

**TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE NUEVO LEÓN**



**TESIS**

**ANÁLISIS COMPARATIVO PARA LA INTEGRACIÓN Y MONITOREO  
DE DATOS ENTRE UN SISTEMA DE TIEMPO REAL Y UN SISTEMA  
DE INFORMACIÓN.**

**PARA OBTENER EL GRADO DE**

**INGENIERO EN SISTEMAS COMPUTACIONALES**

**PRESENTA**

**ANDRES GONZALEZ RODRIGUEZ**

**NO. DE CONTROL 14480653**

**DIRECTOR DE TESIS**

**M.C. JOSÉ ISIDRO HERNÁNDEZ VEGA**

**CO-DIRECTORA DE TESIS**

**M.C. DOLORES GABRIELA PALOMARES GORHAM**

**Cd. Guadalupe, Nuevo León, México, Enero de 2019**

SEP

SECRETARÍA DE  
EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO  
Instituto Tecnológico de Nuevo León

## División de Estudios Profesionales Ingeniería en Sistemas Computacionales

TESIS DEFENDIDA POR  
**ANDRES GONZALEZ RODRIGUEZ**

Y APROBADA POR EL SIGUIENTE COMITÉ

---

**M.C José Isidro Hernández Vega**

*Director del Comité*

---

**M.C. Dolores Gabriela Palomares  
Gorham**

*Miembro del Comité*

---

**M.C. Elda Reyes Varela**

*Miembro del Comité*

---

**Ing. Magaly Benítez Tamez**  
*Jefa del Departamento de Sistemas y  
Computación*



**RESUMEN** del proyecto de tesis de **ANDRES GONZALEZ RODRIGUEZ** presentada como requisito parcial para la obtención del grado de INGENIERO EN SISTEMAS COMPUTACIONALES. Guadalupe, Nuevo León. Enero de 2019.

## **TITULO DE TESIS**

**ANÁLISIS COMPARATIVO PARA LA INTEGRACIÓN Y MONITOREO  
DE DATOS ENTRE UN SISTEMA DE TIEMPO REAL Y UN SISTEMA DE  
INFORMACION**

Resumen aprobado por:

---

M.C. JOSE ISIDRO HERNANDEZ VEGA

## **RESUMEN**

En este proyecto de tesis se presenta el desarrollo de una base de datos integrada a un sistema en tiempo real para el monitoreo de contaminantes criterio. Se diseña la base de datos y se evalúa con respecto a un sistema para el procesamiento de transacciones desde el punto de vista del diseño de base de datos.

El desarrollo de la base de datos se integra a un sistema web que interactúa en tiempo real con los datos recibidos desde los sensores integrados a un Vehículo Aéreo no Tripulado (VANT).

Con la aplicación desarrollada se realizan pruebas experimentales del monitoreo de contaminantes, se comparan dos gestores de base de datos para conocer su eficiencia y se evalúan los resultados obtenidos en las pruebas.

Se exploran los conceptos del desarrollo de Base de Datos, y se realiza una comparación con respecto a los sistemas transacciones y el sistema de monitoreo de contaminantes desarrollado, en el contexto de base de datos.

Se presentan resultados de la importancia del diseño de la base de datos tanto en tiempo real y la de procesamiento de transacciones.

### **Palabras Clave:**

VANT, contaminantes criterio, Base de Datos, Sistemas en Tiempo Real, Base de Datos para el procesamiento de transacciones.

Abstract of the thesis presented by **ANDRES GONZALEZ RODRIGUEZ** as a partial requirement to obtain the Engineering in Computer System. Guadalupe, Nuevo Leon, Mexico, January 2019.

**TITLE OF THESIS**

**COMPARATIVE ANALYSIS FOR INTEGRATION AND MONITORING  
OF DATA BETWEEN A REAL TIME SYSTEM AND A SYSTEM  
OF INFORMATION**

Abstract approved by:

---

M.C. JOSE ISIDRO HERNANDEZ VEGA

**ABSTRACT**

This thesis project presents the development of an integrated database to a real-time system for monitoring criteria pollutants. The database is designed and evaluated compared to a system for transactions processing from the point of view of the database design.

The database development integrates a web system that interacts in real time with the data received from the integrated sensors on an Unmanned Aerial Vehicle (UAV).

With the developed application, experimental tests of contaminant monitoring are carried out, two database managers are compared to find out their efficiency and the results obtained are evaluated.

The concepts of database development are explored, and a comparison is made between the systems for transaction processing and the contaminant monitoring system developed in the context of databases.

The obtained results demonstrate the importance of the database design in a real time system and in a transactional system as well.

**Key Words:** VANT, criteria pollutants, Database, real-time systems, transactional system.

## **DEDICATORIAS**

Le dedico esta tesis a mi familia que me han apoyado en esta Tesis de forma económica, así como emocional.

## **AGRADECIMIENTOS**

Al Instituto Tecnológico de Nuevo León que me ha permitido realizar mis estudios de licenciatura, así como utilizar su instalación para el desarrollo de mi proyecto. Al Programa de Desarrollo Profesional Docente (PRODEP) por la beca otorgada al incorpórame al proyecto de investigación: Monitoreo de contaminantes en el aire utilizando un sistema mecatrónica inteligente para plumas industriales en el área metropolitana de Monterrey, aprobado en la Convocatoria 2017 para el Fortalecimiento del Cuerpos Académico: Tecnologías Mecatrónicas Sustentables e Inteligentes.

Quisiera agradecerle Ing. Magaly Benítez Tamez por ayudarnos con los trámites de la tesis, así como dando el seguimiento a lo largo de esta para su cumplimiento.

Al M.C. José Isidro Hernández Vega el cual nos estuvo coordinando para poder realizar el proyecto como dándonos las herramientas necesarias para entender el funcionamiento de las VANT como su ayuda en el desarrollo del proyecto.

A la Co-Directora M.C. Dolores Gabriela Palomares Gorham por brindar su apoyo como asesor en el proyecto de tesis,

A mis compañeros P.T.B. Mario Humberto Uriegas De León, Rodrigo Aldahyr G. Marroquín Rodríguez, P.T.B Migue Ángel Nava Martínez por su ayuda durante el proyecto de tesis donde esto no sería posible sin el trabajo en equipo durante estos años.



## CONTENIDO

### Contenido

RESUMEN .....	III
ABSTRACT .....	V
DEDICATORIAS .....	VI
AGRADECIMIENTOS .....	VII
CONTENIDO.....	VIII
LISTA DE FIGURAS .....	X
LISTA DE TABLAS .....	XI
INTRODUCCIÓN .....	1
1.1 Motivación del problema.....	2
1.2 Planteamiento del problema.....	3
1.3. Definición del problema .....	4
1.4. Justificación .....	4
1.5 Objetivos.....	5
1.5.1 Objetivo general.....	5
1.5.2 Objetivos particulares.....	5
1.6 Hipótesis.....	6
1.7 Alcances y limitaciones.....	7
1.8 Estructura de la tesis .....	8
MARCO TEÓRICO .....	9
2.1 Antecedentes.....	9
2.2 Conceptos Relacionados.....	12
2.2.1 Base de datos. ....	12
2.2.1.1 SQL.....	13
2.2.1.2 NoSQL .....	14
2.2.1.3 PostgreSQL .....	15
2.2.1.4 ECOLEX .....	16
2.6 Trabajos relacionados .....	17
METODOS Y PROCEDIMIENTOS.....	19
3.1 Procedimiento para el desarrollo del sistema de tiempo real para contaminantes. ....	19

3.2 Desarrollo .....	30
3.2.1 VANTS.....	30
3.3 Dificultades .....	34
3.4 Diseño de la base de datos .....	36
3.5 Material y equipo utilizado (SW) .....	37
Dron .....	37
Computadora de vuelo.....	37
GPS .....	38
Antena.....	38
Batería .....	39
Motores.....	39
Radio.....	40
Estación de tierra .....	40
Raspberri pi 3.....	41
Dronekit.....	42
Nodejs.....	42
Postgresql .....	42
SublimeText .....	42
Mission Planner .....	42
3.6 Consideraciones para el diseño .....	44
Resultados .....	45
4.1 Experimentación del prototipo en toma de muestras .....	50
4.2 Análisis de resultados .....	52
4.3. Análisis comparativo entre Bases de Datos para sistemas en tiempo real. ...	52
4.3.1 Otras pruebas con B.D.'s.....	53
Conclusiones.....	56
5.1 Conclusiones.....	56
5.2 Recomendaciones .....	57
5.3 Trabajos futuros .....	58
6 REFERENCIAS.....	59
ANEXOS .....	61

## LISTA DE FIGURAS

Figura 1. Base de datos en la nube. ....	10
Figura 2. Base de datos. ....	12
Figura 3. SQL. ....	13
Figura 4. NoSQL. ....	14
Figura 5. PostgretSQL. ....	15
Figura 6. Ecolex. ....	16
Figura 7. New Login. ....	19
Figura 8. Nombre B.D. ....	19
Figura 9. Definición. ....	20
Figura 10. Roles. ....	20
Figura 11. New data. ....	21
Figura 12. Nombre New data. ....	21
Figura 13. dentro de la B.D. ....	21
Figura 14. New Table. ....	22
Figura 15. Nombre New table. ....	22
Figura 16. New column. ....	23
Figura 17. Declarar Variables. ....	23
Figura 18. Todas las Variables. ....	24
Figura 19. New Primary Key. ....	24
Figura 20. Primary key. ....	24
Figura 21. Definición de la primary key. ....	25
Figura 22. Primary key en la tabla. ....	25
Figura 23. Datos de la Base de datos diseñada. ....	26
Figura 24. Hexacoptero. ....	30
Figura 25. Cuadricoptero. ....	31
Figura 26. Base de los sensores. ....	31
Figura 27. Sensor MQ. ....	32
Figura 28. Diagrama. ....	32
Figura 29. Programas necesarios. ....	34
Figura 30. Diagrama de la Base de datos. ....	36
Figura 31. Dron de cuatro hélices. Cuadricoptero. ....	37

Figura 32. Computadora de vuelo Pixhawk. ....	37
Figura 33. GPS. ....	38
Figura 34. Antenas de Telemetría. ....	38
Figura 35. Batería de Litio.....	39
Figura 36. Motores para los drones. ....	39
Figura 37. Radiocontrol para operar el dron. ....	40
Figura 38. Una laptop que hace función de estación de tierra. ....	40
Figura 39. Placa raspberri pi 3 versión B. ....	41
Figura 40. Símbolo de distribución Rabian Jessie de Linux.....	41
Figura 41. Pantalla de inicio del programa.....	43
Figura 42. Base de datos en tiempo real. ....	45
Figura 43. Base de datos tradicional.....	45
Figura 44. Radio control.....	50
Figura 45. Dron en tierra.....	51
Figura 46. Dron en vuelo. ....	51
Figura 47. Datos almacenados en la tabla “tbldata” de la base de datos diseñada. ....	52
Figura 48. Conteo de registros con PostgreSQL. ....	53
Figura 49. Conteo de registros con MySQL.....	53
Figura 50. PostgreSQL count. ....	54
Figura 51. MySQL count. ....	54
Figura 52. PostgreSQL consulta de una palabra aleatoria. ....	55
Figura 53. MySQL consulta de una palabra aleatoria.....	55
Figura 54. Aplicación móvil. ....	58
Figura 55. Photon. ....	58

## LISTA DE TABLAS

Tabla 1 Diccionario de datos diseñado .....	27
Tabla 2 Diseño de archivos para la base de datos.....	46
Tabla 3 Interacción en la base de datos.....	48

## Capítulo I

---

### INTRODUCCIÓN

---

A lo largo de la historia siempre ha sido importante la información, debido a esta necesidad se buscó desarrollar una forma para poder almacenar estos datos y que posteriormente se tuviera fácil acceso a ellos para visualizarlos o manipularlos.

Las primeras formas en que se llegaban a almacenar datos eran en medios físicos (por ejemplo, en papel, tablas, entre otras) donde estos tenían una gran cantidad de limitaciones teniendo como resultado un pobre manejo de los datos, así como demasiado tiempo de respuesta en la búsqueda de información. Conforme el paso del tiempo y el avance de la tecnología se fueron implementando mejoras dentro de su manejo, así como mejorar su capacidad de almacenamiento agregando distintas formas de consultar y/o registrar la información.

Los gestores de bases de datos dan un paso más adelante, permitiendo ser manipulados, visualizados, almacenados o eliminados de acuerdo al uso que estos tengan. Las bases de datos (B.D) pueden ser utilizadas en una gran variedad de campos como en los negocios, la educación, el entretenimiento, los deportes, entre otros.

