UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA ELECTRONICA Y TELECOMUNICACIONES

"Trabajo de grado previo a la obtención del Título de Ingeniero en Electrónica y Telecomunicaciones"

TRABAJO DE GRADUACIÓN

Título del proyecto:

"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA Y TELEMANDO UTILIZANDO TECNOLOGÍA MACHINE TO MACHINE (M2M) PARA CONTROL Y MONITOREO DEL LABORATORIO DE TELEVISIÓN DIGITAL DE LA UNACH."

Autores: Galo Eduardo Santillán Meneses

Daniel Leonardo Haro Cuadrado

Director: Ing. Anibal Llanga

Riobamba – Ecuador

AÑO 2015

Los miembros del Tribunal de Graduación del proyecto de investigación de título: **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA Y TELEMANDO UTILIZANDO TECNOLOGÍA MACHINE TO MACHINE (M2M) PARA CONTROL Y MONITOREO DEL LABORATORIO DE TELEVISIÓN DIGITAL DE LA UNACH** presentado por: Galo Eduardo Santillán Meneses, Daniel Leonardo Haro Cuadrado y dirigida por: Ing. Anibal Llanga.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

CARLOS PEÑARIEL
Presidente del Tribunal

Firma

Miembro del Tribunal

Firma

DEYSI JUCA BALSECA
Miembro del Tribunal

Firma

AUTORÍA DE LA INVESTIGACIÓN

"La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Galo Eduardo Santillán Meneses, Daniel Leonardo Haro Cuadrado; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo."

Galo Eduardo Santillán Meneses
C.I. 0604433102

Daniel Leonardo Haro Cuadrado
C.I. 1600708687

AGRADECIMIENTO

Agradezco a la Facultad de Ingeniería en especial a los profesionales que formaron y forman parte de la Carrera de Ingeniería en Electrónica y Telecomunicaciones, y de manera especial al ingeniero Daniel Santillán, por la ayuda intelectual y material recibida para la realización de la presente investigación. Además a mis padres, familiares y amigos por el entendimiento y continuo apoyo a lo largo de mi carrera universitaria.

Galo Eduardo Santillán Meneses.

Agradezco a Dios y a mi familia que ha sido pilar fundamental y fuente de apoyo constante e incondicional en toda mi vida.

A mi querida Facultad de Ingeniería y a la Escuela de Ingeniería en Electrónica y Telecomunicaciones, a los docentes profesionales que formaron parte de mi vida estudiantil.

Al ingeniero Daniel Santillán, por sus conocimientos, su manera de trabajar, su paciencia y motivación para la culminación de este proyecto de investigación.

Daniel Leonardo Haro Cuadrado

DEDICATORIA

A Dios por ser mi guía, protección y fuente de sabiduría a mis padres, a mis hermanas, a mis apreciados profesores y a todos mis amigos y seres queridos, quienes me brindaron su apoyo total e incondicional para concluir mi tesis.

Galo Eduardo Santillán Meneses.

A Dios por permitirme llegar a este momento tan especial en mi vida. Por darme fuerza y humildad ante los triunfos y caídas, que han hecho que valore más la vida.

A mis padres, a mis hermanos, a mi esposa y en especial a mi querida hija Daniela Marisol quienes me han dado su apoyo incondicional y demostrado que tienen fe en mí.

Daniel Leonardo Haro Cuadrado.

ÍNDICE GENERAL

AUTORÍA DE LA INVESTIGACIÓN ii
AGRADECIMIENTO iv
DEDICATORIA vi
INDICE DE FIGURAS xi
RESUMEN xiv
SUMMARY xv
INTRODUCCIÓN 1
CAPÍTULO I 3
1. FUNDAMENTACIÓN TEÓRICA 3
1.1. Antecedentes del tema 3
1.2. Enfoque teórico. 4
1.2.1. La telemetría 4
1.2.2. El telecontrol 4
1.2.3. M2M 5
1.2.3.1. Arquitectura de M2M 6
1.2.3.2. Elementos de la arquitectura M2M 6
1.2.3.3. Tarjetas SIM para M2M 11
1.2.3.4. Etapas de trabajo M2M 12
1.2.3.5. Funcionamiento M2M 12
1.2.3.6. Características de los dispositivos finales. 14
1.2.3.7. Normas M2M. 15
1.2.3.8. Aplicaciones de M2M. 17
1.2.3.9. M2M en la industria. 21
1.2.3.10. Nuevos requerimientos potenciales. 22
1.2.4. Televisión digital. 23
1.2.4.1. Transmisión y recepción de sistema ISDB-TB. 24
1.2.4.2. Medición y monitoreo de ISDB-TB. 24
1.2.5. Internet of things “IoT” (internet de la cosas). 25
1.2.6. Equipos 27
1.2.6.1. STM32F4 Discovery 27
1.2.6.2. KAISSEN PRO 8000 28

