	Dato de Predicción (P)

Luego se obtuvo los siguientes resultados.

Error relativo de los datos de predicción del día 1(ver Tabla 33).

Tabla 33 *Error relativo de la predicción del día 1*

PUNTO DE CONTROL	P01	DATO REAL	ERROR ABSOLUTO	ERROR RELATIVO
UPN	21.15	23.37	2.22	9%
RECOLETA	17.63	19.04	1.41	7%
PUENTE AMARILLO	21.96	21.75	0.21	1%
FONAVI II	23.85	21.4	2.45	11%
PLAZUELA BOLOGNESI	33.75	37.28	3.53	9%
URRELO - M.	28.6	32.91	4.31	13%
URTEAGA	28.5	32.52	4.02	12%
OVALO MUSICAL	6.13	11	4.87	44%
PLAZA DE ARMAS	28.21	32.79	4.58	14%
QUINDE	20.77	19.67	1.1	6%
SUCRE	28.71	31.18	2.47	8%
OVALO EL INCA	30.67	25.95	4.72	18%
2 DE MAYO - M. URTEAGA	17.26	20.26	3	15%
REAL PLAZA	29.79	28.21	1.58	6%
CRUCE DE JESÚS	28.15	27.16	0.99	4%
M. SEOANE - H. RUBIO	0.65	0.58	0.07	12%
QHAPAC ÑAN				

Error relativo de los datos de predicción del día 2 (ver Tabla 34).

Tabla 34 *Error relativo de la predicción del día 2*

Punto de Control	P02	Dato Real	Error Absoluto	Error Relativo (%)
UPN	25.13	19.65	5.48	28%
RECOLETA	14.09	11.51	2.58	22%
PUENTE AMARILLO	24.02	24.41	0.39	2%
FONAVI II	20.78	17.32	3.46	20%
PLAZUELA BOLOGNESI	41.22	37.02	4.2	11%
URRELO - M. URTEAGA	28.54	33.36	4.82	14%
OVALO MUSICAL	22.19	21.27	0.92	4%
PLAZA DE ARMAS	5.9	3.07	2.83	92%
QUINDE	27.98	24.67	3.31	13%
SUCRE	26.6	23.32	3.28	14%
OVALO EL INCA	20.21	22.89	2.68	12%
2 DE MAYO - M. URTEAGA	35.93	40.87	4.94	12%
REAL PLAZA	18.2	21.13	2.93	14%
CRUCE DE JESÚS	28.16	32.52	4.36	13%
M. SEOANE - H. RUBIO	26.83	28.89	2.06	7%
QHAPAC ÑAN	0.53	0.48	0.05	10%

Error relativo de los datos de predicción del día 3 (ver Tabla 35).

Tabla 35 *Error relativo de la predicción del día 3*

Punto de Control	P03	Dato Real	Error Absoluto	Error Relativo (%)
UPN	22.17	24.23	2.06	9%
RECOLETA	16.4	13.84	2.56	18%
PUENTE AMARILLO	24.04	29.02	4.98	17%
FONAVI II	25.12	31.44	6.32	20%
PLAZUELA	33.81	40.55	6.74	17%
BOLOGNESI				
URRELO - M.	30.56	35.63	5.07	14%
URTEAGA				
OVALO MUSICAL	26.37	29.02	2.65	9%
PLAZA DE ARMAS	6.32	9.41	3.09	33%
QUINDE	22.87	22.7	0.17	1%
SUCRE	20.44	23.77	3.33	14%
OVALO EL INCA	25.77	21.67	4.1	19%
2 DE MAYO - M.	34.2	28.18	6.02	21%
URTEAGA				
REAL PLAZA	14.96	20.99	6.03	29%
CRUCE DE JESÚS	32.5	36.17	3.67	10%
M. SEOANE - H. RUBIO	31.82	34.49	2.67	8%
QHAPAC ÑAN	0.58	0.55	0.03	5%

Como se observa se obtuvieron datos aceptables; no obstante, aún no se los puede considerar confiables, puesto que se tuvo un margen de error relativo máximo del 92%, ya que los periodos (días) en los que se ha realizado la toma de datos, son muy pocos (12 días), por el tiempo limitado que hubo para realizar la investigación, pero en la mayoría de casos, el margen de error relativo es menor al 30%, siendo este un dato aceptable.

1.10. Contratación de la Hipótesis

Con el fin de llevar a cabo una contratación de hipótesis se observó en las Tabla 35 que cuenta con dos muestras relacionadas, las cuales han sido obtenidas por medio de toma de datos reales y por medio de predicción a través del sistema EcoFavBen. Así mismo, con el fin de determinar la normalidad de la muestra, se utiliza la prueba de normalidad Shapiro-Wilk, ya que la muestra está compuesta por menos de 30 datos. Como se observa en la Figura 26 obtenemos P-Valor menor a 0.05, lo que plica que la muestra se comporta normalmente (Wasserstein, 2016).

Pruebas de normalidad

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Diferencia_Dato	,182	16	,160	,919	16	,160

a. Corrección de significación de Lilliefors

Figura 26. Pruebas de normalidad de los datos reales y la predicción a través del sistema EcoFavBen

Fuente: IBM SPSS Statistics (versión 23).

Para realizar una contratación de hipótesis se plantea lo siguiente:

Hipótesis nula (H_0) donde P-Valor > 0.05 : El sistema EcoFavBen NO tiene un impacto positivo en la predicción de contaminación por CO en la ciudad de Cajamarca.

Hipótesis alternativa (H_a) donde P-Valor ≤ 0.05 : El sistema EcoFavBen tiene un impacto positivo en la predicción de contaminación por CO en la ciudad de Cajamarca.

En la prueba T-Student realizada en el software IBM SPSS Statistics (versión 23), se obtienen que la significancia o P-Valor es de 0.041, tal como se muestra en la Figura 27; dicho resultado es menor que 0.05, lo que implica que se rechaza la hipótesis nula. Consecuentemente se afirma la hipótesis alterna: El sistema EcoFavBen tiene un impacto positivo en la predicción de contaminación por CO en la ciudad de Cajamarca.

Prueba de muestras emparejadas

		Diferencias emparejadas				t	gl	Sig. (bilateral)	
		Media	Desviación estándar	Media de error estándar	95% de intervalo de confianza de la diferencia				
					Inferior				Superior
Par 1	EcoFavBen - Real	-2,10813	3,77062	,94266	-4,11735	-.09890	-2,236	15	,041

Figura 27. Resultado de prueba T-Student

Fuente: IBM SPSS Statistics (versión 23).

CAPÍTULO IV. DISCUSIÓN Y CONCLUSIONES

4.1. Discusión

En la implementación de prototipo de monitoreo remoto de contaminación ambiental de forma automática mediante sensores se adquieren datos de acuerdo a un intervalo de tiempo de monitoreo, como mencionan Estrada y Reinoso (2009), donde concluyeron que el intervalo para la toma de datos sea de menor a 5 minutos, en contrastación a lo dicho, en la presente investigación el intervalo de tiempo para la toma de datos será distinto ya que se tomaron datos por cada segundo, con la finalidad de contar con mayor dato que redunde en la precisión tanto de valores de CO como de la posición de manera positiva.

En base a los resultados obtenidos se acepta lo mencionado por Salazar Soria (2009), quien concluyó que los datos obtenidos bajo ciertas condiciones son más precisos tomando la medición cada segundo. Por lo que se aplicó de igual manera en el sistema EcoFavBen.

Según Puerta (2015), concluyó que la API para la descripción de servicios tiene como finalidad recopilar la información necesaria para que los servicios se puedan ejecutar. La API para el monitoreo y la gestión sirve, por una parte, para controlar el estado en el que se encuentran los servicios y así poder extraer información relevante para la toma de decisiones; y, por otra parte, para ejecutar servicios locales de forma cómoda. Entonces en referencia a lo dicho se acepta, ya que conforme a lo implementado en el sistema EcoFavBen, usamos la API para recopilar y consumir la información del prototipo sensorial y mostrar los datos de los niveles de contaminación por monóxido de carbono (CO) según zonas de Cajamarca.

Según Matla (2014) y Haylinos y Gonzales (2017), concluyen respectivamente que es posible involucrar de manera directa al cliente (Métodos Ágiles), con lo cual se puede obtener un producto más cercano a las necesidades de este, y en consecuencia maximizando los beneficios para él. Permitiendo a los pequeños grupos de desarrollo concentrarse en la tarea de construir software fomentando prácticas de fácil adopción y un entorno ordenado que ayude a que las personas trabajen mejor y permita que los proyectos finalicen exitosamente.

Se aceptan las afirmaciones antes mencionadas, porque al realizar la elaboración del sistema EcoFavBen, mediante la metodología ágil SCRUM se facilitó cada una de las etapas de desarrollo implementadas.

Según León (2018), concluyó que el flujo algorítmico de árbol de decisión establecido para la predicción en base a los resultados previos, permitió ser más específico y también se verificó que la exactitud de la predicción de las clases depende del tamaño del conjunto de datos al cual pertenece la clase. En base a lo mencionado se comprobó que la implementación del algoritmo árbol de decisión para la predicción de los niveles de monóxido de carbono del sistema EcoFavBen, tiene un registro de predicción aceptable, dentro de su contexto, pues se tiene poca cantidad de datos, respecto al periodo de tiempo, puesto que según el análisis de error estadístico relativo que se aplicó, se tuvo un margen máximo del 92%, lo que apoya León, cuando menciona que la cantidad de datos es importante para procesos de predicción.

4.2. Conclusiones

La implementación del sistema EcoFavBen impactó de manera positiva en la medición y predicción de datos de los niveles de CO en la ciudad de Cajamarca, ya que se pudo tomar en campo los puntos de ubicación con sus respectivas medidas de nivel de CO, además se pudo obtener estos datos en tiempo real, de esta manera se logra el objetivo. Obteniendo un margen de confiabilidad mayor al 70%.

Se logró implementar el sistema EcoFavBen de manera satisfactoria, permitiendo que se puedan tomar datos en entorno real sobre los niveles de CO, procesarlas y tener una análisis conciso.

Se logró identificar las zonas de mayor concentración de CO en la ciudad de Cajamarca, siendo el punto “PLAZUELA BOLOGNESI” con mayor promedio en concentración de monóxido de carbono.

Los datos obtenidos no son muy constantes, ya que no se rigen al Decreto Supremo N° 003-2017-MINAM, el cual menciona valores menores de 9 PPM en lugares cerrados, mientras que los valores captados por el sistema EcoFavBen sobrepasan este límite ya que las mediciones de estos niveles se obtuvieron de manera in situ del parque automotor.

Se logró realizar una predicción de datos aceptable, sin embargo, aún no se los puede considerar confiables, puesto que según el análisis de error estadístico relativo que se aplicó, se tuvo un margen máximo del 92%, ya que los periodos (días) en los que se ha realizado la toma de datos, son muy pocos (12 días), por el tiempo limitado que hubo para realizar la investigación, pero en la mayoría de casos, el margen de error relativo es menor al 30%, siendo este un dato aceptable.

REFERENCIAS

- Agency, E. P. (3 de Enero de 2020). *EPA*. Obtenido de EPA: <https://www.epa.gov/>
- Bank, T. W. (15 de Noviembre de 2019). <https://www.bancomundial.org/>. Obtenido de <https://datos.bancomundial.org/indicador/EN.ATM.CO2E.PC>
- bookdown. (13 de Enero de 2020). *bookdown.org*. Obtenido de bookdown.org: <https://bookdown.org/content/2031/arboles-de-decision-parte-i.html>
- Brown, T. (2004). *química la ciencia central 9na edición*. México: Prentice Hall.
- COMSCORE. (2015). *IMS MOBILE*. Nueva York: Publicaciones COMSCORE.
- Efrainguerrero. (4 de Enero de 2020). <https://wwwefrainguerrero.blogspot.com/>. Obtenido de <https://wwwefrainguerrero.blogspot.com/>: <https://wwwefrainguerrero.blogspot.com/2012/04/ciclo-de-vida-scrum.html>
- Espinoza, B., & Gutierrez, N. (2015). *Sistema De Información Para La Toma De Decisiones, Usando Técnicas De Analisis Predictivo Para La Empresa Iasacorp International S.A.* Lima.
- Estrada Jiménez, J. C., & Reinoso Chisaguano, D. J. (2009). *Diseño e implementación de un prototipo de monitoreo remoto de contaminación ambiental utilizando tecnología GSM*. Madrid.
- Garcia, E., & Valverde, E. (2013). *Toxicología Clínica. Toxicología Clínica*.
- geekbotelectronics. (15 de Diciembre de 2019). <http://www.geekbotelectronics.com/>. Obtenido de <http://www.geekbotelectronics.com/>: <http://www.geekbotelectronics.com/producto/mq-9-modulo-sensor-de-monoxido-de-carbono/>
- Gómez, J., & Valcarce, F. (2014). *Toxicología. Toxicología*. Pamplona.

google. (12 de Febrero de 2020). *google.com*. Obtenido de *google.com/maps*:
<https://www.google.com/maps/@-7.1567554,-78.5166834,15z/data=!5m1!1e1>

Guerrero, E. (10 de Enero de 2020). *efrainguerrero.blogspot*. Obtenido de
efrainguerrero.blogspot: www.efrainguerrero.blogspot.com.org

Hernandez, R. (2014). *Metodología de la Investigación*. Mexico D.F: Graw Hill.

Huanca, L. A. (2015). *Revisión Sistemática De La Calidad Del Software En* . Lima.

Huaylinos Gonzales, E. G. (2017). *Metodologías Ágiles En La Implementación De Una
Aplicación Para La Gestión De Citas En La clínica Dental “Perio Dent” - Huancayo*.
Huancayo.

Huaylinos Gonzales, E. G. (2017). *Metodologías Ágiles En La Implementación De Una
Aplicación Para La Gestión De Citas En La clínica Dental “Perio Dent” - Huancayo*.
Huancayo.

INECC. (10 de Diciembre de 2019). *instituto nacional de ecología y cambio climático*.
Obtenido de
http://www2.inecc.gob.mx/publicaciones2/libros/419/cap1_metodologia.html

INEI. (07 de Noviembre de 2018). Censos 2017. *Censos 2017*. Lima, Jesús María, Perú.

infootec. (10 de Enero de 2020). *infootec.net*. Obtenido de *infootec.net*:
<https://www.infootec.net/arduino/>

Instituto Nacional de Estadística e Informática. (03 de Enero de 2019). *Cajamarca es el
quinto departamento con mayor población del país*. Cajamarca, Cajamarca, Perú.