En los siguientes apartados se describen las actividades realizadas para la creación, manipulación, visualización y comunicación de una base de datos con el resto de las aplicaciones. La base de datos se crea mediante la comunicación entre un vehículo aéreo no tripulado (VANT) y la estación de tierra(GCS), los cuales proporcionan los datos a manipular y se realiza el comparativo entre el desarrollo con un sistema de información y un sistema en tiempo real, obteniendo como conclusiones cuadros comparativos con ventajas y desventajas.

### **1.1 Motivación del problema**

El procesamiento de datos en sistemas computacionales es de suma importancia, ya que con los datos se generan reportes, monitoreo de variables e incluso la toma de decisiones.

En los sistemas de tiempo real se tienen problemas para manipular los datos por la falta de herramientas adecuadas, lo cual requiere de realización de aplicaciones flexibles y de fácil manejo tanto para el programador como para el usuario final.

En el contexto del proyecto de investigación trabajado se requirió de un sistema de información en tiempo real para el monitoreo de contaminantes, su base de datos se diseñó de manera flexible para el almacenamiento e interactuar con variables del medio ambiente en un tiempo de respuesta rápido, situación que los sistemas de información no consideran como prioritario este requisito.

## **1.2 Planteamiento del problema**

El monitoreo de contaminantes por medio de sensores es manejado gracias a las estaciones fijas proporcionadas por el SIMA [Sistema Integral de Monitoreo Ambiental] en Nuevo León desde el año de 1992 y hoy en día proporciona los datos de calidad del aire por medio de su página web en la que la información es presentada por medio de gráficas e imágenes que son fáciles de interpretar por el usuario.

Un punto en contra en cuanto a las estaciones de monitoreo es que se encuentran fijas en distintos puntos de la ciudad/estado y la información proporcionada puede ser solo relevante según las zonas donde se encuentran instaladas.

El hecho que el monitoreo solo se haga a partir de estaciones fijas puede ocasionar que la información que se muestra por medio de una página web no sea tan relevante y que no abarque todas las zonas dentro de un área determinada. Tener un sistema de monitoreo de contaminantes móviles podría en primer lugar abarcar mayores áreas en las cuales monitorear las condiciones de la zonas, en segundo ver que contaminantes predominan y por último ver qué soluciones se pueden ofrecer para reducir la contaminación según los resultados obtenidos.

El avance de la tecnología aporta a que existan herramientas en ocasiones más económicas que puedan presentar resultados concretos en temas específicos. El uso de la tecnología en cuanto al monitoreo de los contaminantes puede hacer que en un área más grande se puedan obtener resultados más concisos y estos puedan ser presentados gráficamente al público en general.

### **1.3. Definición del problema**

La falta de una Base de Datos para el almacenamiento de los valores de los contaminantes criterio, provoca falta de tratamiento de los datos para su consulta y análisis en el monitoreo de contaminantes

### **1.4. Justificación**

No se contaba con un medio donde se pudieran almacenar los datos, orientado a las necesidades del proyecto de monitoreo de contaminantes.

El uso de una B.D. es de gran importancia para el almacenamiento de la información donde esos datos sean posteriormente manipulados para su uso en proyectos de control del medio ambiente. De acuerdo a los datos obtenidos de la VANT se pueden realizar diferentes operaciones para posteriormente tomar acciones de acuerdo con los datos que serán almacenados, estos pueden ser representados por diferentes medios desde forma gráfica así como por medio de una aplicación web.

Gracias a la implementación de la B.D. en tiempo real, los datos pueden ser manejados oportunamente conforme a las necesidades que de estos se tengan en éste o en futuros proyectos relacionados con contaminantes del medio ambiente.



## **1.5 Objetivos**

### **1.5.1 Objetivo general**

Desarrollar una aplicación de software que permita procesar los datos enviados por radiofrecuencia entre un VANT y una estación de tierra, con la finalidad de manipularlos y almacenarlos para su explotación para el proyecto monitoreo de contaminantes, analizando las ventajas sobre el desarrollo de aplicaciones de un sistema de información tradicional para el procesamiento de transacciones.

### **1.5.2 Objetivos particulares**

- Analizar cómo es la integración de datos en un sistema de tiempo real y un sistema de información.
- Desarrollar una aplicación para el almacenamiento de datos en una B.D.
- Creación de tablas con una serie de campos donde se determinarán los datos recibidos.
- Implementar la aplicación para el registro y almacenamiento de los datos
- Documentar los procesos detalladamente.

## **1.6 Hipótesis**

Comparando un sistema de información en tiempo real con un sistema de información tradicional, de almacenamiento de datos, se contará con criterios para el desarrollo de software entre estos dos tipos de aplicaciones.

### **1.7 Alcances y limitaciones**

El alcance que se buscó de este proyecto de Tesis fue la comunicación de una aplicación que almacene información obtenida por los vuelos de un VANT integrado con sensores MQ en una B.D.

Con la base de datos desarrollada tomar en cuenta los requisitos que se deben de considerar en sistemas de tiempo real.

Realizar un comparativo entre un sistema de información tradicional y un sistema de información desarrollado para el proyecto.

Limitaciones para la ejecución de la base de datos desarrollada:

- El límite de registros dependiendo del Sistema Gestor de Base de Datos.
- El programa desarrollado para la comunicación y almacenamiento de los datos solamente puede ser ejecutado en una versión en específico de la B.D.
- El programa desarrollado para la comunicación y almacenamiento de los datos solamente puede ser ejecutado en una versión en específico del Node.Js.
- Velocidad del almacenamiento de la información.
- Velocidad de procesamiento de la GCS para la recepción de datos

### **1.8 Estructura de la tesis**

En el capítulo 1 encontrarán los temas que dieron el porqué del proyecto sus objetivos y planteamiento del problema.

En el capítulo 2 encontrará todos los antecedentes con los que se contaron para la realización de este proyecto.

En el capítulo 3 encontrará la descripción de la metodología que se llevó a cabo para la realización del proyecto.

En el capítulo 4 se muestran los resultados alcanzados en las comparaciones entre el sistema de tiempo real y el sistema de información que se comparó.

Finalmente en el capítulo 5 se encuentran las conclusiones de este trabajo de tesis.

## Capítulo II

---

### MARCO TEÓRICO

---

#### 2.1 Antecedentes.

##### **En la década del 1970**

Edgar Frank Codd definió el modelo relacional y publicó una serie de reglas para la evaluación de administradores de sistemas de datos relacionales y así nacieron las bases de datos relacionales.

A partir de los aportes de Codd el multimillonario Larry Ellison desarrolló la base de datos Oracle, la cual es un sistema de administración de Base de Datos, que se destaca por sus transacciones, estabilidad, escalabilidad y multiplataforma.

Inicialmente no se usó el Modelo Relacional debido a que tenía inconvenientes por el rendimiento, ya que no podían ser competitivas con las bases de datos Jerárquicas y de Red [1].

##### **En la década del 1980**

Las Bases de Datos Relacionales con su sistema de Tablas, Filas y Columnas, pudieron competir con las Bases de Datos Jerárquicas y de Red, ya que su nivel de programación era bajo y su uso muy sencillo.

En esta década el Modelo Relacional ha conseguido posicionarse en el mercado de las Bases de Datos. También en este tiempo se iniciaron grandes investigaciones, como los Sistemas de Gestión de Bases de Datos Orientadas a Objetos SGBDOO (System Management Object Oriented Databases) [1].

### **Principios década de los 1990**

Para la toma de decisiones se crea el lenguaje SQL (Structured Query Language), que es un lenguaje programado para consultas. El programa de alto nivel SQL es un lenguaje de consulta estructurado que analiza grandes cantidades de información, el cual permite especificar diversos tipos de operaciones frente a la misma información, a diferencia de las bases de datos de los 1980 que eran diseñadas para las aplicaciones de procesamiento de transacciones. Los grandes distribuidores de bases de datos incursionaron con la venta de bases de datos orientadas a objetos [1].

### **Finales de la década de los 1990**

El boom de esta década fue la aparición de la WWW “Word Wide Web” ya que por este medio se facilitaba la consulta de las bases de datos. Actualmente tienen una amplia capacidad de almacenamiento de información, también una de las ventajas es el servicio de siete días a la semana las veinticuatro horas del día, sin interrupciones a menos que haya planificaciones de mantenimiento de las plataformas o el software [1].



Figura 1. Base de datos en la nube.

## **Sistemas Transaccionales**

Es un tipo de sistema de información diseñado para recolectar, almacenar, modificar y recuperar todo tipo de información que es generada por las transacciones en una organización

Los sistemas transaccionales son uno de los muchos tipos de sistemas de información que existen, son especialmente usados como su nombre lo indica para realizar transacciones o transferencias de datos, información de una persona a otra, o dinero de una cuenta a otra.

La respuesta es rápida en este tipo de sistemas resulta crítico que exista un rendimiento elevado con tiempos de respuesta cortos.

Tiene basada su fiabilidad en los SPT (Sistema de procesamiento de transacciones), esto convierte en esencial la existencia procedimientos de copia de seguridad y de recuperación ante fallos correctamente diseñados.

Es inflexible un SPT requiere que todas las transacciones sean procesadas exactamente de la misma forma, independientemente del usuario, el cliente o la hora del día. Si los SPT fuesen flexibles, habría entonces demasiadas posibilidades de ejecutar operaciones no estándar [4].

## 2.2 Conceptos Relacionados

### 2.2.1 Base de datos.

Los orígenes de las bases de datos se remontan a la Antigüedad donde ya existían bibliotecas y toda clase de registros. Además, también se utilizaban para recoger información sobre las cosechas y censos. Sin embargo, su búsqueda era lenta y poco eficaz y no se contaba con la ayuda de máquinas que pudiesen reemplazar el trabajo manual.

Posteriormente, el uso de las bases de datos se desarrolló a partir de las necesidades de almacenar grandes cantidades de información o datos. Sobre todo, desde la aparición de las primeras computadoras, el concepto de bases de datos ha estado siempre ligado a la informática [10].



Figura 2. Base de datos.



### 2.2.1.1 SQL

El motor de base de datos proporciona un modelo relacional basado en el Lenguaje de consulta Estructurado (SQL) estándar, utilizándose fundamentalmente para ejecutar consultas de manipulación y definición de datos mediante los objetos correspondientes de registros de la base de datos, bien para diseñar o modificar directamente la estructura de una base de datos.

Los sistemas de Bases de Datos relacionales tradicionales soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas [5].



Figura 3. SQL.

### 2.2.1.2 NoSQL

Como alternativas del lenguaje SQL el NoSQL, Son un tipo de bases de datos estructurales que no usan como lenguaje de consulta SQL (aunque pueden soportarlo) como aspecto más destacado, ya que los datos no requieren de un almacenamiento estructurado fijo, como las tablas.

Un lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado [12].



Figura 4. NoSQL.

### 2.2.1.3 PostgreSQL

Es un manejador de bases de datos relacionales, de tipo cliente / servidor. Este ofrece una mezcla única de características que lo equiparan con los grandes manejadores de bases de datos. Una de las mayores ventajas que ofrece PostgreSQL es que su código es abierto ya que PostgreSQL no es propiedad de una sola compañía, sino que es mantenido, desarrollado y actualizado por un grupo de desarrolladores voluntarios alrededor del mundo.

Con el paquete de Microsoft de SQL Server para Node.js, las aplicaciones de servidor pueden conectarse a un servidor de SQL Azure directamente. Esto permite que podamos utilizar toda la potencia de un motor de base de datos relacionar durante el desarrollo de aplicaciones web basadas en Node.js

Se reciben datos recibidos atreves de radio frecuencia se parte la línea de datos donde son posteriormente separado en pequeños fragmentos y almacenados en variables para su posterior almacenamiento en la Base de datos [9].



Figura 5. PostgretSQL.

#### 2.2.1.4 ECOLEX

Es una base de datos exhaustiva sobre derecho ambiental administrada conjuntamente por la FAO, UICN y PNUMA. La base de datos ECOLEX incluye información sobre los tratados, derecho internacional y otros documentos de política no vinculante, documentos de orientación técnica, legislación nacional, decisiones judiciales y obras sobre derecho y políticas [8].



Figura 6. Ecolex.

## 2.6 Trabajos relacionados

### 1. Aplicación de los drones como herramienta tecnológica innovadora para la investigación de los sitios arqueológicos.

En el trabajo de Gavilanes Parra [6]. Se desarrolló una herramienta informática en los estudios arqueológicos. Por este medio podremos encontrar sitios sin la necesidad de correr riesgo. Los nuevos dispositivos vía (Wireless fidelity/fidelidad inalámbrica) WIFI o (sistema de posicionamiento global) GPS nos permite observar en tiempo real por conexión con el (vehículo aéreo no tripulado) VANT.

