

**DISEÑO Y DESARROLLO DE UNA APLICACIÓN PARA  
MONITOREAR LA CONCENTRACIÓN DE  $CO$  Y  $CH_4$  EN  
DISPOSITIVOS MÓVILES ANDROID.**

**JONATHAN MAURICIO NOVA BECERRA**



**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERIA  
FACULTAD DE INGENIERIA ELECTRONICA  
BUCARAMANGA, COLOMBIA  
2018**

**DISEÑO Y DESARROLLO DE UNA APLICACIÓN PARA  
MONITOREAR LA CONCENTRACIÓN DE  $CO$  Y  $CH_4$  EN  
DISPOSITIVOS MÓVILES ANDROID.**

**JONATHAN MAURICIO NOVA BECERRA**

**Trabajo de grado presentado para optar al título de:  
Ingeniero Electrónico**

**Esp. Juan Carlos Mantilla Saavedra  
Director**

**Esp. Cynthia Katherine Vega Pérez  
Co-Director Corporación CDT de Gas**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA  
ESCUELA DE INGENIERIAS  
FACULTAD DE INGENIERIA ELECTRONICA  
BUCARAMANGA, COLOMBIA  
2018**

**Nota de Aceptación:**

---

---

---

---

---

---

---

---

**Firma del Presidente del Jurado**

---

**Firma del Jurado**

---

**Firma del Jurado**

**Bucaramanga, 17 de Abril 2018**

## **AGRADECIMIENTOS**

A Dios por las bendiciones y oportunidades que me ha brindado en mi proceso de formación como ser humano y profesional.

A mi madre Elsa Becerra Calderón una mujer excepcional de quien aprendí el valor de la disciplina y la perseverancia, su apoyo y amor incondicional me ha permitido cumplir una de las metas más importantes en mi vida.

A mi nonita Ana María Calderón viuda de Becerra que siempre fue un apoyo desde mis primeros años hasta su fallecimiento Q.E.P.D.

A mi novia Andrea Marcela Quintero Carrillo, por creer siempre en mí, su compañía, comprensión, cariño y amor en los momentos difíciles me dieron la fuerza para superar las adversidades y hoy realizarme como profesional.

A mi amigo de infancia Felipe Aguirre por brindarme su amistad y asesoría para la resolución de dudas e inquietudes acerca de la realización de este proyecto.

A la corporación CDT de GAS y especialmente a mi supervisor Cynthia Vega, por confiar en mis habilidades y permitirme mediante el desarrollo de la práctica llevar al máximo mi potencial como Ingeniero.

A la Universidad Pontificia Bolivariana por abrirme sus puertas para formarme profesionalmente, a mis amigos y compañeros de estudio que me acompañaron en este proceso y a mis profesores quienes con su orientación y motivación fueron piezas fundamentales para hacer de este sueño una realidad.

# CONTENIDO

1. INTRODUCCION.....	12
2. CARACTERISTICAS GENERALES.....	13
2.1. PLANTEAMIENTO DEL PROBLEMA.....	13
2.2. JUSTIFICACION DEL PROYECTO .....	13
2.3. OBJETIVOS.....	14
2.3.1 Objetivo General.....	14
2.3.2 Objetivos Específicos. ....	14
2.4. DESCRIPCION DE LA EMPRESA.....	14
2.4.1 Reseña Histórica .....	14
2.4.2 Misión.....	15
2.4.3 Visión .....	15
2.4.4 Actividad Económica, Productos y Servicios .....	15
3. MARCO TEORICO.....	17
3.1. ENTORNO DE DESARROLLO (SOFTWARE) .....	17
3.1.1 Descripción de los principales entornos de Desarrollo para Android..	18
3.1.2 Selección del Software.....	21
3.1.3 Componentes de Android Estudio Utilizados .....	23
3.2. MARCO CONCEPTUAL APLICACIÓN .....	25
3.3. SISTEMA DE DETENCION DE GASES.....	26
3.3.1 Principio de medición de los sensores electroquímicos.....	26
3.3.2 Principio de Detección Infrarroja.....	27
3.3.3 Principio de Detección Catalítica .....	28
3.4. SENSORES IMPLEMENTADOS.....	29
3.4.1 Sensor de Metano.....	29
3.4.2 Sensor de Monóxido .....	30
3.5. PLACA DE DESARROLLO .....	32
3.6. PROTOCOLO DE COMUNICACION .....	33
3.6.1 Módulo HC-06 .....	35
4. ESTADO DEL ARTE.....	36
4.1. ANTECEDENTES .....	36
4.1.1 Sense+.....	36

4.1.2	Nest Protect .....	36
4.1.3	Los sensores de Gas presentes en los Smartphone's .....	39
5.	DESARROLLO PRÁCTICA EMPRESARIAL.....	40
5.1.	Requerimientos práctica empresarial .....	40
5.2.	METODOLOGIA .....	42
5.3.	PLAN DE TRABAJO.....	43
5.3.1	Selección de los Sensores.....	43
5.3.2	Desarrollo para la adecuación del sensor .....	44
5.3.3	Área de trabajo Laboratorio de calidad de gas .....	46
5.3.4	Caracterización del sensor .....	52
5.3.5	Implementación del protocolo Bluetooth en la Aplicación.....	68
6.	INSTRUCCIONES PARA UTILIZAR LA APLICACIÓN.....	73
	CONCLUSIONES.....	78
	OBSERVACIONES Y/O RECOMENDACIONES.....	79
	REFERENCIAS BIBLIOGRAFICAS.....	80
	ANEXOS .....	82
	ANEXO A CODIGO DE PROGRAMACION DE LA APLICACIÓN EN ANDROID STUDIO .....	82
	ANEXO B CODIGO DE PROGRAMACION ARDUINO .....	82

## LISTA DE FIGURAS

Figura 1.Ventana principal de Android Studio.....	21
Figura 2.Principio de Medición Electroquímica .....	27
Figura 3.Principio de Medición Infrarroja.....	27
Figura 4.Sensor basado en el principio de la oxidación catalítica .....	28
Figura 5.Esquema Puente de Wheastone .....	29
Figura 6.Sensor de gases inflamables MC113C .....	30
Figura 7.Sensor MQ-7 .....	31
Figura 8.Conexión sensor al Arduino.....	32
Figura 9.Arduino Pro mini .....	32
Figura 10.Conexión Modulo Bluetooth al Arduino .....	35
Figura 11.App Sense+ .....	36
Figura 12.Dimensiones Nest Protect.....	37
Figura 13.Cambridge CMOS Sensors.....	39
Figura 14.Requerimiento pantalla inicial App .....	40
Figura 15.Requerimiento Pantalla secundaria App .....	41
Figura 16.Requerimiento página de ingreso de coeficientes de calibración .....	41
Figura 17. Metodología. ....	42
Figura 18.Plan de Trabajo tareas a realizar .....	43
Figura 19.Diagrama de Bloques adecuación del sensor .....	44
Figura 20.Circuito Puente de Wheatstone recomendado por el fabricante.....	44
Figura 21.Circuito para operación a bajos voltajes recomendado .....	45
Figura 22.Espacio de Trabajo.....	46
Figura 23.Dilutor de Gas TGD2000 .....	46
Figura 24.Sistema adquisición de datos Dilutor de Gas TGD2000.....	47
Figura 25.Capilares para control de caudal de gas .....	47
Figura 26.Cilindro de gas $CH_4$ Metano o gas natural .....	48
Figura 27.Regulador de presión.....	48
Figura 28.Cámara de gas para caracterizar el sensor .....	49
Figura 29.Sensor MC113 marca Winsen utilizado para $CH_4$ .....	49
Figura 30. Sensor MQ7 para Arduino .....	50
Figura 31.Circuito Implementado para realizar las pruebas .....	50
Figura 32.Multímetro a la derecha lectura de la diferencia del puente de Wheatstone y a la izquierda lectura después del amplificador de instrumentación.....	51
Figura 33.Osciloscopio para realizar seguimiento a la señal.....	51
Figura 34.Utilizando un detector calibrado para corroborar el %mol de concentración de gas.....	52
Figura 35. Almacenamiento de datos en Microsoft Excel.....	54
Figura 36.Prueba 1 Sensor MC113 .....	55
Figura 37. Prueba 2 Sensor MC113.....	57
Figura 38. Prueba 3 Sensor MC113.....	59
Figura 39. Curva de la ecuación característica para el sensor MC113 de Metano .....	59
Figura 40. Prueba MQ7 .....	62
Figura 41. Valor de Estabilización MQ7 .....	62

Figura 42. Ecuación característica MQ7 .....	63
Figura 43. Datos leídos por el Arduino visualizados en el monitor serie .....	63
Figura 44. Monitor Arduino y Android Studio.....	64
Figura 45.Registro de Android Studio con la aplicación en ejecución .....	64
Figura 46.Concentración ascendente y Concentración descendente.....	65
Figura 47.Descripción 1 Fragments, Handlers, Helpers .....	66
Figura 48.Descripción 2 Adapter.....	66
Figura 49.Descripción 3 Activity, Fragment.....	67
Figura 50.Descripción 4 Main Activity .....	67
Figura 51.Implementación de protocolo Bluetooth 1 .....	68
Figura 52.Implementación de protocolo Bluetooth 2 .....	69
Figura 53.Implementación de protocolo Bluetooth 3 .....	70
Figura 54.Implementación de protocolo Bluetooth 4 .....	70
Figura 55.Programación grafica para la pantalla de “Log In” o Ingreso de Usuario .....	71
Figura 56.Programación en lenguaje JAVA para la funcionalidad de la App .....	71
Figura 57.Diseño para la actividad principal de visualización de concentración de gas ...	72
Figura 58.Puntos Fijados para la gráfica debido a que a falta de implementación del sensor y el modulo Bluetooth en esta fase inicial.....	72
Figura 59.Permiso para acceder a Bluetooth .....	73
Figura 60.Pantalla de ingreso o bienvenida .....	74
Figura 61.Acceso Administrador creación usuarios .....	74
Figura 62.Añadir Sensor nuevo o modificar .....	75
Figura 63.Sensores almacenados activos.....	75
Figura 64.Selección del dispositivo Bluetooth .....	76
Figura 65.Coeficientes de Calibración .....	77
Figura 66.Concentración de Gas en el tiempo en estado estable .....	77



## LISTA DE TABLAS

Tabla 1.Comparación entre Android Studio y ADT.....	22
Tabla 2.Diferencias entre Wifi y Bluetooth Low Energy.....	34
Tabla 3.Presiones según porcentaje de concentración.....	53
Tabla 4. Prueba 1 MC113.....	55
Tabla 5.Prueba 2 MC113.....	57
Tabla 6.Prueba 3 MC113.....	58
Tabla 7.Valores promedio de las 3 pruebas para obtener la ecuación.....	59
Tabla 8 Prueba MQ7 .....	61
Tabla 9 Promedio de las pruebas .....	62

**RESUMEN GENERAL DE TRABAJO DE GRADO**

**TITULO:** DISEÑO Y DESARROLLO DE UNA APLICACIÓN PARA MONITOREAR LA CONCENTRACIÓN DE CO Y CH4 EN DISPOSITIVOS MÓVILES ANDROID.

**AUTOR(ES):** JONATHAN MAURICIO NOVA BECERRA

**PROGRAMA:** Facultad de Ingeniería Electrónica

**DIRECTOR(A):** JUAN CARLOS MANTILLA SAAVEDRA

**RESUMEN**

En la Corporación CDT de GAS se planteó como proyecto interno el desarrollo de un dispositivo portable para la detección de gas CO y CH4 en el ambiente, como método de alerta temprana para evitar posibles situaciones de riesgo. Por ello se diseñó un circuito de adecuación que envía la señal de salida del sensor a una aplicación sobre el sistema operativo Android la cual, realiza el monitoreo constante y guarda los datos para su posterior análisis. El hardware utilizado está compuesto por la tarjeta Arduino Pro Mini, un módulo de comunicación Bluetooth HC06 y un Smartphone con sistema operativo Android. A nivel de software se utilizaron los entornos de desarrollo Android Studio y Arduino. Para agregar un sensor primero se debe obtener la ecuación característica del mismo y esta se implementa sobre la aplicación, posteriormente, se establece la comunicación para transmitir la señal entregada por el sensor y se grafica con frecuencia de actualización de 1 segundo en el Smartphone. Se comprueba que el prototipo del dispositivo es funcional brindando lecturas parecidas a los detectores comerciales, con bajo costo de producción y compatible con dispositivos Android desde versiones de Android 4.4 hasta la última versión actual 8.

**PALABRAS CLAVE:**

Android Studio, Arduino, Bluetooth, Detector gas, Dispositivo portable, HC06, Smartphone.

V° B° DIRECTOR DE TRABAJO DE GRADO

**GENERAL SUMMARY OF WORK OF GRADE**

**TITLE:** DESIGN AND DEVELOPMENT OF AN APPLICATION TO MONITOR THE CONCENTRATION OF CO AND CH<sub>4</sub> IN ANDROID MOBILE DEVICES.

**AUTHOR(S):** JONATHAN MAURICIO NOVA BECERRA

**FACULTY:** Facultad de Ingeniería Electrónica

**DIRECTOR:** JUAN CARLOS MANTILLA SAAVEDRA

**ABSTRACT**

In the CDT Corporation of GAS, the development of a portable device for the detection of CO and CH<sub>4</sub> gas in the environment was proposed as an internal project, as an early warning method to avoid possible risk situations. For this reason, an adaptation circuit was designed that sends the output signal of the sensor to an application on the Android operating system which performs constant monitoring and saves the data for later analysis. The hardware used is composed of the Arduino Pro Mini card, a Bluetooth HC06 communication module and a smartphone with Android operating system. At the software level, the Android Studio and Arduino development environments were used. To add a sensor first you must obtain the characteristic equation of it and this is implemented on the application, subsequently, the communication is established to transmit the signal delivered by the sensor and is graphed with a refresh rate of 1 second on the Smartphone. It is verified that the prototype of the device is functional, providing readings similar to commercial detectors, with low production costs and compatible with Android devices from Android 4.4 versions to the latest version 8.

**KEYWORDS:**

Android Studio, Arduino, Bluetooth, Gas Detector, Portable Device, HC06, Smartphone.

V° B° DIRECTOR OF GRADUATE WORK

# 1. INTRODUCCION

La Corporación Centro de Desarrollo Tecnológico del Gas (CDT de GAS) con el fin de dar cumplimiento a su compromiso de ofrecer a la industria servicios tecnológicos de alta calidad busca estar continuamente optimizando los procesos que garantizan la competitividad del sector gas y otros sectores productivos, implementando y desarrollando a lo largo de su extensa trayectoria la infraestructura, procedimientos y competencias necesarias para mejorar las condiciones de seguridad industrial y salud ocupacional de sus clientes.

Como líder en investigación y desarrollo tecnológico se ha propuesto incorporar en su portafolio de servicios procesos que sean accesibles a cualquier tipo de usuario.

Es por ello que el CDT de Gas busca con la elaboración de este proyecto crear un dispositivo que permita realizar el monitoreo constante de concentración de CO y CH<sub>4</sub> en entornos domésticos, y ofrecer con ello un sistema de seguridad confiable y de fácil implementación, que para su funcionamiento solo requerirá de un Smartphone con Sistema Operativo Android y conexión por medio de protocolo Bluetooth.

## **2. CARACTERISTICAS GENERALES**

### **2.1. PLANTEAMIENTO DEL PROBLEMA**

Con el fin de dar cumplimiento a los parámetros de calidad del servicio de Gas Natural, las empresas prestadoras de este servicio deben contar con protocolos de actuación que permitan verificar que los niveles de concentración de CO y CH<sub>4</sub> en espacios con poca ventilación como son los hogares, se encuentren dentro de los estándares legales establecidos, ya que la acumulación y exposición a gases tóxicos como el monóxido de carbono y el metano, se considera de alto riesgo y puede llegar a ser mortal.

Teniendo en cuenta la necesidad de ofrecer un sistema de Seguridad adecuado a las necesidades de los usuarios que se encuentran en los entornos residenciales, el presente proyecto propone el desarrollo e implementación de una App Android asociada a un detector de CO y CH<sub>4</sub> que permita al usuario visualizar la concentración de los gases mencionados y con ello evitar situaciones de riesgo que puedan desencadenar pérdidas materiales y vidas humanas.

### **2.2. JUSTIFICACION DEL PROYECTO**

La Corporación Centro de Desarrollo Tecnológico del Gas (CDT de GAS) con el fin de dar cumplimiento a su compromiso de ofrecer a la industria servicios tecnológicos de alta calidad busca estar continuamente optimizando los procesos que garantizan la competitividad del sector gas y otros sectores productivos, implementando y desarrollando a lo largo de su extensa trayectoria la infraestructura, procedimientos y competencias necesarias para mejorar las condiciones de seguridad industrial y salud ocupacional de sus clientes.

Es por ello que el CDT de Gas en alianza con la Universidad Pontificia Bolivariana Seccional Bucaramanga busca con la elaboración de este proyecto crear un dispositivo que permita realizar el monitoreo constante de concentración de CO y CH<sub>4</sub> en entornos domésticos, y ofrecer con ello un sistema de seguridad confiable y de fácil implementación que para su funcionamiento solo requerirá de un Smartphone con Sistema Operativo Android y conexión Bluetooth.

## **2.3. OBJETIVOS**

### **2.3.1 Objetivo General**

- Diseñar y desarrollar una aplicación para establecer comunicación entre un sistema de sensado de concentración de CO y CH<sub>4</sub> y un dispositivo móvil basado en el sistema operativo Android.

### **2.3.2 Objetivos Específicos.**

- Diseñar la aplicación para móviles Android según requerimientos.
- Desarrollar y programar la App en la plataforma Android Studio.
- Efectuar pruebas de funcionamiento en la App para comparar y verificar su comportamiento con respecto a la comunicación serial de la plataforma Arduino.

## **2.4. DESCRIPCION DE LA EMPRESA**

### **2.4.1 Reseña Histórica**

[1]La Corporación CDT de GAS fue creada el 19 de diciembre de 1999 en la ciudad de Bucaramanga como una entidad de derecho privado y participación mixta, sin ánimo de lucro, de carácter Científico y Tecnológico, sin ánimo de lucro, con patrimonio propio dentro del marco legal de la Ley 29 de febrero de 1990, Ley Marco de Ciencia y Tecnología.

El CDT de GAS ha experimentado dos etapas claves :la primera (2000-2002)como entidad de carácter virtual en la cual se destaca el resultado del estudio prospectivo del sector gas (2002-2015),realizado con recursos de COLCIENCIAS–SENA; la segunda etapa (2003 hasta nuestros días )ha estado caracterizada por decisiones estratégicas tal como el aprovechamiento de los recursos competidos provistos por COLCIENCIAS –SENA para consolidar la Infraestructura Metrológica con el fin de brindar trazabilidad a las mediciones de gases en Colombia.

## **2.4.2 Misión**

[2] En los laboratorios de calibración y ensayos de la Corporación Centro de Desarrollo Tecnológico del Gas, contribuimos al mejoramiento de la calidad de los procesos, productos y servicios en beneficio de las empresas del sector gas y de la seguridad social en general; realizamos servicios de ingeniería y formación de personal externo a través de jornadas técnicas de intercambio nacional e internacional, fundamentados en nuestro liderazgo tecnológico y en la permanente evaluación de cada una de nuestras actividades manteniendo criterios de competitividad técnica y principios éticos.