Jimenez, A. (2013). *Arboles de Decision*. Sevilla.

- Leon, H. (2018). *Desarrollo De Un Modelo Algorítmico Basado En Árboles De Decisión Para La Predicción De La Permanencia De Un Paciente En Un Proceso Psicoterapéutico*. Lima.
- Leon, H. (2018). *Desarrollo De Un Modelo Algorítmico Basado En Árboles De Decisión Para La Predicción De La Permanencia De Un Paciente En Un Proceso Psicoterapéutico*. Lima.
- Lepkowski, J. (2008). *Advances in telephone survey*. Washington,DC: Jhon Wiley.
- Managements Solutions. (2018). *Machine Learning, una pieza clave en la transformación de los modelos de negocio*. Madrid.
- Matla, E. (2014). *Desarrollo de software guiado por la norma ISO/IEC 29110 Y SCRUM: SIDEPE*. Mexico D.F.
- Microsoft. (28 de Enero de 2020). *microsoft.com*. Obtenido de microsoft.com: <https://docs.microsoft.com/es-es/dotnet/machine-learning/how-does-mldotnet-work>
- Ministerio del Ambiente - Perú. (17 de junio de 2017). Decreto Supremo 003-2017-MINAM. *Decreto Supremo 003-2017-MINAM*.
- Municipalidad de Cajamarca. (11 de Enero de 2020). *municaj.gob.pe*. Obtenido de municaj.gob.pe: <http://www.municaj.gob.pe/archivos/pdu/07-CAP-IV-S-5.pdf>
- Muñoz Pellissier, C. O. (2017). *Implementación Del Software Tareo Para Mejorar El Tiempo Del Seguimiento Del Mantenimiento Preventivo Y Correctivo De Maquinaria Pesada, En La Empresa Assac*. Cajamarca.
- NJ Health. (01 de Septiembre de 2011). Hoja Informativa Sobre Sustancias Peligrosas. *Monóxido de Carbono*. New York, New York, USA.

- Ochoa, C. (3 de Febrero de 2020). *netquest.com*. Obtenido de netquest.com:
<https://www.netquest.com/blog/es/blog/es/muestreo-por-conveniencia>
- ONU. (2019 de marzo de 2019). *Noticial ONU*. Obtenido de
<https://news.un.org/es/story/2019/03/1452171>
- Pérez, J. (4 de Enero de 2020). *definicion.de*. Obtenido de definicion.de:
<https://definicion.de/prediccion/>
- Puerta, J. M. (2015). *Desarrollo de una API para la descripción y gestión de Servicios Web Rest*. Valencia.
- Reyes Mena, D. V. (2014). *Diseño De La Red De Sensores Inalámbricos Mediante La Evaluación Del Desempeño De Las Plataformas Arduino Y Waspote Bajo El Protocolo De Comunicaciones Zigbee Para El Monitoreo Y Control De Los Embalses De Agoyán Y Pisayambo*. Riobamba, Chimborazo, Ecuador.
- Rivero, F. (2015). *informe mobile en españa y en el mundo 2015*. Madrid: prensa ditrendia.
- RoboVent. (15 de Enero de 2020). *robovent.com*. Obtenido de robovent.com:
<https://robovent.com/es/frequently-asked-questions/diferencias-osha-acgih/>
- Salazar Soria, L. A. (2009). *Análisis y medición de contaminación acústica en sectores de alta densidad vehicular de la ciudad de Quito*. Sangolquí, Pichincha, Ecuador.
- Salud, M. (5 de Enero de 2020). <https://www.murciasalud.es/principal.php>. Obtenido de
<https://www.murciasalud.es/principal.php>:
<https://www.murciasalud.es/pagina.php?id=180398&idsec=1573>
- scielosp. (27 de Enero de 2020). *scielosp.org*. Obtenido de scielosp.org:
<https://www.scielosp.org/article/rsap/2006.v8n1/108-117/>

- Scrum. (5 de Enero de 2020). <https://proyectosagiles.org/>. Obtenido de <https://proyectosagiles.org/>: <https://proyectosagiles.org/que-es-scrum/>
- SUNARP. (20 de Enero de 2018). Boletín Estadístico. *Boletín Estadístico*. Lima, Lima, Perú.
- Téllez. (10 de diciembre de 2019). *Contaminación por Monóxido de Carbono: un Problema de Salud Ambiental*. Obtenido de Revista de Salud Pública: <https://www.scielosp.org/article/rsap/2006.v8n1/108-117/>
- Téllez. (24 de Noviembre de 2019). *Salud Publica*. Obtenido de <https://www.scielosp.org/>: <https://www.scielosp.org/article/rsap/2006.v8n1/108-117/>
- Tellez, J. (2012). Contaminación por Monóxido de Carbono. *Contaminación por Monóxido de Carbono*. Bogota.
- Valencia, P. (2012). Infraestructura Polivalente Basada En Sensores Inalámbricos Aplicada A La Monitorización Medioambiental Georreferenciada. Gran Canaria, Las Palmas, España.
- Vivanco, M. (2005). *Muestreo Estadístico Diseño y Aplicaciones*. Santiago de Chile: Universitaria.
- Wasserstein, R. (2016). *La declaración de la ASA sobre los valores de p : contexto, proceso y propósito*. Washington: The American.

ANEXOS

ANEXO N° 1. Definición de Monóxido de Carbono (CO)

Con el uso creciente de electrodomésticos y gas domésticos, además de otros elementos como chimeneas y sistemas de calefacción, las viviendas se han tornado en un factor de riesgo importante de intoxicación intradomiciliaria por monóxido de carbono.

En los Estados Unidos las emisiones de monóxido de carbono al ambiente son producidas en un 95 % por los vehículos automotores. La Comisión Nacional del Medio Ambiente de Chile informó que para el año 2014, el 91 % de las emisiones de monóxido de carbono en la región metropolitana de Santiago de Chile fueron producidas por el transporte automotor (Agency, 2020).

En México, los vehículos automotores también son responsables de la mayor cantidad de emisiones de monóxido de carbono a la atmósfera; allí se ha fijado como concentración media de este compuesto para 1 hora el valor de 11 partes por millón, que equivale a 100 puntos del Índice Metropolitano de Calidad del Aire (scielosp, 2020).

En Brasil, el monóxido de carbono es el contaminante emitido en mayor cantidad a la atmósfera. En Sao Paulo 1,5 millones de toneladas son lanzadas al aire anualmente; de estos el 78 % son producidos por los automotores a gasolina o diésel, 15 % por automotores con combustible alcohol, 3 % por motocicletas, 2 % por taxis y 2 % resultan de procesos industriales.

En Colombia, el Instituto de Hidrología, Meteorología y Estudios Ambientales, estimó para 1996 las emisiones atmosféricas de gases con efecto local en 8 612 kilotoneladas, de las cuales el monóxido de carbono representó el 58 %. En Bogotá, el Departamento

Administrativo del Medio Ambiente estima que la concentración media de monóxido de carbono atmosférico en un día normal se encuentra entre 30 y 35 partes por millón. En el informe sobre monitoreo realizado en el día sin carro en 2013, se estimó que la concentración ambiental de monóxido de carbono disminuyó en un 76 % respecto a un día normal (scielosp, 2020).

Efectos tóxicos en la salud humana; la Asociación Americana de Centros para el Control de Intoxicaciones y el Sistema de Vigilancia de Exposición a Tóxicos, informaron sobre 17 006 casos de intoxicación y 35 muertes relacionados con exposición a monóxido de carbono. Omaye, informa que anualmente en Estados Unidos entre 10 000 a 40 000 personas demandan atención médica o faltan al trabajo debido a intoxicación por monóxido de carbono (García & Valverde, 2013).

Se realizó sobre 882 personas fallecidas en el periodo de 1991 a 2000, encontraron que los periodos del año que con mayor frecuencia se presentaron casos relacionados con monóxido de carbono, son los correspondientes a otoño e invierno, en una estrecha relación con el uso de calderas, calefacciones, braseros y chimeneas en los hogares (Gómez & Valcarce, 2014).

En España 31 casos de intoxicación por monóxido de carbono; Portoles y Algarra informaron en 1992 una serie de 13 casos de intoxicación aguda por monóxido de carbono en España. Revert y Brotons describieron en 1995 una epidemia de intoxicación aguda por monóxido de carbono en la Región de Badía, relacionada con escapes en equipos de calefacción en temporada de invierno. Dueñas, Ruiz-Mambrilla y otros en 1997 informaron sobre 154 casos de intoxicación por monóxido de carbono presentados en 2 hospitales de la

región de Valladolid en España en un periodo de 18 meses; de estos casos 5 fueron mortales (Tellez, 2012).

El monóxido de carbono, que por sus características físico-químicas es llamado el asesino silencioso, utiliza múltiples mecanismos de toxicidad que explican sus potenciales efectos adversos en la salud humana. Entre ellos se encuentran los siguientes: Compite con el oxígeno y altera la curva de disociación de la hemoglobina. Una vez penetra al organismo, el monóxido de carbono se une a las enzimas del grupo Hem de la hemoglobina, desplazando al oxígeno de la misma. De esta manera se forma en la sangre un complejo que se denomina carboxihemoglobina, que dificulta el transporte de oxígeno a las células y tejidos, lo que va a producir una hipoxia celular generalizada. La molécula de hemoglobina dispone de cuatro sitios de unión con el oxígeno, cuando uno de estos sitios es ocupado por el monóxido de carbono, la hemoglobina se altera de tal forma que impide que los otros sitios se unan al oxígeno, produciendo así una desviación de la curva de disociación oxígeno-hemoglobina hacia la izquierda. Esta alteración empeora la hipoxia celular producida por la formación de carboxihemoglobina. Los síntomas y signos iniciales de la intoxicación aguda se presentan muy rápidamente y se han relacionado con niveles de carboxihemoglobina en sangre superiores a 10 %.

Interfiere la utilización del oxígeno por los tejidos. El carácter fuertemente ligante del monóxido de carbono le permite unirse a los átomos de hierro de los citocromos, especialmente a los que intervienen en la cadena respiratoria mitocondrial, lo que impide una adecuada utilización del oxígeno por los tejidos orgánicos, agravando la hipoxia tisular originada en el déficit de oxígeno en la hemoglobina, produce peroxidación lipídica cerebral.

El monóxido de carbono reacciona con la enzima xantina deshidrogenasa convirtiéndola en xantina oxidasa, la cual a su vez reacciona con la hipoxantina, produciendo superóxidos; estos reaccionan con el óxido nítrico producido por el endotelio formando peroxinitrito, ácido peroxinitroso y/o peroxinitratos, potentes oxidantes que interactúan con los ácidos grasos insaturados, produciendo peroxidación lipídica y originando desmielinización progresiva de las neuronas del sistema nervioso central. Sherral Devine y otros presentan un caso clínico en el cual una mujer que laboró durante un largo periodo de tiempo como cocinera en un restaurante, presentó lesiones neuropsicológicas caracterizadas por dificultades en la lectura, la escritura y en el lenguaje verbal; estas alteraciones fueron correlacionadas con múltiples lesiones bilaterales en los ganglios basales detectadas mediante Resonancia Nuclear Magnética. Estos autores sugieren que la neurotoxicidad por monóxido de carbono, también involucra otros mecanismos de toxicidad como la inhibición de la respiración mitocondrial, acidosis metabólica y peroxidación lipídica cerebral, lo que conlleva a una desmielinización progresiva de la sustancia blanca cerebral

Inhibe la síntesis aeróbica de adenosina trifosfato. La alteración en el transporte de electrones en la mitocondria produce estrés oxidativo, que se manifiesta por un aumento en la fracción de radicales libres hidroxilo.

Formación de grupos sulfatides en cerebro. La actividad de la enzima arilsulfatasa A es esencial en el metabolismo de la mielina, ya que participa en su degradación normal e impide la acumulación de grupos sulfatides que son neurotóxicos. Cuando la actividad de la enzima arilsulfatasa A es inferior al 10 %, se produce acumulación de los grupos sulfatides, generándose una leucoencefalopatía. Individuos que presenten una disminución previa de la actividad de esta enzima en porcentajes entre 10 % y 30 % que no se ha expresado

clínicamente, al exponerse a monóxido de carbono, desarrollan un cuadro clínico de encefalopatía retardada. La encefalopatía retardada por monóxido de carbono es un cuadro clínico poco conocido y de rara presentación; se describe que tiene una frecuencia de 2,7 %, sus características clínicas más frecuentes son deterioro cognitivo, alteraciones de la marcha, incontinencia de esfínteres, mutismo, hiponimia, hipertonía y retropulsión. Este cuadro se suele presentar en mayores de 40 años de edad, su patogénesis se ha relacionado con disminución de la actividad de la enzima arilsulfatasa A, por debajo del 10 %, lo que permite la acumulación de grupos sulfatidos, que son considerados neurotóxicos al generar un cuadro de leucoencefalopatía (Salud, 2020).

Monitoreo de la exposición a monóxido de carbono. El monitoreo de la exposición a monóxido de carbono, se puede realizar a través de la determinación de biomarcadores y medición de concentraciones ambientales. El término biomarcador es usado para incluir casi cualquier medida que refleje una interacción entre un sistema biológico y un riesgo potencial. Los biomarcadores también han sido definidos como cambios inducidos por un contaminante en los componentes bioquímicos o fisiológicos que pueden ser medidos en un sistema biológico.

Carboxihemoglobina en sangre. Es un biomarcador de exposición de alta especificidad, refleja la dosis interna de monóxido de carbono en sangre. También se considera un biomarcador de efecto precoz en cuanto refleja alteraciones bioquímicas y fisiológicas en la estructura de la hemoglobina y en el proceso fisiológico de oxigenación celular y tisular. La American Conference Governmental Industrial Hygienist, fija el valor biológico tolerable para carboxihemoglobina en sangre en 3,5 %.

Actividad de la enzima arilsulfatasa A. Este es un biomarcador de susceptibilidad. La enzima arilsulfatasa A es esencial en el metabolismo de la mielina, ya que participa en su degradación normal e impide la acumulación de grupos sulfatidos. Cuando la actividad de la enzima arilsulfatasa A es inferior al 10 %, se produce acumulación de los grupos sulfatidos. Individuos que presenten una disminución previa de la actividad de esta enzima en porcentajes entre 10 % y 30 % que no se ha expresado clínicamente, al exponerse a monóxido de carbono, desarrollan un cuadro clínico de encefalopatía retardada.