Una vez obtenida la información por medio de video y fotos se procede a editar con aplicaciones evitando falencia de las imágenes, estos pequeños errores pueden ser corregidos de manera profesional. En la actualidad existen bases de datos que para su requerimiento se puede hacer por programas o internet por medio de páginas web o Blogs.

Las diversas tecnologías han permitido al hombre adaptarse y organizarse desde sus inicios y aun en la actualidad, la base de datos ha cubierto una infinidad de distintas labores para acumular datos y para la investigación, estos son utilizados por el arqueólogo.

### 2. Las bases de datos geoespaciales en el análisis para la reducción de riesgo de desastres en áreas urbanas

En el trabajo de Benedetti [7]. Los especialistas de análisis de riesgo hacen un intenso uso de las geotecnologías, y para ello recurren a la información geográfica disponible.

La compleja dinámica de los espacios urbanos amerita una consideración especial al momento de modelizar la vulnerabilidad social y las amenazas para el análisis de riesgos de desastres.

Con este trabajo se propone una introducción metodológica para la construcción de las bases de datos geoespaciales que mejoren la modelización de la información geográfica para el análisis de riesgos a escala urbana.

Información Geoespacial es el conjunto de información que sirve de referencia y encuadre de contexto. Normalmente producida por organismos con funciones

específicas tales como el Instituto Geográfico Nacional, Catastros Municipales, etc.

El área de cobertura deberá ser motivo de análisis particular para definir los límites del entorno a representar, y los principales rasgos necesarios para su representación (límites, calles, parcelario, zonificaciones, distribución de la población, principales rasgo físicos, imagen de alta resolución, otros).

### **3. Diseño de un Modelo de Agricultura de Precisión Utilizando Drones y un Sistema de Captura, Almacenamiento y Análisis de Datos que Permita Identificar a Tiempo**

En el trabajo de Medina García y otros<sup>[13]</sup>. Proponen la implementación de un modelo de agricultura de precisión utilizando drones y un sistema de captura, almacenamiento y análisis de datos que permita identificar a tiempo la generación de la pudrición del cogollo en cultivos de palma de aceite que se desarrollan en los departamentos de Bolívar y Santander.

Drone Services Colombia: Empresa Colombiana cuya especialidad está en determinar el Índice de Vegetación de Diferencia Normalizada (NDVI).

Cuentan con drones equipados de un sistema de cámaras de última tecnología.

Los drones están en capacidad de observar las condiciones de los cultivos, además de monitorear aspectos relacionados con su crecimiento, nutrientes, identificación de deficiencias, variaciones, para cultivos. Se realiza el proceso de extracción de la información la información correspondiente será almacenada en la nube con el fin de que ésta se encuentre a disposición del cliente cuando sea necesario.

## Capítulo III

### METODOS Y PROCEDIMIENTOS

La metodología utilizada es la de espiral, dividiéndose las tareas durante todo el desarrollo del proyecto entre los diferentes participantes, realizando cada uno sus actividades de forma independiente. Finalmente el conjunto de los trabajos se unificaron en un sistema integral.

#### 3.1 Procedimiento para el desarrollo del sistema de tiempo real para contaminantes.

El procedimiento seguido fue: una vez instalado el sistema gestor de base de datos, en este caso el PostgreSQL, se debe inicializar de acuerdo a los pasos que se describen a continuación, habiéndose instalado por primera vez el programa. Esto se hace para definir el lugar de almacenamiento de los datos que se reciben del VANT.

Primero, se deben asignar los roles de los usuarios que podrán acceder a la base de datos. Para hacer esto se da clic derecho y asignar nuevo rol (New Login Role)

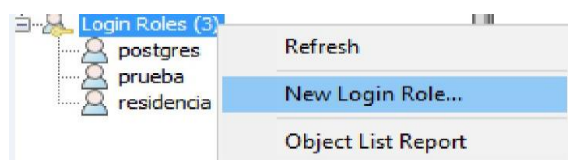


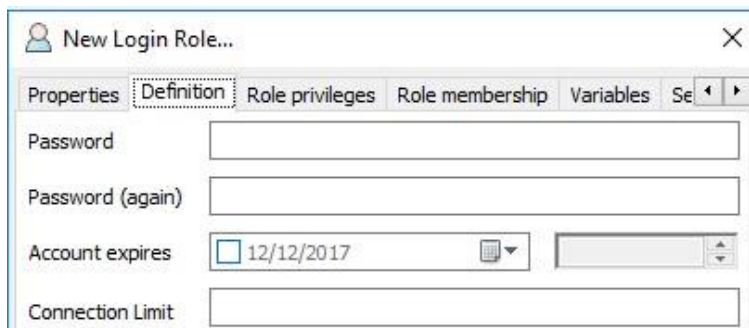
Figura 7. New Login.

Se desplegará la siguiente pantalla en la cual se declara el nombre del usuario que se va a asignar a la base de datos que se utilizará.



Figura 8. Nombre B.D.

Segundo, se define una contraseña mediante la cual el usuario correspondiente ingresa a la base de datos. Esta contraseña también funciona para el programa base “index” el cual llama a la base de datos.

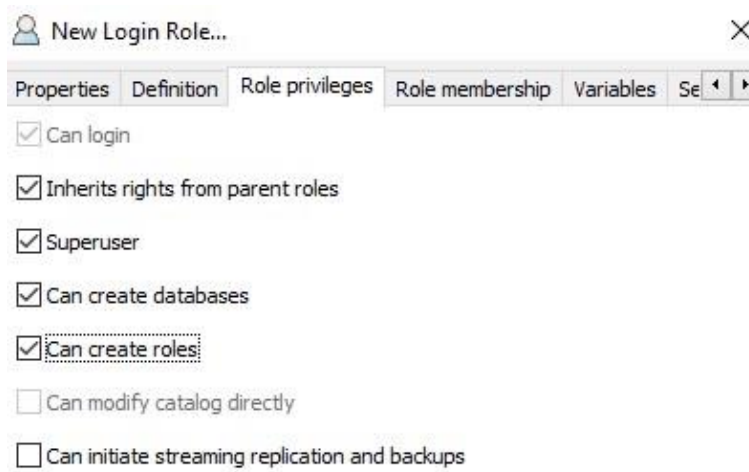


The screenshot shows a dialog box titled "New Login Role...". It has several tabs: "Properties", "Definition", "Role privileges", "Role membership", "Variables", and "Security". The "Definition" tab is active. It contains the following fields:

- Password: A text input field.
- Password (again): A text input field.
- Account expires: A date picker set to 12/12/2017.
- Connection Limit: A text input field.

Figura 9. Definición.

Tercero, al nuevo usuario se le asignan los permisos para que actúe como administrador en la base de datos. Estos permisos le dan la opción de crear bases de datos, asignar roles a usuarios y heredar los privilegios de los usuarios.



The screenshot shows the same dialog box, but with the "Role privileges" tab selected. It displays a list of permissions with checkboxes:

- Can login
- Inherits rights from parent roles
- Superuser
- Can create databases
- Can create roles
- Can modify catalog directly
- Can initiate streaming replication and backups

Figura 10. Roles.

Cuarto, se crea una base de datos específica dentro del sistema gestor. Esta nueva base de datos se asigna a un usuario definido (creado anteriormente). Para crear la nueva base de datos se da clic derecho en la pestaña de Base de Datos (Databases) y nueva Base de Datos (new Database).



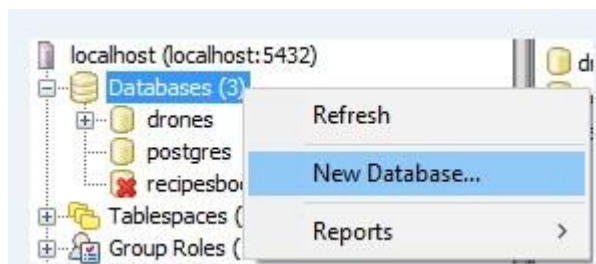


Figura 11. New data.

Quinto, se despliega una pantalla en la cual se van a llenar diferentes campos. En la primera pestaña se declara el nombre de la nueva base de datos y el usuario al que le pertenece.



Figura 12. Nombre New data.

Sexto, una vez creada la base de datos se pueden desplegar todas las características que posee. En la pestaña de Esquemas (Schemas) se encuentra la opción de tablas.

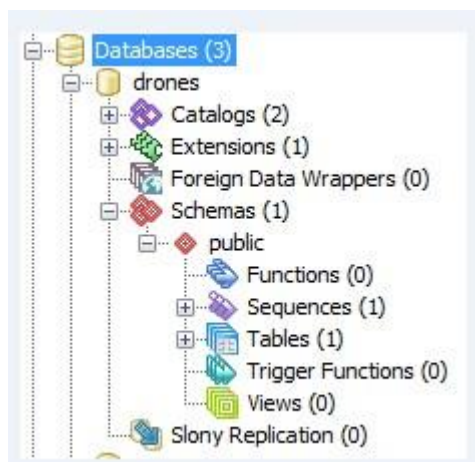


Figura 13. dentro de la B.D.

Séptimo, se crea la tabla. En la pestaña de tablas se da clic derecho con el puntero y aparece un menú con la opción de crear una nueva tabla (New Table).

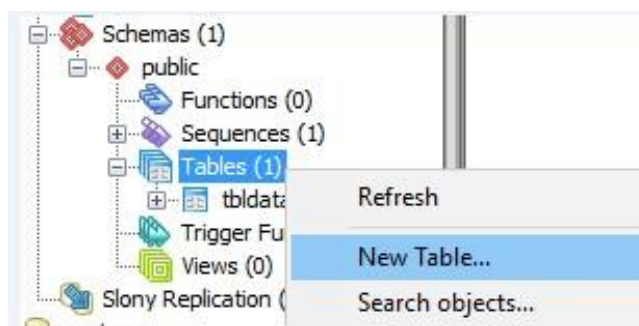


Figura 14. New Table.

Octavo, al darle clic se despliega una pantalla en el cual se declara el nombre de la tabla y a quien le pertenece. El nombre de la tabla (al igual que la contraseña) influye en el programa base y en la forma que éste tiene para hacer una conexión con la base de datos.

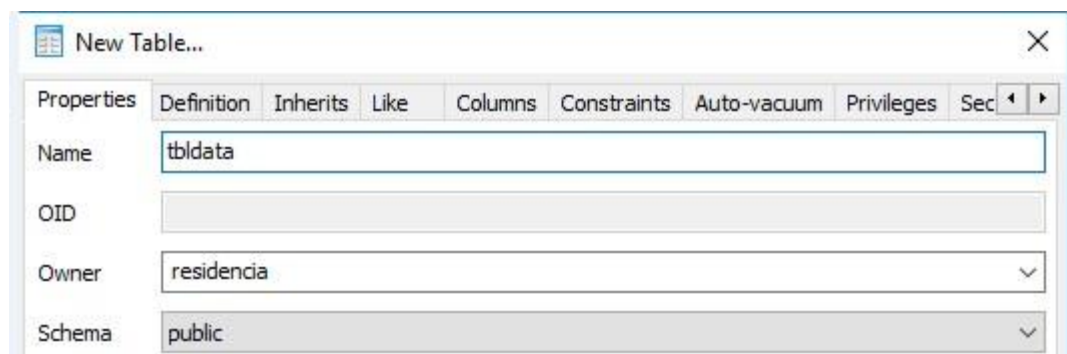


Figura 15. Nombre New table.

Noveno, una vez creada la tabla se definen las columnas de la tabla una a una con sus respectivos atributos.

Para la creación de las columnas, se da clic derecho en la tabla que se acaba de crear, se selecciona la opción de nuevo objeto (New Object) y luego nueva columna (New Column).