## **2.4.3 Visión**

[2]Ser el laboratorio líder en investigación y desarrollo tecnológico a nivel nacional, y reconocido en el entorno internacional, por el proceso continuo de aplicación de las ventajas que ofrece la metrología de los fluidos para el desarrollo industrial, mediante la prestación de servicios especializados de calibración y ensayos, acreditados por Organismos Reconocidos, cumpliendo con los requisitos legales aplicables y buscando satisfacer las necesidades del sector con enfoque práctico hacia la sociedad.

## **2.4.4 Actividad Económica, Productos y Servicios**

[2]Desde su creación La Corporación CDT de GAS , se ha mantenido como un ente independiente, para brindar a sus clientes resultados libres de influencias económicas o comerciales, garantizando así, la transparencia de las soluciones entregadas, manteniendo el liderazgo tecnológico en el sector gas, aplicando conceptos inherentes a la metrología de fluidos, mediante la utilización de la investigación aplicada y la constante formación de su talento humano, destacándose por la eficacia y eficiencia de sus soluciones, contribuyendo así, a la apropiación de los avances de la ciencia y la tecnología que hacen posible el crecimiento sostenido del sector gas y de la industria colombiana en general.

### Sectores de Influencia

El principal foco de acción ha sido el sector gas; sin embargo, los desarrollos y competencias institucionales se han puesto al servicio del país, cubriendo hoy día sectores tales como:

- Producción, refinación, procesamiento, transporte y distribución de gas e hidrocarburos.

- Grandes consumidores de gas natural: termoeléctricas y petroquímicas.
- Usuarios industriales (alimentos, metalmecánica) y usuarios comerciales del gas.
- Sector vehicular (EDS-GNV, talleres de conversión, etc.).
- Sector del gas licuado del petróleo.
- Entes Reguladores y de Control.
- Entes del sistema nacional de calidad: Acreditadores, Normalizadores, Certificadores, y Organismos de Inspección.
- Laboratorios de investigación, calidad, calibración y ensayos.
- Fabricantes y proveedores de insumos, equipos y servicios para el sector gas y la industria en general.
- Universidades y sector académico.

Entre los Servicios que ofrece la Corporación CDT de GAS se destacan:

- Metrología Física: Servicios acreditados por ONAC: Calibración Volumen y Flujo de Gases, Volumen y Caudal de Líquidos, Calibración Presión, Temperatura, Señales Eléctricas, Velocidad de Gases, Otros servicios: Calibración Analizadores de Gases de Chimenea (isocinéticos), Calibración Masas y Balanzas
- Metrología Química: Servicios acreditados por ONAC: Composición de Gas Natural por Cromatografía, Calibración de Medidores de CO y  $CH_4$  para inspecciones Residenciales. Otros servicios: Muestreo de Gases Combustibles, Calibración de Medidores de Humedad, Medición de Contenido de Humedad en Gas Natural, Medición de Contenido de  $H_2S$  en
- Laboratorio de ensayos: pruebas hidrostáticas y neumáticas a tuberías, equipos y accesorios de sistemas de medición.
- Soluciones Tecnológicas: Comprometidos con el incremento de la competitividad del sector gas y otros servicios productivos ofrece suministro e integración de sistemas de medición de flujo de fluidos, Bancos de Calibración para uso en laboratorios o in situ, Consultorías en Re-ingeniería de Sistemas de Medición, Diseño e Implementación de programas de aseguramiento metrológico para sistemas de Medición, Asistencia Técnica (Pre-commissioning, Commissioning, Start-Up), Verificación y estimación de incertidumbre en mediciones estática y dinámica.
- Desarrollo de Software: MDM (Metrological Data Management), Gas Balance Manager (Balance en Redes de Gas), FluxPro (Propiedades del Gas Natural), StoFlow (Herramienta Estadística como apoyo en la optimización de Sistemas de Medición) SIMED (Sistema de Información para inventario de Sistemas de Medición).
- Responsabilidad ambiental: Detección de Emisiones Fugitivas, Emisiones de Gases de Efecto Invernadero, Energía y Optimización de procesos.



## 3. MARCO TEORICO

### 3.1. ENTORNO DE DESARROLLO (SOFTWARE)

[3]Un entorno de desarrollo integrado (IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

Algunos ejemplos de entornos integrados de desarrollo (IDE) son los siguientes:

- Eclipse
- NetBeans
- IntelliJ IDEA
- JBuilder de Borland
- JDeveloper de Oracle
- KDevelop
- Anjuta
- Clarion
- MS Visual Studio
- Visual C++

Un IDE debe tener las siguientes características:

- Multiplataforma
- Soporte para diversos lenguajes de programación
- Integración con Sistemas de Control de Versiones
- Reconocimiento de Sintaxis
- Extensiones y Componentes para el IDE

- Integración con Framework populares
- Depurador
- Importar y Exportar proyectos
- Múltiples idiomas
- Manual de Usuarios y Ayuda

### **Ventajas de los IDEs.**

- La curva de aprendizaje es muy baja.
- Es más ágil y óptimo para los usuarios que no son expertos en manejo de consola.
- Formato de código.
- Funciones para renombrar variables.
- Warnings y errores de sintaxis en pantalla de algo que no va a funcionar al interpretar o compilar.
- Poder crear proyectos para poder visualizar los archivos de manera gráfica.
- Herramientas de refactoring como por ejemplo sería extraer una porción de código a un método nuevo.
- No es recomendado, pero posee un navegador web interno por si queremos probar las cosas dentro de la IDE.

### **3.1.1 Descripción de los principales entornos de Desarrollo para Android.**

#### 3.1.1.1 Eclipse

[4]Eclipse es una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-ins. Fue concebida desde sus orígenes para convertirse en una plataforma de integración de herramientas de desarrollo.

Principales características de Eclipse

- Perspectivas, editores y vistas: en Eclipse el concepto de trabajo está basado en las perspectivas, que no es otra cosa que una pre configuración de ventanas y editores, relacionada entre sí, y que nos permiten trabajar en un determinado entorno de trabajo de forma óptima.
- Gestión de proyectos: el desarrollo sobre Eclipse se basa en los proyectos, que son el conjunto de recursos relacionados entre sí, como puede ser el código fuente,

documentación, ficheros configuración, árbol de directorios. El IDE nos proporcionará asistentes y ayudas para la creación de proyectos. Por ejemplo, cuando creamos uno, se abre la perspectiva adecuada al tipo de proyecto que estemos creando, con la colección de vistas, editores y ventanas preconfigurada por defecto.

- Depurador de código: se incluye un potente depurador, de uso fácil e intuitivo, y que visualmente nos ayuda a mejorar nuestro código. Para ello sólo debemos ejecutar el programa en modo depuración (con un simple botón). De nuevo, tenemos una perspectiva específica para la depuración de código, la perspectiva depuración, donde se muestra de forma ordenada toda la información necesaria para realizar dicha tarea.
- Extensa colección de plug-ins: están disponibles en una gran cantidad, unos publicados por Eclipse, otros por terceros. Al haber sido un estándar de facto durante tanto tiempo (no el único estándar, pero sí uno de ellos), la colección disponible es muy grande. Los hay gratuitos, de pago, bajo distintas licencias, pero casi para cualquier cosa que nos imaginemos tenemos el plug-in adecuado.

[5] Android ofrece un complemento personalizado para Eclipse IDE, llamado Android Development Tools (ADT). Este complemento proporciona un entorno potente e integrado en el que desarrollar aplicaciones de Android. Extiende las capacidades de Eclipse para permitirle configurar rápidamente nuevos proyectos de Android, crear una interfaz de usuario de aplicación, depurar su aplicación y exportar paquetes de aplicaciones firmadas (o sin firmar) para su distribución.

### 3.1.1.2 Android Studio

[6] Android Studio es un entorno de desarrollo integrado (IDE), basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin ADT (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.

Fue presentado por Google el 16 de mayo del 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. Se trata pues de una alternativa real a Eclipse, el IDE recomendado por Google hasta la fecha, pero que presentaba problemas debido a su lentitud en el desarrollo de versiones que solucionaran las carencias actuales (es indispensable recordar que Eclipse es una plataforma de desarrollo, diseñada para ser extendida a través de plugins).

A partir de la versión estable de Android Studio 1.0, Google ha pasado a recomendarlo como el IDE para desarrollar aplicaciones para su sistema operativo, dejando el plugin ADT para Eclipse de estar en desarrollo activo.

Principales características que incluye Android Studio:

- Soporte para programar aplicaciones para Android Wear (sistema operativo para dispositivos corporales como por ejemplo un reloj).
- Herramientas Lint (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.
- Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con limitaciones de memoria interna).
- Integración de la herramienta Gradle encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado.
- Nuevo diseño del editor con soporte para la edición de temas.
- Nueva interfaz específica para el desarrollo en Android.
- Permite la importación de proyectos realizados en el entorno Eclipse, que a diferencia de Android Studio (Gradle) utiliza ANT.
- Posibilita el control de versiones accediendo a un repositorio desde el que se puede descargar versiones: Mercurial, Git, Github o Subversión.
  
- Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- Vista previa en diferentes dispositivos y resoluciones.
- Integración con Google Cloud Platform, para el acceso a los diferentes servicios que proporciona Google en la nube.
- Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo XML.

Interfaz de Usuario

[7]La ventana principal de Android Studio consta de varias áreas lógicas que se identifican en la figura 1.

1. La **barra de herramientas** le permite llevar a cabo una amplia gama de acciones, incluyendo el funcionamiento de su aplicación y el lanzamiento de herramientas de Android.
2. La **barra de navegación** le ayuda a navegar a través de su proyecto y los archivos abiertos para su edición. Proporciona una visión más compacta de la estructura visible en el **Proyecto** ventana.
3. La **ventana de edición** es donde puede crear y modificar el código. Dependiendo del tipo de archivo actual, el editor puede cambiar. Por ejemplo, al ver un archivo de diseño, el editor muestra el editor de diseño.
4. La **barra de la ventana de herramientas** corre alrededor de la parte exterior de la ventana IDE y contiene los botones que le permiten expandir o contraer las ventanas de herramientas individuales.

5. Las **ventanas de herramientas** dan acceso a tareas específicas como la gestión de proyectos, búsqueda, control de versiones, y mucho más. Puede expandirlos y contraerlos.
6. La **barra de estado** muestra el estado de su proyecto y el propio IDE, así como mensajes o avisos.

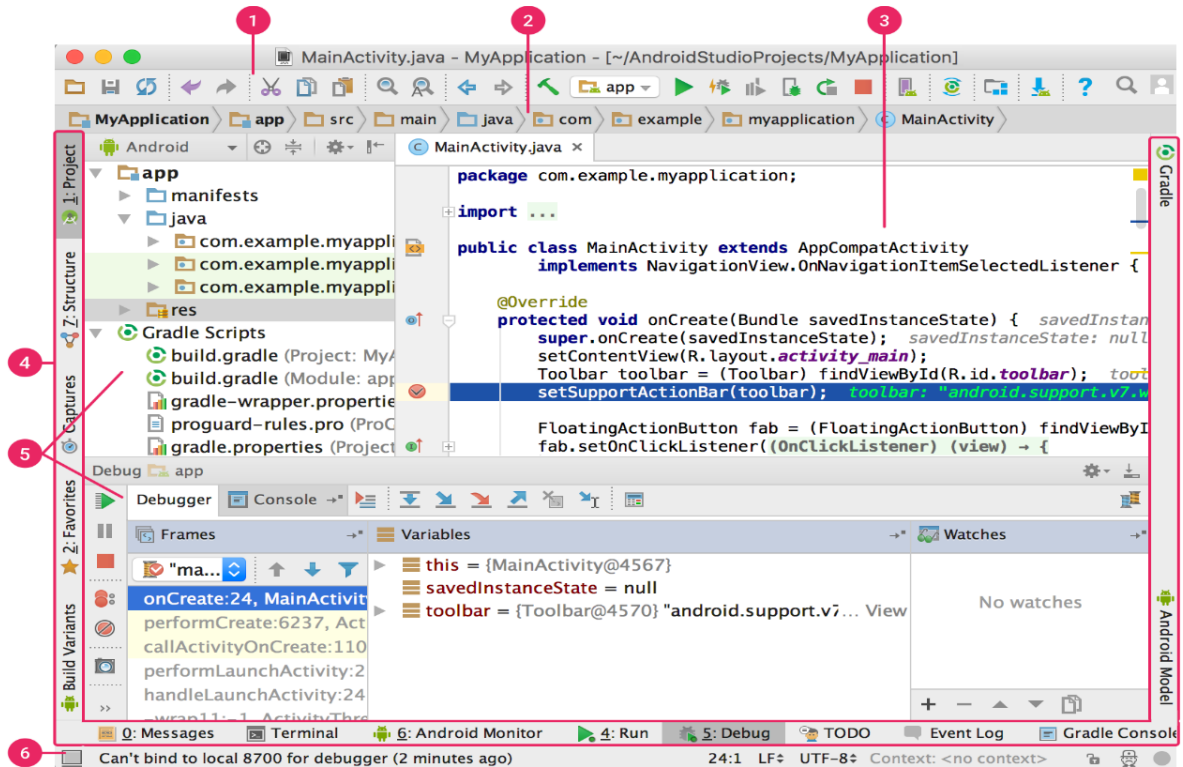


Figura 1. Ventana principal de Android Studio.<sup>1</sup>

### 3.1.2 Selección del Software

Para la selección del software fue necesario establecer las diferencias y novedades que presenta Android Studio con respecto al IDE Eclipse, y más concretamente con el ADT para Android, se propone la siguiente tabla comparativa entre ambas opciones Tabla 1.

<sup>1</sup> Fuente: <https://developer.android.com/studio/intro/index.html>

Características	A. Studio	ADT
Sistema de construcción	Gradle	ANT
Construcción y gestión de proyectos basado en Maven (Software para gestión y construcción de proyectos Java, similar a Apache ANT, más simple basado en XML)	Si	No (es necesario instalar un plugin auxiliar)
Construir variantes y generación de múltiples APK (útil para Android Wear)	Si	No
Refactorización y completado avanzado de código Android	Si	No
Diseño del editor gráfico	Si	Si
Firma APK, gestión de almacén de claves	Si	Si
Soporte para NDK (Native Development Kit: herramientas para implementar código nativo escrito en C y C++)	Próximas versiones	Si
Soporte para Google Cloud Platform	Si	No
Vista en tiempo real de layouts	Si	No
Nuevos módulos en proyecto	Si	No
Editor de navegación	Si	No
Generador de assets	Si	No
Datos de ejemplo en diseño de layout (sin renderizar en tiempo de ejecución)	Si	No
Visualización de recursos desde editor de código	Si (Línea de asignación)	No

Tabla 1. Comparación entre Android Studio y ADT.<sup>2</sup>

<sup>2</sup> Fuente: <https://developer.android.com/studio/intro/index.html>

## [6]Ventajas uso Android Studio

- Android Studio ha pasado a ser el entorno recomendado para el desarrollo de aplicaciones en Android, al tratarse de un IDE oficial de Google en colaboración con JetBrains (compañía de desarrollo software especializada en diseño de IDEs).
- Android Studio permite la creación de nuevos módulos dentro de un mismo proyecto, sin necesidad de estar cambiando de espacio de trabajo para el manejo de proyectos, algo habitual en Eclipse.
- Con la simple descarga de Android Studio se disponen de todas las herramientas necesarias para el desarrollo de aplicaciones para la plataforma Android.
- Su nueva forma de construir los paquetes apk, mediante el uso de Gradle, proporciona una serie de ventajas más acorde a un proyecto Java:
  - Facilita la distribución de código, y por lo tanto el trabajo en equipo.
  - Reutilización de código y recursos.
  - Permite compilar desde línea de comandos, para aquellas situaciones en las que no esté disponible un entorno de desarrollo.
- Mayor facilidad para la creación de diferentes versiones de la misma aplicación, que proporciona numerosas ventajas como puede ser la creación de una versión de pago y otra gratuita, o por ejemplo diferentes dispositivos o almacén de datos.

Teniendo en cuenta lo anterior se decidió seleccionar el entorno de desarrollo Android Studio, debido a que es el IDE oficial de google para el desarrollo de aplicaciones en Android, brinda facilidad de actualización para cambiar entre versiones de Android, cuenta con todas las herramientas necesarios para desarrollar sin necesidad de adicionar plugins, posee sincronización con la nube para que todo el equipo de trabajo pueda desarrollar sobre el mismo proyecto y es software libre, por tanto gratuito.

### 3.1.3 Componentes de Android Estudio Utilizados

#### 3.1.3.1 Lenguaje de Programación XML

XML (*eXtensible Markup Language*) o "Lenguaje de Marcado Extensible"

[8]XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML, pero su función principal es describir datos y no mostrarlos como es el caso

de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

### Diseño en XML para desarrollador Android

Un diseño define la estructura visual para una interfaz de usuario, como la IU para una actividad o widget de una app. Puedes declarar un diseño de dos maneras:

- Declarar elementos de la IU en XML. Android proporciona un vocabulario XML simple que coincide con las clases y subclases de vistas, como las que se usan para widgets y diseños.
- Crear una instancia de elementos del diseño en tiempo de ejecución. Tu aplicación puede crear objetos View y ViewGroup (y manipular sus propiedades) programáticamente.

El framework de Android te ofrece la flexibilidad de usar uno de estos métodos o ambos para declarar y administrar la IU de tu aplicación. Luego podrías agregar código en tu aplicación para modificar el estado de los objetos de la pantalla, incluidos los declarados en XML, en tiempo de ejecución. La ventaja de declarar tu IU en XML es que te permite separar mejor la presentación de tu aplicación del código que controla su comportamiento. Tus descripciones de la IU son externas al código de tu aplicación, lo que significa que puedes modificarlo o adaptarlo sin tener que modificar el código fuente y volver a compilar. Por ejemplo, puedes crear diseños XML para diferentes orientaciones de pantalla, diferentes tamaños de pantalla de dispositivos y diferentes idiomas. Además, declarar el diseño en XML facilita la visualización de la estructura de tu IU, de modo que sea más fácil depurar problemas.

#### 3.1.3.2 Componentes adicionales dentro de las actividades

##### Fragmentos (Fragments)

- Nacen a partir de la necesidad de usar múltiples tamaños de pantalla.
- Son secciones modulares para la navegación entre pantallas o actividades, se pueden definir varios fragmentos dentro de una misma actividad.
- Representan el comportamiento de una porción de la interfaz de usuario asociada a la actividad.



- Al pausar una actividad se pausarán todos los fragmentos definidos dentro de esta.
- Si se destruye una actividad, todos los fragmentos que contenga serán también destruidos.
- Si la actividad se encuentra en ejecución, es posible manipular de manera independiente cada fragmento, inclusive destruirlo.

#### Adaptadores (Adapters)

- Un adaptador es un objeto que comunica a elementos nativos de Android, por ejemplo, un ListView los datos necesarios para crear las filas de la lista. Es decir, conecta la lista con una fuente de información como si se tratase de un adaptador de corriente que alimenta a un televisor.
- Interacción ListView-Adapter: cuando se relaciona un adaptador a una lista, inmediatamente comienza un proceso de comunicación interno para poblarla con una fuente de datos.

#### Handlers

- Permiten poner en cola una acción que se realiza en un subproceso distinto al suyo.
- Comunicación del hilo secundario con el hilo principal a través de un controlador.
- Se encarga de la manipulación entre la lógica y los datos, por ejemplo, una clase intermedia para insertar valores en una base de datos.