Monóxido de carbono exhalado. Marcador de exposición de alta especificidad, que permite tener una medición de dosis interna. La American Conference Governmental Industrial Hygienist, fija su valor límite biológico en 25 partes por millón (Téllez, Salud Publica, 2019).

ANEXO N° 2. Instrumento: Ficha de Registro

FICHA DE REGISTRO

**Impacto del Sistema EcoFavBen en la Predicción de Contaminación por CO
en la Ciudad de Cajamarca**

Lectura y registro de niveles de monóxido de carbono en la ciudad de Cajamarca.

Nro.	Latitud	Longitud	Zona	Contaminante	Lectura	Fecha	Hora
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							

Información validada por: Briones Chilón Noe Elías; Hernández Valencia Erick Omar

ANEXO N° 3. Metodología Scrum

El Scrum es un proceso de desarrollo iterativo e incremental enfocado a la gestión de procesos de desarrollo de software, aunque también puede ser utilizado en equipos de mantenimiento de software, que sirve para administrar y controlar el desarrollo de sistemas. Esta metodología, inicialmente documentada por dos japoneses (Takeuchi y Nonaka) en 1986 y con aportes de especialistas (Sutherland y Schwaber) a lo largo de la última década, tiende a hacer que los procesos que influyen en el desarrollo de tecnología se incrementen en rapidez y flexibilidad, siempre tomando en cuenta los tiempos y reglas de negocio que dan motivo al desarrollo del proyecto. Se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software, usualmente con equipos chicos y multidisciplinarios.

El desarrollo se realiza en forma iterativa e incremental. Cada iteración termina con una pieza de software ejecutable que incorpora nueva funcionalidad. Estas iteraciones tienen en general una duración entre 2 y 4 semanas. Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos (Scrum, 2020).

Los requerimientos y las prioridades se revisan y ajustan durante el proyecto en intervalos muy cortos y regulares. De esta forma se puede adaptar en tiempo real el producto que se está construyendo a las necesidades del cliente o, dicho de otra manera, incorporar cambios con rapidez y en cualquier fase del proyecto.

Hay estudios estadísticos de la Industria del Software que muestran que casi la mitad de la funcionalidad que se implementa no es utilizada. En Scrum se construye primero la funcionalidad que resulte en el mayor beneficio para el negocio y se evita en todo momento desarrollar cosas que no serán de utilidad para el cliente. De esta manera se puede desarrollar un sistema en menos tiempo evitando el trabajo innecesario. Una ventaja de esta metodología es que es muy fácil de aprender y requiere muy poco esfuerzo para comenzar a utilizarse. Scrum se utiliza como marco para otras prácticas de ingeniería de software como RUP o Extreme Programming.

Introducción a la metodología Scrum

El Scrum es un proceso que incluye un conjunto de actores o roles y de acciones o prácticas predefinidas.

Los roles principales en Scrum son el Scrum Master, que mantiene los procesos y trabaja de forma similar al director de proyecto, Product Owner, que representa a los Stakeholders (clientes externos o internos), y el Scrum Team que incluye a los desarrolladores. Luego, los usuarios o clientes, que deben estar involucrados con el proyecto para que este tenga éxito. Durante cada Sprint, un periodo entre 15 y 30 días, el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada Sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que dan forma al trabajo a realizar. Los elementos del Backlog que forman parte del Sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner informa al equipo de los elementos en el Product Backlog que quiere ver completados. El equipo entonces determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente Sprint. Durante el Sprint, nadie puede cambiar el Sprint Backlog (una

lista donde se detalla cómo se van a construir los diferentes requisitos del producto), lo que significa que los requisitos están congelados durante el Sprint (Scrum, 2020).

Scrum en mayor detalle. Sus actores y acciones. Su forma de trabajo

Los actores son los que ejecutarían obviamente las acciones.

El Scrum Máster es quien facilita el Scrum; su trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del Sprint y por lo tanto facilitar el Scrum. El Scrum Máster no es el líder del equipo (porque ellos se auto organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. Se asegura que el proceso Scrum se utiliza como debe. Es quien hace que las reglas se cumplan. Es la persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que pueda aparecer.

El Product Owner representa la voz del cliente y aporta la visión de negocio. Ellos se aseguran de que el equipo de Scrum trabaja de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario (sencillas tarjetas en las que se recoge de forma esquemática y en un lenguaje claro qué es lo que queremos hacer), las prioriza y las coloca en el Product Backlog.

El Scrum Team tiene la responsabilidad de entregar el producto. Un pequeño equipo de 5 a 9 personas con una mezcla de habilidades necesarias para realizar el trabajo (desarrolladores, diseñadores, etc.) de implementar la funcionalidad o funcionalidades elegidas por el Product Owner.

Los demás involucrados en el proceso, son los usuarios del producto o aplicación, los clientes y vendedores, y los gestores y directivos. No son actores esenciales pero si implicados y deben ser tenidos en cuenta. Aportan ideas, sugerencias o necesidades.

Las acciones tienen relación directa con los actores. En Scrum se indican claramente las acciones a acometer y cómo acometerlas (Scrum, 2020).

Las acciones en Scrum forman un ciclo iterativo repetitivo que tiene como objetivo minimizar el esfuerzo y maximizar el rendimiento en el desarrollo. El Product Backlog corresponde con todas las tareas, funcionalidades o requerimientos a realizar. El Product Owner es la persona que se encarga de marcar las prioridades, mantener y actualizar el Product Backlog.

El Sprint Planning Meeting es una reunión que tiene por objetivo, planificar el Sprint a partir del Product Backlog. El objetivo de esta reunión es la de mover las tareas del Product Backlog al Sprint Backlog. En esta reunión, suelen participar el Product Backlog, el Scrum Master y el Scrum Team. Del Sprint Planning Meeting sale también el Sprint Goal, que es un pequeño documento o una breve descripción que indica lo que el Sprint intentará alcanzar. El Sprint Backlog corresponde con una o más tareas que provienen del Product Backlog. Del Product Backlog se saca una o más tareas que van a formar parte del Sprint Backlog. Estas tareas se deben acometer en unas 2 o 4 semanas.

Una vez que el Sprint Backlog se inicia, este no puede ser alterado o modificado. Hay que esperar a que este concluya para realizar la correspondiente modificación o alteración, cuya tarea formaría parte de otro Sprint Backlog (Scrum, 2020).

El Daily Scrum Meeting es una tarea iterativa que se realiza todos los días que dure el Sprint Backlog con el equipo de desarrollo. Se trata de una reunión diaria, operativa, informal y ágil, de un máximo de 30 minutos, en la que se le hacen preguntas a cada integrante del equipo. Preguntas sobre las tareas que ha realizado desde la última reunión, preguntas sobre qué va a realizar en el día actual y preguntas de identificación de obstáculos o riesgos que

impiden o pueden impedir el normal avance. El Scrum Master debe eliminar aquí cualquier obstáculo que encuentre.

En el Sprint Review se revisa el Sprint finalizado. En esta reunión suelen asistir el Product Owner, el Scrum Master y personas que podrían estar involucradas con el proyecto. El Scrum Team es quien muestra los avances realizados en el Sprint. Al finalizar el Sprint Backlog y el Sprint Review, se inicia el Sprint Retrospective. El Product Owner revisará con el equipo los objetivos marcados inicialmente en el Sprint Backlog concluido, se aplicarán los cambios y ajustes si son necesarios, y se marcarán los aspectos positivos (para repetirlos) y los aspectos negativos (para evitar que se repitan) del Sprint.

Ventajas

- Se obtiene software lo más rápido posible y este cumple con los requerimientos más importantes.
- Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.
- Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.
- Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.
- Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.
- Permite producir software de una forma consistente, sostenida y competitiva.
- Las reuniones se dedican a inconvenientes recientes, evitando el estancamiento

Desventajas

- Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.

- Es una metodología que difiere del resto, y esto causa cierta resistencia en su aplicación para algunas personas

Scrum por sus características no es válido para cualquier proyecto ni para cualquier persona o equipo de personas. Es más, Scrum según muchos especialistas de esta metodología, es óptima para equipos de trabajo de hasta 8 personas, aunque hay empresas que han utilizado Scrum con éxito con equipos más grandes (ver Figura 28).

Se puede decir que para el 90% de los proyectos y empresas, es una metodología válida, pero no es una metodología válida al 100%. Es más, no hay metodología mejor que otra ni válida al 100% para todas las personas y empresas (Scrum, 2020).

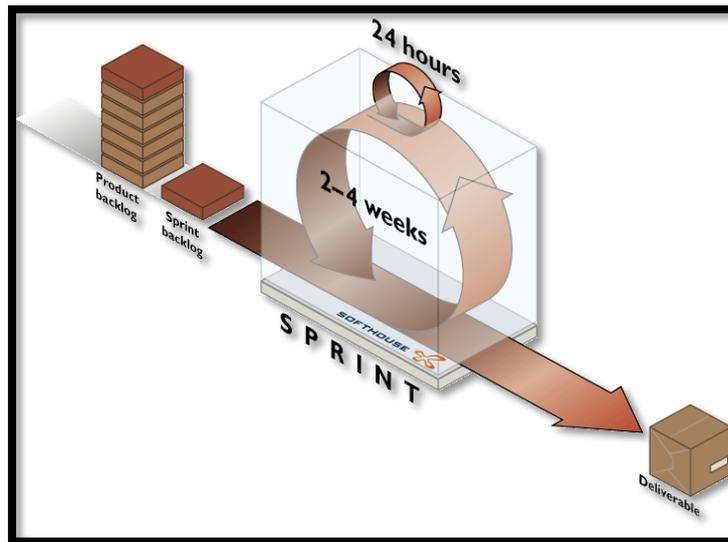


Figura 28. Diagrama Scrum

Fuente: (Guerrero, 2020)

La idea de la metodología ágil es fundamentalmente que adopte los cambios, que se pueda reconducir el proyecto en un momento dado, y que afecte lo menos posible a los costes, los tiempos y al equipo de trabajo (ver Figura 29).

Diagrama ciclo de vida de Scrum

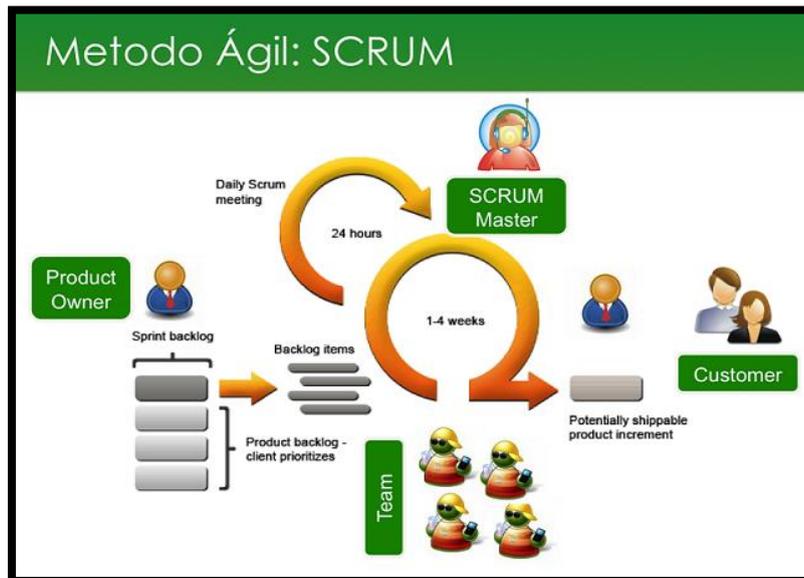


Figura 29. Diagrama ciclo de vida de Scrum

Fuente: (Guerrero, 2020)

ANEXO N° 4. Materiales para el prototipo del circuito sensorial

Arduino Uno R3

Arduino Uno R3 es una placa electrónica de las muchas que tiene Arduino y con la que es muy fácil introducirse en el mundo de la programación electrónica, Arduino es una plataforma de código abierto (open-source) lo que permite realizar proyectos y modificaciones tanto de hardware como de software a cualquier persona sin ningún problema.

En la imagen siguiente se detallan las diferentes partes y características que tiene la placa electrónica Arduino Uno R3 (ver Tabla 36).

Tabla 36 *Partes y características de Arduino Uno R3*

Partes	Características
Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
Digital pines I/O	14 (de los cuales 6 proporcionan una salida PWM)
PWM digital pines I/O	6
Pines de entrada analógica	6
Corriente DC por Pin I/O	20mA
Corriente DC para Pin 3.3V	60mA
Memoria flash	32KB ATmega328P
SRAM	2KB ATmega328P
EEPROM	1KB ATmega328P
Velocidad de reloj	16 MHz
Longitud	68,6 mm
Anchura	53,4 mm
Peso	25 g

Fuente: (infootec, 2020)

En las siguientes imágenes se muestran cada una de las partes que posee el Arduino Uno R3, siendo las partes usadas para esta investigación los pines digitales, los pines análogos, alimentador de energía tanto de 5 como de 3.3 voltios y el puerto usb el cual comunica con el ordenador (ver Figura 30 y 31).