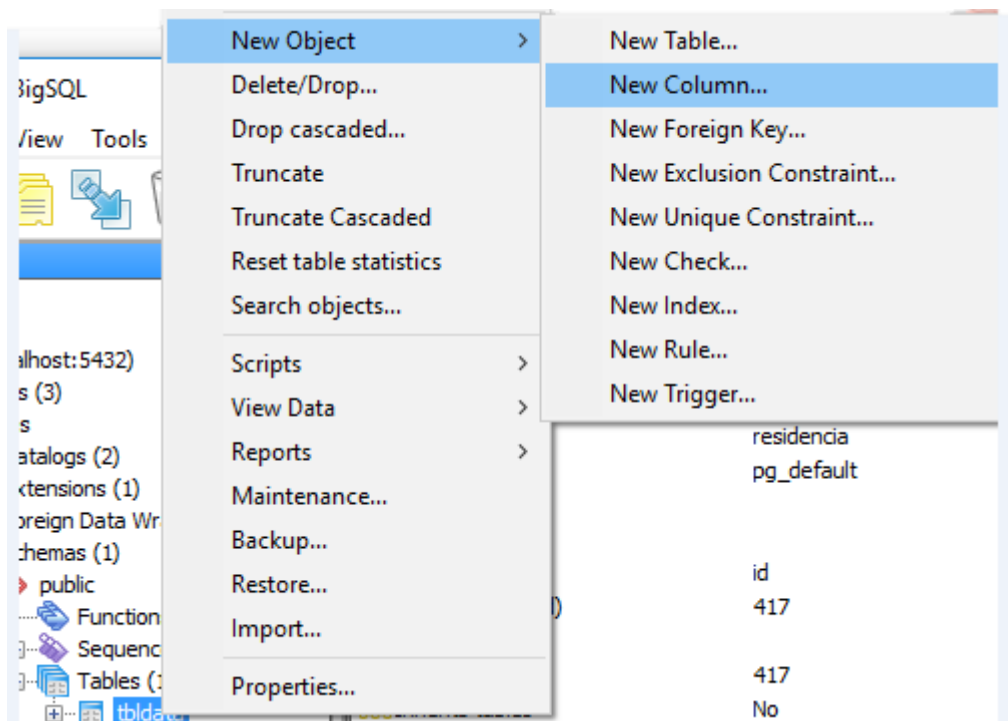


Figura 16. New column.

Décimo, se declara el nombre de la variable y el tipo de dato que es, en caso de que se requiera se debe señalar la longitud del tipo de dato.



Figura 17. Declarar Variables.

Onceavo, se repite el mismo procedimiento hasta tener todos los campos que se utilizarán con su respectivo tipo de dato.

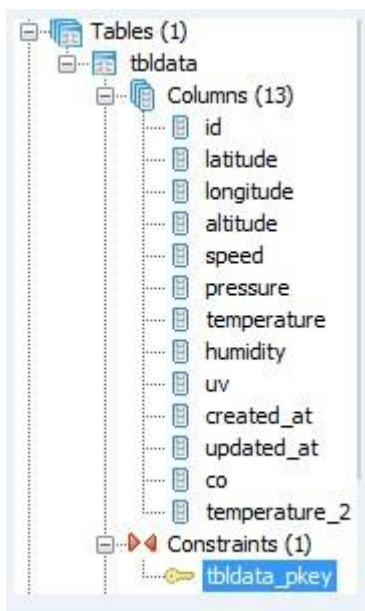


Figura 18. Todas las Variables.

Doceavo, para la creación de la clave primaria se da clic derecho dentro de la pestaña de restricciones (constraints), nuevo objeto y como no se tiene una llave primaria definida sale la opción para crearla, se da clic en ella (New Primary Key).

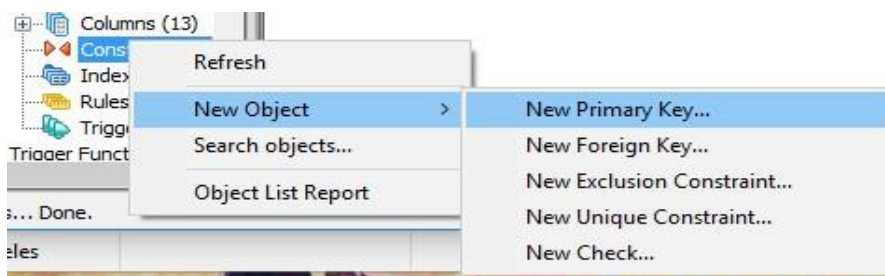


Figura 19. New Primary Key.

Treceavo, se le asigna el nombre que tendrá la llave primaria .

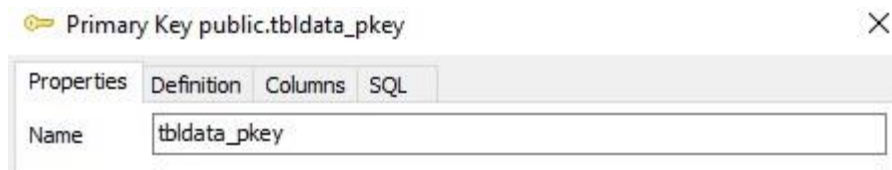


Figura 20. Primary key.

Catorceavo, en la definición de las características de la clave primaria se selecciona el espacio en que va a funcionar (pg\_default), ya que es donde quedan guardadas las bases de datos locales.

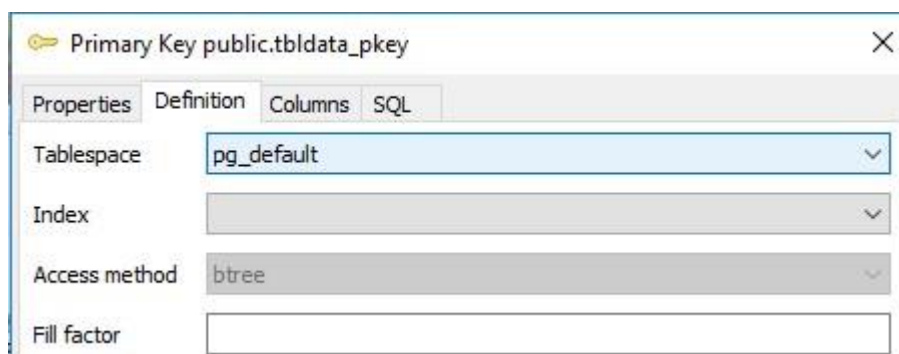


Figura 21. Definición de la primary key.

Quinceavo, por último definir la columna que va a ser identificada como la llave primaria, en este caso la primera columna de "ID".

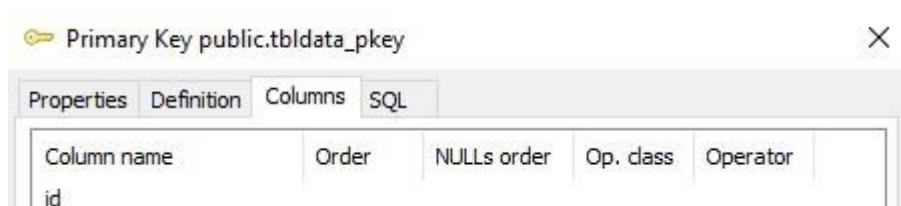



Figura 22. Primary key en la tabla.

Para poder visualizar los datos almacenados en la base de datos se seleccionará el siguiente icono.



Edit Data - localhost (localhost:5432) - drones - public.tbldata

File Edit View Tools Help

No limit

	id [PK]	serial	latitude text	longitude text	altitude text	speed numeric	pressure numeric	temperature numeric	humidity numeric	uv integer	created_at timestamp without time zone	updated_at timestamp without time zone	co numeric	temperature_2 numeric
1	1032		-100.2443875	S11.65	[-0.03	NaN	0.03	-0.11	NaN	0	2017-12-07 14:55:41.893119	2017-12-07 14:55:41.893119	NaN	0.0527
2	1033		-100.2443872	S11.62	[-0.06	NaN	0.06	-0.10	NaN	0	2017-12-07 14:55:43.032857	2017-12-07 14:55:43.032857	NaN	0.0896
3	1034		-100.2443844	S11.66	[-0.13	NaN	0.14	-0.08	NaN	0	2017-12-07 14:55:44.165558	2017-12-07 14:55:44.165558	NaN	0.1942
4	1035		-100.2443815	S11.72	[-0.18	NaN	0.17	-0.07	NaN	0	2017-12-07 14:55:45.332697	2017-12-07 14:55:45.332697	NaN	0.2523
5	1036		-100.2443793	S11.75	[-0.12	NaN	0.16	-0.08	NaN	0	2017-12-07 14:55:46.490139	2017-12-07 14:55:46.490139	NaN	0.2014
6	1037		-100.2443779	S11.79	[-0.12	NaN	0.12	-0.08	NaN	0	2017-12-07 14:55:47.630554	2017-12-07 14:55:47.630554	NaN	0.1758
7	1038		-100.2443745	S11.84	[-0.12	NaN	0.13	-0.06	NaN	0	2017-12-07 14:55:48.782854	2017-12-07 14:55:48.782854	NaN	0.1827
8	1039		-100.244371	S11.89	[-0.04	NaN	0.12	-0.06	NaN	0	2017-12-07 14:55:49.938349	2017-12-07 14:55:49.938349	NaN	0.1362
9	1040		-100.2443674	S11.93	[-0.06	NaN	0.10	-0.05	NaN	0	2017-12-07 14:55:51.085896	2017-12-07 14:55:51.085896	NaN	0.1188
10	1041		-100.2443653	S11.95	[-0.05	NaN	0.05	-0.07	NaN	0	2017-12-07 14:55:52.223496	2017-12-07 14:55:52.223496	NaN	0.0766
11	1042	25.6649932	-100.2444651	S04.74	0.06	NaN	0.03	-0.1200	0	2017-12-07 15:41:44.276806	2017-12-07 15:41:44.276806	0.43	NaN	
12	1043	25.6649932	-100.2444649	S04.7	0.07	NaN	0.05	-0.1300	0	2017-12-07 15:41:44.537529	2017-12-07 15:41:44.537529	0.19	NaN	
13	1044	25.664993	-100.2444651	S04.67	0.06	NaN	0.04	-0.1400	0	2017-12-07 15:41:44.73308	2017-12-07 15:41:44.73308	0.28	NaN	

Scratch pad

Figura 23. Datos de la Base de datos diseñada.

Tabla 1. Diccionario de datos diseñado.

Campo	Datos Nulos	Tipo de dato	Descripción
id	no	serial	Id de cada dato almacenado
latitude	Si	text	Latitud del dron
longitude	Si	text	Longitud del dron
altitude	Si	text	Altitud del dron
speed	Si	numeric	Velocidad del aire
pressure	Si	numeric	Presión del aire
temperature	Si	numeric	Temperatura del aire
humidity	Si	numeric	Humedad del aire
uv	Si	integer	Radiación uv
created_at	no	timestamp without time zone	Fecha y hora de creación del dato
updated_at	no	timestamp without time zone	Fecha y Hora de actualización del dato
co	Si	numeric	Dióxido de Carbono del aire
temperature_2	Si	numeric	Temperatura2

Para que el programa ejecute el sistema gestor de B.D., en este caso PostgreSQL se deben instalar una serie de librerías para el correcto funcionamiento y comunicación entre ambas aplicaciones. Estas librerías deben ser instaladas después de la instalación del programa NodeJS y en el símbolo del sistema (CMD) se introducen los comandos necesarios. Para hacer funcionar la base de datos en Postgre se introduce la siguiente línea de comando:

```
npm install pg
```

Dentro del programa base “index” se declaran las instrucciones para que esta aplicación se comunice con la base de datos y las tablas y columnas a utilizar.

Para hacer la conexión con nuestra base de datos se especifica mediante la instrucción:

```
var connectionString =
  "postgres://residencia:123456@localhost/drones"
```

Con esta instrucción se declara el sistema gestor (postgres), el usuario que va a ingresar (residencia) la contraseña con la que va a entrar (123456), el host local con el que va a conectar y por último el nombre de la base de datos que va a utilizar.

```
var client=pg.Client(connectionString;
client.connect(function(err) {  if(err) {
return console.error('Could not connect to
mysql', err);
  }
client.query('SELECT id AS "indx" FROM tbldata',
function(err, result) {  if(err) {  return
console.error('error running query', err);

  }  if
(result.length >
0) {
console.log(result[0].indx);
  }
}
```

El siguiente comando se utiliza para insertar los datos en la base de datos después de haberlos declarado y leído dentro del programa.



```
client.query("INSERT INTO tblldata(latitude, longitude,  
altitude, speed, pressure, temperature, humidity,  
temperature_2, uv, co, created_at, updated_at)  
values($1, $2, $3, $4, $5, $6, $7, $8, 0, $9, NOW(),  
NOW())", [lat, long, alt, speed, pressure, temp,  
humidity, temp2, co]); console.log('Data inserted!');
```

A continuación, se realiza una consulta que selecciona todos los datos de la fuente para traerlos al programa posteriormente guardarlos en la base de datos.

```
var query = client.query("SELECT *, cast(created_at as  
time) AS time FROM tblldata ORDER BY id DESC LIMIT 15");
```

### 3.2 Desarrollo

El llegar a la idea de la realización de una aplicación para almacenar datos fue a causa de la idea general de un gran proyecto dividido en diferentes tareas, todos ellos enfocados en el monitoreo de contaminante por medio de un VANT y cada proyecto forma el conjunto de actividades realizadas para la finalización sin contratiempos de la tesis. Se podría decir que el proyecto fue dividido en tres etapas.