#### Helpers

Ayudan para gestionar la base de datos, es decir:

- Crearla
- Modificarla
- Introducir datos

Son una clase propia que se debe personalizar para adaptarse a las necesidades concretas de la aplicación.

## 3.2. MARCO CONCEPTUAL APLICACIÓN

#### Android Standard Development Kit (SDK)

Es el núcleo de desarrollo de Android se trata de los componentes necesarios para la creación de una herramienta software. Dentro de ella están contenidos todos los elementos para comunicación software/hardware, control del dispositivo y los controles base de interacción con el usuario. Cada una de las versiones de este contienen un nivel de

características diferentes por lo cual entre más alta la versión se encuentran disponibles una gama más alta de herramientas a disposición del desarrollador. Dentro de lo utilizado para este desarrollo se encuentran las siguientes características:

- Biblioteca appcompat v7: Biblioteca que agrega las características de Material Design a todas las versiones Android tal y como el manejo de Fragmentos, Botones con Imágenes, y el control de Flujo de Datos Bluetooth.
- Biblioteca de compatibilidad Design: Esta se encuentra de habilitar el uso de los controles de Material Design para el ambiente de diseño de la aplicación.

Herramientas Colaborativas (GIT)

Software de control de versiones para el almacenamiento de las fuentes de la aplicación. Se encarga de mantener un control de versiones de cada uno de los archivos del proyecto permitiendo monitorear cada uno de los cambios realizados en los archivos del mismo.

### **3.3. SISTEMA DE DETENCION DE GASES**

[9]Los sistemas de detección de gases están dedicados a detectar concentraciones de gas peligrosas, para activar alarmas y hasta donde sea posible activar contramedidas antes de que se pueda producir una situación peligrosa para empleados, instalaciones y medioambiente.

Los equipos para detección de gases pueden ser portátiles (o semi-portátiles) o sistemas fijos de detección de gases. La seguridad de una zona potencialmente afectada por gases y vapores peligrosos depende principalmente de la fiabilidad del sistema de detección de gases, y especialmente de la calidad de los sensores utilizados.

Los sensores para la detección de gases y vapores son transductores que usan ciertas propiedades de los gases para la conversión en una señal eléctrica adecuada. Especialmente tres principios de medición se han hecho dominantes en las décadas recientes de la detección industrial de gases: sensores electroquímicos, sensores de perla catalítica y sensores infrarrojos.

#### **3.3.1 Principio de medición de los sensores electroquímicos.**

El sensor electroquímico es un micro-reactor, que con la presencia de gases reactivos produce electrones exactamente como una batería. El flujo de electrones es una corriente eléctrica muy baja, pero medible.

La célula electroquímica se compone de un electrodo de medición (1), de un contra electrodo (2) y de un electrodo de referencia (3). Estos electrodos están bañados en un

electrolito situado en el interior de la carcasa de la célula, provista a su vez de una membrana permeable a los gases (4). El gas que se difunde en el interior de la célula provoca una reacción química en el electrodo de medición y en el contra electrodo. De ello resulta una corriente eléctrica, proporcional a la concentración del gas presente, que circula entre estos dos electrodos. Esta corriente es medida por el circuito externo (5) al que está conectada la célula. El tercer electrodo sirve de referencia de medición estable.

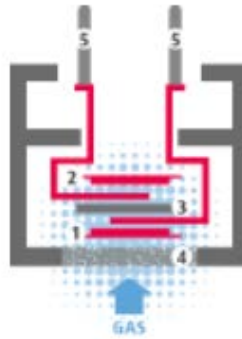


Figura 2.Principio de Medición Electroquímica<sup>3</sup>

### 3.3.2 Principio de Detección Infrarroja

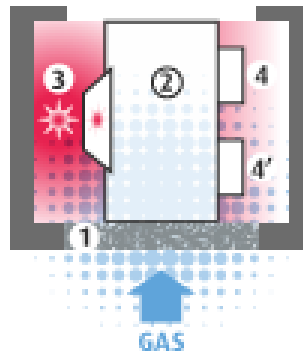


Figura 3.Principio de Medición Infrarroja<sup>4</sup>

La célula infrarroja funciona según el principio del infrarrojo no dispersivo (NDIR). Está constituida por una carcasa que comprende:

- una membrana de difusión (1),
- una cámara de medición (2),
- una fuente de radiación IR (3),
- un sensor activo (4) y
- un sensor de referencia (4').

<sup>3</sup> Fuente: <http://www.dalemans.com/es/introduction/>

<sup>4</sup> Fuente: <http://www.dalemans.com/es/introduction/>

El gas que penetra en la cámara de medición absorbe, en una gama de longitudes de onda muy precisa, una parte de la radiación emitida por la fuente de IR. El sensor activo mide la radiación IR restante y determina, de este modo, la concentración de gas presente. El sensor de referencia mide la radiación IR en una gama de longitudes de onda no influida por el gas entrante. Su señal sirve para compensar cualquier variación de la radiación IR que no se deba a la absorción causada por el gas diana, como una variación de la T, de la tasa de humedad, etc.

Esto permite obtener una medición precisa y fiable en todas las condiciones.

### 3.3.3 Principio de Detección Catalítica

[10]La detección requiere dos elementos:

- un filamento "detector"
- un filamento "compensador"

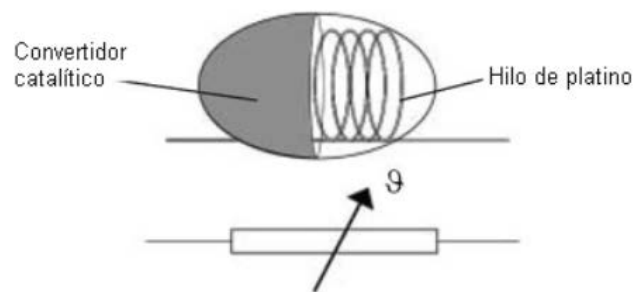


Figura 4. Sensor basado en el principio de la oxidación catalítica<sup>5</sup>

Se trata de 2 filamentos de Pt montados en espiral y recubiertos de alúmina para aumentar la superficie de catálisis. La alúmina del filamento compensador está tratada para evitar la oxidación catalítica. El filamento compensador simplemente evita la influencia de las variaciones en la humedad, en la temperatura ambiente y en la presión barométrica. Estos dos filamentos y sus soportes están acondicionados en una envolvente "antideflagrante".

Los dos filamentos se sitúan formando un Puente de WHEASTONE. Se alimentan mediante una corriente constante en el caso de los sensores fijos, y un voltaje constante para los sensores portátiles con el fin de mantener la temperatura de catálisis requerida de 450 °C. A esta temperatura, los gases inflamables son oxidados al entrar en contacto con el catalizador.

Cuando un gas inflamable se pone en contacto con la superficie caliente del elemento detector se produce una combustión. El calor generado aumenta la temperatura del

<sup>5</sup> Fuente: <http://www.academiatesto.com.ar/cms/conductividad-termica>

filamento de platino y varía su resistencia eléctrica. La medida de la variación de la resistencia eléctrica nos indica la concentración de gas inflamable presente en el ambiente.

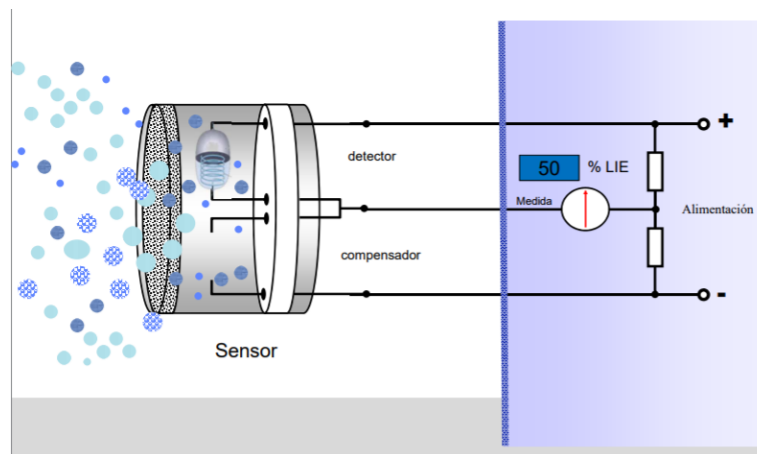


Figura 5. Esquema Puente de Wheatstone<sup>6</sup>

### 3.4. SENSORES IMPLEMENTADOS

#### 3.4.1 Sensor de Metano

- **MC113 Figura 6.**

*Sensor de gases inflamables MC113C / Sensor de gases C3H8 / Sensor de gas metano*

[11] El sensor MC113 / MC113C adopta el principio de combustión catalítica. Por otra parte, sus dos brazos de puente eléctrico se componen de un elemento de prueba y un elemento de compensación.

La resistencia del elemento de prueba aumenta una vez que alcanza los gases combustibles, al mismo tiempo, la tensión de salida del puente cambia y la variación de voltaje aumenta en proporción directa a la concentración de gas. El elemento de compensación, como referencia, tiene la función de compensar la temperatura y la humedad.

Aplicaciones:

Es ampliamente utilizado en ocasiones industriales para detectar la concentración de gas natural, GLP, CO y alcanos.

También se usa en el sistema de alarma de fuga de gas combustible, detector de gas combustible y medidor de concentración de gas, y así sucesivamente.

<sup>6</sup> Fuente: <https://www.ucm.es/data/cont/docs/3-2015-03-16-Principios%20de%20detecci%C3%B3n.pdf>



Figura 6. Sensor de gases inflamables MC113C <sup>7</sup>

### Especificaciones Técnicas

- Modelo MC113 / MC113C
- Tipo de sensor Catalítico
- Encapsulación estándar: Metal, metalurgia en polvo de malla
- Voltaje de trabajo (V):  $2.8 \pm 0.1$
- Corriente de trabajo (mA): MC113:  $90 \pm 10$ ; MC113C:  $95 \pm 10$
- Sensibilidad (mV): 1% de CH<sub>4</sub> MC113: 20 a 40; MC113C: 16 a 35
  - 1% C<sub>3</sub>H<sub>8</sub> MC113: 30 a 60; MC113C: de 25 a 55
- Linealidad:  $\leq 5\%$
- Rango de medición (% LEL): 0 a 100
- Tiempo de respuesta (90%):  $\leq 10s$
- Tiempo de recuperación (90%):  $\leq 30s$
- Ambiente de trabajo: -40 a + 70 °C, menos del 95% de HR
- Entorno de almacenamiento: -20 a + 70 °C, menos del 95% de HR
- Tamaño (mm): MC113: 19 x 9,5 x 14; MC113C: 14 x 8 x 10
- Marca anti-exposición: Exdibl

### 3.4.2 Sensor de Monóxido

- **MQ-7** Figura 7.

[12] Los sensores de gas de la serie MQ son sensores analógicos por lo que son fáciles de implementar con cualquier microcontrolador.

Estos sensores son electroquímicos y varían su resistencia cuando se exponen a determinados gases, internamente posee un calentador encargado de aumentar la temperatura interna y con esto el sensor pueda reaccionar con los gases provocando un cambio en el valor de la resistencia.

---

<sup>7</sup>Fuente: [www.winsen-sensor.com/products/catalytic-gas-sensor-for-industrial-application/mc113.html](http://www.winsen-sensor.com/products/catalytic-gas-sensor-for-industrial-application/mc113.html)



Figura 7. Sensor MQ-7<sup>8</sup>

La salida del sensor es una resistencia análoga. El circuito de interfaz es muy simple, todo lo que se necesita hacer es alimentarlo con 5V, añadir una resistencia de carga y conectar la salida al conversor análogo – digital.

[13] El MQ-7 puede detectar concentraciones de CO de 20 a 2000 ppm (partes por millón).

Este sensor tiene una alta sensibilidad y un tiempo de respuesta rápido.

#### Especificaciones Técnicas

- Voltaje de Operación: 5V DC
- Voltaje de Calentamiento: 5V (alto) y 1.4V (bajo)
- Resistencia de carga: regulable
- Resistencia de calentamiento: 33 Ohm
- Tiempo de Calentamiento: 60s (alto) 90s (bajo)
- Consumo de Resistencia: aprox. 350mW
- Concentración de Oxígeno: 21%
- Dimensión: 35mm x 20mm x 11mm
- Temperatura de trabajo: -10 ~ 50 °C (temperatura nominal: 20 °C)

Para conectar el sensor, hay 4 cables, 2 de ellos son para el encendido. El terminal + 5V del sensor se conecta al terminal de 5V de la placa Arduino. El terminal GND del sensor se conecta al terminal GND del Arduino. Esto establece la potencia para el sensor.

Las otras 2 conexiones son la salida analógica y digital del sensor. Estos se conectan al pin analógico A0 y al pin digital D8, respectivamente.

---

<sup>8</sup> Fuente: <https://www.luisllamas.es/arduino-detector-gas-mq/>

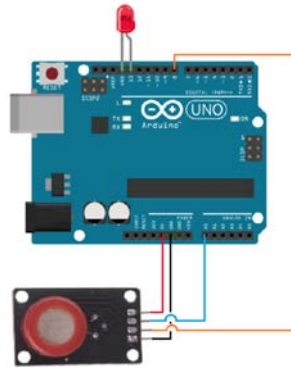


Figura 8. Conexión sensor al Arduino<sup>9</sup>

### 3.5. PLACA DE DESARROLLO

#### Arduino Pro mini

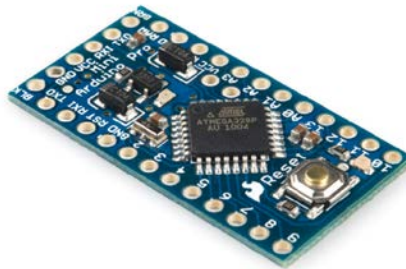


Figura 9. Arduino Pro mini<sup>10</sup>

[14] El Arduino Pro Mini es una tarjeta pequeña, perteneciente a la gran familia Arduino, ideal para proyectos donde el tamaño es una limitante, ya que está orientada a proyectos del tipo wearables (tecnología en vestuario o uso corporal). Esta tarjeta está dirigida a usuarios que requieran trabajar con sensores que usen tensiones de 3,3V, como por ejemplo acelerómetros, sensor de Presión Barométrica o que dispongan de ese voltaje, además, no trae conectores, de modo que pueden soldar en la orientación que se necesite.

---

<sup>9</sup> Fuente: <http://www.learningaboutelectronics.com/Articles/MQ-7-carbon-monoxide-sensor-circuit-with-arduino.php>

<sup>10</sup> Fuente: <http://www.electronicoscaldas.com/boards/744-sistema-de-desarrollo-arduino-pro-mini-328-5v-clone.html>



Para programar el Arduino Pro Mini se requiere de un FTDI Basic Breakout a 3,3V la cual se conecta directamente en los pines donde sale indicado GRN y BLK, energizando el Arduino Pro mini y reiniciándolo para la programación.

Características:

- Microcontrolador de 8 bits ATmega328P-AU
- Reloj de 16 MHz
- Memoria de programa (ROM): 32 kB Flash (2 kB usados por el bootloader)
- Memoria RAM: 2 kB SRAM
- Memoria ROM de datos: 1 kB EEPROM
- 22 pines entradas/salidas digitales (8 pueden actuar como entradas analógicas y 6 pueden actuar como canales PWM)
- ADC de 10 bits
- 1 USART para comunicación serial full dúplex
- 1 bus de comunicaciones TWI (2-Wire) (I<sup>2</sup>C Compatible)
- 1 bus de comunicaciones SPI
- 1 LED para uso del usuario on-board
- Interface serial (Requiere conversor USB a serial TTL para su programación)
- Botón de reset
- Operación a 5 V. Regulador on-board, corriente Max. 150 mA
- Voltaje fuente externa recomendada: 5 V a 12 V DC
- Voltaje fuente externa mínima: 5 V DC
- Voltaje máximo en cualquier entrada / salida: 5.5 V DC (o negativo: -0.5 V DC)
- Corriente máxima absoluta en cualquier salida: 40 mA
- Corriente máxima total del microcontrolador: 200 mA con fuente externa, 120 mA usando el regulador on-board. (La suma de la corriente de todas las salidas en cualquier momento no debe superar este valor)
- Tamaño aproximado: 1.8 cm x 3.4 cm

### **3.6. PROTOCOLO DE COMUNICACION**

Para el protocolo de comunicación se tuvo en cuenta dos opciones, por medio de Bluetooth

- Wi-Fi directo, se eligió el protocolo Bluetooth debido a la facilidad de implementación, menos hardware requerido, bajo consumo energético, no requiere acceso a la red y es ampliamente conocido por la mayoría de usuarios representando una menor curva de aprendizaje. Esta decisión se tomó en base a los siguientes aspectos

[15] Tanto la conexión Wi-Fi como la Bluetooth son ampliamente utilizadas para establecer conexiones inalámbricas en una amplia variedad de teléfonos móviles, computadoras y otros dispositivos de comunicación personales. Aunque las dos usan tecnología de

radiofrecuencia inalámbrica, Wi-Fi y Bluetooth realizan funciones diferentes y se distinguen por diferencias importantes.

Además de su diferente funcionalidad, hay algunas diferencias en las especificaciones entre Wi-Fi y Bluetooth. Wi-Fi tiene un alcance mucho más largo que Bluetooth, alcanzando de manera fiable hasta 300 pies (91 m), mientras que una conexión Bluetooth generalmente sostiene sólo unos 30 pies (9 m).

En cuanto a implementación Wi-Fi es más complicado y con más hardware que Bluetooth, involucrando tarjetas Wi-Fi, módems, routers y software; mientras que Bluetooth sólo requiere que los dispositivos se enciendan y se "enlacen" juntos. Bluetooth requiere un ancho de banda y alcance mucho menor que Wi-Fi y por lo tanto más es más eficiente en cuanto a energía.

- A comparación de Wifi directo, módulos Xbee, módulos Ant el Bluetooth demostró tener el menor consumo energético debido a que requiere menor ancho de banda y tiene menor alcance.
- Wifi requiere mayor hardware (tarjetas wifi, módems, routers) y software, mientras que Bluetooth solo requiere que los dispositivos se enciendan y se enlacen.
- Presenta mayor seguridad debido a que no está conectado a internet directamente.

En la siguiente tabla se resumen los aspectos más importantes de BLE (Bluetooth Low Energy) y Wi-Fi.