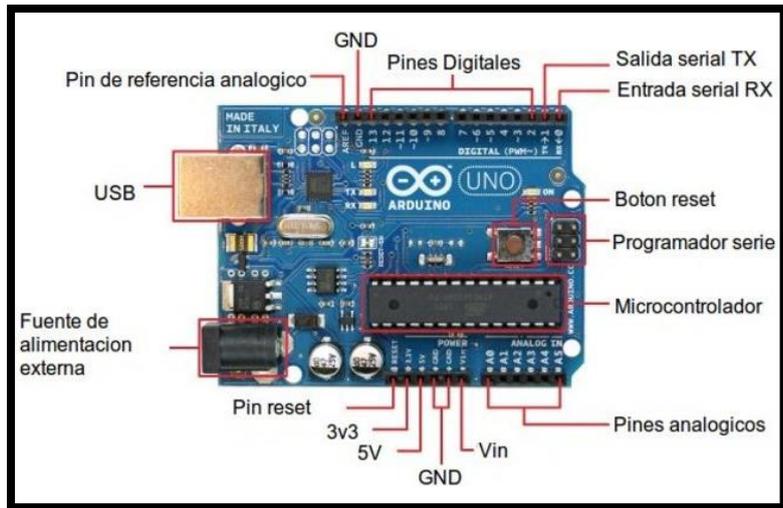


Figura 30. Partes de Arduino Uno R3

Fuente: infootec

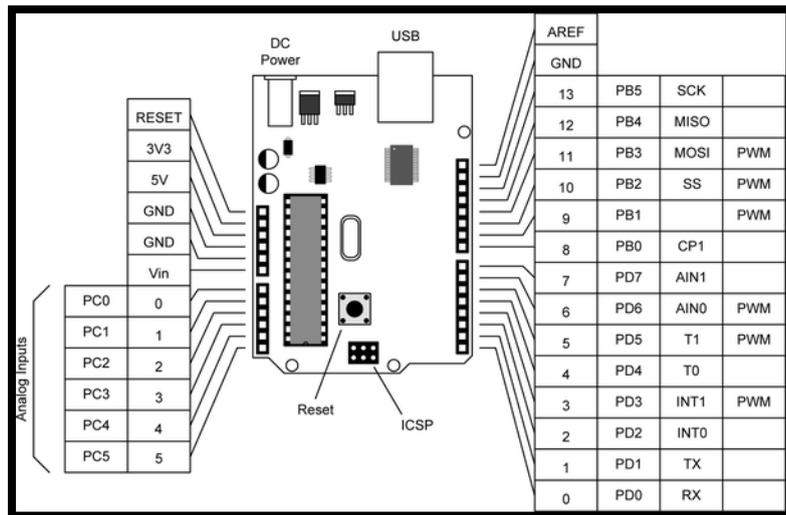


Figura 31. Descripción de los pines de Arduino Uno.

Fuente: infootec

MQ-9 Módulo Sensor de Monóxido de Carbono

Sensor de gas MQ-9 CO monóxido de carbono ideal para detección del mismo gas. Tiene la sensibilidad ajustable para tener una lectura adecuada al sistema particular. El sensor tiene un tamaño pequeño y es muy práctico y sencillo de utilizar. El módulo tiene una salida digital a través de un comparador con umbral ajustable y una salida analógica que la puedes medir con cualquier microcontrolador o tarjeta de desarrollo con ADC (geekbotelectronics, 2019) (ver Figura 32).

Especificaciones:

- Voltaje de operación: 5V
- Tipo de interfaz: Analógica
- Definición de pines: 1-Output 2-GND 3-VCC
- Tipo de Sensor: Semiconductor
- Buena sensibilidad a CO / gas combustible
- Alta sensibilidad al metano, propano y CO
- Larga vida útil y bajo costo
- Tamaño: 40x20mm

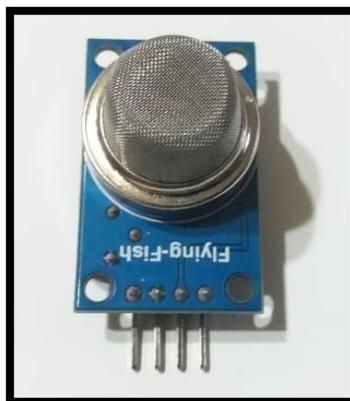


Figura 32. Sensor MQ-9

En las imágenes se muestran todas las características de sensibilidad típicas, estructura y configuración del MQ-9, ordenada significa relación de resistencia del sensor (R_s / R_o), la abscisa es la concentración de gases (ver Figura 33 y 34).

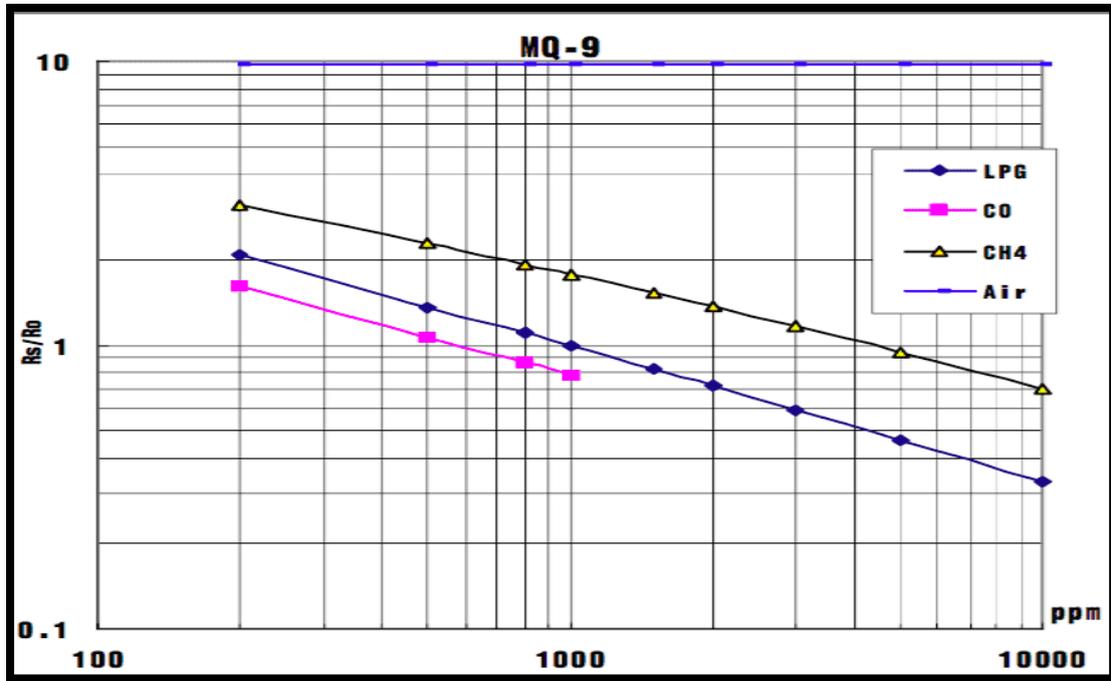


Figura 33. Sensibilidad del sensor MQ-9

Fuente: seed

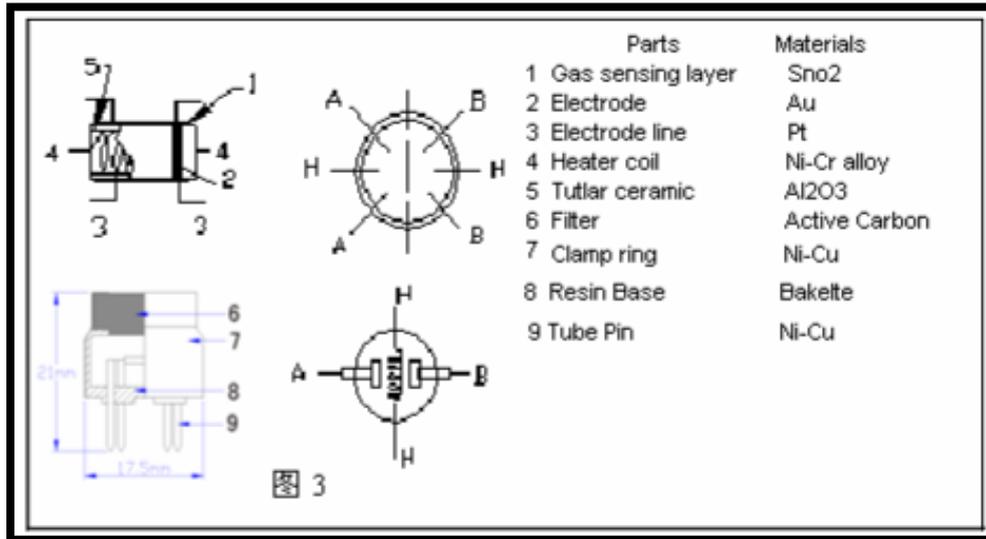


Figura 34. Estructura y configuración del sensor MQ-9

Fuente: seed

Módulo Bluetooth HC-05

Para empezar el módulo de bluetooth HC-06 solo opera de modo esclavo, a diferencia de su hermano HC-05. Primeramente el HC-05 ofrece una mejora con respecto a precio y características, ya que es un módulo Maestro-Eslavo, esto quiere decir, que además de recibir conexiones desde una PC o tablet, (objetivo de este primer Tutorial), también es capaz de generar conexiones hacia otros dispositivos bluetooth (ver Figura 35).

Para empezar, esto nos permite por ejemplo, conectar dos módulos bluetooth y formar una conexión punto a punto, para transmitir datos entre dos microcontroladores o dispositivos.

Comunicación App-Android A Arduino Mediante Bluetooth HC-05

Este tutorial puede servir desde prender y apagar LEDs al mismo tiempo que controlar un Robot de manera remota. Por consiguiente cualquier modulo bluetooth es bueno. Para esta parte usaremos lo siguiente, App Android, un Arduino, un LED y el Modulo Bluetooth HC-

06 (esclavo). Ya que este módulo no ocupa ninguna clase de configuración, se destaca por solo tener 4 pines.

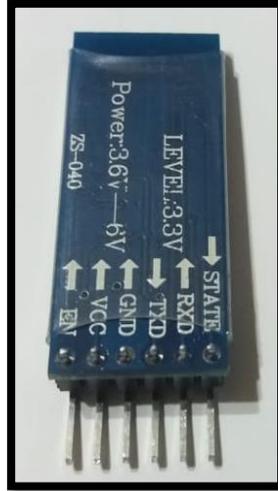


Figura 35. Modulo Bluetooth HC-05

Diseño e implementación del circuito sensorial de monóxido de carbono (CO).

Se detalla cada uno de las tareas que se tuvo que desarrollar en el Sprint 1 (ver Tabla 37).

Tabla 37 *Sprint Backlog Del Sprint 1*

ID	Historia de usuario	Tarea
H01	Como un usuario, capturar los datos captados por el sensor	Diseño e implementar el circuito en Arduino (Arduino Uno, sensor MQ-9 Y Modulo Bluetooth)
		Codificar las funciones en el IDE de Arduino y cargar el código en el circuito de Arduino
H02	Como un usuario, procesar los datos obtenidos por el sensor	Calibrar el sensor Comprobar funcionalidad del circuito Arduino con el monitor del puerto serial

Prototipos Arduino

En la siguiente figura se muestra la implementación del prototipo de circuito sensorial, el cual permitió captar los niveles del monóxido de carbono (CO), a través del sensor MQ-9 (ver Figura 37).

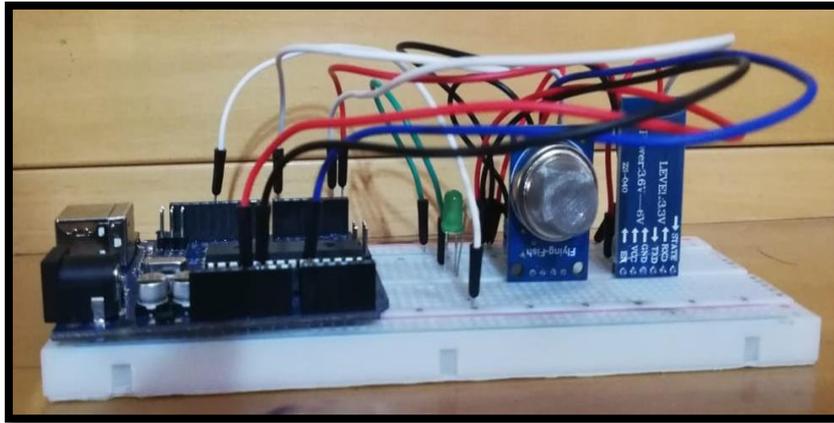


Figura 37. Circuito sensorial armado

Prototipos IDE Arduino

En la Figura 19 se muestra el código fuente utilizado en Arduino (ver Figura 38).

```

final_final_final.txt
#include <SoftwareSerial.h> // Incluimos la libreria SoftwareSerial
SoftwareSerial BT(10,11); // Definimos los pines RX y TX del Arduino conectados al Bluetooth

//*****Hardware Related Macros*****
const int calibrationLed = 13; //When the calibration start , LED pin 13 will light up , off when finish calibrating
const int MQ_PIN = A0; //define which analog input channel you are going to use
int RL_VALUE = 1; //define the load resistance on the board, in kilo ohms
float RO_CLEAN_AIR_FACTOR=9.86; //RO_CLEAN_AIR_FACTOR=(Sensor resistance in clean air)/RO,
//which is derived from the chart in datasheet

//*****Software Related Macros*****
int CALIBRATION_SAMPLE_TIMES=50; //define how many samples you are going to take in the calibration phase
int CALIBRATION_SAMPLE_INTERVAL=500; //define the time interval(in milisecond) between each samples in the
//calibration phase
int READ_SAMPLE_INTERVAL=50; //define how many samples you are going to take in normal operation
int READ_SAMPLE_TIMES=5; //define the time interval(in milisecond) between each samples in
//normal operation

//*****Application Related Macros*****
#define GAS_LPG 0
#define GAS_CO 1
#define GAS_SMOKE 2

//*****Global*****
float LPGCurve[3] = {2.3,0.3,-0.43}; //two points are taken from the curve.
//with these two points, a line is formed which is "approximately equivalent"
//to the original curve.
//data format: { x, y, slope}; point1: (1q200, 0.21), point2: (1q10000, -0.59)
float COCurve[3] = {2.3,0.21,-0.43}; //two points are taken from the curve.

```

Figura 38. Código Arduino

Inmediatamente después se verifico los datos ya calibrados obtenidos por el sensor (ver Figura 41).