#### 3.2.1 VANTS.

La primera etapa se trató de aprender sobre lo que íbamos a estar trabajando, es decir, los VANTS.

- Aprendiendo generalmente sobre de los drones, sus componentes y conexiones, así como datos específicos: los tipos de drones con los que íbamos a trabajar (cuadricoptero, hexacopteros), motores, baterías, hélices, estructura, materiales, computadoras de vuelo, telemetría, GPS, tipos de arquitectura de la computadora de vuelo (abierta, semi-abierta, cerrada).
- Se armaron los drones y como se ajustan sus partes.
- Se calibraron los sensores de la computadora de vuelo y el radiocontrol.
- Se hicieron y monitorearon diversas pruebas de vuelo



Figura 24. Hexacoptero.



Figura 25. Cuadricoptero.

### 3.2.2 Placas.

- La instalación del sistema operativo jessie raspbian en la raspberry pi 3 modelo B.
- La configuración del dronekit en la raspberry para la la transmisión de datos.

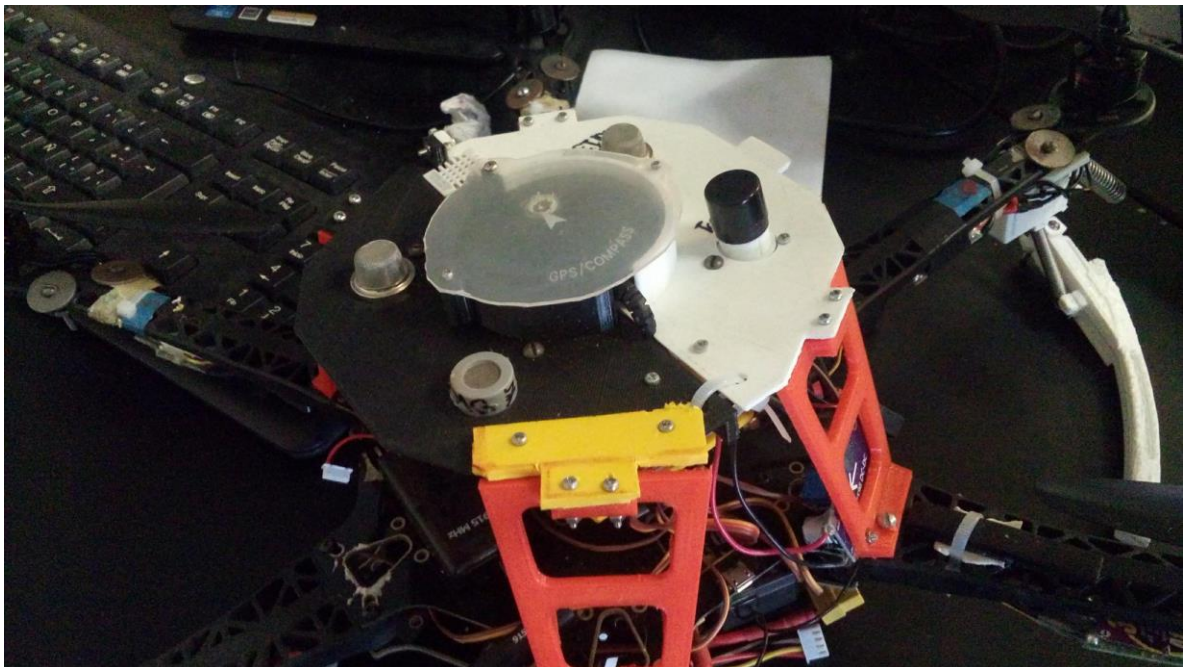


Figura 26. Base de los sensores.

- El ensamble del Arduino con los respectivos sensores MQ.



Figura 27. Sensor MQ.

- La integración de las placas en el VANT.

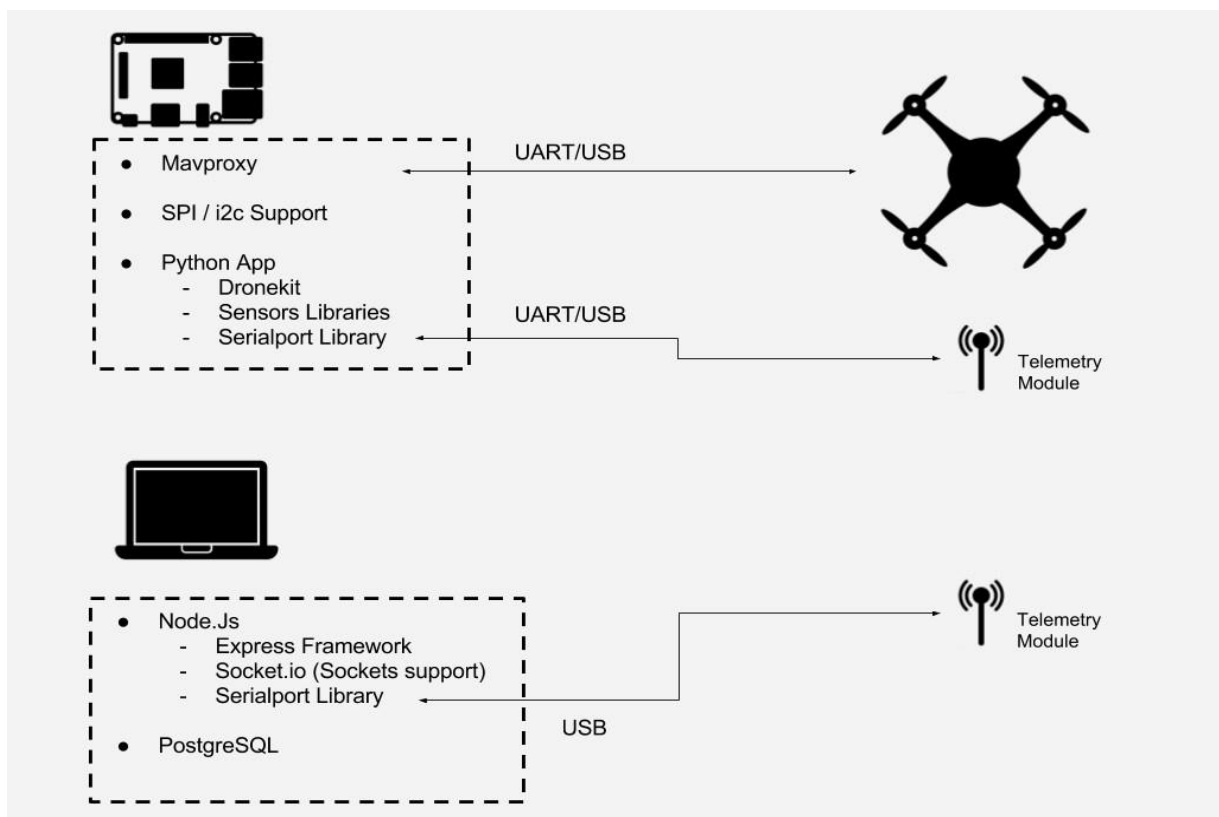


Figura 28. Diagrama.

### 3.2.3 Programas

La tercera etapa fue la configuración de la estación de tierra con los programas necesarios para la recepción de datos y almacenamiento de los mismos en una base de datos:

- Se instalaron los programas necesarios para la configuración de la estación de tierra (NodeJS versión 4.x.) y para el almacenamiento de los datos (PostgreSQL).
- Se configuró el programa principal en NodeJS para actuar como la estación de tierra (receptor de los datos), así como de interpretarlos para ser almacenados, conectarse a una base de datos y servir como una interfaz web.
- Dentro de archivos del programa principal se cargaron las librerías y recursos necesarios para que, al desplegar la página web, esta tuviera el contenido obtenido de la base de datos.

### 3.3 Dificultades

Los problemas que surgieron durante el desarrollo de la aplicación, se debieron a la incompatibilidad de versiones del programa principal con las de nodeJS y PostgreSQL, lo cual impedía ejecutar el programa. Esto era debido a que el programa principal fue desarrollado antes que la versión de nodeJS 10.13 con la que inicialmente fue probado.

La solución fue utilizar las versiones específicas compatibles con la versión del programa principal, quedando funcionando con JS versión 4.8 y PostgreSQL 9.6.5, las cuales se muestran en la siguiente imagen.

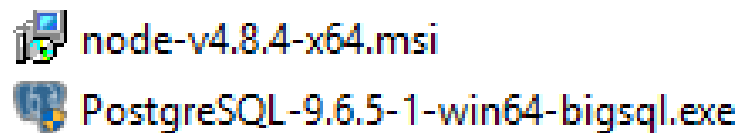


Figura 29. Programas necesarios.

En el código del programa se deben especificar algunos puertos, los cuales se pueden modificar directamente desde el código, de la siguiente manera.

```
var client=mysql.createConnection({  
  
host:'localhost',  
  
user:'root',  
  
password: 'udooer',  
  
database: 'node'  
  
})
```

En esta parte se declararía el nombre del host como su usuario además del nombre de la base de datos que se utilizaría

```
var server = http.createServer(app);  
  
server.listen(8080);
```

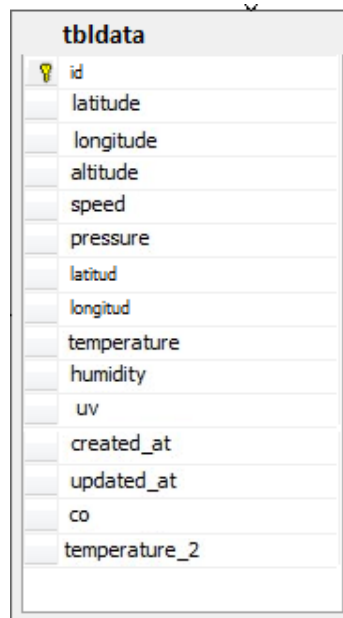
En esta parte se declara el puerto de la Base de datos, mismo que escuchará el programa, en este caso se utilizó el puerto por defecto: el 8080.

```
var serialPort = new SerialPort("COM3", {  
    autoOpen: false,  
    baudrate: 57600,  
    parser: serialport.parsers.readline("\n")  
});
```

Este será el puerto donde conectaremos la antena de telemetría a nuestra estación de tierra. Hay que asegurarse de que esté conectado al COM3, en caso de ser diferente es necesario cambiar el COM o cambiar el código.

### 3.4 Diseño de la base de datos

La base de datos lleva el nombre de: “**drones**” la cual consta de una única tabla llamada “**tbldata**”. En esa tabla se crean los campos correspondientes a las variables recibidas de la VANT. Ver Tabla 1 diccionario de datos.

The image shows a screenshot of a database table named 'tbldata'. The table has 15 columns. The first column, 'id', is marked as a primary key with a yellow key icon. The other columns are: 'latitude', 'longitude', 'altitude', 'speed', 'pressure', 'latitud', 'longitud', 'temperature', 'humidity', 'uv', 'created\_at', 'updated\_at', 'co', and 'temperature\_2'. Each column has a small grey square icon to its left.
















tbldata	
	id
	latitude
	longitude
	altitude
	speed
	pressure
	latitud
	longitud
	temperature
	humidity
	uv
	created_at
	updated_at
	co
	temperature_2

Figura 30. Diagrama de la Base de datos.



### 3.5 Material y equipo utilizado (SW)

#### Dron

Un dron es esencialmente un robot que vuela. Puede ser controlado vía remota o puede volar de manera autónoma a través de unos planes de vuelo controlados por el software que está integrado al sistema y el cual trabaja en conjunto con el GPS.



Figura 31. Dron de cuatro hélices. Cuadricoptero.

#### Computadora de vuelo

Es el sistema o computadora que dirige los movimientos del dron a través de instrumentos de medición como el giroscopio y el acelerómetro, y en otros casos, también utilizando GPS o barómetro.

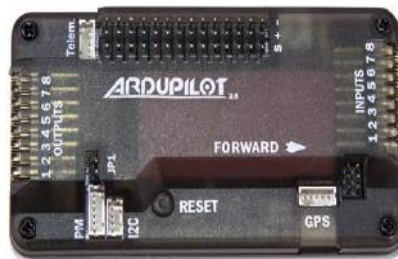


Figura 32. Computadora de vuelo Pixhawk.

**GPS**

El GPS es el sistema de posicionamiento global, el cual nos puede dar la ubicación del dron ya que está basado por satélites y cuenta con 24 satélites puestos en órbita.