### What difference of BLE and Wi-Fi

BLE is reduce time to connect and power consumption

Aspect	Wi-Fi	BLE
Standardization body	IEEE LAN/MAN	Bluetooth SIG
Network Standard	IEEE 802.11	IEEE 802.15.1
Network Type	WPAN	WPAN (wireless personal area network)
Cryptography	WEP, WPA, WPA2	available
Range	50 - 100 m	~50 m
Bit rate	1Mbit - 6.75Gbit/s	~1.0 Mbit/s
Set-up time	Variant by devices	< 0.006 s
Power consumption	Related with range	< 15 mA (read and transmit)
Pairing	Yes	No (but can use, if you want)
Internet Require	Yes	No

- Wi-Fi greater transfer data than BLE
- Wi-Fi greater to support multiple security standards
- Wi-Fi power consumption is more than that of BLE
- Wi-Fi greater data range than BLE
- Wi-Fi must use internet for transfer data to application (except Wi-Fi direct)

Inspired by Chitpong Wittanan @ifew

Tabla 2. Diferencias entre Wifi y Bluetooth Low Energy<sup>11</sup>

<sup>11</sup> Fuente: <https://prezi.com/bq230-nsnhmp/cual-es-la-diferencia-entre-wi-fi-y-bluetooth/>

### 3.6.1 Módulo HC-06

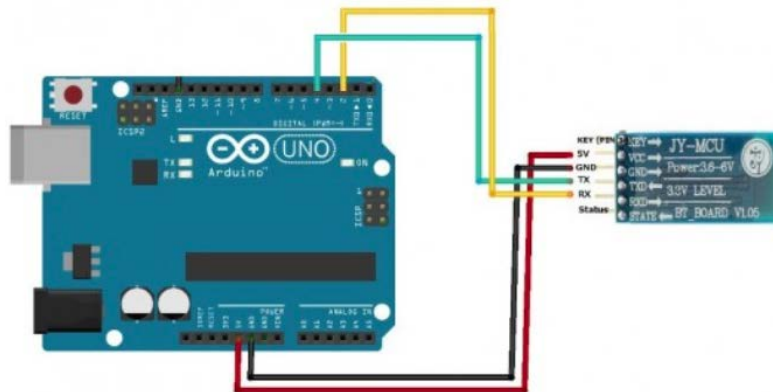


Figura 10. Conexión Módulo Bluetooth al Arduino<sup>12</sup>

[16] Un módulo Bluetooth HC-06 se comporta como esclavo, esperando peticiones de conexión, Si algún dispositivo se conecta, el HC-06 transmite a este todos los datos que recibe del Arduino y viceversa.

EL HC-06 tiene 4 pines:

- Vcc, Voltaje positivo de alimentación, se debe tener cuidado, aunque la mayoría de módulos trabajan en el rango de 3.3V a 6V es necesario revisar los datos técnicos del módulo antes de hacer las conexiones.
- GND, Voltaje negativo de alimentación, se tienen que conectar al GND del Arduino o al GND de la placa que se esté usando.
- TX, Pin de Transmisión de datos, por este pin el HC-06 transmite los datos que le llegan desde la PC o Móvil mediante Bluetooth, este pin debe ir conectado al pin RX del Arduino.
- RX, pin de Recepción, a través de este pin el HC-06 recibirá los datos del Arduino los cuales se transmitirán por Bluetooth, este pin va conectado al Pin TX del Arduino.

#### Especificaciones Técnicas

- Compatible con el protocolo Bluetooth V2.0.
- Voltaje de alimentación: 3.3VDC – 6VDC.
- Voltaje de operación: 3.3VDC.
- Baud rate ajustable: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
- Tamaño: 1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)
- Corriente de operación: < 40 mA
- Corriente modo sleep: < 1Ma

---

<sup>12</sup> Fuente: <https://www.prometec.net/bt-hc06/>

## 4. ESTADO DEL ARTE

### 4.1. ANTECEDENTES

Con el uso creciente de electrodomésticos y gasodomésticos, 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.

Alrededor del planteamiento del problema propuesto se encontraron los siguientes trabajos que se relacionan con la idea planteada

#### 4.1.1 Sense+



Figura 11.App Sense+ <sup>13</sup>

[17]Sense + es una app que funciona a partir de un sensor acoplado al móvil que en conjunto trabajan como un detector de gases. Sense + tiene un sensor de humo fotoeléctrico integrado que puede detectar incendios latentes antes de que estén fuera de control, también detecta monóxido de carbono, un gas inodoro que puede ser fatal para las personas en ciertas concentraciones Toda la información detectada pasa a través de una aplicación, que avisa del peligro al propietario. Si no hay respuesta, la aplicación llama automáticamente a los números de contacto de emergencia especificados.

#### 4.1.2 Nest Protect

[18]El detector de humos y monóxido de carbono Nest Protect es capaz de enviar una notificación a un Smartphone o Tablet, tanto con Android como iOS, donde indica qué

---

<sup>13</sup> Fuente:<http://prevenblog.com/te-interesaria-una-app-para-movil-que-detecta-humo-y-monoxido-de-carbono/>

ocurre. Así se puede identificar si se trata de una alarma por humos o porque el ambiente está sobrecargado de monóxido de carbono.

Nest Protect se conecta a nuestra red wifi para poder desplegar su sistema de notificaciones, así como la conexión al termostato Nest con el cual es compatible.

Nest Protect cuenta con un sensor de monóxido de carbono de larga duración con una vida útil de hasta una década.

En caso de emergencia, Nest Protect puede indicarle al termostato Nest que apague la calefacción para evitar que el peligro se propague.

Los tableros de partículas y gomaespuma de muchos de los muebles actuales hacen que el fuego tenga una combustión más rápida. La tecnología no puede quedarse atrás. El Sensor de Espectro Disperso usa luz de dos longitudes de onda para detectar el humo de fuegos de combustión rápida y lenta.

### Especificaciones Técnicas



Figura 12. Dimensiones Nest Protect<sup>14</sup>

<sup>14</sup> Fuente: <https://nest.com/es/smoke-co-alarm/tech-specs/>

## Funciones

- Alarmas de voz con ubicación personalizada
- Sensor de Espectro Disperso
- Detección de monóxido de carbono
- Vida útil de hasta 10 años
- Avisos
- Prueba de sonido
- Indicación nocturna
- Luz piloto
- Control de vapor
- Prueba automática
- Interconexión inalámbrica
- Apagado de emergencia con Nest Learning Thermostat
- Grabación de clip de emergencia con Nest Cam Indoor y Nest Cam IQ
- Informe de Nest
- Aplicación Nest
- Las alertas por teléfono incluyen:
  - Humo
  - Monóxido de carbono
  - Pila/batería baja
  - Fallo del sensor
- Aplicación Silenciar
- Comprobación de seguridad
- Historial de seguridad
- Qué hacer

## Sensores

- Sensor de Espectro Disperso, longitud de onda de 450 nm y 880 nm
  - Sensor de monóxido de carbono electromecánico de 10 años
  - Sensor de calor,  $\pm 1$  °C
  - Sensor de humedad, humedad relativa de  $\pm 3\%$
  - Sensor de ocupación, campo de visión de 120° a 6 metros
  - Sensor de luz ambiental, de 1 a 100.000 lx
  - Rango dinámico
-

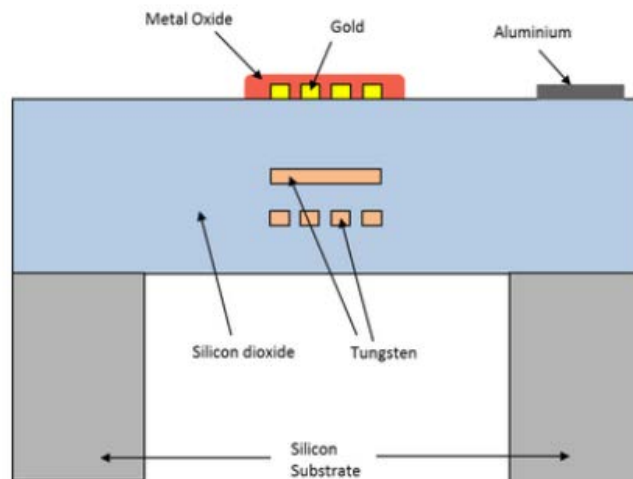
### 4.1.3 Los sensores de Gas presentes en los Smartphone's

[19]EL primer sensor de gas lo suficientemente pequeño para cualquier Smartphone fue demostrado en el MENS Executive Congress 2015. Fabricado por Cambridge CMOS Sensors Ltd. (UK), mide 1mm cuadrado el empaquetado, es lo suficientemente pequeño y económico para convertirse en ubicuo.

El sensor MENS-CMOS CCS811 puede ser configurado para detectar CO por debajo de los 20 ppm con precisión  $\pm 25\%$ ,  $CH_2O$  o como alcoholímetro.

Se espera que este sensor se incluya en un billón de unidades del mercado masivo de Smartphone, dispositivos vestibles e internet de las cosas.

[20]Cambridge CMOS Sensors tiene un sistema Microelectromecánico compuesto por sensores de gas complementarios basados en placas de semiconductores de óxido de metal (MEMS-CMOS), que se pueden escalar a tamaños más pequeños, de acuerdo con el International Technology Roadmap for Semiconductors (ITRS) esto se logra utilizando el óxido de metal superior, para ajustarlo a la detección de gases específicos se necesita de un calentamiento previo realizado por la placa ubicada debajo de él.



## Cross-section of micro-hotplate

Figura 13. Cambridge CMOS Sensors<sup>15</sup>

<sup>15</sup> (Fuente: [http://www.eetimes.com/document.asp?doc\\_id=1328199](http://www.eetimes.com/document.asp?doc_id=1328199))

## 5. DESARROLLO PRÁCTICA EMPRESARIAL

### 5.1. Requerimientos práctica empresarial

La Corporación Centro de Desarrollo Tecnológico del Gas (CDT de GAS) solicitó el desarrollo de una aplicación para telefonía móvil con sistema Operativo Android. La idea general es conectar un sensor al puerto micro USB del celular y que de esa forma se transmita los datos de medición y se visualice en el celular, a través de una app.

De forma específica, la empresa anteriormente nombrada solicito que la aplicación posea las siguientes características: (las siguientes imágenes son ilustrativas)

- **PANTALLA INICIAL:** Login y contraseña, Logo de la empresa, nombre de la aplicación Figura 14.



Figura 14.Requerimiento pantalla inicial App

- **PANTALLAS SECUNDARIAS:** Visualizar y graficar en tiempo real los valores de concentración de monóxido y metano, permitir el cambio de las unidades de medición; historización de las mediciones cuando el usuario lo indique y que el archivo se almacene en el celular o se envíe por correo. Figura 15.





Figura 15.Requerimiento Pantalla secundaria App

- **PAGINA PARA EL INGRESO DE COEFICIENTES DE CALIBRACION:** al ingresar dichos valores se ajusta el cálculo de concentración

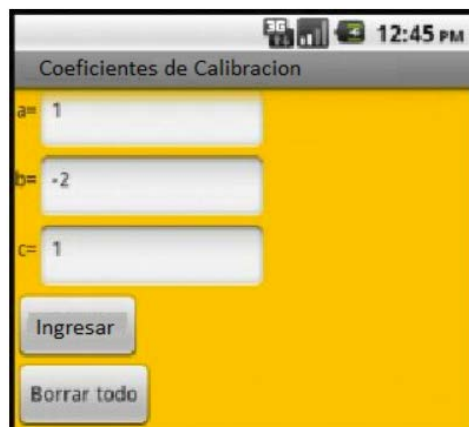


Figura 16.Requerimiento página de ingreso de coeficientes de calibración

- Alarmas (audio con mensaje de aviso) para indicar que se ha pasado los límites superior e inferior de concentración permitidos, además que permita modificar los limites.
- Funcionamiento de la aplicación en segundo plano
- 1 solo idioma (español)

## 5.2. METODOLOGIA

Para el desarrollo de este proyecto, se tomó como base una metodología de desarrollo en Espiral, ajustándola a las necesidades del proyecto. Según (Pressman, 2006) la metodología en Espiral comprende cuatro fases: Análisis (Definición de las Necesidades), Diseño (Construcción de los Diagramas que describen el software propuesto), Desarrollo (Codificación de lo representado en los diagramas) y Pruebas (Entrega al usuario e Inicio de una nueva iteración en caso de ser necesario).

Para este proyecto se definió “Requerimientos” como la etapa análisis, y se realizó la creación de una fase adicional que permite desagregar la Implementación facilitando la creación de una iteración entre Implementación y Desarrollo para la realización de pruebas unitarias que faciliten la corrección de errores.

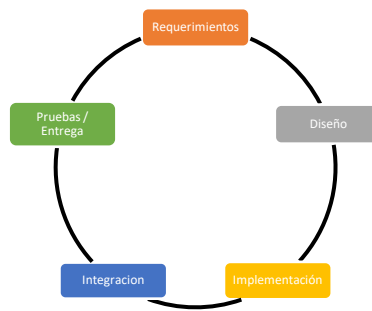


Figura 17. Metodología.<sup>16</sup>

- **Requerimientos:** Etapa inicial del proceso, en la que se realiza la obtención de los requerimientos y el análisis de los mismos con el propósito de generar el diseño. En esta etapa se definirán los diagramas de Casos de Uso.
- **Diseño:** En esta se generan los diseños de los elementos a nivel de modelo y documentación que permite a los desarrolladores contar con un amplio entendimiento de los requerimientos y de ser necesario replantear en conjunto con el cliente los requerimientos para alcanzar una viabilidad en el proyecto. La documentación del diseño de este proyecto debe contemplar como mínimo el diagrama de la arquitectura
- **Desarrollo:** En esta etapa se realiza la codificación modulo a modulo, de cada uno de los componentes lo que permite disminuir el acoplamiento sin afectar la cohesión. Esto a la larga facilita la recodificación de los módulos y el mantenimiento.

---

<sup>16</sup> Fuente: Autores. Adaptado de (Pressman, 2006)

- Implementación: Teniendo como partida los módulos ya desarrollados se realiza la integración de los mismos dentro del sistema global con el propósito de realizar pruebas de unificación y comportamiento por parte del desarrollador lo que conduce a un producto viable.
- Pruebas/Entrega: Esta etapa es una de las más importante debido a que mediante esta se da la finalización de la iteración por parte del cliente, lo que permite involucrarlo a mayor nivel y realizar de paso una validación de los requerimientos y de ser necesario el replanteamiento de los mismos dando inicio a una nueva iteración.

### 5.3. PLAN DE TRABAJO

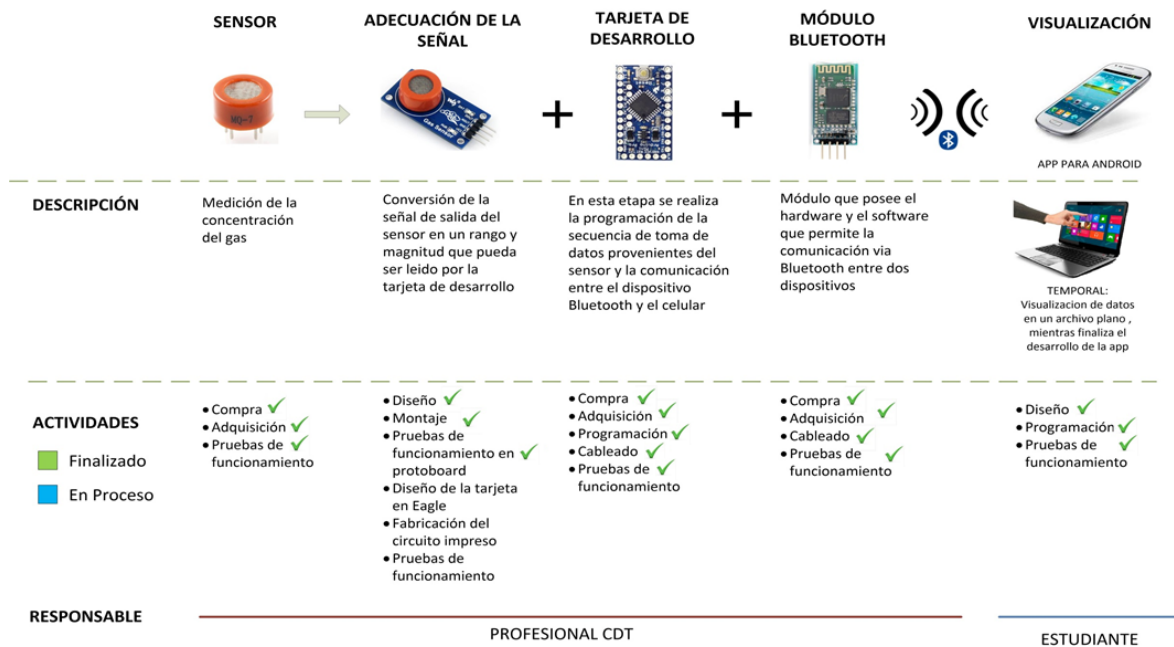


Figura 18. Plan de Trabajo tareas a realizar

#### 5.3.1 Selección de los Sensores

Estos fueron escogidos por el profesional del CDT con apoyo de sus compañeros de trabajo de acuerdo a características propias que se debían cumplir para esta aplicación específicamente: porcentaje LEL (Límite Inferior de Explosión) y detección de gas ambiental.

### 5.3.2 Desarrollo para la adecuación del sensor

*Diagrama de Bloques*

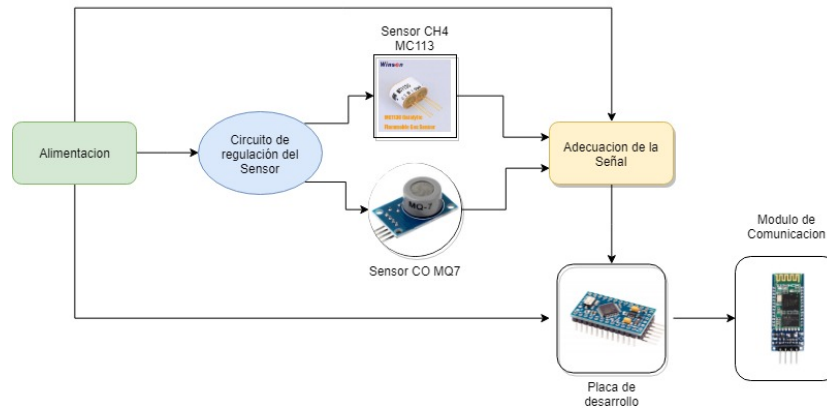


Figura 19. Diagrama de Bloques adecuación del sensor

- Etapa de Alimentación

En esta etapa se utilizó un cargador de 5.0V para la alimentación del circuito excepto la placa de desarrollo y el módulo de comunicación Bluetooth.

- Circuito de regulación del Sensor MC113

Se aplicó el circuito propuesto por el fabricante del sensor dado en el datasheet [Anexo #] siguiendo sus recomendaciones. Este es un puente de Wheatstone con las características mostradas en la Figura 18.

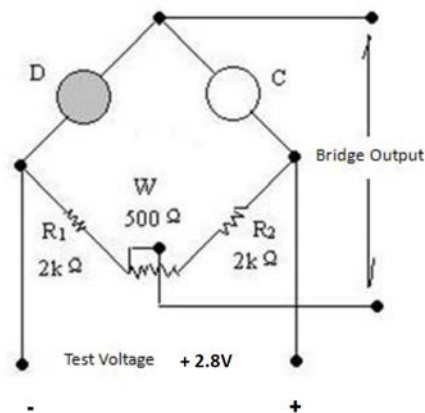


Figura 20. Circuito Puente de Wheatstone recomendado por el fabricante.<sup>17</sup>

<sup>17</sup> Fuente: [http://www.winsen-sensor.com/d/files/residential-application/mc113/mc113-mc113c-manual-\(ver1\\_3\).pdf](http://www.winsen-sensor.com/d/files/residential-application/mc113/mc113-mc113c-manual-(ver1_3).pdf)

- Adecuación de la señal

Para adecuar la señal entregada por el sensor se requiere amplificar y realizar la diferencia de las señales entregadas por el puente Wheatstone, debido a estos requisitos se escogió un amplificador de instrumentación conectado en su configuración típica [REF#], se deben tener en cuenta las recomendaciones para operación a bajos voltajes dadas en el ítem 9.1 del datasheet Figura 19. Debido a que se está alimentando a 5V se escogió una ganancia de 2.5 para que el amplificador no pierda el comportamiento lineal deseado.