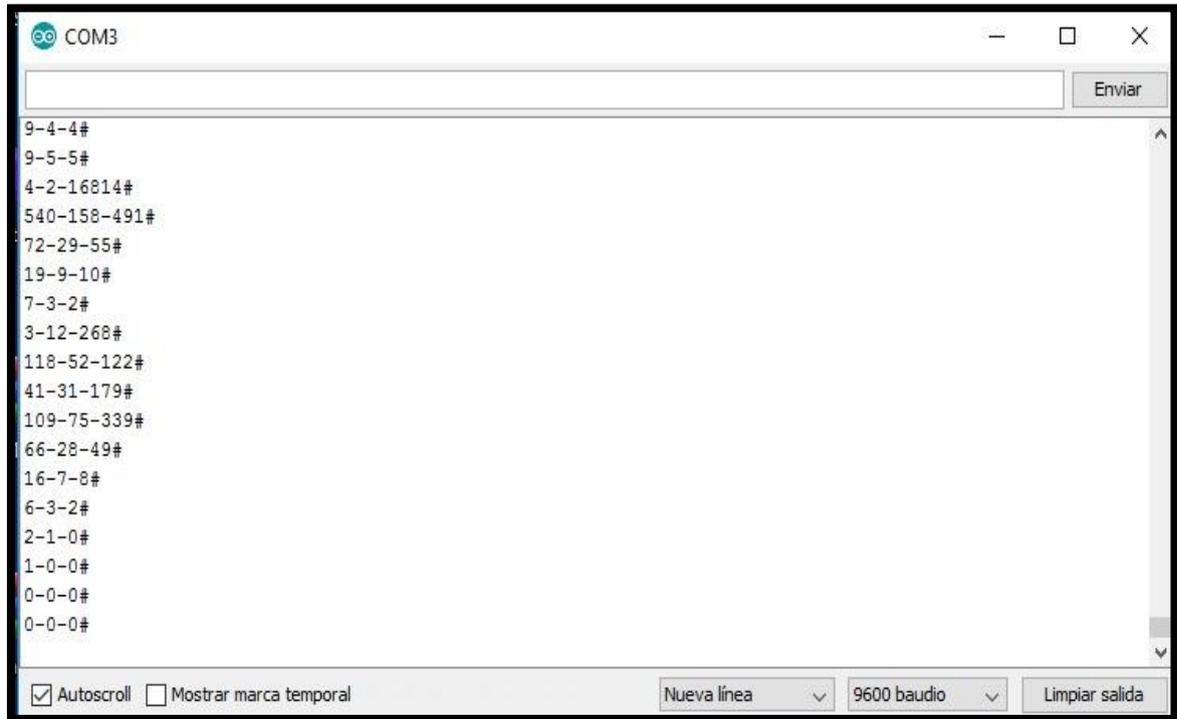


Figura 41. Datos de Sensor calibrado

ANEXO N° 5. Decreto Supremo N° 003-2017-MINAM

6

NORMAS LEGALES

Miércoles 7 de junio de 2017 /  **El Peruano**

AMBIENTE

Aprueban Estándares de Calidad Ambiental (ECA) para Aire y establecen Disposiciones Complementarias

**DECRETO SUPREMO
N° 003-2017-MINAM**

EL PRESIDENTE DE LA REPÚBLICA

CONSIDERANDO:

Que, el numeral 22 del artículo 2 de la Constitución Política del Perú establece que toda persona tiene derecho a gozar de un ambiente equilibrado y adecuado al desarrollo de su vida;

Que, de acuerdo a lo establecido en el artículo 3 de la Ley N° 28611, Ley General del Ambiente, en adelante

la Ley, el Estado, a través de sus entidades y órganos correspondientes, diseña y aplica, entre otros, las normas que sean necesarias para garantizar el efectivo ejercicio de los derechos y el cumplimiento de las obligaciones y responsabilidades contenidas en la citada Ley;

Que, el numeral 31.1 del artículo 31 de la Ley, define al Estándar de Calidad Ambiental (ECA) como la medida que establece el nivel de concentración o del grado de elementos, sustancias o parámetros físicos, químicos y biológicos, presentes en el aire, agua o suelo, en su condición de cuerpo receptor, que no representa riesgo significativo para la salud de las personas ni al ambiente; asimismo, el numeral 31.2 del artículo 31 de la Ley, establece que el ECA es obligatorio en el diseño de las normas legales y las políticas públicas y es un referente obligatorio en el diseño y aplicación de todos los instrumentos de gestión ambiental;

Que, de acuerdo con lo señalado en el numeral 33.1 del artículo 33 de la Ley, la Autoridad Ambiental Nacional dirige el proceso de elaboración y revisión de ECA y Límites Máximos Permisibles (LMP) y, en coordinación con los sectores correspondientes, elabora o encarga las propuestas de ECA y LMP, los que serán remitidos a la Presidencia del Consejo de Ministros para su aprobación mediante Decreto Supremo;

Que, en virtud a lo dispuesto por el numeral 33.4 del artículo 33 de la Ley, en el proceso de revisión de los parámetros de contaminación ambiental, con la finalidad de determinar nuevos niveles de calidad, se aplica el principio de gradualidad, permitiendo ajustes progresivos a dichos niveles para las actividades en curso;

Que, de conformidad con lo establecido en el literal d) del artículo 7 del Decreto Legislativo N° 1013, Ley de Creación, Organización, y Funciones del Ministerio del Ambiente, este ministerio tiene como función específica elaborar los ECA y LMP, los cuales deberán contar con la opinión del sector correspondiente y ser aprobados mediante Decreto Supremo;

Que, mediante Decreto Supremo N° 074-2001-PCM se aprueba el Reglamento de Estándares Nacionales de Calidad Ambiental del Aire, el cual tiene por objetivo establecer los ECA para Aire y los lineamientos de estrategia para alcanzarlos progresivamente;

Que, a través del Decreto Supremo N° 069-2003-PCM, se adiciona el valor anual de concentración de Plomo a los ECA para Aire establecidos en el Reglamento de Estándares Nacionales de Calidad Ambiental del Aire, aprobado por Decreto Supremo N° 074-2001-PCM;

Que, mediante Decreto Supremo N° 003-2008-MINAM, se aprueban nuevos parámetros y valores en los ECA para Aire y se modifica, entre otros, el valor del Dióxido de Azufre;

Que, mediante Decreto Supremo N° 006-2013-MINAM, se aprueban las disposiciones complementarias para la aplicación del ECA de Aire para el Dióxido de Azufre;

Que, a través de la Resolución Ministerial N° 205-2013-MINAM se establecen las cuencas atmosféricas a las cuales les será aplicable los numerales 2.2 y 2.3 del artículo 2 del Decreto Supremo N° 006-2013-MINAM;

Que, mediante Resolución Ministerial N° 331-2016-MINAM se crea el Grupo de Trabajo encargado de establecer medidas para optimizar la calidad ambiental, estableciendo como una de sus funciones específicas, el analizar y proponer medidas para mejorar la calidad ambiental en el país;

Que, en mérito del análisis técnico realizado por el citado Grupo de Trabajo se ha identificado la necesidad de actualizar y unificar la normatividad vigente que regula los ECA para Aire;

Que, por otro lado, mediante Resolución Suprema N° 768-98-PCM, modificada por Resolución Suprema N° 588-99-PCM y Resolución Suprema N° 007-2004-VIVIENDA, se creó el Comité de Gestión de la Iniciativa del Aire Limpio para Lima y Callao, con la finalidad de proponer mecanismos de coordinación interinstitucional y los cambios normativos orientados a la mejora de la calidad del aire de Lima y Callao;

Que, resulta necesario que el referido Comité se enmarque dentro de lo dispuesto en el numeral 3 del artículo 36 de la Ley N° 29158, Ley Orgánica del Poder Ejecutivo, que establece que las Comisiones Multisectoriales de

naturaleza permanente son creadas con fines específicos para cumplir funciones de seguimiento, fiscalización, o emisión de informes técnicos. Se crean formalmente mediante decreto supremo refrendado por el Presidente del Consejo de Ministros y los titulares de los Sectores involucrados. Cuentan con Reglamento Interno aprobado por Resolución Ministerial del Sector al cual están adscritas;

Que, mediante Resolución Ministerial N° 094-2017-MINAM, el Ministerio del Ambiente dispuso la prepublicación del proyecto de Decreto Supremo que aprueba los ECA para aire y establece disposiciones complementarias, en cumplimiento del Reglamento sobre Transparencia, Acceso a la Información Pública Ambiental y Participación y Consulta Ciudadana en Asuntos Ambientales, aprobado por Decreto Supremo N° 002-2009-MINAM, y el artículo 14 del Reglamento que establece disposiciones relativas a la publicidad, publicación de Proyectos Normativos y difusión de Normas Legales de Carácter General, aprobado por Decreto Supremo N° 001-2009-JUS; en virtud de la cual se recibieron aportes y comentarios al mismo;

De conformidad con lo dispuesto en el numeral 8 del artículo 118 de la Constitución Política del Perú, así como el numeral 3 del artículo 11 de la Ley N° 29158, Ley Orgánica del Poder Ejecutivo;

DECRETA:

Artículo 1.- Aprobación de los Estándares de Calidad Ambiental para Aire

Apruébase los Estándares de Calidad Ambiental (ECA) para Aire, que como Anexo forman parte integrante del presente Decreto Supremo.

Artículo 2.- Los Estándares de Calidad Ambiental para Aire como referente obligatorio

2.1 Los ECA para Aire son un referente obligatorio para el diseño y aplicación de los instrumentos de gestión ambiental, a cargo de los titulares de actividades productivas, extractivas y de servicios.

2.2 Los ECA para Aire, como referente obligatorio, son aplicables para aquellos parámetros que caracterizan las emisiones de las actividades productivas, extractivas y de servicios.

Artículo 3.- Financiamiento

El financiamiento para la aplicación de lo dispuesto en la presente norma, se realizará con cargo al presupuesto institucional de los pliegos involucrados, sin demandar recursos adicionales al Tesoro Público.

Artículo 4.- Refrendo

El presente Decreto Supremo es refrendado por la Ministra del Ambiente, la Ministra de Salud, el Ministro de Transportes y Comunicaciones, el Ministro de Energía y Minas, el Ministro de la Producción y el Ministro de Vivienda, Construcción y Saneamiento.

DISPOSICIONES COMPLEMENTARIAS FINALES

Primera.- Aplicación de los ECA para Aire en los instrumentos de gestión ambiental aprobados

La aplicación de los ECA para Aire en los instrumentos de gestión ambiental aprobados, que sean de carácter preventivo, se realiza en la actualización o modificación de los mismos, en el marco de la normativa vigente del Sistema Nacional de Evaluación del Impacto Ambiental (SEIA). En el caso de instrumentos correctivos, la aplicación de los ECA para Aire se realiza conforme a la normativa ambiental sectorial.

Segunda.- Monitoreo de la calidad del aire

Mediante Decreto Supremo, a propuesta del Ministerio del Ambiente, en coordinación con las autoridades competentes, se aprobará el Protocolo Nacional de Monitoreo de la Calidad Ambiental del Aire, en un plazo máximo de ciento ochenta (180) días calendario, contados desde la entrada en vigencia del presente dispositivo.

En tanto se apruebe el citado Protocolo Nacional, el monitoreo de la calidad del aire se realizará conforme a la normativa vigente.

Tercera.- Grupos de Estudio Técnico Ambiental de Calidad del Aire

El Ministerio del Ambiente, mediante resolución ministerial, en el plazo máximo de noventa (90) días calendario, contados desde la entrada en vigencia del presente Decreto Supremo, aprobará los lineamientos para fortalecer e incorporar a los Grupos de Estudio Técnico Ambiental de la Calidad del Aire en las Comisiones Ambientales Municipales (CAM) Provinciales, en el marco del Sistema Nacional de Gestión Ambiental.

En tanto se apruebe los citados lineamientos, los Grupos de Estudio Técnico Ambiental de la Calidad del Aire continuarán ejerciendo las siguientes funciones: a) Supervisar los diagnósticos de línea base; b) Formular los planes de acción para el mejoramiento de la calidad del aire; y c) Proponer las medidas inmediatas que deban realizarse en los estados de alerta nacionales para contaminantes del aire.

Cuarta.- Zonas de Atención Prioritaria

Las Zonas de Atención Prioritaria son aquellos centros poblados que cuenten con actividades económicas que planteen real o potencial afectación en la calidad del aire, que posean actividad vehicular ambientalmente relevante, o que cuenten con una dinámica urbana que implique un potencial incremento de emisiones atmosféricas.

El Ministerio del Ambiente, mediante resolución ministerial, desarrollará los lineamientos para la determinación de las Zonas de Atención Prioritaria en un plazo máximo de ciento veinte (120) días calendario, contados desde la entrada en vigencia del presente decreto supremo. Asimismo, el Ministerio del Ambiente, mediante resolución ministerial, establecerá las Zonas de Atención Prioritaria, en coordinación con las autoridades competentes, Las Zonas de Atención Prioritaria creadas con anterioridad mantienen su vigencia.

Quinta.- Planes de Acción para el Mejoramiento de la Calidad del Aire

Los Planes de Acción para el Mejoramiento de la Calidad del Aire de las provincias conurbadas de Lima y el Callao, son formulados por la Comisión Multisectorial de Gestión de la Iniciativa de Aire Limpio para Lima y Callao, y aprobados por el Ministerio del Ambiente.

Los Planes de Acción para el Mejoramiento de la Calidad del Aire de las demás provincias serán aprobados mediante Ordenanza Municipal del Gobierno Local competente. En tanto se aprueben los lineamientos a los que se hacen referencia en la tercera disposición complementaria final, los Planes de Acción aprobados con anterioridad, deberán continuar con su implementación.

La Dirección General de Calidad Ambiental del Ministerio del Ambiente realizará el seguimiento de la implementación de los Planes de Acción.

Sexta.- Estados de Alerta Nacionales para contaminantes del aire

El Ministerio de Salud es la autoridad competente para declarar los Estados de Alerta Nacionales para contaminantes del aire que tengan por objeto activar, en forma inmediata, un conjunto de medidas destinadas a prevenir el riesgo a la salud y evitar la exposición excesiva de la población a los contaminantes del aire, durante episodios de contaminación aguda.

El Ministerio de Salud, en coordinación con el Ministerio del Ambiente, establece los niveles de Estados de Alerta.

Séptima.- Estándar de Calidad Ambiental para Aire de Mercurio Gaseoso Total

El ECA para Aire del parámetro Mercurio Gaseoso Total, aprobado mediante el presente Decreto Supremo, entrará en vigencia al día siguiente de la publicación del Protocolo Nacional de Monitoreo de la Calidad Ambiental del Aire.

DISPOSICIÓN COMPLEMENTARIA TRANSITORIA**Única.- Instrumento de gestión ambiental en trámite ante la Autoridad Competente**

Los titulares que antes de la fecha de entrada en vigencia de la norma, hayan iniciado un procedimiento

administrativo para aprobación del instrumento de gestión ambiental ante la autoridad competente, tomarán en consideración los ECA para Aire vigentes a la fecha de inicio del procedimiento.

Luego de aprobado el instrumento de gestión ambiental por la autoridad competente, los titulares deberán considerar lo establecido en la Primera Disposición Complementaria Final, a efectos de aplicar los ECA para Aire aprobados mediante el presente Decreto Supremo.