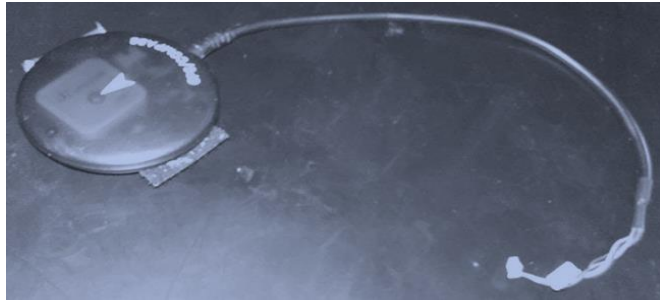


Figura 33. GPS.

**Antena**

Permite la comunicación entre el dron y la estación de tierra y/o la radio.



Figura 34. Antenas de Telemetría.

### Batería

Las baterías utilizadas para dar poder al dron. Es un tipo de batería recargable de tecnología de ion de litio en un empaque suave, haciéndolas más ligeras pero también quitándole rigidez a la cobertura.



Figura 35. Batería de Litio.

### Motores

Los motores son parte del sistema de propulsión. Los motores que se utilicen tienen un impacto enorme en la capacidad de carga máxima que el dron pueda soportar, así como el tiempo de vuelo.



Figura 36. Motores para los drones.

### Radio

Utilizado para controlar el dron en modo manual o de respaldo cuando se vuela en modo automático.



Figura 37. Radiocontrol para operar el dron.

### Estación de tierra

La estación de tierra es el equipo que un segundo operador debe mantener en constante vigilancia para poder vigilar el estado del dron, el estado de los aparatos que el dron este cargando (cámaras, sensores) y la ruta de vuelo que se haya trazado.



Figura 38. Una laptop que hace función de estación de tierra.

### Rasberri pi 3

Placa de computadora simple integra conectividad BlueTooth y Wi-Fi. Se le pueden instalar diversos sistemas operativos y aplicaciones hechos especialmente para este tipo de placas.

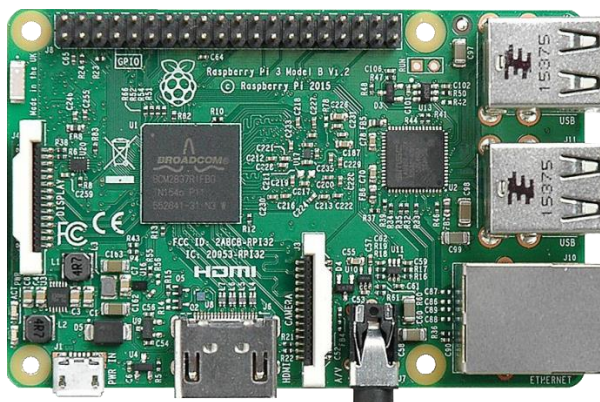


Figura 39. Placa rasberri pi 3 versión B.

Los cables se utilizan para interconectar a la computadora de vuelo lo necesario para volar (gps, motores, buzzer), así como también conectar el dron a la estación de tierra para verificar que este en un estado óptimo para el vuelo.

- Raspbian Jessi OS

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Stretch (Debian 9.4) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática.

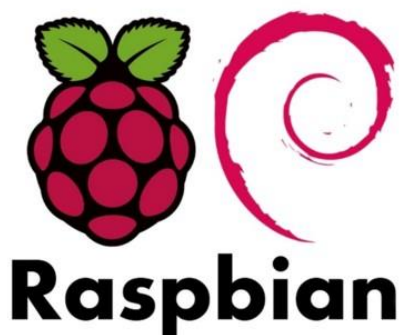


Figura 40. Símbolo de distribución Rabian Jessie de Linux.

**Dronekit**

Utilerías que ofrecen un SDK y API web para el fácil desarrollo de aplicaciones para drones.

**Nodejs**

Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node está diseñado para construir aplicaciones en red escalables.

**Postgresql**

Es un manejador de base de datos libre, basado en el modelo de SQL y permite la recopilación y almacenamiento de los volúmenes de datos entrantes.

**SublimeText**

Es un editor de texto que permite trabajar con diversos lenguajes de programación, haciendo que la codificación se haga de una manera más ágil y dinámica.

**Mission Planner**

Mission Planner tiene como objetivo ofrecer a sistemas abiertos de controladores de vuelo la posibilidad de trazar rutas para el vuelo de sus vehículos, así como también dar la posibilidad de calibrar los aparatos para comprobar su buen funcionamiento antes, durante y al finalizar la operación.



Figura 41. Pantalla de inicio del programa.

### **3.6 Consideraciones para el diseño**

El tipo de base de datos que se diseñó está orientado a objetos, de tal modo que esta pueda contener objetos de base de datos.

Las bases de datos orientadas a objetos se definen como una base de datos inteligente que soporta el paradigma orientado a objetos, almacenando métodos y datos, y no solamente datos.

La base de datos diseñada permite una integración más fácil con la propia arquitectura de los programas empleados para el manejo de la base de datos.

Se tomaron en cuenta las variables a utilizar para poder almacenar todos los datos necesarios para este proyecto.

Los datos enviados por el VANT a través de sus sensores, manejan un formato analógico por lo que fue necesario realizar conversiones a formato digital para ser almacenados en la base de datos diseñada. La conversión se hizo de analógica-digital donde ésta consiste básicamente en realizar de forma periódica medidas de la amplitud de una señal, redondear sus valores a un conjunto finito de niveles preestablecidos de tensión y registrarlos como números enteros en cualquier tipo de memoria o soporte.

Al momento de almacenarse deben estar en el mismo formato y guardarse en el orden establecido, debido a que en la base de datos si alguno de éstos no llegara a estar almacenado, ocurriría un error.

Para la creación de la aplicación que almacena los datos provenientes de la VANT se utilizó una herramienta llamada Sublime Text, la cual nos permite editar y visualizar código en una gran cantidad de lenguajes de programación. Para efectos de este proyecto se utilizó JavaScript, ya que este lenguaje tiene facilidad de almacenar los datos en la B.D, además de contar con un soporte en dado caso de que este tuviera problemas en el almacenamiento de datos



## Capítulo IV

### Resultados

#### ANÁLISIS COMPARATIVO PARA LA INTEGRACIÓN Y MONITOREO DE DATOS ENTRE UN SISTEMA DE TIEMPO REAL Y UN SISTEMA DE INFORMACION

Al analizar las dos diferentes maneras en las que se pueden almacenar los datos: de forma tradicional y de forma en tiempo real, pueden verse diferencias al momento de generar resultados dependiendo de su aplicación.

id [PK] serial	latitude text	longitude text	altitude text	speed numeric	pressure numeric	temperature numeric	humidity numeric	uv integer	created_at timestamp with time zone	updated_at time with time zone	co numeric	temperature_2 numeric
37	25.6649976	-100.244381	1.9	0.02	96.00	12.00	65.0000	0	2018-08-28 14:25:44.545863-05	14:25:44.545863-05	188.00	27.0000
38	25.6650007	-100.2443683	1.98	0.01	97.00	26.00	32.0000	0	2018-08-28 14:25:47.642661-05	14:25:47.642661-05	122.00	32.0000
39	25.6650053	-100.2443734	1.99	0.01	95.00	10.00	4.0000	0	2018-08-28 14:25:50.637495-05	14:25:50.637495-05	147.00	46.0000
40	25.6650111	-100.2443783	2.02	0.01	1.00	13.00	25.0000	0	2018-08-28 14:25:53.737575-05	14:25:53.737575-05	71.00	0.0000
41	25.66502	-100.2443743	2.01	0.01	98.00	98.00	32.0000	0	2018-08-28 14:25:56.8397-05	14:25:56.8397-05	32.00	39.0000
42	25.6650246	-100.2443740	2.03	0.01	23.00	14.00	80.0000	0	2018-08-28 14:25:59.94125-05	14:25:59.94125-05	147.00	4.0000
43	25.6650288	-100.2443772	2.05	0.01	0.00	22.00	32.0000	0	2018-08-28 14:26:03.037377-05	14:26:03.037377-05	183.00	6.0000
44	25.6650324	-100.2443803	2.08	0.01	8.00	20.00	39.0000	0	2018-08-28 14:26:06.189961-05	14:26:06.189961-05	148.00	10.0000
45	25.6650329	-100.2443812	2.04	0.01	87.00	43.00	42.0000	0	2018-08-28 14:26:09.277327-05	14:26:09.277327-05	172.00	3.0000
46	25.6650343	-100.2443877	2.0	0.02	14.00	41.00	48.0000	0	2018-08-28 14:26:12.399455-05	14:26:12.399455-05	184.00	44.0000
47	25.6650401	-100.2443925	2.09	0.02	44.00	30.00	75.0000	0	2018-08-28 14:26:15.544563-05	14:26:15.544563-05	24.00	30.0000
48	25.6650405	-100.2443951	2.09	0.02	97.00	3.00	96.0000	0	2018-08-28 14:26:18.479786-05	14:26:18.479786-05	20.00	32.0000
49	25.6650437	-100.244397	2.03	0.02	16.00	19.00	45.0000	0	2018-08-28 14:26:21.594399-05	14:26:21.594399-05	118.00	12.0000
50	25.6650448	-100.2443958	2.21	0.03	0.00	25.00	44.0000	0	2018-08-28 14:26:24.657049-05	14:26:24.657049-05	97.00	7.0000
51	25.6650500	-100.2443964	2.13	0.02	78.00	19.00	71.0000	0	2018-08-28 14:26:27.79476-05	14:26:27.79476-05	196.00	7.0000
52	25.6650509	-100.244392	2.09	0.01	89.00	21.00	45.0000	0	2018-08-28 14:26:30.882564-05	14:26:30.882564-05	158.00	3.0000
53	25.6650519	-100.2443924	2.11	0.02	40.00	16.00	66.0000	0	2018-08-28 14:26:33.964661-05	14:26:33.964661-05	182.00	41.0000
54	25.6650547	-100.2443937	2.13	0.02	40.00	11.00	97.0000	0	2018-08-28 14:26:37.046361-05	14:26:37.046361-05	188.00	48.0000

Figura 42. Base de datos en tiempo real.

	codigo [PK] serial	nombre character var	sueldo double precis	fechnac date	categoria character(1)
1	1	Chema	55555.55	1964-03-19	J
2	2	Petra	44444.44	1967-07-23	J
3	3	Luis	1234.56	1975-12-12	E
4	4	Fernanda	1345.67	1984-04-13	E
5	5	Mario	1222.22	1987-01-01	E
6	6	Laura	1567.89	1996-02-22	E
7	7	Petronio	1456.32	1996-01-11	E

Figura 43. Base de datos tradicional.

Con la base de datos en tiempo real es posible realizar un análisis en segundos, porque se trabaja con cantidades pequeñas de datos a medida que estos ingresan, haciendo posible ejecutar cálculos intensos, como la búsqueda en paralelo y combinar consultas sobre la marcha. De esta forma se pueden obtener resultados mucho más rápidamente que con los sistemas de base de datos transaccionales, en los que los datos deban ser almacenados e indexados antes de procesarse para su consulta.

Una de las virtudes de la B.D. en tiempo real es el flujo de entrada de datos al servidor, el cual da una respuesta en tiempo real a las cambiantes condiciones de los datos, haciendo posible actualizar una base de datos y la visión continua y en vivo de los mismos.

A continuación se muestra las tablas comparativas entre ambos sistemas de B.D, que contrastan tanto el diseño de archivos como su interacción con la B.D:

Tabla 2. Diseño de archivos para la base de datos.

<b>Criterio</b>	<b>Sistema para el procesamiento de transacciones (tradicional)</b>	<b>Sistema en tiempo real (el desarrollado para el proyecto)</b>
<b>Diseño lógico usado en datos</b>	Consiste en almacenar los datos en archivos individuales, exclusivos para cada aplicación particular. En este sistema los datos pueden ser redundantes y la actualización de los archivos es más lenta que en una base de datos.	Los datos se almacenan en varias tablas con igual esquema. La Unión de las tablas particionadas debe ser igual al conjunto original.
<b>Flexibilidad en el manejo de registros</b>	Debido a la probabilidad de inconsistencia, cada modificación se debe repetir en	Gracias a los metadatos es posible describir la información de la base de

	todas las copias del dato.	datos. Ej. Metadato DronDatos. Longitud ,Latitud ,Altitud.
<b>Creación de llaves de registro</b>	Cada uno de los registros almacenados en un archivo convencional.	Los índices pueden ser creados con cualquier combinación de campos de una tabla.
<b>Tipo de diseño arquitectónico para los archivos de la BD</b>	Jerárquicos. En red.	Relacionales. Multidimensionales. De objetos.
<b>Flexibilidad en diagramas para estructura de datos</b>	Las relaciones entre registros forman una estructura en árbol.	Almacena los datos en filas o registros (tuplas) y columnas o campos (atributos). Estas tablas pueden estar conectadas entre sí por claves comunes.
<b>Tipos de archivos manejados</b>	Archivos secuenciales.	Archivos Orientados a objetos.
<b>Redundancia de archivos</b>	Datos redundantes, ya que se repiten continuamente.	Menor redundancia. No hace falta tanta repetición de datos.
<b>Flexibilidad en estructuras de almacenamiento</b>	Tiempos de procesamiento elevados al no poder optimizar el espacio de almacenamiento.	Menor espacio de almacenamiento, gracias a una mejor estructuración de los datos.