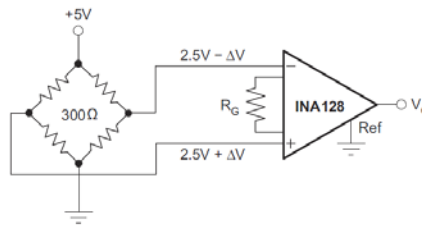


Figura 21. Circuito para operación a bajos voltajes recomendado<sup>18</sup>

- Placa de Desarrollo

El profesional del CDT escogió la placa de desarrollo de hardware Arduino Pro Mini, debido a su pequeño tamaño con Dimensiones de 18 x 33mm y bajo costo aproximadamente \$35.000 COP por la placa y su módulo programador.

Esta placa cuenta con un ADC (Convertor análogo a digital) embebido de 10 bits para discretizar la señal analógica, una vez discretizada esta será transmitida por Bluetooth a la aplicación del Smartphone para procesar los datos y entregar información útil de concentración de gas CO o  $CH_4$  al usuario.

- Módulo de transmisión de datos

Se escogió el protocolo de comunicación Bluetooth para transmitir los datos leídos por el ADC del Arduino al Smartphone, por esta razón se necesitó un módulo complementario, el elegido fue el HC06 que cumple con los requerimientos además es de fácil consecución y a precios muy asequibles.

<sup>18</sup> Fuente: <http://www.ti.com/lit/ds/symlink/ina128.pdf>

### 5.3.3 Área de trabajo Laboratorio de calidad de gas



Figura 22.Espacio de Trabajo

- Equipos utilizados para la realización de las pruebas



Figura 23.Dilutor de Gas TGD2000

El dispositivo generador de trazas TGD 2000, fue parte fundamental en el desarrollo del proyecto, debido a que el brinda la posibilidad de diluir la concentración de gas y generar los porcentajes necesarios para realizar las pruebas que permitirán obtener la ecuación característica del sensor.



Figura 24. Sistema adquisición de datos Dilutor de Gas TGD2000

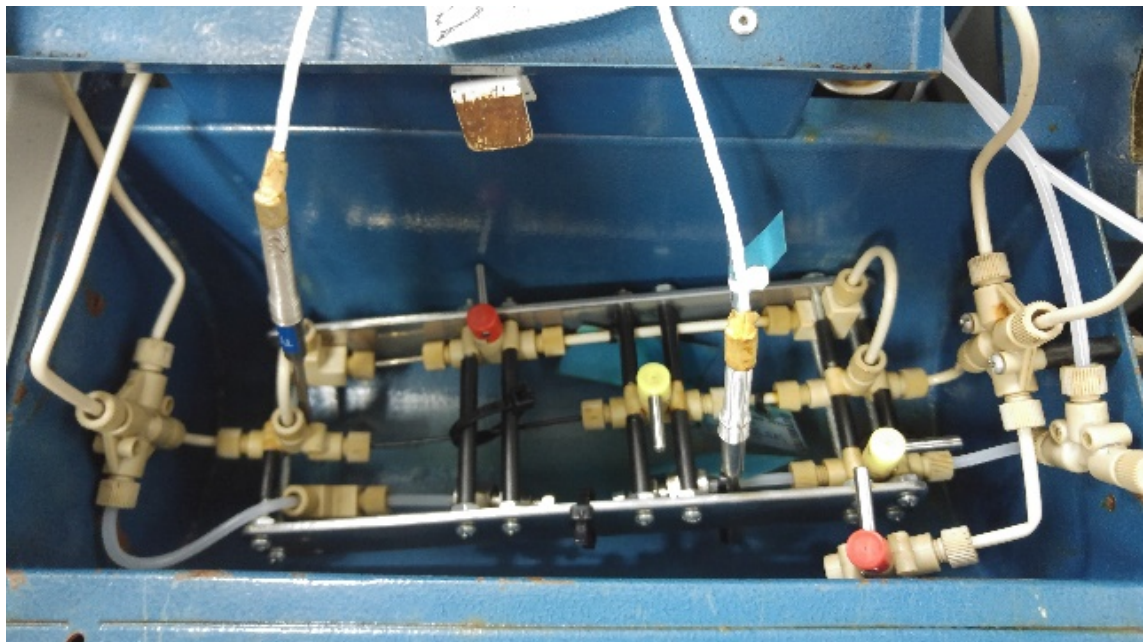


Figura 25. Capilares para control de caudal de gas



Figura 26. Cilindro de gas  $CH_4$  Metano o gas natural



Figura 27. Regulador de presión





Figura 28. Cámara de gas para caracterizar el sensor

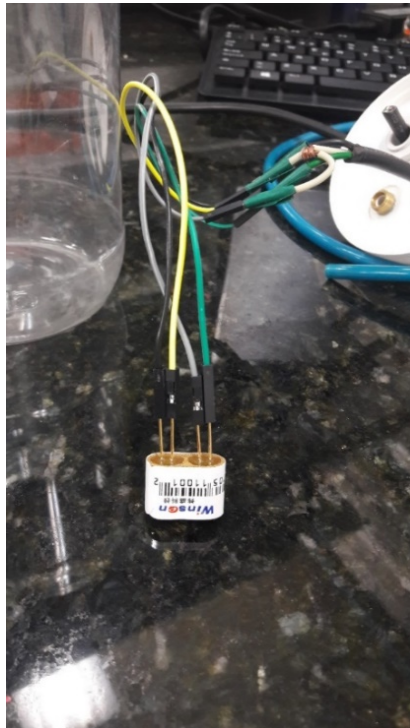


Figura 29. Sensor MC113 marca Winsen utilizado para  $CH_4$



Figura 30. Sensor MQ7 para Arduino

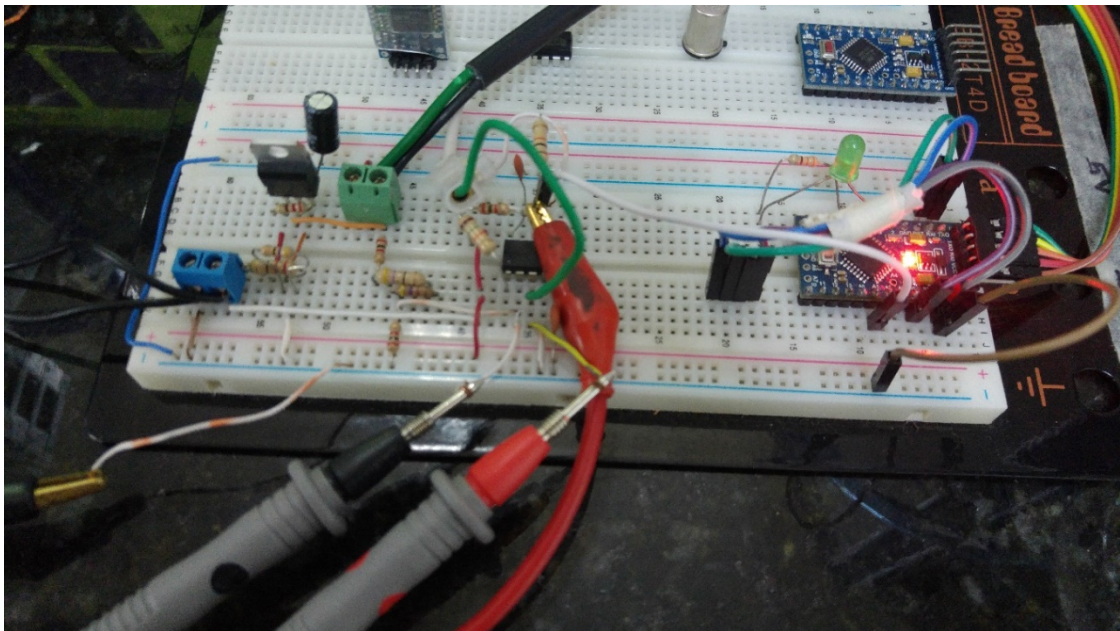


Figura 31. Circuito Implementado para realizar las pruebas



Figura 32. Multímetro a la derecha lectura de la diferencia del puente de Wheatstone y a la izquierda lectura después del amplificador de instrumentación.

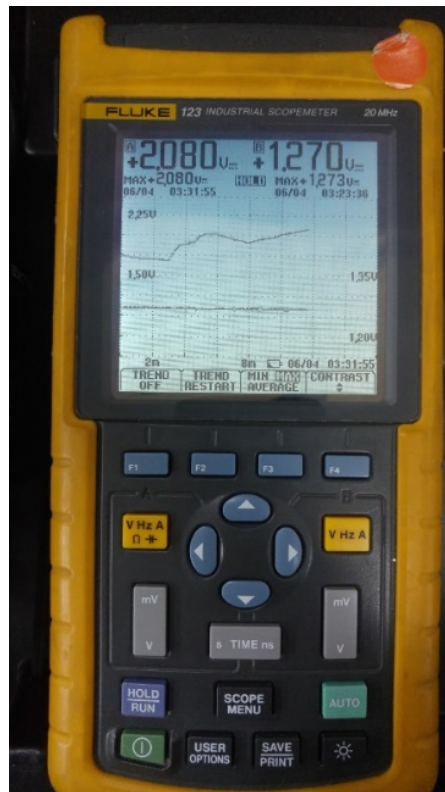


Figura 33. Osciloscopio para realizar seguimiento a la señal



Figura 34. Utilizando un detector calibrado para corroborar el %mol de concentración de gas.

#### 5.3.4 Caracterización del sensor

Para la caracterización del sensor se realizaron como mínimo 3 pruebas a diferentes horas del día y cambiando de día con el fin de evaluar la reproducibilidad del sensor.

Cada prueba consistió en pasar concentraciones que van desde 0% mol hasta 3% para el sensor de  $\text{CH}_4$  y tomar la lectura dada por el monitor serie del Arduino con una frecuencia de 10 segundos hasta completar un lapso de tiempo de 7 minutos, de los cuales 5 minutos se utilizan para el llenado de la cámara y 2 minutos para obtener una lectura estable.

La prueba de 1% mol es realizada con el capilar de 50µm de diámetro y las demás concentraciones se hacen fluir a través del capilar de 75µm.

#### 5.3.4.1 Metodología para realizar las pruebas

Una vez en funcionamiento el Dilutor y alimentado el circuito de adecuación de la señal del sensor, se procede a realizar los siguientes pasos para ejecutar las pruebas y de esa forma obtener la ecuación característica de la planta. Es recomendable generar una concentración inicial de 0% mol con el fin de definir el valor equivalente al Zero del sistema

1. Se debe Regular la presión de Aire y del gas, para asegurar que la concentración de la mezcla equivale al porcentaje que se desea medir.
2. Definir la presión de cada uno de los gases de acuerdo a la tabla a continuación:

Puntos	Valor [% mol]	<b>Metano</b>	<b>Aire</b>
		Presión [Psi]	Presión [Psi]
1	0		75
2	1	27	75
3	2	46	50
4	3	55	50

Tabla 3.Presiones según porcentaje de concentración

Una vez estén estables las presiones empezar la toma de datos

3. Realizar la toma de las 42 muestras equivalentes a 7 minutos tomando lectura cada 10 segundos.
4. Al finalizar cada prueba esperar un lapso mínimo de 5 minutos o más haciendo circular solo aire para descontaminar la cámara antes de iniciar la prueba con el siguiente porcentaje de concentración
5. Realizar el siguiente conjunto de pruebas con un lapso de diferencia mínimo de 4 a 6 horas para que las condiciones varíen, de preferencia de un día para otro.
6. Realizar mínimo de tres conjuntos de pruebas antes de generar la ecuación característica.
7. Obtener la ecuación.



Figura 35. Almacenamiento de datos en Microsoft Excel

5.3.4.1.1 Prueba 1

Número de Muestra	Concentración			
	0%	1%	2%	3%
1	0,393	0,415	0,404	0,393
2	0,392	0,421	0,418	0,393
3	0,393	0,426	0,427	0,392
4	0,392	0,429	0,436	0,399
5	0,393	0,432	0,444	0,421
6	0,393	0,433	0,448	0,435
7	0,392	0,436	0,454	0,444
8	0,391	0,438	0,457	0,452
9	0,394	0,437	0,46	0,466
10	0,393	0,437	0,462	0,474
11	0,393	0,439	0,464	0,479
12	0,393	0,439	0,465	0,484
13	0,393	0,439	0,468	0,486
14	0,393	0,44	0,469	0,488
15	0,393	0,44	0,47	0,491
16	0,393	0,439	0,469	0,494
17	0,393	0,439	0,471	0,496
18	0,392	0,439	0,471	0,496
19	0,393	0,44	0,472	0,497
20	0,393	0,439	0,473	0,499
21	0,394	0,44	0,473	0,499
22	0,393	0,439	0,473	0,499

23	0,392	0,44	0,473	0,499
24	0,392	0,44	0,474	0,5
25	0,392	0,44	0,474	0,502
26	0,392	0,44	0,474	0,502
27	0,394	0,441	0,473	0,502
28	0,393	0,44	0,473	0,501
29	0,392	0,44	0,474	0,502
30	0,393	0,44	0,473	0,503
31	0,393	0,44	0,474	0,503
32	0,391	0,439	0,475	0,504
33	0,393	0,44	0,474	0,503
34	0,394	0,44	0,474	0,503
35	0,393	0,44	0,473	0,502
36	0,393	0,441	0,476	0,503
37	0,393	0,44	0,473	0,504
38	0,394	0,44	0,474	0,505
39	0,392	0,44	0,475	0,503
40	0,394	0,44	0,475	0,504
41	0,393	0,439	0,474	0,503
42	0,392	0,44	0,475	0,503
Promedio	0,39282609	0,43991304	0,47386957	0,50208696

Tabla 4. Prueba 1 MC113

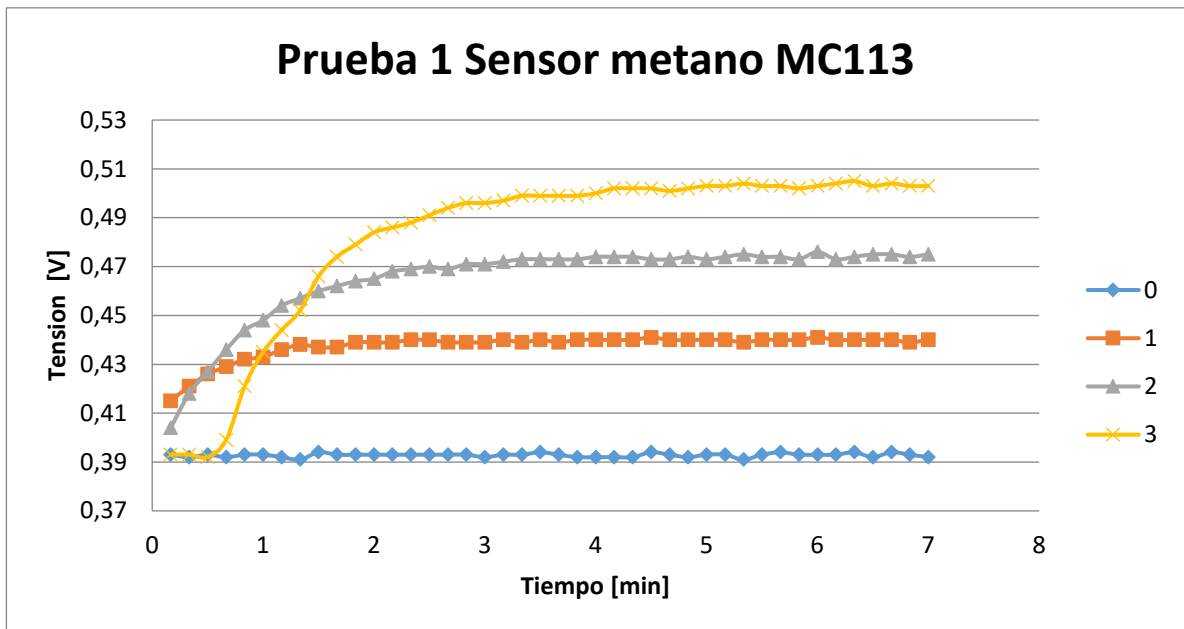


Figura 36. Prueba 1 Sensor MC113

5.3.4.1.2 Prueba 2

Número de Muestra	Concentración			
	0%	1%	2%	3%
1	0,406	0,406	0,405	0,406
2	0,406	0,406	0,41	0,416
3	0,405	0,412	0,424	0,434
4	0,405	0,418	0,435	0,446
5	0,405	0,425	0,444	0,458
6	0,406	0,431	0,45	0,469
7	0,405	0,435	0,456	0,476
8	0,405	0,438	0,461	0,482
9	0,406	0,44	0,465	0,487
10	0,405	0,443	0,468	0,491
11	0,405	0,444	0,47	0,494
12	0,406	0,444	0,471	0,497
13	0,404	0,447	0,473	0,499
14	0,405	0,447	0,476	0,5
15	0,405	0,448	0,478	0,502
16	0,405	0,448	0,478	0,503
17	0,405	0,447	0,478	0,505
18	0,405	0,448	0,479	0,506
19	0,405	0,448	0,478	0,506
20	0,406	0,448	0,479	0,506
21	0,406	0,448	0,48	0,508
22	0,405	0,448	0,481	0,508
23	0,406	0,448	0,481	0,507
24	0,405	0,449	0,48	0,508
25	0,405	0,448	0,48	0,509
26	0,406	0,449	0,481	0,509
27	0,406	0,449	0,48	0,508
28	0,406	0,448	0,48	0,509
29	0,406	0,448	0,48	0,509
30	0,406	0,449	0,48	0,508
31	0,405	0,448	0,48	0,509
32	0,405	0,449	0,48	0,51
33	0,406	0,448	0,479	0,51
34	0,405	0,449	0,481	0,51
35	0,406	0,449	0,479	0,509
36	0,406	0,45	0,481	0,509
37	0,405	0,449	0,481	0,51
38	0,404	0,45	0,481	0,51



39	0,405	0,449	0,481	0,509
40	0,404	0,449	0,48	0,51
41	0,406	0,449	0,481	0,509
42	0,405	0,448	0,48	0,509
Promedio	0,40543478	0,44865217	0,48026087	0,50882609