DISPOSICIONES COMPLEMENTARIAS MODIFICATORIAS**Primera.- Adecuación del Comité de Gestión de la Iniciativa del Aire Limpio para Lima y Callao a Comisión Multisectorial de naturaleza permanente**

Modifícase el artículo 1 de la Resolución Suprema N° 768-98-PCM, modificada por las Resoluciones Supremas N° 588-99-PCM y N° 007-2004-VIVIENDA, así como los artículos 2, 3 y 4 de la Resolución Suprema N° 007-2004-VIVIENDA, e incorpórase el artículo 5 en la Resolución Suprema N° 007-2004-VIVIENDA, los mismos que quedan redactados conforme al siguiente texto:

"Artículo 1.- Constituir la Comisión Multisectorial para la Gestión de la Iniciativa del Aire Limpio para Lima y Callao, la cual está adscrita al Ministerio del Ambiente e integrado por:

- El/la Viceministro (a) de Gestión Ambiental del Ministerio del Ambiente, quien lo presidirá.
- El/la Directora(a) General de Asuntos Ambientales de Industria del Ministerio de la Producción.
- El/la Director(a) General de Asuntos Ambientales Pesqueros y Acuícolas del Ministerio de la Producción.
- El/la Director(a) General de Asuntos Ambientales Energéticos del Ministerio de Energía y Minas.
- El/la Director(a) General de la Dirección General de Calidad Ambiental del Ministerio del Ambiente.
- El/la Director(a) General de la Dirección General de Asuntos Ambientales del Ministerio de Vivienda, Construcción y Saneamiento.
- El/la Director(a) General de la Dirección General de Transporte Terrestre del Ministerio de Transportes y Comunicaciones.
- El/la Director(a) General de la Dirección General de Asuntos Socio-Ambientales del Ministerio de Transportes y Comunicaciones.
- El/la Director(a) General de la Dirección General de Salud Ambiental e Inocuidad Alimentaria del Ministerio de Salud.
- El/la Presidente(a) Ejecutivo(a) del Servicio Nacional de Meteorología e Hidrología del Perú - SENAMHI.
- Dos representantes de la Municipalidad Metropolitana de Lima.
- Dos representantes de la Municipalidad Provincial del Callao.
- Dos representantes de la Confederación Nacional de Instituciones Empresariales Privadas - CONFIEP.
- Un representante de la Asociación Peruana de Consumidores y Usuarios - ASPEC.

Los representantes podrán acreditar ante la Secretaría Técnica a sus representantes alternos.

La Comisión podrá convocar, a personas naturales o jurídicas de derecho privado o público, a participar en sus reuniones, de acuerdo a la materia a ser tratada".

"Artículo 2.- Creación de la Secretaría Técnica

Créase la Secretaría Técnica de la Comisión Multisectorial para la Gestión de la Iniciativa del Aire Limpio para Lima y Callao, la cual dependerá del Viceministerio de Gestión Ambiental."

"Artículo 3.- Reglamento Interno

La Comisión Multisectorial para la Gestión de la Iniciativa del Aire Limpio para Lima y Callao, formulará su Reglamento Interno, el cual será aprobado mediante

Resolución Ministerial expedida por el Ministerio del Ambiente."

"Artículo 4.- Finalidad

La Comisión Multisectorial para la Gestión de la Iniciativa del Aire Limpio para Lima y Callao emitirá los informes técnicos que contengan las propuestas de mecanismos de coordinación interinstitucional y las modificaciones normativas orientadas a mejorar la calidad del aire de Lima y Callao."

"Artículo 5.- Financiamiento

El cumplimiento de las funciones de la Comisión Multisectorial para la Gestión de la Iniciativa del Aire Limpio para Lima y Callao se financia con cargo al presupuesto institucional del Ministerio del Ambiente, sin demandar recursos adicionales al Tesoro Público. Asimismo, los gastos que pueda involucrar la participación de los representantes de la citada Comisión Multisectorial se financian con cargo al presupuesto de las entidades a las cuales pertenecen".

Segunda.- Modificación del Reglamento Interno de la Comisión Multisectorial para la Gestión de la Iniciativa del Aire Limpio para Lima y Callao

Mediante Resolución Ministerial emitida por el Ministerio del Ambiente, en el plazo máximo de treinta (30) días hábiles contados desde la entrada en vigencia del presente Decreto Supremo, la Comisión Multisectorial para la Gestión de la Iniciativa del Aire Limpio para Lima y Callao modificará su Reglamento Interno, aprobado por Resolución Ministerial N° 229-2013-VIVIENDA.

**DISPOSICIÓN COMPLEMENTARIA
DEROGATORIA**

Única.- Derogación de normas referidas al ECA para Aire

Derógase el Decreto Supremo N° 074-2001-PCM, el Decreto Supremo N° 069-2003-PCM, el Decreto Supremo N° 003-2008-MINAM y el Decreto Supremo N° 006-2013-MINAM.

Dado en la Casa de Gobierno, en Lima, a los seis días del mes de junio del año dos mil diecisiete.

PEDRO PABLO KUCZYNSKI GODARD
Presidente de la República

ELSA GALARZA CONTRERAS
Ministra del Ambiente

GONZALO TAMAYO FLORES
Ministro de Energía y Minas

PEDRO OLAECHEA ÁLVAREZ CALDERÓN
Ministro de la Producción

PATRICIA J. GARCÍA FUNEGRA
Ministra de Salud

BRUNO GIUFFRA MONTEVERDE
Ministro de Transportes y Comunicaciones

EDMER TRUJILLO MORI
Ministro de Vivienda, Construcción y Saneamiento

**Anexo
Estándares de Calidad Ambiental para Aire**

Parámetros	Periodo	Valor [µg/m³]	Criterios de evaluación	Método de análisis ¹⁾
Benceno (C ₆ H ₆)	Anual	2	Media aritmética anual	Cromatografía de gases
Dióxido de Azufre (SO ₂)	24 horas	250	NE más de 7 veces al año	Fluorescencia ultravioleta (Método automático)
Dióxido de Nitrógeno (NO ₂)	1 hora	200	NE más de 24 veces al año	Quimioluminiscencia (Método automático)
	Anual	100	Media aritmética anual	
Material Particulado con diámetro menor a 2,5 micras (PM _{2,5})	24 horas	50	NE más de 7 veces al año	Separación inercial/filtración (Gravimetría)
	Anual	25	Media aritmética anual	
Material Particulado con diámetro menor a 10 micras (PM ₁₀)	24 horas	100	NE más de 7 veces al año	Separación inercial/filtración (Gravimetría)
	Anual	50	Media aritmética anual	
Mercurio Gaseoso Total (Hg) ²⁾	24 horas	2	No exceder	Espectrometría de absorción atómica de vapor frío (CVAAS) o Espectrometría de fluorescencia atómica de vapor frío (CVAFS) o Espectrometría de absorción atómica Zeeman. (Métodos automáticos)
Monóxido de Carbono (CO)	1 hora	30000	NE más de 1 vez al año	Infrarrojo no dispersivo (NDIR) (Método automático)
	8 horas	10000	Media aritmética móvil	
Ozono (O ₃)	8 horas	100	Máxima media diaria NE más de 24 veces al año	Fotometría de absorción ultravioleta (Método automático)
Plomo (Pb) en PM ₁₀	Mensual	1,5	NE más de 4 veces al año	Método para PM ₁₀ (Espectrofotometría de absorción atómica)
	Anual	0,5	Media aritmética de los valores mensuales	
Sulfuro de Hidrógeno (H ₂ S)	24 horas	150	Media aritmética	Fluorescencia ultravioleta (Método automático)

NE: No Exceder.

¹⁾ o método equivalente aprobado.

²⁾ El estándar de calidad ambiental para Mercurio Gaseoso Total entrará en vigencia al día siguiente de la publicación del Protocolo Nacional de Monitoreo de la Calidad Ambiental del Aire, de conformidad con lo establecido en la Séptima Disposición Complementaria Final del presente Decreto Supremo.

1529835-1

ANEXO N° 6. Algoritmo Árbol de Decisión

Machine Learning ML.NET

ML.NET ofrece la posibilidad de agregar aprendizaje automático en aplicaciones de .NET, ya sea en escenarios en línea o sin conexión. Con esta funcionalidad, puede realizar predicciones automáticas con los datos disponibles para su aplicación.

La base de ML.NET es un **modelo** de aprendizaje automático. El modelo especifica los pasos necesarios para transformar los datos de entrada en una predicción. Con ML.NET, puede entrenar un modelo personalizado mediante la especificación de un algoritmo, o bien puede importar modelos TensorFlow y ONNX previamente entrenados.

Una vez que tiene un modelo, puede agregarlo a la aplicación para realizar las predicciones.

ML.NET se ejecuta en Windows, Linux y macOS con .NET Core o en Windows mediante .NET Framework. La versión de 64 bits es compatible con todas las plataformas. La versión de 32 bits es compatible con Windows, salvo para la funcionalidad relacionada con TensorFlow, LightGBM y ONNX (Microsoft, 2020) (ver Tabla 38).

Algunos ejemplos del tipo de predicciones que puede hacer con ML.NET:

Tabla 38 *Tipo de predicciones que puede hacer con ML.NET*

Clasificación y categorización	Clasifique automáticamente los comentarios del cliente en positivos y negativos.
Valores continuos de regresión y predicción	Prediga el precio de la vivienda según el tamaño y la ubicación.
Detección de anomalías	Detecte transacciones bancarias fraudulentas.
Recomendaciones	Sugiera productos que los compradores en línea pueden comprar, en función de sus compras anteriores.
Series temporales y datos secuenciales	Haga una previsión del tiempo y de las ventas de productos
Clasificación de la imagen	Clasifique las patologías de imágenes médicas

Fuente: (Microsoft, 2020)

Flujo de trabajo del código

El siguiente diagrama representa la estructura del código de aplicación, así como el proceso iterativo de desarrollo de modelos (ver Figura 42).

- Recopilar y cargar datos de entrenamiento en un objeto IDataView
- Especificar una canalización de operaciones para extraer características y aplicar un algoritmo de aprendizaje automático
- Entrenar un modelo mediante una llamada a Fit() en la canalización
- Evaluar el modelo e iterar para mejorar

- Guardar el modelo en formato binario, para su uso en una aplicación
- Cargar el modelo en un objeto ITransformer
- Realizar predicciones mediante una llamada a CreatePredictionEngine.Predict()

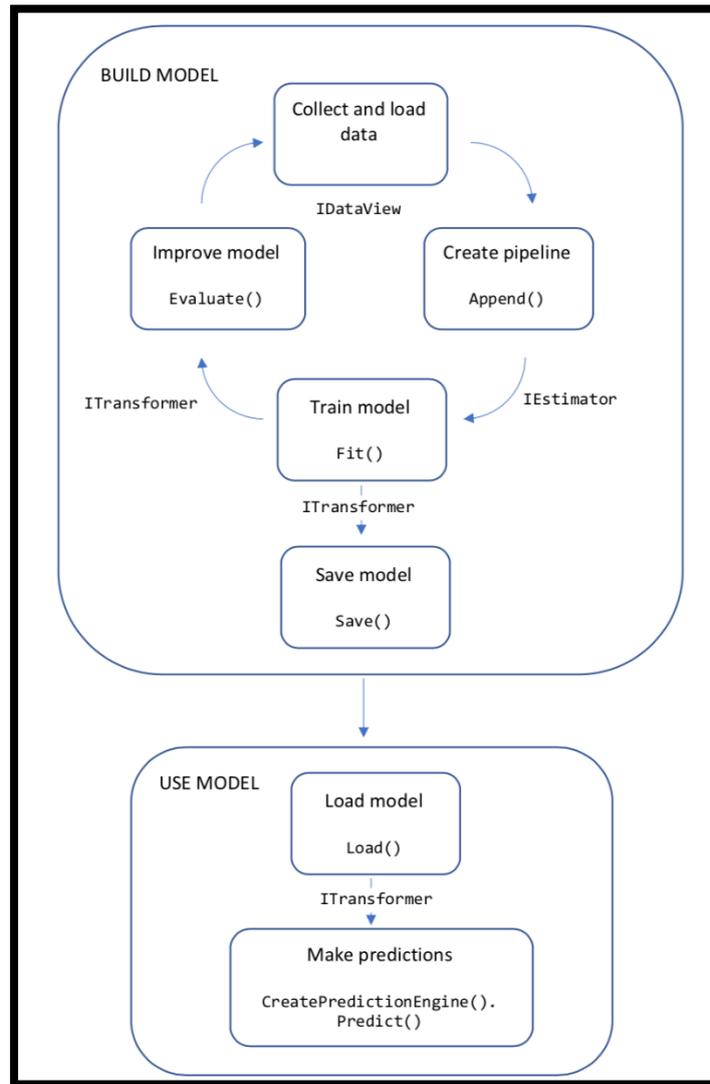


Figura 42. Flujo De Trabajo Del Código

Fuente: (Microsoft, 2020)

Modelo de Machine Learning

Un modelo de ML.NET es un objeto que contiene las transformaciones que se realizarán en los datos de entrada para llegar a los resultados previstos.

Básico

El modelo más básico es una regresión lineal bidimensional, donde una cantidad continua es proporcional a otra, como se muestra en el ejemplo anterior del precio de la vivienda (ver Figura 43).