<b>Respaldo y recuperación de archivos</b>	Datos no documentados.	Datos más documentados Gracias a los metadatos es posible describir la información de la base de datos.
<b>Pérdida de datos</b>	Es posible perder la integridad de los datos. Mayor facilidad de perder los datos.	Mayor dificultad de perder los datos o de realizar incoherencias con ellos.

Tabla 3. Interacción en la base de datos.

<b>Criterio</b>	<b>Sistema para el procesamiento de transacciones (tradicional)</b>	<b>Sistema en tiempo real (El desarrollado para el proyecto)</b>
<b>Relación entre datos</b>	Unívoca, Uno a varios, Varios a varios	Unívoca, Uno a varios, Varios a varios
<b>Diagramas de estructura de datos</b>	Jerárquicos En red	Relacionales. Multidimensionales. De objetos.
<b>Relación entre entidades u objetos</b>	Las relaciones entre registros forman una estructura en árbol.	Los datos de filas o registros y columnas o campos estas tablas pueden estar conectadas entre sí por claves comunes.
<b>Modelo de datos</b>	Las entidades se representan en registros que se agrupan en archivos	Estáticas se definen en un esquema. Dinámicas se definen

		como especificaciones de transacciones.
<b>Normalización</b>	Se normalizan para tener la información de forma más eficaz y estructurada.	Se normalizan para tener la información de forma no tener redundancia de datos así como más velocidad de respuesta.
<b>Manipulación de datos</b>	Permite la inserción, consulta de datos, eliminación y actualización.	Permite la inserción, actualización, eliminación y consulta de datos mediante el lenguaje de manejo o manipulación de datos.
<b>Ambiente de la base de datos</b>	Enfoque estándar para desarrollo y mantenimiento de programas de aplicación. Modelo de datos y propiedad de datos.	Enfoque estándar para desarrollo y mantenimiento de programas de aplicación. Modelo de datos y propiedad de datos. Acceso a la base de datos. Segregación de funciones. Seguridad de los datos y recuperación de la base de datos.

#### 4.1 Experimentación del prototipo en toma de muestras

Se realizaron pruebas de vuelo con el dron para tomar muestras para la base de datos y así valorar la transmisión y recepción de datos en el momento del vuelo.



Figura 44. Radio control.

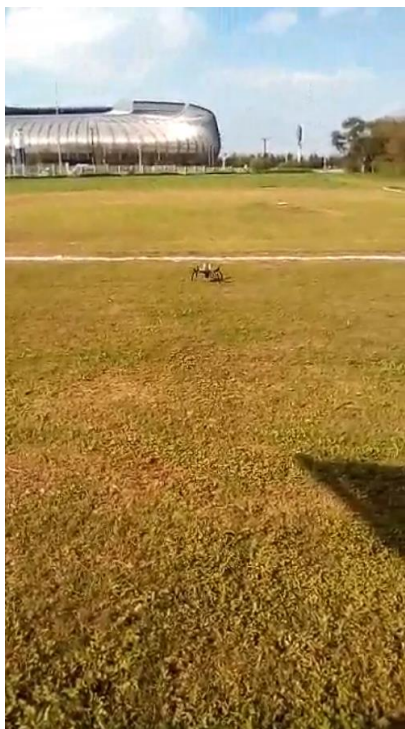


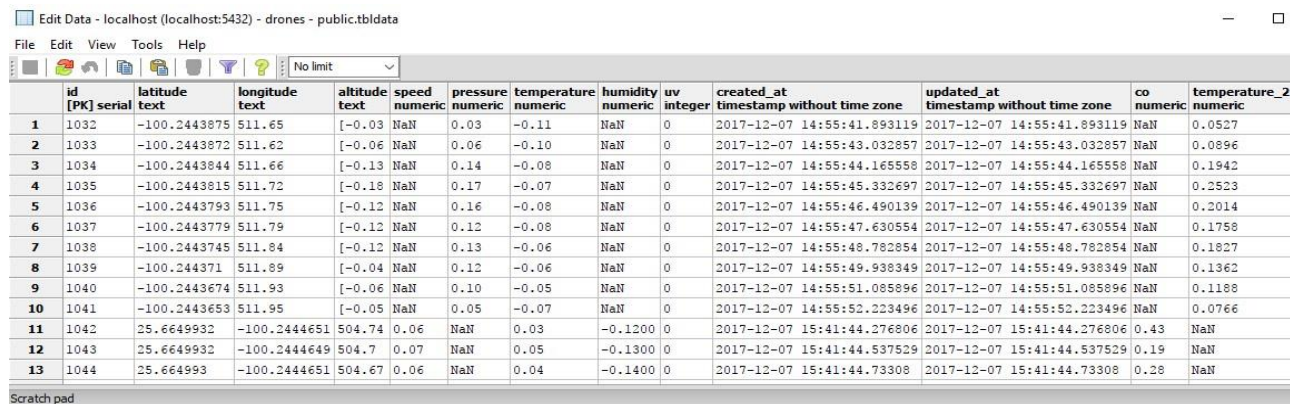
Figura 45. Dron en tierra.



Figura 46. Dron en vuelo.

## 4.2 Análisis de resultados

Los datos obtenidos de la lectura que hizo el dron, fueron almacenados correctamente en sus respectivos campos al ejecutarse las pruebas de vuelo, como se puede constatar en siguiente tabla.



	id [PK]	serial	latitude text	longitude text	altitude text	speed numeric	pressure numeric	temperature numeric	humidity numeric	uv integer	created_at timestamp without time zone	updated_at timestamp without time zone	co numeric	temperature_2 numeric
1	1032		-100.2443875	511.65	[-0.03	NaN	0.03	-0.11	NaN	0	2017-12-07 14:55:41.893119	2017-12-07 14:55:41.893119	NaN	0.0527
2	1033		-100.2443872	511.62	[-0.06	NaN	0.06	-0.10	NaN	0	2017-12-07 14:55:43.032857	2017-12-07 14:55:43.032857	NaN	0.0896
3	1034		-100.2443844	511.66	[-0.13	NaN	0.14	-0.08	NaN	0	2017-12-07 14:55:44.165558	2017-12-07 14:55:44.165558	NaN	0.1942
4	1035		-100.2443815	511.72	[-0.18	NaN	0.17	-0.07	NaN	0	2017-12-07 14:55:45.332697	2017-12-07 14:55:45.332697	NaN	0.2523
5	1036		-100.2443793	511.75	[-0.12	NaN	0.16	-0.08	NaN	0	2017-12-07 14:55:46.490139	2017-12-07 14:55:46.490139	NaN	0.2014
6	1037		-100.2443779	511.79	[-0.12	NaN	0.12	-0.08	NaN	0	2017-12-07 14:55:47.630554	2017-12-07 14:55:47.630554	NaN	0.1758
7	1038		-100.2443745	511.84	[-0.12	NaN	0.13	-0.06	NaN	0	2017-12-07 14:55:48.782854	2017-12-07 14:55:48.782854	NaN	0.1827
8	1039		-100.244371	511.89	[-0.04	NaN	0.12	-0.06	NaN	0	2017-12-07 14:55:49.938349	2017-12-07 14:55:49.938349	NaN	0.1362
9	1040		-100.2443674	511.93	[-0.06	NaN	0.10	-0.05	NaN	0	2017-12-07 14:55:51.085896	2017-12-07 14:55:51.085896	NaN	0.1188
10	1041		-100.2443653	511.95	[-0.05	NaN	0.05	-0.07	NaN	0	2017-12-07 14:55:52.223496	2017-12-07 14:55:52.223496	NaN	0.0766
11	1042	25.6649932	-100.2444651	504.74	0.06	NaN	0.03	-0.1200	0	2017-12-07 15:41:44.276806	2017-12-07 15:41:44.276806	0.43	NaN	
12	1043	25.6649932	-100.2444649	504.7	0.07	NaN	0.05	-0.1300	0	2017-12-07 15:41:44.537529	2017-12-07 15:41:44.537529	0.19	NaN	
13	1044	25.664993	-100.2444651	504.67	0.06	NaN	0.04	-0.1400	0	2017-12-07 15:41:44.73308	2017-12-07 15:41:44.73308	0.28	NaN	

Figura 47. Datos almacenados en la tabla “tbldata” de la base de datos diseñada.

Los datos almacenados en la tabla “tbldata” son los valores captados por los sensores en el dron, los cuales son transmitidos en tiempo real, como la latitud, longitud, altitud, así como los datos de los sensores de los contaminantes criterios manejados en este proyecto.

Con esta tabla se pudieron comprobar tanto el funcionamiento de los sensores como la recuperación de sus datos vía una página web en colaboración con otro trabajo de investigación, la cual es responsable de mostrar de manera gráfica y en tiempo real los datos de la B.D.

## 4.3. Análisis comparativo entre Bases de Datos para sistemas en tiempo real.

Haciendo la comparativa entre PostgreSQL y MySQL, podremos ver cuál de los dos gestores responde de forma más rápida. A continuación se muestra diferentes aspectos de desempeño entre ambos gestores.



### 4.3.1 Otras pruebas con B.D.'s

#### Primer test

Citado de bitácora de @macool como apoyo a los resultados [\[3\]](#). La primera parte fue crear una base de datos en cada uno de los (SGBD) la base de datos se llama tweets con 12313 users.

```
benchmark_1=# select count(*) from tweets;
count
-----
 69255
(1 row)

benchmark_1=# select count(*) from users;
count
-----
 12313
(1 row)
```

Figura 48. Conteo de registros con PostgreSQL.

```
mysql> select count(*) from tweets;
+-----+
| count(*) |
+-----+
|    69255 |
+-----+
1 row in set (0.01 sec)

mysql> select count(*) from users;
+-----+
| count(*) |
+-----+
|    12313 |
+-----+
1 row in set (0.00 sec)
```

Figura 49. Conteo de registros con MySQL.

**Segundo test**

Contar (diez mil) veces (count(\*))

```
cuxibamba:postgresql macool$ ruby count.rb
Rehearsal -----
count 10000 times  5.620000  0.390000  6.010000 (165.826635)
----- total: 6.010000sec

                user      system      total      real
count 10000 times  5.630000  0.390000  6.020000 (164.970709)
```

Figura 50. PostgreSQL count.

```
cuxibamba:mysql macool$ ruby count.rb
Rehearsal -----
count 10000 times  6.250000  0.330000  6.580000 (151.418105)
----- total: 6.580000sec

                user      system      total      real
count 10000 times  6.290000  0.320000  6.610000 (151.667663)
```

Figura 51. MySQL count.

**Resultados** de las pruebas de conteo:

PostgreSQL: 164.97 segundos

MySQL: 151.67 segundos

### 4.3.3 Segundo test

Consultando una palabra aleatoria 500 veces (where text like '%word%').

Se lleva a cabo en dos partes:

1. Contar cuántas veces aparece dentro del contenido de los tweets.
2. Hacer la proyección de los resultados; no solamente el conteo.

```
cuxibamba:postgresql macool$ ruby query.rb
Rehearsal -----
count query for random word 500 times  0.380000  0.020000  0.400000 ( 21.020610)
select query for random word 500 times  0.360000  0.020000  0.380000 ( 20.506485)
----- total: 0.780000sec

                user      system      total      real
count query for random word 500 times  0.360000  0.020000  0.380000 ( 20.714749)
select query for random word 500 times  0.300000  0.020000  0.320000 ( 20.773904)
```

Figura 52. PostgreSQL consulta de una palabra aleatoria.

```
cuxibamba:mysql macool$ ruby query.rb
Rehearsal -----
count query for random word 500 times  0.440000  0.020000  0.460000 ( 28.596938)
select query for random word 500 times  0.730000  0.030000  0.760000 ( 74.751770)
----- total: 1.220000sec

                user      system      total      real
count query for random word 500 times  0.350000  0.010000  0.360000 ( 26.407030)
select query for random word 500 times  0.490000  0.020000  0.510000 ( 69.910461)
```

Figura 53. MySQL consulta de una palabra aleatoria.