Tabla 5.Prueba 2 MC113

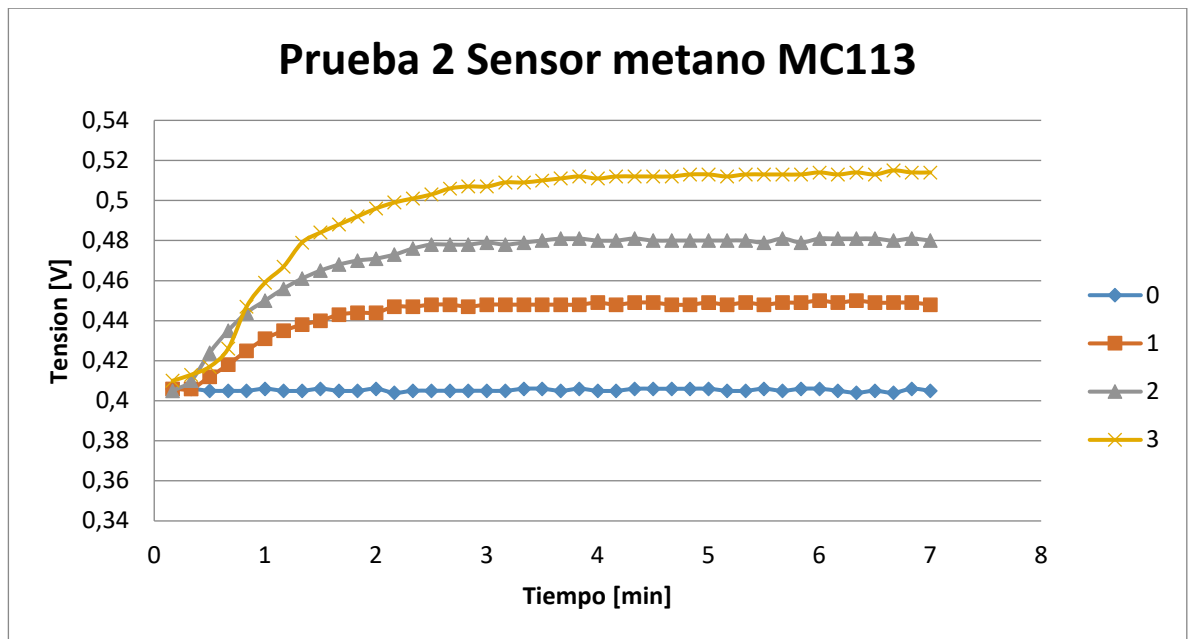


Figura 37. Prueba 2 Sensor MC113

Prueba 3

Número de Muestra	Concentración			
	0%	1%	2%	3%
1	0,411	0,41	0,412	0,41
2	0,41	0,411	0,411	0,413
3	0,412	0,413	0,413	0,417
4	0,411	0,42	0,428	0,426
5	0,411	0,431	0,44	0,447

6	0,411	0,435	0,449	0,459
7	0,411	0,441	0,453	0,467
8	0,411	0,443	0,461	0,479
9	0,411	0,448	0,464	0,484
10	0,411	0,448	0,468	0,488
11	0,412	0,45	0,471	0,492
12	0,411	0,452	0,475	0,496
13	0,411	0,451	0,477	0,499
14	0,411	0,452	0,478	0,501
15	0,411	0,452	0,479	0,503
16	0,41	0,452	0,48	0,506
17	0,411	0,452	0,481	0,507
18	0,41	0,451	0,483	0,507
19	0,411	0,451	0,483	0,509
20	0,411	0,452	0,483	0,509
21	0,41	0,452	0,483	0,51
22	0,41	0,453	0,485	0,511
23	0,411	0,453	0,484	0,512
24	0,41	0,454	0,484	0,511
25	0,411	0,453	0,485	0,512
26	0,41	0,453	0,484	0,512
27	0,411	0,454	0,485	0,512
28	0,411	0,454	0,485	0,512
29	0,41	0,454	0,485	0,513
30	0,411	0,453	0,486	0,513
31	0,411	0,454	0,486	0,512
32	0,41	0,454	0,485	0,513
33	0,411	0,454	0,486	0,513
34	0,41	0,454	0,484	0,513
35	0,41	0,453	0,485	0,513
36	0,411	0,454	0,485	0,514
37	0,41	0,454	0,485	0,513
38	0,41	0,454	0,486	0,514
39	0,41	0,455	0,485	0,513
40	0,411	0,455	0,485	0,515
41	0,411	0,454	0,485	0,514
42	0,411	0,454	0,484	0,514
Promedio	0,41052174	0,45365217	0,48478261	0,51252174

Tabla 6.Prueba 3 MC113

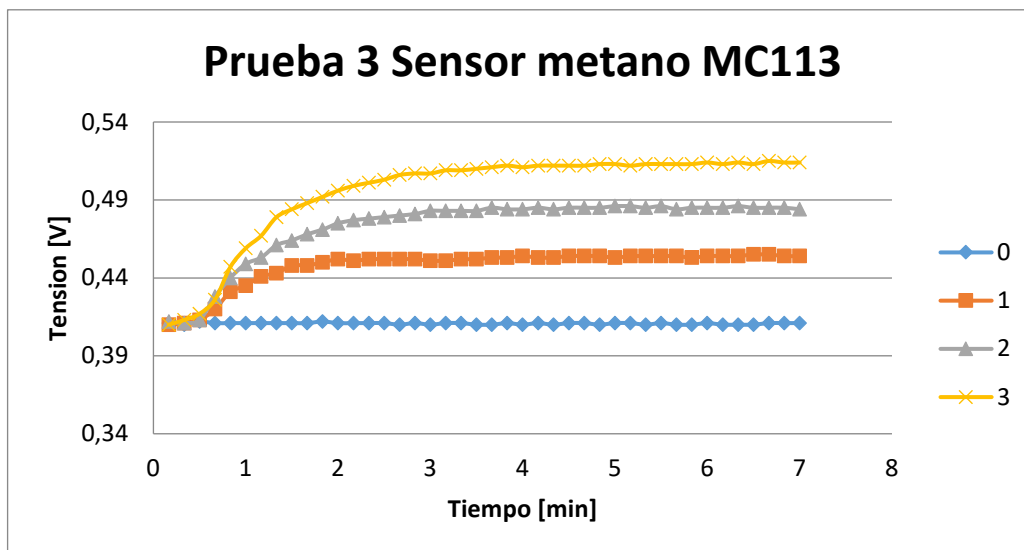


Figura 38. Prueba 3 Sensor MC113

Ecuación	
Valor [% mol]	Promedio
0	0,407978261
1	0,451152174
2	0,482521739
3	0,510673913

Tabla 7. Valores promedio de las 3 pruebas para obtener la ecuación

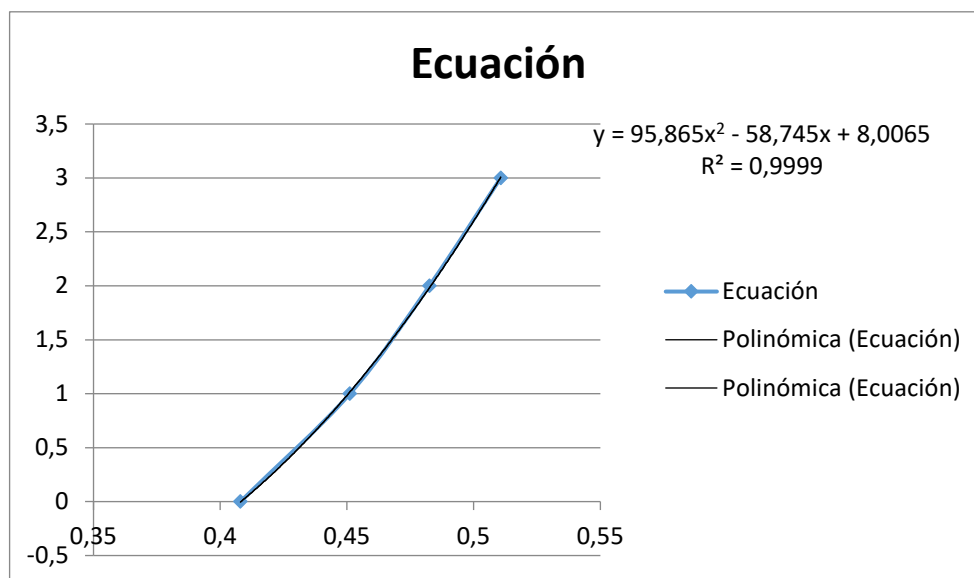


Figura 39. Curva de la ecuación característica para el sensor MC113 de Metano

Una vez obtenidos los valores promedio de las pruebas se calcula la ecuación por medio de Microsoft Excel insertando una línea de tendencia polinómica de segundo orden

$$y = 95,865x^2 - 58,745x + 8,0065$$

#### 5.3.4.2 Caracterización del Sensor de Monóxido

Se caracterizó con el mismo método del sensor de metano  $CH_4$ .

#### Prueba Final

Para la realización de las pruebas este sensor requiere de precalentamiento, toma 60 muestras y luego se desactiva por 90 segundos, la alimentación se realizó por PWM para su activación y desactivación

<b>Dato</b>	<b>0 ppm</b>	<b>15 ppm</b>	<b>50 ppm</b>	<b>95 ppm</b>
1	217	<b>232</b>	257	277
2	224	239	263	285
3	230	244	267	290
4	235	249	271	295
5	239	254	275	299
6	243	257	279	302
7	247	261	281	305
8	249	263	283	308
9	252	265	286	310
10	253	267	287	311
11	255	267	288	313
12	257	268	290	314
13	258	269	291	315
14	259	270	292	316
15	259	272	293	317
16	260	272	294	317
17	261	272	294	318
18	262	273	294	319

19	262	273	294	319
20	263	274	295	319
21	263	274	296	320
22	264	274	295	320
23	264	275	296	320
24	264	275	296	320
25	264	275	296	321
26	264	275	296	321
27	264	276	297	321
28	264	276	296	321
29	264	276	297	321
30	264	276	297	321
31	264	276	297	321
32	265	276	297	320
33	265	276	297	321
34	265	276	297	320
35	266	277	297	320
36	266	276	297	320
37	265	276	297	320
38	265	276	297	320
39	265	276	298	320
40	265	276	297	320
41	265	276	297	320
42	265	277	297	321
43	265	277	298	321
44	265	277	297	321
45	265	277	297	321
46	265	277	298	321
47	265	277	297	321
48	265	277	297	321
49	265	277	297	321
50	265	277	297	320
51	265	277	297	320
52	265	277	297	320
53	265	277	297	320
54	265	277	297	320
55	265	277	297	320
56	264	277	297	320
57	264	277	297	320
58	264	277	297	320
59	264	277	297	320
60	264	277	297	320

Tabla 8 Prueba MQ7

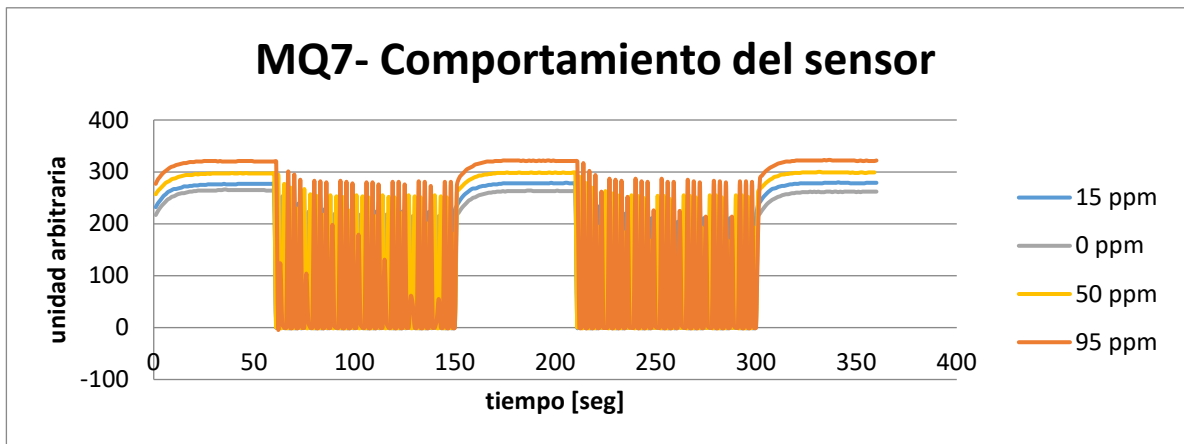


Figura 40. Prueba MQ7

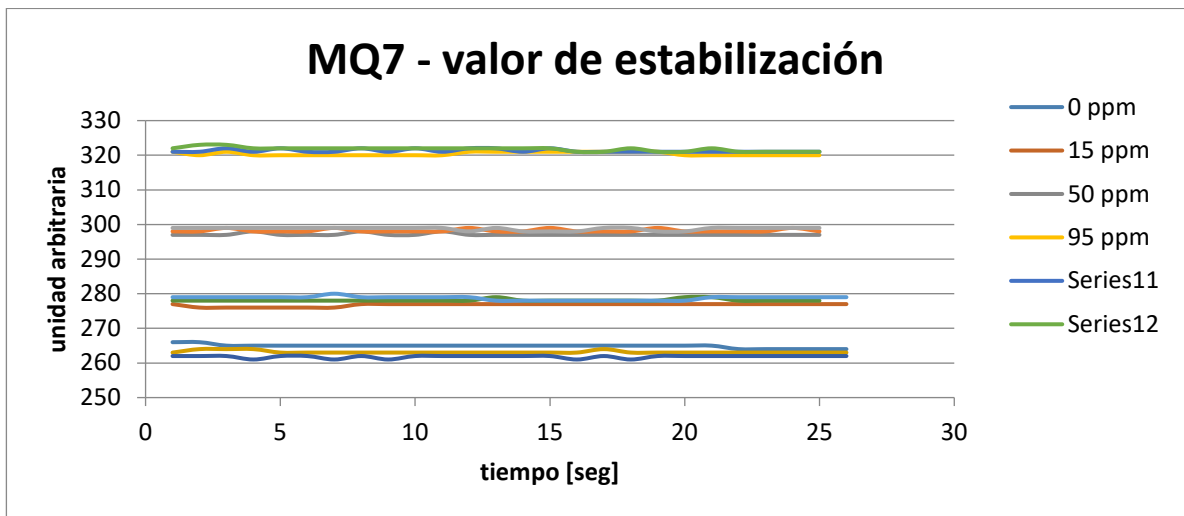


Figura 41. Valor de Estabilización MQ7

Promedio				
0	15	50	95	
265	277	297	320	
263	278	287	321	
262	279	299	322	
263	278	294	321	

Tabla 9 Promedio de las pruebas

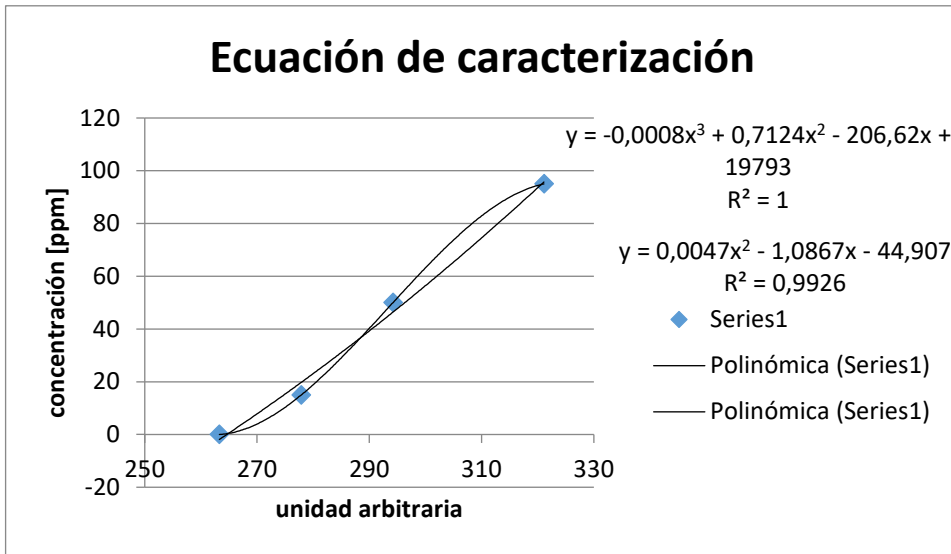


Figura 42. Ecuación característica MQ7

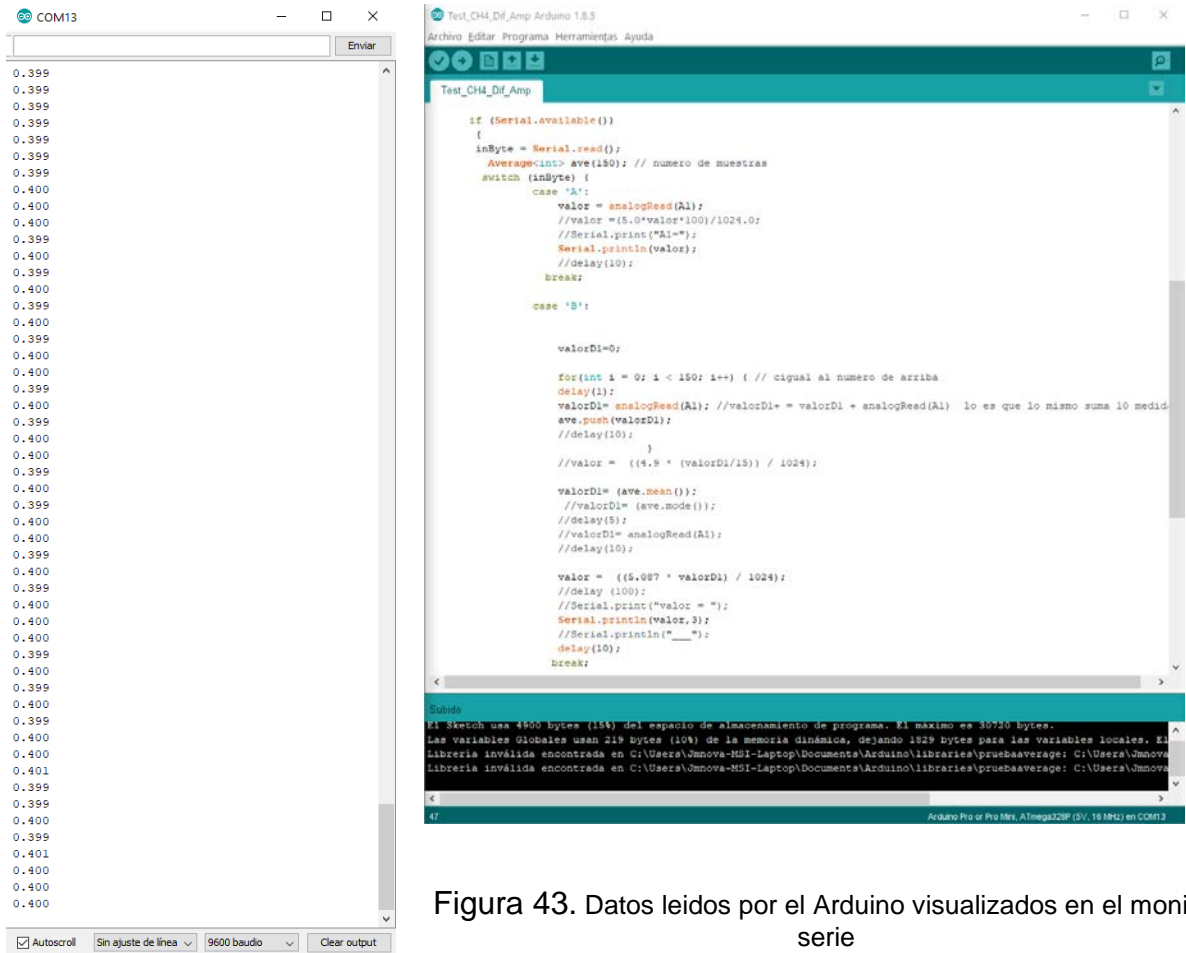


Figura 43. Datos leídos por el Arduino visualizados en el monitor serie

En la siguiente imagen se puede observar el dato leído por el Arduino y comprobar que es el mismo recibido por Android Studio y con el que se va a aplicar el calculo para entregar la grafica de concentracion.