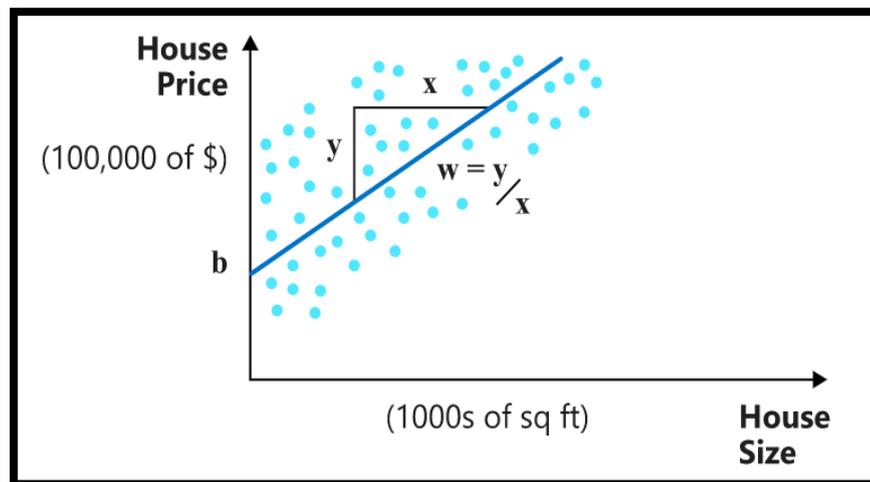


Figura 43. El modelo más básico es una regresión lineal

Fuente: (Microsoft, 2020)

El modelo es simplemente: **Precio= b+Tamaño*w**. Los parámetros b y w se calculan ajustando una línea en un conjunto de pares (tamaño, precio). Los datos que se usan para buscar los parámetros del modelo se denominan datos de aprendizaje. Las entradas de un modelo de Machine Learning se denominan características. En este ejemplo, Tamaño es la única característica. Los valores de datos reales en el terreno que se usan para entrenar un

modelo de Machine Learning se denominan **etiquetas**. En este caso, los valores de Precio en el conjunto de datos de entrenamiento son las etiquetas (Microsoft, 2020).

Más complejo

Un modelo más complejo clasifica las transacciones financieras en categorías utilizando la descripción de texto de la transacción.

La descripción de cada transacción se divide en un conjunto de características quitando palabras y caracteres redundantes, y contando combinaciones de palabras y caracteres. El conjunto de características se utiliza para entrenar un modelo lineal según el conjunto de categorías en los datos de aprendizaje. Cuanto más similar es una nueva descripción a las del conjunto de entrenamiento, más probabilidades tienen que se le asigne la misma categoría (ver Figura 44).

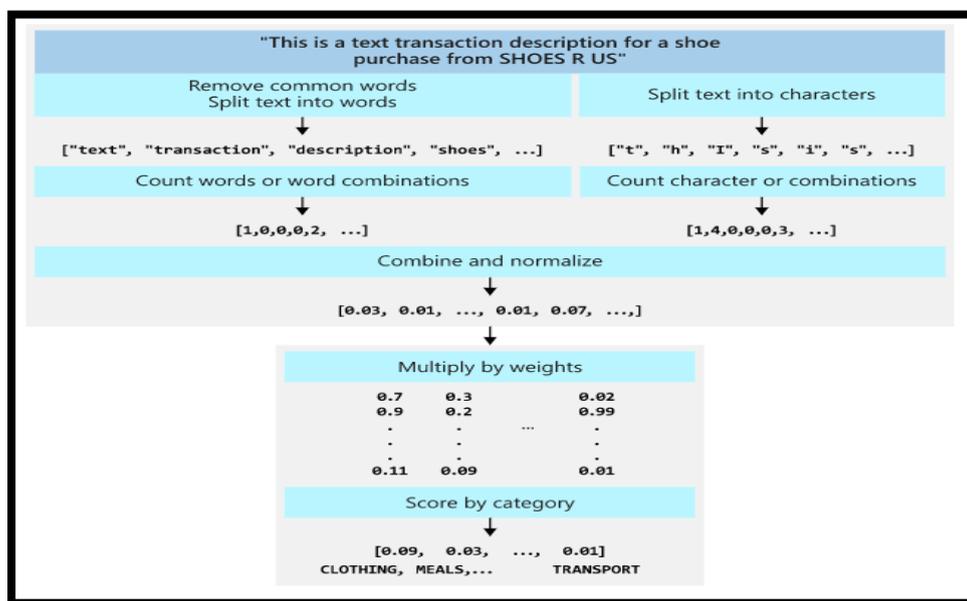


Figura 44. Descripción a las del conjunto de entrenamiento

Fuente: (Microsoft, 2020)

Tanto el modelo de precios de vivienda como el modelo de clasificación de textos son modelos **lineales**. Según la naturaleza de los datos y el problema que se va a resolver, también puede usar modelos de **árbol de decisión** y **modelos aditivos generalizados**, entre otros. Para obtener más información sobre los modelos, vea Tareas (Microsoft, 2020).

Preparación de datos

En la mayoría de los casos, los datos que tiene a su disposición no son aptos para usarse directamente en el entrenamiento de un modelo de Machine Learning. Los datos sin procesar deben prepararse o procesarse previamente antes de que se puedan usar para buscar los parámetros del modelo. Es posible que los datos deban convertirse de valores de cadena a una representación numérica. Puede que tenga información redundante en los datos de entrada. Tal vez necesite reducir o expandir las dimensiones de los datos de entrada. Puede que los datos deban normalizarse o escalarse.

Los tutoriales de ML.NET le proporcionan información sobre las diferentes canalizaciones de procesamiento de datos para texto, imágenes, datos numéricos y de series temporales que se usan en tareas específicas de aprendizaje automático.

Cómo preparar los datos le explica cómo aplicar la preparación de datos de manera más general (Microsoft, 2020).

Encontrará un apéndice de todas las transformaciones disponibles en la sección de recursos.

Evaluación de modelo

Una vez que ha entrenado el modelo, ¿cómo sabe hasta qué punto realizará predicciones futuras? Con ML.NET, puede evaluar su modelo con algunos datos de prueba nuevos.

Cada tipo de tarea de aprendizaje automático tiene métricas que se usan para evaluar la precisión y exactitud del modelo en el conjunto de datos de prueba.

En nuestro ejemplo de precios de vivienda, hemos usado la tarea **Regresión**. Para evaluar el modelo, agregue el código siguiente en el ejemplo original.

Las métricas de evaluación indican que el error es bastante bajo y que la correlación entre la salida de predicción y la salida de prueba es alta. ¡Qué sencillo! En los ejemplos reales, se necesitan más ajustes para lograr las métricas del modelo adecuado (Microsoft, 2020).

Arquitectura de ML.NET

En esta sección, veremos los patrones arquitectónicos de ML.NET. Si es un desarrollador experimentado de .NET, algunos de estos patrones le resultarán familiares y otros serán menos conocidos. ¡Sujétese bien mientras nos sumergimos!

Una aplicación de ML.NET se inicia con un objeto `MLContext`. Este objeto singleton contiene catálogos. Un catálogo es una fábrica para cargar y guardar datos, transformaciones, instructores y componentes de la operación de modelos. Cada objeto de catálogo tiene métodos para crear los diferentes tipos de componentes (ver Tabla 39):

Tabla 39 *Patrones arquitectónicos de ML.NET*

Carga y guardado de datos		DataOperationsCatalog	
Preparación de datos		TransformsCatalog	
Algoritmos de entrenamiento	Clasificación binaria	BinaryClassificationCatalog	
	Clasificación multiclase	MulticlassClassificationCatalog	
	Detección de anomalías	AnomalyDetectionCatalog	
	Agrupación en clústeres	ClusteringCatalog	
	Previsión	ForecastingCatalog	
	Clasificación	RankingCatalog	
	Regresión	RegressionCatalog	
	Recomendación	RecommendationCatalog	Adición del paquete NuGet Microsoft.ML.Recommender
TimeSeries	TimeSeriesCatalog	Adición del paquete NuGet Microsoft.ML.TimeSeries	
Uso del modelo		ModelOperationsCatalog	

Fuente: (Microsoft, 2020)

Puede navegar a los métodos de creación en cada una de las categorías anteriores. Con Visual Studio, los catálogos se muestran a través de IntelliSense (ver Figura 45).

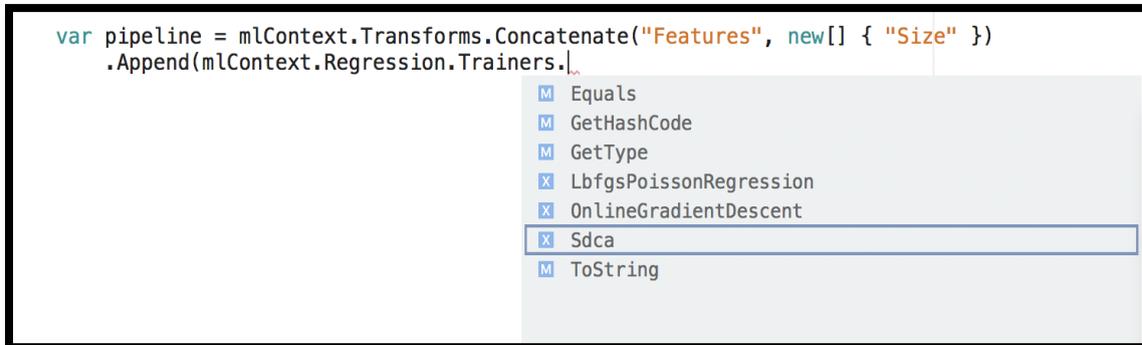


Figura 45. Catálogos que muestran de IntelliSense

Fuente: (Microsoft, 2020)

Crear la canalización

Dentro de cada catálogo hay un conjunto de métodos de extensión. Echemos un vistazo a la forma en que se usan los métodos de extensión para crear una canalización de entrenamiento.

```
C#Copiar

var pipeline = mlContext.Transforms.Concatenate("Features", new[] { "Size" })
    .Append(mlContext.Regression.Trainers.Sdca(labelColumnName: "Price",
maximumNumberOfIterations: 100));
```

En el fragmento de código, Concatenate y Sdca son métodos en el catálogo. Cada uno crea un objeto IEstimator que se anexa a la canalización.

En este momento, solo se crean los objetos. No se ha producido ninguna ejecución.

Entrenar el modelo

Una vez que se han creado los objetos en la canalización, se pueden usar datos para entrenar el modelo.

```
C#Copiar  
  
var model = pipeline.Fit(trainingData);
```

Una llamada a Fit() usa los datos de entrenamiento de entrada para calcular los parámetros del modelo. Esto se conoce como entrenar el modelo. Recuerde que el modelo de regresión lineal anterior tenía dos parámetros de modelo: sesgo y peso. Después de la llamada de Fit(), se conocen los valores de los parámetros. La mayoría de los modelos tendrá muchos más parámetros que esto.

Puede obtener más información acerca del entrenamiento del modelo en [Cómo entrenar el modelo](#).

El objeto del modelo resultante implementa la interfaz ITransformer. Es decir, el modelo transforma datos de entrada en predicciones.

```
C#Copiar  
  
IDataView predictions = model.Transform(inputData);
```

Uso del modelo

Puede transformar datos de entrada en predicciones de forma masiva o una entrada a la vez. En el ejemplo de precios de vivienda, lo hicimos de ambas maneras: de forma masiva

con el fin de evaluar el modelo y de uno en uno para realizar una predicción nueva. Echemos un vistazo a la realización de predicciones únicas.

```
C#Copiar
```

```
var size = new HouseData() { Size = 2.5F };  
  
var predEngine = mlContext.CreatePredictionEngine<HouseData, Prediction>(model);  
  
var price = predEngine.Predict(size);
```

El método `CreatePredictionEngine()` toma una clase de entrada y una clase de salida. Los atributos de código o los nombres de campo determinan los nombres de las columnas de datos utilizadas durante el entrenamiento del modelo y la predicción. Puede leer más sobre cómo realizar una sola predicción en la sección de instrucciones (Microsoft, 2020).

Esquema y modelos de datos

En el núcleo de una canalización de aprendizaje automático de ML.NET están los objetos `DataRow`.

Cada transformación en la canalización tiene un esquema de entrada (nombres, tipos y tamaños de datos que la transformación espera ver en su entrada), así como un esquema de salida (nombres, tipos y tamaños de datos que la transformación produce después de la misma). En el siguiente documento se ofrece una explicación detallada de la interfaz de `IDataView` y su sistema de tipos.

Si el esquema de salida de una transformación en la canalización no coincide con el esquema de entrada de la siguiente transformación, ML.NET producirá una excepción.

Un objeto de vista de datos tiene columnas y filas. Cada columna tiene un nombre, un tipo y una longitud. Por ejemplo, las columnas de entrada en el ejemplo de precios de vivienda son Tamaño y Precio. Ambas son tipos y cantidades escalares, en lugar de vectores (ver Figura 46).

	Size Single Scalar	Price Single Scalar	Features Single Vector(1)	Score Single Scalar
Rows {	1.1	1.2	[1.1]	1.29
	1.9	2.3	[1.9]	2.14
	2.8	3.0	[2.8]	3.10
	3.4	3.7	[3.4]	3.75

Columns

Figura 46. Esquema y modelos de datos

Fuente: (Microsoft, 2020)

Todos los algoritmos ML.NET buscan una columna de entrada que sea un vector. De forma predeterminada, esta columna de vector se denomina Características. Este es el motivo por el que concatenamos la columna Tamaño en una nueva columna denominada Características en nuestro ejemplo de precios de vivienda (Microsoft, 2020).

C#Copiar

```
var pipeline = mlContext.Transforms.Concatenate("Features", new[] { "Size" })
```

Todos los algoritmos también crean nuevas columnas una vez que han realizado una predicción. Los nombres fijos de estas nuevas columnas dependen del tipo de algoritmo de aprendizaje automático. Para la tarea de regresión, una de las nuevas columnas se denomina

Puntuación. Este es el motivo por el que atribuimos nuestros datos de precios con este nombre.

```
C#Copiar

public class Prediction{

    [ColumnName("Score")]

    public float Price { get; set; }}
```

Si desea obtener más información acerca de las columnas de salida de diferentes tareas de aprendizaje automático, vea la guía Tareas de Machine Learning.

Una propiedad importante de los objetos `DataGridView` es que se evalúan **de forma diferida**. Las vistas de datos solo se cargan y utilizan en las operaciones durante el entrenamiento del modelo, la evaluación y la predicción de datos. Mientras escribe y prueba la aplicación de ML.NET, puede utilizar el depurador de Visual Studio para echar un vistazo a cualquier objeto de vista de datos llamando al método `Preview`.

```
C#Copiar

var debug = testPriceDataGridView.Preview();
```

Puede ver la variable `debug` en el depurador y examinar su contenido. No utilice el método `Preview` en el código de producción, ya que reduce significativamente el rendimiento.