#### Resultados:

PostgreSQL:

- **20.71** segundos al contar cuántos resultados (count()),
- **20.77** segundos al hacer la proyección de los resultados y asignar a una variable cada uno.

MySQL:

- **26.4** segundos al contar cuántos resultados (count());
- **69.91** segundos al hacer la proyección y asignar los resultados a variables.

## Capítulo V

---

### Conclusiones

---

#### 5.1 Conclusiones

La hipótesis se cumplió debido a que se compararon un sistema de información en tiempo real con un sistema de información tradicional de almacenamiento de datos, por lo que se pudieron ver las diferencias que se tienen desde cómo se debe desarrollar una B.D. en tiempo real, su flexibilidad para el manejo de registros, de cómo se deben crear sus llaves de registro o id, si se normalizan para minimizar la redundancia a fin de agilizar su integración y monitoreo.

Los datos que se almacenaron en tiempo real se hizo por medio de los sensores, por lo que la información que se muestra está de acuerdo a las condiciones en que se hicieron las mediciones al momento del monitoreo.

El diseño de la tabla “tbldata” se hizo de acuerdo a la cantidad de sensores que se incluyeron en el dron, cantidad relacionada con el número de contaminantes criterio que se quieren monitorear en el presente proyecto, de tal manera que si la cantidad de sensores cambia, el diseño de la tabla se verá afectado aumentado o disminuyendo la cantidad de sus campos.

En una comparación de dos SGBD como son MySQL y PostgreSQL para decidir cuál de los dos poder usar en el proyecto, PostgreSQL dio mejores tiempos de respuesta al momento de realizar búsquedas, por otra parte se demostró la compatibilidad al 100% con el programa principal desarrollado para el proyecto, cosa que no se consiguió usando MySQL.

## **5.2 Recomendaciones**

Si se quisieran usar otros manejadores de base de datos para ya sea, mejorar su almacenamiento o para conseguir mayor rapidez de respuesta, se debería modificar el código del programa principal, debido a que este funciona actualmente solo con la version de nodeJS v4.8.4 y PostgreSQL v9.6.5-1.

El programa principal realiza todas las tareas implicadas en el monitoreo de los contaminantes, por lo que mejoraría si se dividiera en módulos de acuerdo ala funcionalidad de las diferentes operaciones, como son: la digitalización de los valores de entrada des de los sensores, la interacción con la base de datos y la habilitación de la página web.

### 5.3 Trabajos futuros

En desarrollos futuras podrían utilizarse Aplicaciones móviles para visualizar la información de la base de datos de una forma más sencilla y a la mano, que desde la estación de tierra.

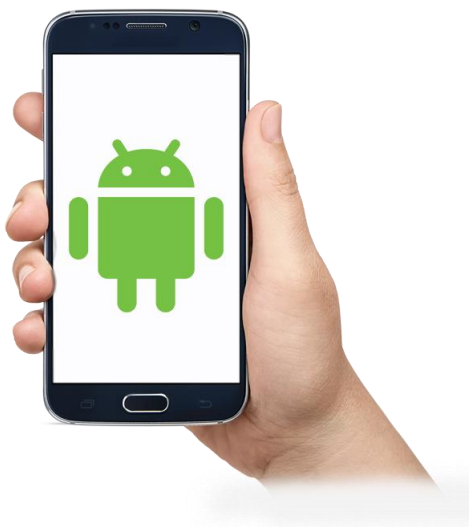


Figura 54. Aplicación móvil.

Con la utilización de herramientas más modernas como el Photon, podría remplazarse el arduino y la raspberry dando una mayor facilidad de conectividad con los sensores, actuando como un arduino con conexión a internet, contando a la vez con una plataforma en la nube en la que se podrían ver los datos en su propia plataforma.

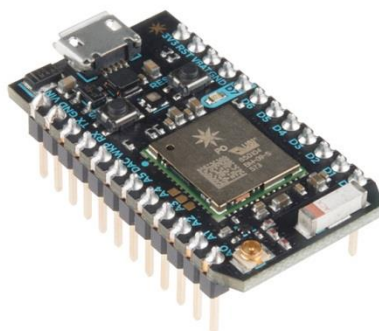


Figura 55. Photon.

## 6 REFERENCIAS

[1]Alfonso. (2015). Orígenes y Antecedentes de las Bases de Datos.10/6/2018, de Dataprix

Sitio web: <http://www.dataprix.com/24-or-genes-antecedentes-las-bases-datos>

[2]Arturo Curiel Anaya. (2017). Lenguaje de Manipulación de Datos

(DML). 01/12/2017, de Microsoft Sitio web:

[http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/53\\_lenguaje\\_de\\_manipulacin\\_de\\_datos\\_dml.html](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/53_lenguaje_de_manipulacin_de_datos_dml.html)

[3] bitácora de @macool.(2013). MySQL vs PostgreSQL vs MongoDB

(velocidad).(10/6/2018),de bitácora de @macool Sitio web:

<http://macool.me/tag/postgresql>

[4]Delgado Jefferson. (2015, agosto 29). Sistemas transaccionales, sistemas de apoyo a las decisiones y sistemas estratégicos. (2/10/2018),

Recuperado de <https://www.gestiopolis.com/sistemas-transaccionales-sistemas-de-apoyo-a-las-decisiones-y-sistemas-estrategicos/>

[5] Enrique Martínez Montejo.(2004). Manual de SQL de Microsoft Jet -

Mvp-access.(3/12/2017). Mvp-access Sitio web: <http://www.mvp-access.es/softjaen/manuales/sql/sjtsqlj001.htm>

[6]Gavilanes Parra Dario Alejandro.(2016). Aplicación de los drones como herramienta tecnológica innovadora para la investigación de los sitios

arqueológicos. 2/10/2018, de UTMACH Sitio web:

<http://repositorio.utmachala.edu.ec/bitstream/48000/9673/1/ECUACS-ESS-DE00018.pdf>

[7]Julio César Benedetti.(2015). "Las bases de datos geoespaciales en el análisis para la reducción de riesgo de desastres en áreas urbanas".

(2/10/2018)De USAL Sitio web:

[http://www.usal.edu.ar/archivos/geousal/docs/sintesis\\_seminario\\_investigacion\\_urbana\\_benedetti.pdf](http://www.usal.edu.ar/archivos/geousal/docs/sintesis_seminario_investigacion_urbana_benedetti.pdf)

[8]Legislación sobre Agricultura y la Alimentación. (2017). Base de datos FAOLEX. 03/12/2017, de FAOLEX Sitio web:

<http://www.fao.org/faolex/thematic-collections/es/>

[9]Lic. Rodolfo Vegas Ana Aguilera, Tineo Leonid, José Cadenas. (02/12/009). POSTGRESQLf.

08/12/2017, de DEPARTAMENTO DE COMPUTACIÓN BASE DE DATOS

Sitio web:

[http://ldc.usb.ve/~itcadenas/PostgreSQLf/Documento\\_PostgreSQLf.pdf](http://ldc.usb.ve/~itcadenas/PostgreSQLf/Documento_PostgreSQLf.pdf)

[10]Lidia Contreras. (4/01/2011). Historia de las Bases de Datos.

02/12/2017, de museo de informática Sitio web:

<http://histinf.blogs.upv.es/2011/01/04/historia-de-las-bases-dedatos/>

[11]Manuel Raigón. (26/06/2015). NoSQL(Not Only SQL). 09/12/2017, de

ticarte Sitio web: <http://www.ticarte.com/contenido/nosqlnot-only-sql-una-alternativa-al-tradicional-sql>

[12]Marcelo A. Giuntoli. (02/06/16). Biblioteca digital. 10/12/2017, de

Facultad de Ciencias Sociales y Administrativas Sitio web:

[http://bibliotecadigital.uda.edu.ar/objetos\\_digitales/525/tesis-4849-confiabilidad.pdf](http://bibliotecadigital.uda.edu.ar/objetos_digitales/525/tesis-4849-confiabilidad.pdf)

[13]Medina García, Pedro Andrés, Niño López, Juan Carlos. (2017). Diseño de un Modelo de Agricultura de Precisión Utilizando Drones y un Sistema de Captura, Almacenamiento y Análisis de Datos que Permita Identificar a Tiempo la

Generación de la Pudrición del Cogollo en Cultivos de Palma de Aceite que se Encuentran en los Departamentos de Bolívar y Santander.(2/10/2018), de RIUD

Sitio web: <http://hdl.handle.net/11349/6306>

[14]Uribe Franco Christian Guillermo. (2008). ESPOL. 08/12/2017, de

Instituto de Ciencias Matemáticas Sitio web:

<https://www.dspace.espol.edu.ec/bitstream/123456789/19249/2/TESIS%20COMPLETA%20CHRISTIAN%20URIBE%20FRANCO.pdf>



## ANEXOS

Departamento: **Sistemas y Computación**  
 Cd. Guadalupe, N.L., a 6 de febrero de 2018

Asunto: **Solicitud de registro de tesis**

**Ing. Magaly Benítez Tamez**  
**Jefa del Departamento de Sistemas y Computación del I.T de Nuevo León**  
 Presente.

Por este conducto solicito a usted sea registrada la tesis que llevaré a cabo bajo la línea de investigación de Tecnologías de Información y Comunicación. La realización de esta tesis servirá como el producto académico para la obtención del grado de Ingeniero en Sistemas Computacionales en este Instituto Tecnológico de Nuevo León.

Los datos de registro de la tesis son:

a) Nombre del estudiante	<b>Andrés González Rodríguez</b> No. de control: <b>14480653</b>
b) Programa Educativo	<b>Ingeniería en Sistemas Computacionales</b>
c) Título de tesis:	<b>Análisis comparativo para la integración y monitoreo de datos entre un sistema de tiempo real y un sistema de información.</b>

Así mismo, le informo que para este proyecto mi comité de seguimiento de tesis es:

**Director de tesis: M.C. José Isidro Hernández Vega**  
**Co-Director: M.C. Jonam Leonel Sánchez Cuevas**

Agradezco de antemano su valioso apoyo para el registro de este proyecto académico.

Atentamente



**C. Andrés González Rodríguez**

Estudiante de la carrera de Ingeniería en Sistemas Computacionales

No. de control: **14480653**

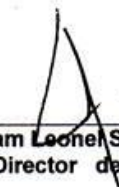


SEP  
 TNM SES  
**INSTITUTO TECNOLÓGICO  
 DE NUEVO LEÓN  
 DEPARTAMENTO DE  
 SISTEMAS Y COMPUTACIÓN**

VoBo.



**M.C. José Isidro Hernández Vega**  
**Director de Tesis**



**M.C. Jonam Leonel Sánchez Cuevas**  
**Co-Director de Tesis**

c.c.p. Archivo  
 c.c.p. Directores y Co-Director de Tesis  
 c.c.p. Estudiante



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Nuevo León

Departamento de Sistemas y Computación

Guadalupe, N.L., a 5 de octubre de 2018

Asunto: Notificación de tesis terminada para revisión

**Ing. Magaly Benítez Tamez**

Jefa de Departamento de Sistemas y Computación del I.T. de Nuevo León

Presente

Por este conducto le notificamos la conclusión de la tesis de Ingeniería en Sistemas Computacionales para revisión, cuyo título es:

**"ANÁLISIS COMPARATIVO PARA LA INTEGRACIÓN Y MONITOREO DE DATOS ENTRE UN SISTEMA DE TIEMPO REAL Y UN SISTEMA DE INFORMACIÓN"**

Se han entregado por correo electrónico dos ejemplares del documento para la consideración del comité revisor conformado por:

**M.C. Eida Reyes Varela, M.C Dolores Gabriela Palomares Gorham**

Se avala la liberación para revisión por el director de proyecto.

Atentamente

**ANDRÉS GONZÁLEZ RODRIGUEZ**  
No. Control: 14480653

Vo.Bo.

**M.C JOSÉ ISIDRO HERNÁNDEZ VEGA**  
Director de Tesis

c.c.p. Departamento de Sistemas y Computación  
c.c.p. Expediente  
c.c.p. Interesados



Av. Eloy Cavazos No. 2001, Col. Toluca, C.P. 67170  
Guadalupe, N.L., Tels. 8157 0500 - 8157 0505, Facebook: Difusión Itnl, Twitter: @TECNL  
[www.itnl.edu.mx](http://www.itnl.edu.mx)