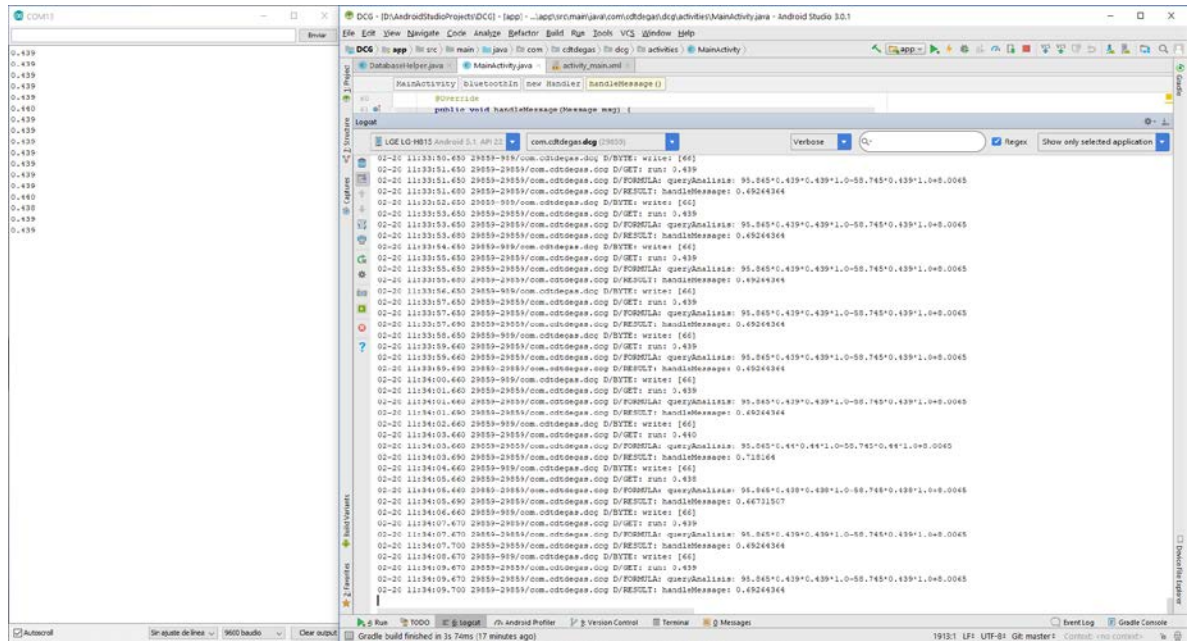


Figura 44. Monitor Arduino y Android Studio

- Una vez en ejecución la app envía una A en código ASCII [66] para indicarle al programa del Arduino a que ítem del case debe ingresar y solicitar el dato, una vez retorne el valor leído del sensor la App procede a realizar el cálculo en la ecuación característica respectiva como se observa en la Imagen [38] y el valor de ese resultado es lo que se muestra en la gráfica.

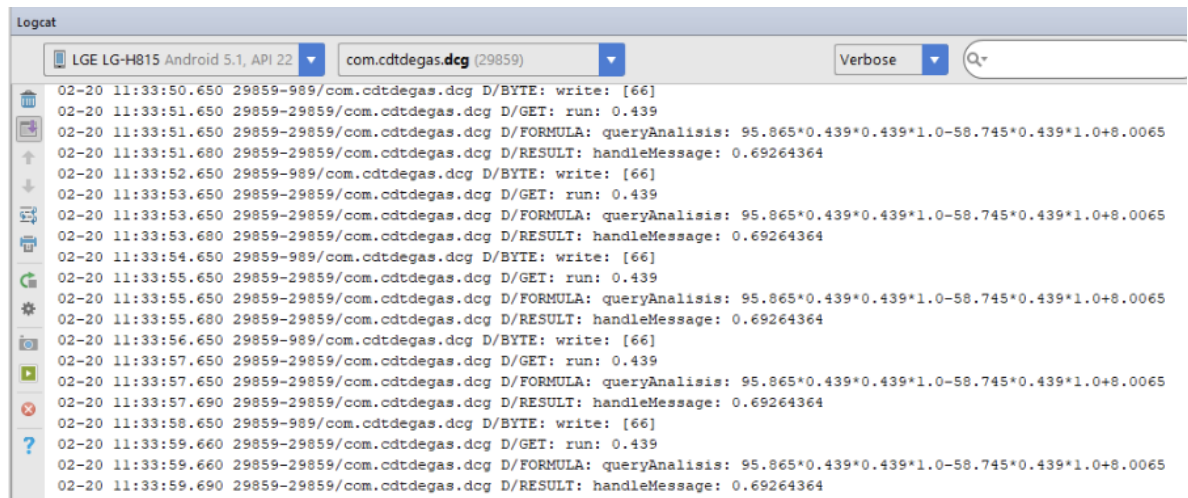


Figura 45.Registro de Android Studio con la aplicación en ejecución



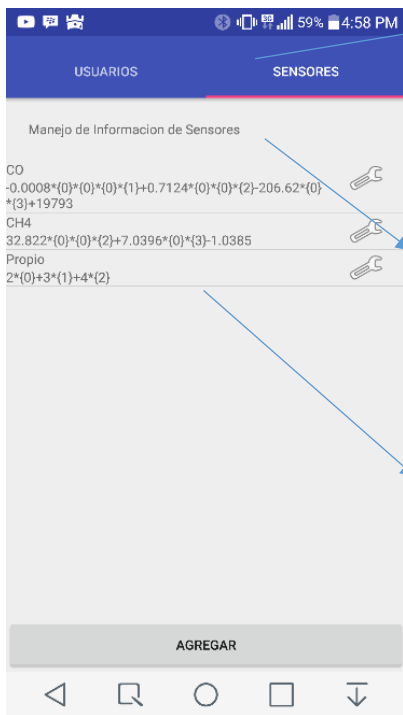


Figura 46. Concentración ascendente y Concentración descendente

#### 5.3.4.2.1 Descripción de la aplicación

En las siguientes imágenes se puede observar el uso de fragments, helpers y adapters con ejemplos de la App en ejecución en donde se indica la aplicación de cada uno de ellos.

En la Figura [47] corresponde al usuario administrador en donde se crean los nuevos usuarios o sensores



Barra de Tabs (AdminTabHelper)

Encargado de controlar el funcionamiento de los tabs y la acción al desplazarse mediante ellos.

Fragmento Principal (UsersFragment y SensorsFragment): Pequeños fragmentos de código que se encargan de simular el comportamiento de una actividad completa. Encargado de dar vida a cada uno de los tabs descritos por el Helper este a su vez puede contener otros fragmentos de código.

Fragmentos del Cuerpo (GasAddFragment, UserAddFragment, GasListFragment y UserListFragment): Pueden ser autocontenidos o como en este caso llamados uno dentro de otro se mueven dentro del contexto de su fragmento padre y no tienen acceso a la actividad principal.

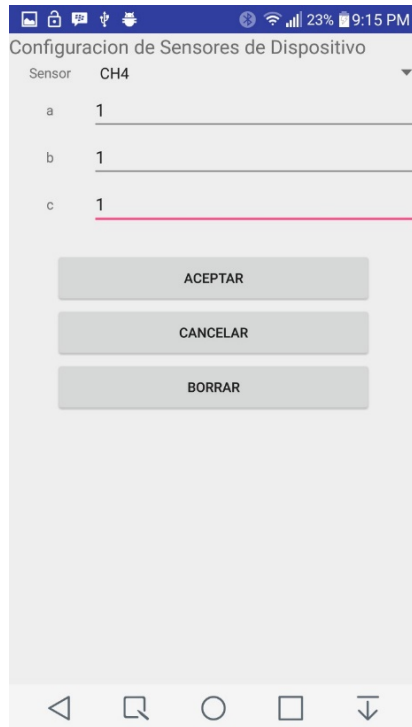
Figura 47.Descripción 1 Fragments, Handlers, Helpers



Adaptador (GasAdapter) un adaptador de Arreglos es el encargado de procesar los elementos de un arreglo y generar un elemento visual por cada uno de ellos, para este caso cada uno de los elementos de lista

Figura 48.Descripción 2 Adapter

La figura [49] corresponde al ingreso de los coeficientes de calibración



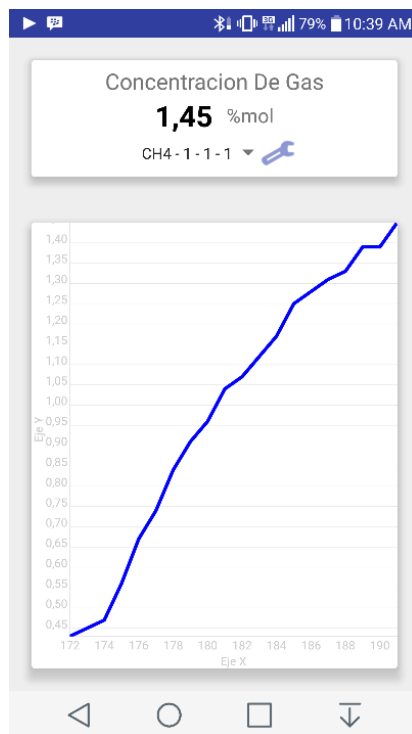
Configuración de Coeficientes:

Pantalla Principal (SensorActivity): Contenedor principal de los fragmentos, relaciona los coeficientes registrados al dispositivo.

Gestionar Coeficientes (SensorListFragment y SensorAddFragment): permiten listar y agregar información de coeficientes para diferentes sensores ya registrados.

Figura 49.Descripción 3 Activity, Fragment

La figura [50] captura la gráfica de concentración con variación ascendente



Actividad Principal (MainActivity):

Esta actividad es la encargada de la articulación del proceso de control de flujo del Bluetooth, se utiliza el plugin de HelloCharts para el proceso de graficar con ajustes predeterminados.

Figura 50.Descripción 4 Main Activity

### 5.3.5 Implementación del protocolo Bluetooth en la Aplicación

- Connected Thread se encarga del envío y recepción de información mediante el uso de un flujo de entrada y uno de salida con control unidireccional. El método run se encarga de realizar el envío y petición de información mediante el uso de los flujos en los métodos read y write mediante el uso de un ciclo que se ejecuta mientras la conexión este activa. Cada una de estas peticiones tiene un intervalo de 1 segundo.

```
private class ConnectedThread extends Thread {
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;

    ConnectedThread(BluetoothSocket socket) {
        //Flujos de Entrada y Salida
        InputStream tmpIn = null;
        OutputStream tmpOut = null;
        try {
            //Con el Socket Abierto se Asocian los Flujos al socket
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException ignored) {}
        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }

    @Override
    public void interrupt() {
        try {
            btSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
            btSocket = null;
        }
    }

    public void run() {
        while (btSocket != null && btSocket.isConnected()) {
            try {
                //Escribir y leer la respuesta del Dispositivo
                //Coeficientes coeficientes = databaseHelper.queryDeviceData(address);
                //write(coeficientes.getValorA() + "," + coeficientes.getValorB() + "," + coeficientes.getValorC());
                write(currentGas.getCodigo());
                Thread.sleep( millis 1000);
                read();
                Thread.sleep( millis 1000);
            } catch (InterruptedException ignored) {
                try {
                    btSocket.close();
                } catch (IOException e) {
                    btSocket = null;
                }
                break;
            }
        }
    }

    void read() {
        byte[] buffer = new byte[1024];
        int bytes;
        try {
            //Se lee lo enviado por el Dispositivo
            bytes = mmInStream.read(buffer);
            //Se notifica al Handler para que manipule el mensaje
            bluetoothIn.obtainMessage(handlerState, bytes, -1, buffer).sendToTarget();
        } catch (IOException ignored) {}
    }

    void write(String input) {
        try {
            //Se envia el Valor para esperar la respuesta.
            Log.d( tag: "BYTE", msg: "write: " + Arrays.toString(input.getBytes()));
            mmOutStream.write(input.getBytes());
        } catch (IOException e) {
            finish();
        }
    }
}
```

Figura 51. Implementación de protocolo Bluetooth 1

- Handler es un elemento manejador de flujos de control de Android que se encarga de la manipulación de información de comunicación entre actividades y elementos como hilos derivados de la aplicación. Acá se recibe el flujo de datos y se procesa de acuerdo a lo necesario. Acá se procesa para que la información de la gráfica se actualice mediante la aplicación de las ecuaciones de ajuste a la información enviada por el dispositivo.

```

//Handler encargado de recibir los mensajes que se envían por el hilo de conexión
Handler bluetoothIn = new Handler(Looper.getMainLooper()) {
    @Override
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case 0: {
                //Recibe el Mensaje
                String message = new String((byte[]) msg.obj, offset 0, msg.arg1);
                Log.d( tag: "GET", msg: "run: " + message);
                //Crea un Valor en Base al String Recibido
                float yValue = databaseHelper.queryAnálisis(currentGas.getFormula(), Float.valueOf(message)
                    , Float.parseFloat(currentSensor.getValorA()), Float.parseFloat(currentSensor.getValorB())
                    , Float.parseFloat(currentSensor.getValorC()));
                Log.d( tag: "RESULT", msg: "handleMessage: " + yValue);
                //Muestra en Pantalla el Valor Leído
                IdBufferIn.setText(String.format(new Locale( language: "es", country: "CO"), format: "%.01f", yValue));
                //Obtiene el arreglo de datos actual en la grafica
                LineChartData data = mChart.getLineChartData();
                //Si se llega a los 20 puntos mostrados elimine el primero
                if (pointValues.size() == 20) {
                    pointValues.remove( 0);
                }
                //Se añade un nuevo punto con el valor capturado
                pointValues.add(new PointValue(i, yValue));
                i++;
                //Se setea los nuevos puntos en la pantalla
                data.getLines().get(0).setValues(pointValues);
                //Se asocian a la Grafica Mostrada
                mChart.setLineChartData(data);
                //Se actualiza el valor de los ejes
                setViewport();
                return;
            }
            default: {
            }
        }
    }
};

```

Figura 52.Implementación de protocolo Bluetooth 2

- Se inicia el proceso de conexión mediante la creación de un BluetoothDevice (Dispositivo Bluetooth) como una representación lógica del dispositivo remoto, mediante el uso de la dirección mac. Mediante este elemento se realiza la creación de un Socket de conexión para el control de flujo. Mediante dicho socket se inicializa la clase ConnectedThread.

```

if (MyConexionBT == null) {

BluetoothDevice device = btAdapter.getRemoteDevice(address);
try {
    //Crea el Socket de Conexion con el Dispositivo
    btSocket = createBluetoothSocket(device);
} catch (IOException e) {
    Toast.makeText(getBaseContext(), text "La creación del Socket fallo", Toast.LENGTH_LONG).show();
}

// Establece la conexión con el socket Bluetooth.
try {
    btSocket.connect();
    //Se crea el hilo para manejar la conexión
    MyConexionBT = new ConnectedThread(btSocket);
    //Se inicia la escucha
    MyConexionBT.start();
} catch (IOException e) {
    try {
        btSocket.close();
    } catch (IOException ignored) {
    }
    Intent intentLogin = new Intent(getBaseContext(), DeviceListActivity.class);
    startActivity(intentLogin);
    finish();
}
}
}

```

Figura 53.Implementación de protocolo Bluetooth 3

- Cierre del flujo de datos y del hilo de control para poder abandonar la pantalla y liberar el dispositivo.

```

try {
    if (btSocket.isConnected()) {
        btSocket.close();
    }
    if (MyConexionBT != null) {
        MyConexionBT.interrupt();
        MyConexionBT = null;
    }
} catch (IOException ignored) {
}
}

```

Figura 54.Implementación de protocolo Bluetooth 4

- Para el diseño (Layout) de la pantalla de ingreso de usuario el lenguaje de programación utilizado por el entorno de desarrollo Android Studio es XML, este permite definir cada uno de los atributos de los botones, cuadros de entrada de

textos, logos y demás características propias de la interfaz gráfica utilizada para esa pantalla relacionada a la actividad “Log In”.

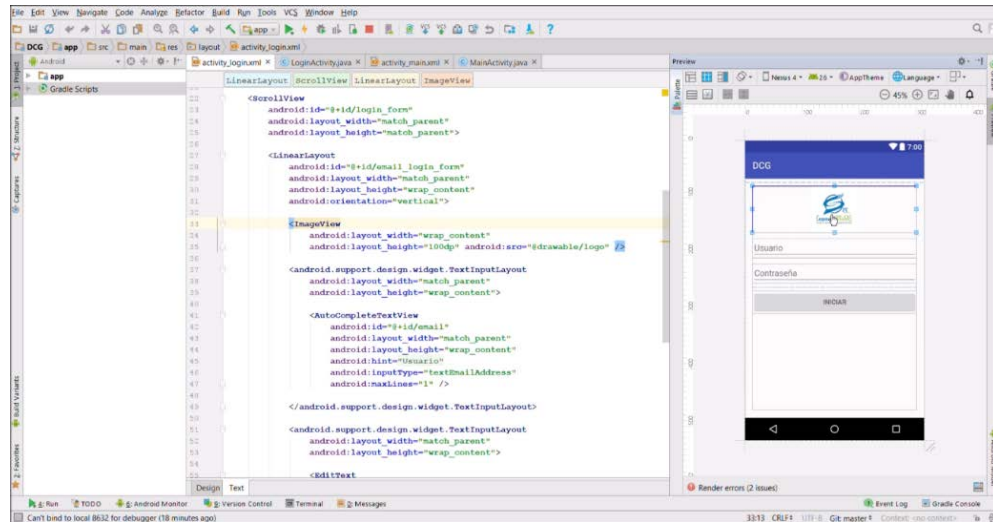


Figura 55. Programación grafica para la pantalla de “Log In” o Ingreso de Usuario

- En la imagen a continuación se observa la clase principal o public class, relacionada a la actividad de ingreso (log in), en ella se definieron dos usuarios con su respectivo nombre y contraseña, además de las acciones relacionadas con cada botón o campo definido para esta actividad.