Implementación de modelos

En las aplicaciones de la vida real, el código de entrenamiento y evaluación del modelo será independiente de la predicción. De hecho, estas dos actividades las realizan a menudo equipos diferentes. El equipo de desarrollo del modelo puede guardar el modelo para su uso en la aplicación de predicción (Microsoft, 2020).

```
C#Copiar
```

```
mlContext.Model.Save(model, trainingData.Schema,"model.zip");
```

Cómo elegir un algoritmo de ML.NET

Para cada tarea de ML.NET, existen varios algoritmos de entrenamiento de donde elegir. La elección depende del problema que intenta resolver, las características de los datos y los recursos de proceso y almacenamiento que tiene a su disposición. Es importante tener en cuenta que entrenar un modelo de Machine Learning es un proceso iterativo. Es posible que deba probar varios algoritmos para encontrar el que mejor funcione.

Los algoritmos funcionan con características. Las características son valores numéricos calculados a partir de los datos de entrada. Son entradas óptimas para los algoritmos de aprendizaje automático. Transforme sus datos de entrada sin formato en características mediante una o varias de las transformaciones de datos. Por ejemplo, los datos de texto se transforman en un conjunto de recuentos de palabras y recuentos de combinación de palabras. Una vez extraídas las características de un tipo de datos sin formato mediante transformaciones de datos, se dice que están caracterizadas. Por ejemplo, texto caracterizado o datos de imagen caracterizados (Microsoft, 2020).

Instructor = algoritmo + tarea

Un algoritmo es la expresión matemática que se ejecuta para generar un modelo. Diferentes algoritmos generan modelos con diferentes características.

Con ML.NET, se puede aplicar el mismo algoritmo a distintas tareas. Por ejemplo, se puede usar el ascenso coordinado dual estocástico para la clasificación binaria, la clasificación multiclase y la regresión. La diferencia radica en cómo se interpreta la salida del algoritmo para que coincida con la tarea.

Para cada combinación de tarea o algoritmo, ML.NET proporciona un componente que ejecuta el algoritmo de entrenamiento y no la interpretación. Estos componentes se denominan instructores. Por ejemplo, `SdcaRegressionTrainer` usa el algoritmo `StochasticDualCoordinatedAscent` que se aplica a la tarea de regresión (Microsoft, 2020).

Algoritmos lineales

Los algoritmos lineales generan un modelo que calcula puntuaciones desde una combinación lineal de los datos de entrada y un conjunto de pesos. Los pesos son parámetros del modelo estimado durante el entrenamiento.

Los algoritmos lineales funcionan bien para las características que se pueden separar de manera lineal.

Antes del entrenamiento con un algoritmo lineal, se deben normalizar las características. Esto evita que una característica tenga más influencia sobre el resultado que otros.

En general, los algoritmos lineales son escalables, rápidos y baratos de entrenar y predecir. Se escalan por el número de características y aproximadamente por el tamaño del conjunto de datos de entrenamiento.

Los algoritmos lineales realizan varios recorridos por los datos de entrenamiento. Si el conjunto de datos cabe en la memoria, la adición de un punto de control de memoria caché a la canalización de ML.NET antes de anexar el instructor hará que el entrenamiento se ejecute más rápidamente (Microsoft, 2020).

Algoritmos de árbol de decisión

Los algoritmos de árbol de decisión crean un modelo que contiene una serie de decisiones: eficazmente un diagrama de flujo a través de los valores de datos.

Para usar este tipo de algoritmo, no es necesario poder separar las características de forma lineal. Tampoco es necesario normalizar las características, porque los valores individuales en el vector de la característica se usan de forma independiente en el proceso de toma de decisiones.

Los algoritmos de árbol de decisión normalmente son muy precisos.

Salvo por los modelos aditivos generalizados (GAM), los modelos de árbol carecen de explicación cuando el número de características es grande.

Los algoritmos de árbol de decisión consumen más recursos y no se escalan tan bien como los lineales. Se desempeñan bien en los conjuntos de datos que pueden caber en la memoria.

Los árboles de decisión potenciados son un conjunto de árboles pequeños, donde cada árbol asigna una puntuación a los datos de entrada y pasa la puntuación al árbol siguiente para generar una mejor puntuación y así sucesivamente, donde cada árbol en el conjunto mejora al anterior (Microsoft, 2020).

Instructores para árbol de decisión

Se muestra en la siguiente tabla a las propiedades e instructores para cada tipo de algoritmo (ver Tabla 40).

Tabla 40 *Instructores para árbol de decisión*

Algoritmo	Propiedades	Instructores
Máquina potenciada de degradado ligero	Instructores de árbol de clasificación binaria y rápidos precisos. Altamente optimizable	LightGbmBinaryTrainer LightGbmMulticlassTrainer LightGbmRegressionTrainer LightGbmRankingTrainer
Árbol rápido	Se usa para los datos de imagen caracterizados. Resistente a los datos no equilibrados. Altamente optimizable	FastTreeBinaryTrainer FastTreeRegressionTrainer FastTreeTweedieTrainer FastTreeRankingTrainer
Bosque rápido	Funciona bien con datos ruidosos	FastForestBinaryTrainer FastForestRegressionTrainer
Modelo aditivo generalizado (GAM)	Ideal para problemas que funcionan bien con los algoritmos de árbol, pero en los que la explicación es una prioridad	GamBinaryTrainer GamRegressionTrainer

Fuente: (Microsoft, 2020)

ANEXO N° 7. Product Backlog del Sistema EcoFavBen

El sistema de trabajo que se desarrolló para la presente investigación está compuesto por Sprint, historias de usuario y tareas, que cuentan con un tiempo de realización y entrega límite, este tiempo está especificado en cada una de las tareas del Sprint (ver Tabla 41).

Tabla 41 *Cuadro de Sprint Backlog del Sistema EcoFavBen*

Ítem De Product Backlog	Enunciado Del Ítem De Product Backlog	Tarea	Dueño / Voluntario	Estatus	Horas Estimadas Totales
1. Diseño e implementación del circuito sensorial de monóxido de carbono (CO)	Como usuario, capturar los datos captados por el sensor	Diseño e implementar el circuito en Arduino (Arduino Uno, sensor MQ-9 Y Modulo Bluetooth)	Noe	Completado	10
		Codificar las funciones en el IDE de Arduino y cargar el código en el circuito de Arduino	Erick	Completado	10
	Como un usuario, procesar los datos obtenidos por el sensor	Calibrar el sensor	Noe	Completado	5
		Comprobar funcionalidad del circuito Arduino con el monitor del puerto serial	Erick	Completado	10
2. Desarrollar app EcoFavBen en	Como un usuario, ingresar al app	Diseñar los prototipos de la App	Noe	Completado	15
		Implementar la funcionalidades de los prototipos	Erick	Completado	25
3. Diseño e implementación del Web API	Como un usuario, mantener un historial de datos	Diseñar de la API	Noe	Completado	15
		Implementar funcionalidades a la API	Erick	Completado	25

4.	Comunicación API con App	Como un usuario, visualizar los datos de niveles de contaminación de monóxido de carbono (CO) en la app	Diseñar interfaz de comunicación del App con el API	Noe	Completado	15
5.	Clasificación de zonas de acuerdo a los niveles de contaminación.	Como un usuario ver los niveles de contaminación de monóxido de carbono (CO) por zonas	Obtener la data de los niveles de contaminación captadas por la app a través del sensor	Erick	Completado	25
6.	Elección e implementación del algoritmo de predicción	Como un usuario, ver la predicción de niveles de contaminación d CO en Cajamarca en el tiempo	Elección del algoritmo de predicción	Noe	Completado	15
			Implementación del algoritmo en el sistema web y en la App	Erick	Completado	25

ANEXO N° 8. Matriz de Consistencia

En la siguiente tabla se muestra la matriz de consistencia de la investigación (ver Tabla 42).

Tabla 42 *Matriz de consistencia*

PROBLEMA	OBJETIVO	HIPOTESIS	VARIABLES	DISEÑO	POBLACION Y MUESTRA
¿Cuál es el impacto del sistema EcoFavBen en la predicción de contaminación por CO en la ciudad de Cajamarca.	Determinar el impacto del sistema EcoFavBen en la predicción de contaminación por CO en la ciudad de Cajamarca.	HIPÓTESIS GENERAL El sistema EcoFavBen tiene un impacto positivo en la predicción de contaminación por CO en la	VARIABLE 1 Predicción de contaminación por CO en la ciudad de Cajamarca	El presente trabajo de investigación es de tipo Aplicativo - Experimental	para esta investigación la población es la ciudad de Cajamarca.

ión por CO	ciudad de	Cajamarca.	HIPÓTESIS	VARIABLE 2	MUESTRA
en la	OBJETIVOS ESPECIFICOS	<p>Diseñar e implementar sistema EcoFavBen para el registro y procesamiento de datos de niveles de monóxido de carbono en la ciudad de Cajamarca.</p> <p>Identificar y clasificar las zonas con mayores niveles de monóxido de carbono en la ciudad de Cajamarca.</p> <p>Realizar predicción de niveles de monóxido de carbono en la ciudad de Cajamarca, en función de los datos obtenidos por el sistema EcoFavBen.</p>	ESPECIFICAS	<p>Sistema EcoFavBen</p>	<p>el tipo de muestra que se utilizará en esta investigación será el muestro no probabilístico por conveniencia, para lo cual se tomó 12 zonas.</p>

ANEXO N° 9. Operacionalización de Variables

En la siguiente tabla se muestra la Operacionalización de variables de la investigación (ver Tabla 43).

Tabla 43 *Operacionalización de variables*

VARIABLE	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES	INSTRUMENTOS
Sistema				
EcoFaBen (Variable Independiente)	Es un sistema en el que el usuario puede acceder a la información desde cualquier lugar, independientemente del tipo de dispositivo que utilice para ello.	Ergonomía Usabilidad	Facilidad Aprendizaje Entendibilidad	de

Comprensibilidad

Satisfacción

Predicción de contaminación por CO en la ciudad de Cajamarca (Variable Dependiente)

Predicción es una expresión que anticipa aquello que, supuestamente, va a suceder. Se puede predecir algo a partir de datos previos. (Pérez, 2020)

Contaminación por CO

Nivel de CO

Ficha de Registro

Predicción

Porcentaje de error

Formulas estadísticas

Pronostico

ANEXO N° 10. Validaciones del Instrumento



FICHA PARA VALIDACION DEL INSTRUMENTO

I. REFERENCIA

- 1.1. Experto: *T. Villume, Necha, Yari Alexis*
- 1.2. Especialidad: *Ing. de Sistemas*
- 1.3. Cargo actual: *Docente T.P. U.P.N.C.*
- 1.4. Grado académico: *Doctor*
- 1.5. Institución: *Universidad Privada del Norte - Cajamarca*
- 1.6. Tipo de instrumento: *Ficha de Registro de Datos*
- 1.7. Lugar y fecha: *Cajamarca, 13 de febrero del 2020*

II. TABLA DE VALORACION POR EVIDENCIAS

N°	EVIDENCIAS	VALORACION					
		5	4	3	2	1	0
1	Pertinencia de indicadores	X					
2	Formulado con lenguaje apropiado	X					
3	Adecuado para los sujetos en estudio		X				
4	Facilita la prueba de hipótesis	X					
5	Suficiencia para medir la variable	X					
6	Facilita la interpretación del instrumento		X				
7	Acorde al avance de la ciencia y tecnología		X				
8	Expresado en hechos perceptibles	X					
9	Tiene secuencia lógica		X				
10	Basado en aspectos teóricos	X					
	Total		30	16			

Coefficiente de valoración porcentual: $c = 92\%$

III. OBSERVACIONES Y/O RECOMENDACIONES

Añadir datos estadísticos al final del instrumento (Nicho, medición, etc.)

.....

.....

.....

[Firma manuscrita]

Firma y sello del Experto

Yuri Alexis Villume Mechi
 INGENIERO DE SISTEMAS
 REG. CIP. N° 91908



UNIVERSIDAD PRIVADA DEL NORTE

FICHA PARA VALIDACION DEL INSTRUMENTO

I. REFERENCIA

- 1.1. Experto: Luis Miguel Cotrina Malca
- 1.2. Especialidad: Ingeniero de sistemas
- 1.3. Cargo actual: Gerente Datos Tecnología y Comunicación SRI
- 1.4. Grado académico: Maestro en Project Management
- 1.5. Institución: University of Maryland, USA
- 1.6. Tipo de instrumento: Ficha de Registro
- 1.7. Lugar y fecha: Cajamarca, 13 de febrero 2020

II. TABLA DE VALORACION POR EVIDENCIAS

N°	EVIDENCIAS	VALORACION					
		5	4	3	2	1	0
1	Pertinencia de indicadores	✓					
2	Formulado con lenguaje apropiado		✓				
3	Adecuado para los sujetos en estudio	✓					
4	Facilita la prueba de hipótesis	✓					
5	Suficiencia para medir la variable	✓					
6	Facilita la interpretación del instrumento	✓					
7	Acorde al avance de la ciencia y tecnología	✓					
8	Expresado en hechos perceptibles	✓					
9	Tiene secuencia lógica	✓					
10	Basado en aspectos teóricos	✓					
	Total	40	8				

Coefficiente de valoración porcentual: c = 96%

III. OBSERVACIONES Y/O RECOMENDACIONES

Se recomienda actualizar el título de la columna "Lectura"

.....

.....

.....

.....
Firma y sello del Experto

FICHA PARA VALIDACION DEL INSTRUMENTO

I. REFERENCIA

- 1.1. Experto: Alejandro Hugo Pérez Quinz
 1.2. Especialidad: Analista de Sistemas
 1.3. Cargo actual: Desarrollador de Aplicaciones
 1.4. Grado académico: Ingeniero de Sistemas
 1.5. Institución: Universidad Nacional de Cajamarca
 1.6. Tipo de instrumento: Ficha de Observación
 1.7. Lugar y fecha: S/13/02/2020

II. TABLA DE VALORACION POR EVIDENCIAS

N°	EVIDENCIAS	VALORACION					
		5	4	3	2	1	0
1	Pertinencia de indicadores		X				
2	Formulado con lenguaje apropiado		X				
3	Adecuado para los sujetos en estudio	X					
4	Facilita la prueba de hipótesis		X				
5	Suficiencia para medir la variable		X				
6	Facilita la interpretación del instrumento	X					
7	Acorde al avance de la ciencia y tecnología		X				
8	Expresado en hechos perceptibles	X					
9	Tiene secuencia lógica	X					
10	Basado en aspectos teóricos		X				
	Total	20	24				

Coefficiente de valoración porcentual: $c = \dots 88\%$

III. OBSERVACIONES Y/O RECOMENDACIONES

.....



 Firma y sello del Experto