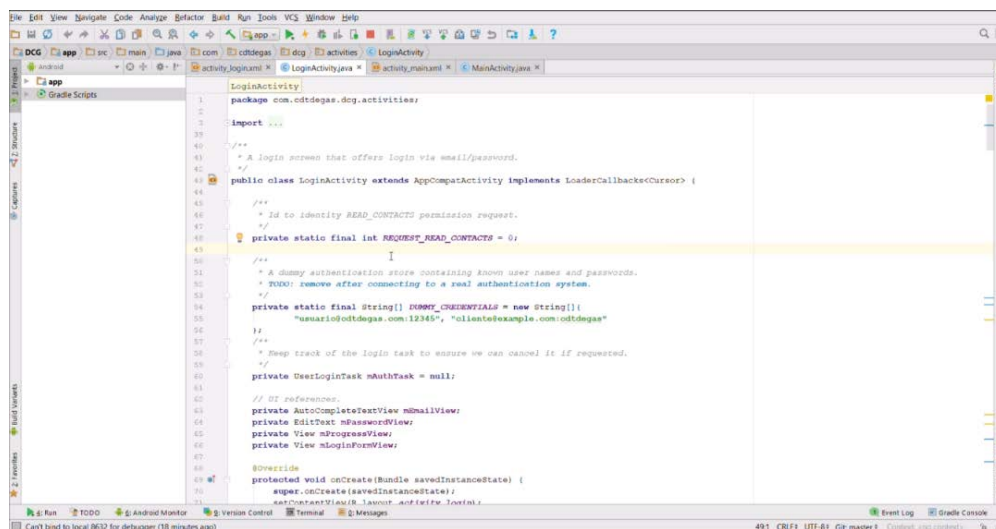


Figura 56. Programación en lenguaje JAVA para la funcionalidad de la App

Layout o esquema grafico de la pantalla asociada a la actividad principal

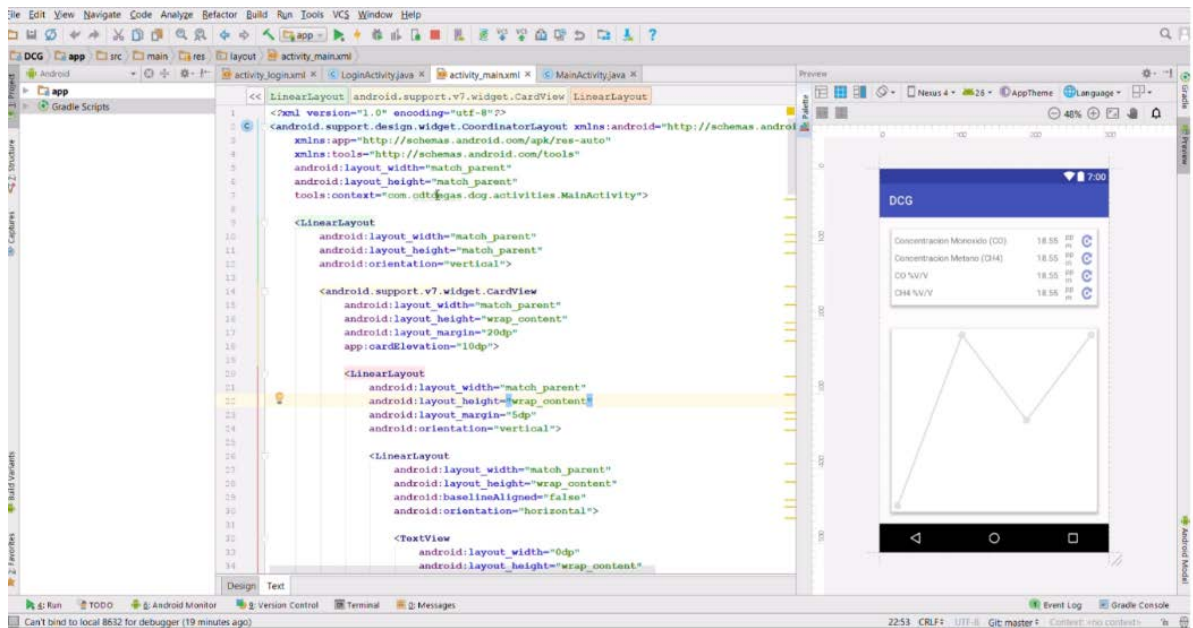
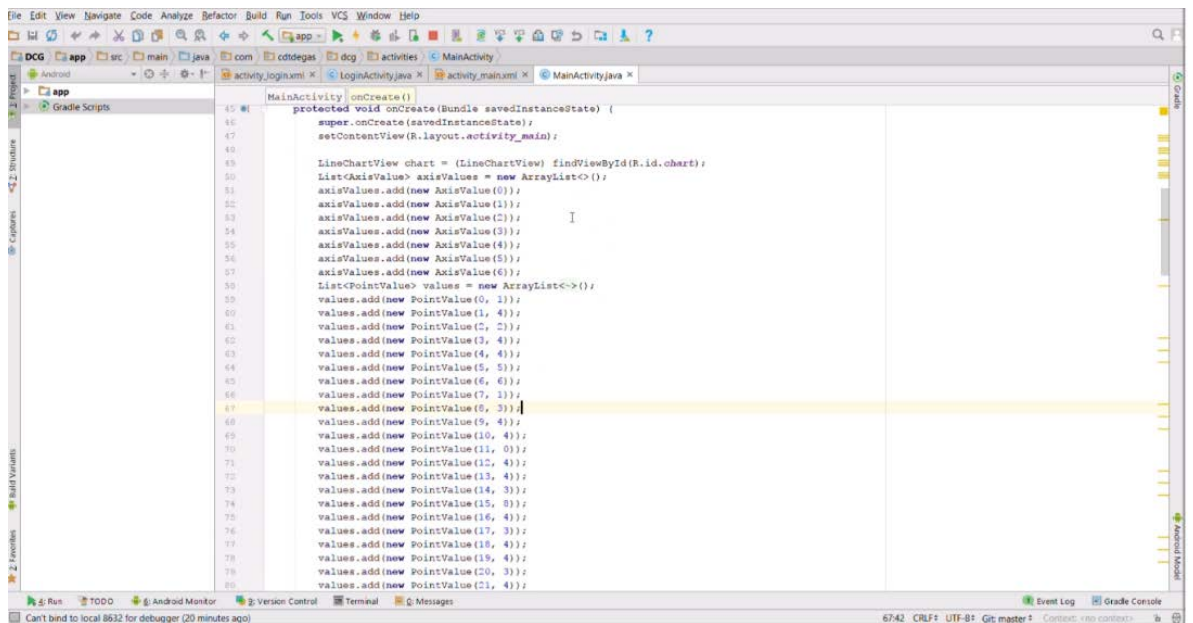


Figura 57. Diseño para la actividad principal de visualización de concentración de gas





## 6. INSTRUCCIONES PARA UTILIZAR LA APLICACIÓN

1. Se realiza verificación de que el Smartphone cuenta con el protocolo de comunicación Bluetooth y se exige que este encendido para acceder a la aplicación. Imagen [#]

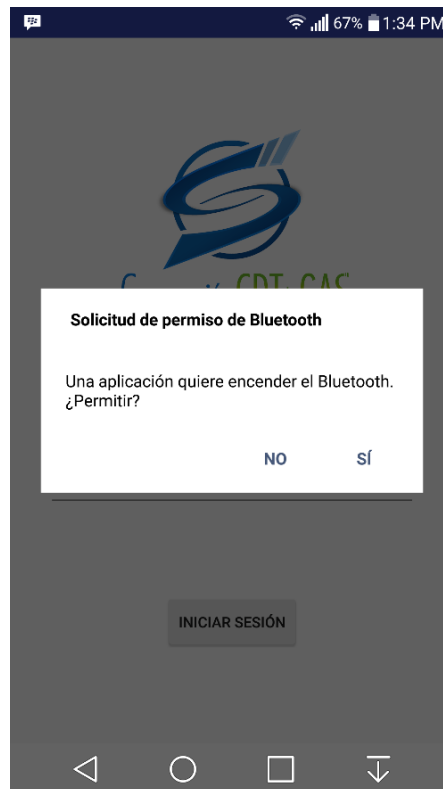


Figura 59. Permiso para acceder a Bluetooth

2. Al ingresar se verá una pantalla como en la imagen [60] la cual brinda acceso de administrador o a un usuario para utilizar la app
  - El ingreso de administrador permite crear usuarios nuevos Imagen [61] y añadir o editar sensores Imágenes [#, #] directamente sin necesidad de ir al entorno de programación.
  - Una vez creado el usuario y al menos ingresados los parámetros de un sensor permite la utilización de la app.



Figura 60. Pantalla de ingreso o bienvenida

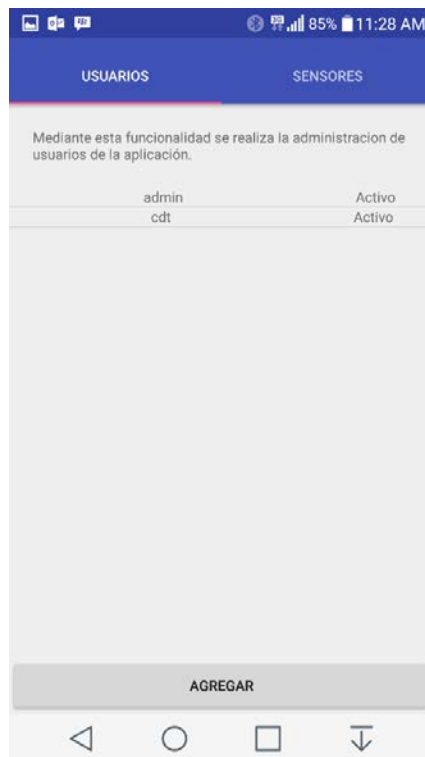


Figura 61. Acceso Administrador creación usuarios

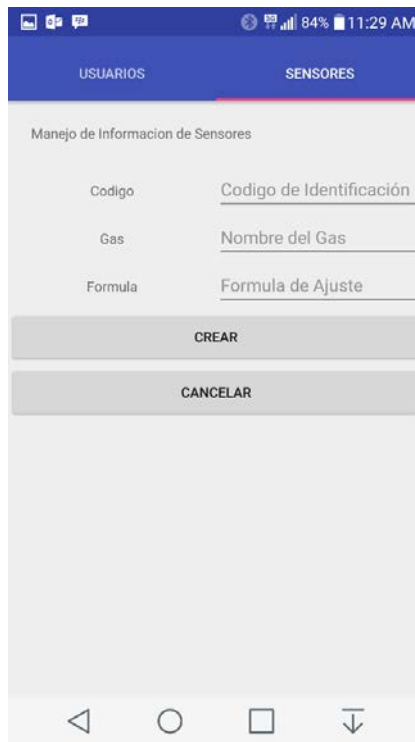


Figura 62. Añadir Sensor nuevo o modificar



Figura 63. Sensores almacenados activos

3. Al ingresar con un usuario, este será direccionado a la lista de dispositivos Bluetooth emparejados con su Smartphone Imagen [#], en el cual debe seleccionar el módulo HC06 Imagen [#] que es el encargado de enviar los datos desde el arduino a la app.

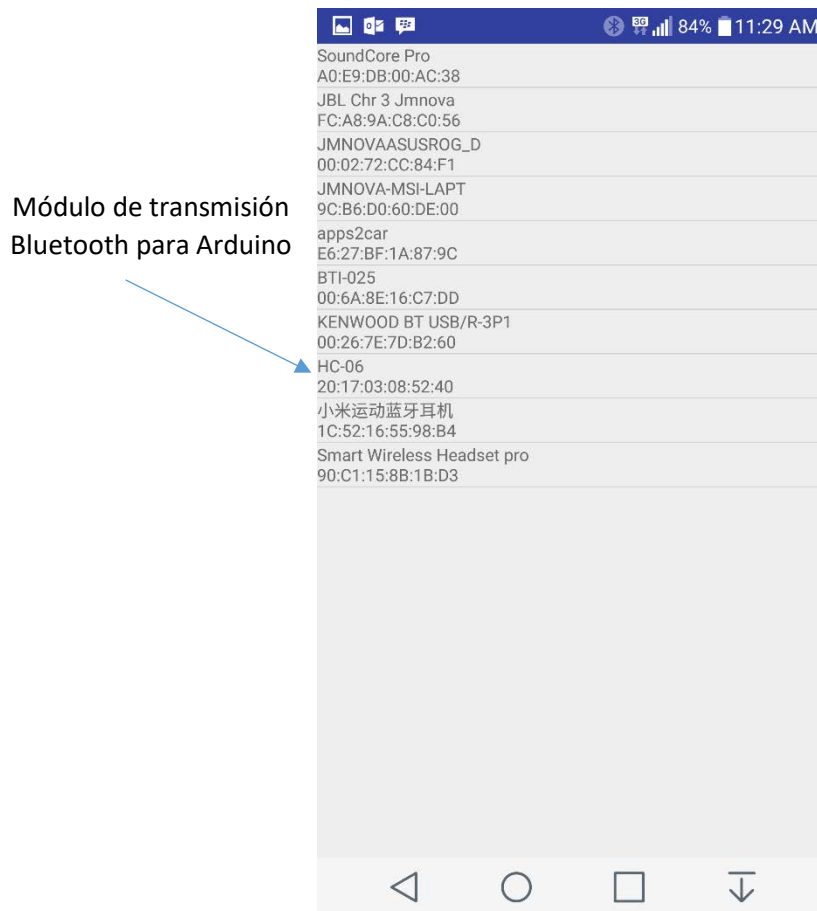


Figura 64. Selección del dispositivo Bluetooth

4. Al ingresar por primera vez se abre la pantalla de coeficientes de calibración Imagen Figura 58, estos son de uso exclusivo del CDT ya que sirven para ajustar la ecuación del sensor corrigiendo posibles diferencias a nivel de hardware porque la idea es masificar el uso del dispositivo.  
Estas diferencias pueden ser propias de un sensor a otro, diferencias en los elementos electrónicos debido a tolerancias propias dadas por los fabricantes o de un Smartphone a otro.  
Al dejar los coeficientes a, b, c en valor 1 se utiliza la ecuación de característica del sensor ingresada por el administrador

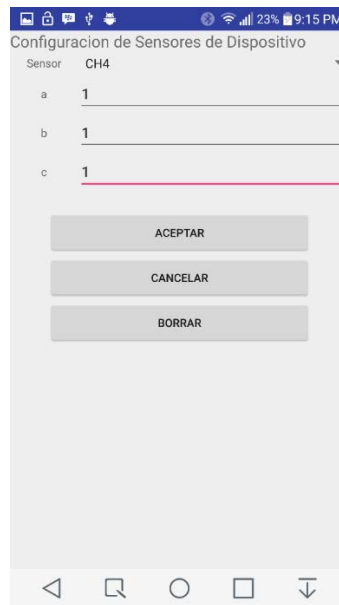


Figura 65. Coeficientes de Calibración

5. Ahora se muestra la pantalla de la actividad principal, esta corresponde a la gráfica de concentración de gas en el ambiente, se observan los cambios de concentración con unidad de %mol en el tiempo y la tasa de muestreo está fijada a 1 segundo.

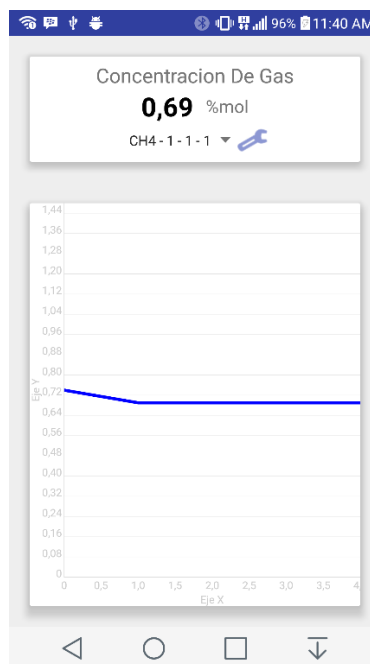


Figura 66. Concentración de Gas en el tiempo en estado estable

## CONCLUSIONES

- La tarjeta Arduino Pro mini implementada cumplió con los requerimientos propuestos, facilitando la integración de los diferentes módulos debido a su bajo consumo energético, facilidad de programación y reducido tamaño.
- La integración del sensor con el hardware diseñado permitió medir la concentración de gas, proporcionando una indicación estable con respecto a un instrumento de referencia.
- Se evidencio que el entorno de desarrollo Android Studio, es la mejor opción para desarrollar aplicaciones sobre el sistema operativo Android debido a su excelente compatibilidad, facilidad de actualización, sincronización, soporte para emulador y facilidad de uso.

## OBSERVACIONES Y/O RECOMENDACIONES

- Teniendo en cuenta que la siguiente fase del proyecto es construcción del dispositivo este debe cumplir con bajo consumo energético, ya que operara con batería y se desea lograr una buena autonomía de funcionamiento.
- Mayor soporte de la empresa y el supervisor al estudiante para la correcta consecución de los objetivos propuestos, brindando los equipos necesarios para ello (Debido a que hubo demasiados retrasos para el préstamo de equipo necesario para el proyecto).
- Cambiar el modulo Bluetooth por uno que cumpla con estándares BLE (Bluetooth low energy)
- Tener especial cuidado en el diseño del circuito de alimentación con batería, porque se observó en las pruebas que el circuito es dependiente del voltaje de alimentación, por lo tanto, si hay variación en esta etapa se introduce un offset al cero del sensor que generara lecturas erróneas

## REFERENCIAS BIBLIOGRAFICAS

- [1] Corporación CDT de gas [Enero 2018] Disponible en: [http://www.cdtdegas.com/images/Descargas/Nuestra\\_revista/MetFlu1/ejemplarN001\\_7CD\\_T\\_GAS.pdf](http://www.cdtdegas.com/images/Descargas/Nuestra_revista/MetFlu1/ejemplarN001_7CD_T_GAS.pdf)
- [2] Corporación CDT de gas. Quienes somos [Enero 2018] Disponible en: <http://www.cdtdegas.com/index.php/nosotros/quienes-somos>
- [3] Fergarciac. Entorno de Desarrollo Integrado (IDE). [Junio 2017] Disponible en: <https://fergarciaac.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>
- [4] Eclipse IDE [Octubre 2017] Disponible en: <https://www.genbetadev.com/herramientas/eclipse-ide>
- [5] Installing the Eclipse Plugin [Noviembre 2017] Disponible en: <https://stuff.mit.edu/afs/sipb/project/android/docs/sdk/installing/installing-adt.html>
- [6] Academia Android. Android Studio v1.0: características y comparativa con Eclipse [Julio 2017] Disponible en: <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
- [7] Android Studio [Julio 2017] Disponible en: <https://developer.android.com/studio/intro/index.html>
- [8] Guía Breve de Tecnologías XML [Agosto 2017] Disponible en: <https://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>
- [9] Dalemans Gas Detection [Agosto 2017] Disponible en: <http://www.dalemans.com/es/introduction/>
- [10] Principios detección de Gas Disponible en: <https://www.ucm.es/data/cont/docs/3-2015-03-16-Principios%20de%20detecci%C3%B3n.pdf>
- [11] Winsen [Septiembre 2017] Disponible en: <http://www.winsen-sensor.com/products/catalytic-gas-sensor-for-industrial-application/mc113.html>
- [12] Detector de gases con arduino y la familia de sensores MQ. [Febrero 2018] Disponible en: <https://www.luisllamas.es/arduino-detector-gas-mq/>
- [13] Technical Data Mq-7 Gas Sensor [Enero 2018] Disponible en: <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>



[14] Electrónicos Caldas [Enero 2018] Disponible en: <http://www.electronicoscaldas.com/boards/744-sistema-de-desarrollo-arduino-pro-mini-328-5v-clone.html>

[15] Prezi. Cuál es la diferencia entre Wi-Fi y Bluetooth [Septiembre 2017] Disponible en: <https://prezi.com/bq230-nsnhmp/cual-es-la-diferencia-entre-wi-fi-y-bluetooth/>

[16] Naylamp Mechatronics. Tutorial básico de uso del módulo Bluetooth Hc-06 y hc-05. Disponible en: <http://www.naylampmechatronics.com/blog/12-Tutorial-B%C3%A1sico-de-Usode-M%C3%B3dulo-Bluetooth-H.html>

[17] PrevenBlog. Una app que convierte el móvil en un detector de gases [Octubre 2017] Disponible en: <http://prevenblog.com/te-interesaria-una-app-para-movil-que-detecta-humo-y-monoxido-de-carbono/>

[18] Nest Protect, el detector de humos y monóxido de carbono del creador del termostato Nest [Octubre 2017] Disponible en: <https://www.xatakahome.com/electrodomesticos-innovadores/nest-protect-el-detector-de-humos-y-monoxido-de-carbono-del-creador-del-termostato-nest>

[19] Nest protect [Noviembre 2017] Disponible en: <https://nest.com/es/smoke-co-alarm/tech-specs/>

[20] EETimes [Noviembre 2017] Disponible en: [http://www.eetimes.com/document.asp?doc\\_id=1328199](http://www.eetimes.com/document.asp?doc_id=1328199)

## **ANEXOS**

**ANEXO A CODIGO DE PROGRAMACION DE LA APLICACIÓN  
EN ANDROID STUDIO**

**ANEXO B CODIGO DE PROGRAMACION ARDUINO**