1.2.7. Software 36
1.2.7.1. CooCox 1.7.8. 36
1.2.7.2. Android Studio 1.0.1 37
1.2.7.3. Eclipse 4.2. 38
1.2.7.4. Android 38
1.2.8. Servicio web 41
1.2.9. PHP 42
1.2.10. Servicio web ligero o peso pluma 46
1.2.11. Base de datos MYSQL 47
2. METODOLOGÍA 48
2.1. Tipo de Estudio. 48
2.2 Población Muestra. 48
2.2.1. Población. 48
2.2.2. Muestra. 49
2.2.3. Hipótesis. 49
2.3 Operacionalización de Variables 49
2.4. Procedimientos 52
2.5. Procesamiento y Análisis 52
2.5.1. Sistemas de telemetría. 52
2.5.2. Desarrollo del prototipo. 53
2.6 Comprobación de la hipótesis 95
CAPÍTULO III 109
3. RESULTADOS 109
CAPÍTULO IV 113
4. DISCUSIÓN 113
CAPÍTULO V 115
5. CONCLUSIONES Y RECOMENDACIONES. 115
5.1 Conclusiones 115
5.2 Recomendaciones. 116
CAPÍTULO VI 117
6. PROPUESTA 117
6.1 Título de la Propuesta. 117

6.2 Introducción 117
6.3 Objetivos 117
6.3.1 General 117
6.3.2 Específicos: 118
6.4 Fundamentación Científica – Técnica. 118
6.5. Descripción de la Propuesta 119
6.6. Diseño Organizacional 120
6.7 Monitoreo y Evaluación de la propuesta. 120
CAPÍTULO VII 121
7. BIBLIOGRAFÍA 121
CAPÍTULO VIII 123
8. ANEXOS 123
Anexo 1 123
Anexo 2 125
Anexo 3 128
Anexo 4 131
Anexo 5 134

INDICE DE FIGURAS

Figura 1. Objetivo de la comunicación M2M 5
Figura 2. Arquitectura básica de M2M 6
Figura 3 Arquitectura y elementos de la comunicación M2M..... 7
Figura 4. Tecnologías usadas de los dominios M2M..... 8
Figura 5. Diagrama de bloques de arquitectura M2M 9
Figura 6. Módems M2M 10
Figura 7. Sim card M2M..... 11
Figura 8. La esencia de M2M 13
Figura 9. Gestión de flota..... 13
Figura 10. Gestión mediada por dispositivos inteligentes 14
Figura 11. Escenario inteligente de implementación de medición típica PLC/celular 18
Figura 12. Escenario de implementación típica de los sistemas de medición inteligente..... 20
Figura 13. Ejemplo de una aplicación M2M..... 21
Figura 14. Ejemplo de un primer escenario: Al accionar el Medio Ambiente 23
Figura 15. Evolución Internet de las Cosas..... 26
Figura 16. Funcionamiento de la SIM..... 26
Figura 17. Placa STM32F4 27
Figura 18. Tablet KAISSEN PRO 8000 28
Figura 19. Sensor LM35 29
Figura 20. Diagrama eléctrico del sensor LM35..... 30
Figura 21. ZFM 20 lector de huellas..... 31
Figura 22. Detector de humo Fotoeléctrico..... 32
Figura 23. Sensor de movimiento MC420s 33
Figura 24. Chapa eléctrica marca DURAL 34
Figura 25. Ventilador 35
Figura 26. Programa CooCox 36
Figura 27. Programa Android Studio 1.0.1..... 37
Figura 28. Programa eclipse JUNO 4.2. 38
Figura 29. Logotipo Android 39
Figura 30. Gráfico pastel distribución de versiones Android. 40
Figura 31. Esquema de intercambio de datos..... 42
Figura 32. Cliente intercambiando información con el servidor web..... 43
Figura 33. Medios para acceso a un web service..... 43
Figura 34. Consulta y respuesta desde un smartphone 44
Figura 35. Acceso desde cualquier dispositivo..... 44
Figura 36. Interacción Cliente Android y Web Service mediante PHP..... 45
Figura 37. Uso del protocolo GET/POST..... 46

Figura 38. Diagrama de bloques del sistema 54
Figura 39. Sensores y actuadores conectados al concentrador 55
Figura 40. Tarjeta STM32f4 56
Figura 41. Conexión placa-tablet. 62
Figura 42. IDE Android Studio 64
Figura 43. Flujograma de programación 65
Figura 44. Pantalla principal 66
Figura 45. Pantalla de identificación de huella 68
Figura 46. Pantalla de identificación del administrador 70
Figura 47. Pantalla de administración 71
Figura 48. Pantalla para agregar usuario 72
Figura 49. Funcionamiento básico del servidor VILLAGEFLOW 74
Figura 50. Espacio de VillageFlow configurado 75
Figura 51. Archivo VF_CONF.XML 76
Figura 52. Parámetros de configuración del Brick EPG 77
Figura 53. Parámetros de configuración del Brick EPG 77
Figura 54. Parámetros de configuración del Brick REMUX SIMPLE 78
Figura 55. Parámetros de configuración del TMCC ENCODING 78
Figura 56. Parámetros de configuración del Brick TMCC ENCODING 79
Figura 57. Parámetros del Brick DEKTEC OUTPUT CARD 79
Figura 58. Funcionamiento y comunicación del servidor web 80
Figura 59. Entorno de MySQL 81
Figura 60. Archivos php en el dominio. 84
Figura 61. Archivos php en el servidor. 84
Figura 62. Almacenamiento de información. 87
Figura 63. Intercambio de información. 89
Figura 64. Pantalla de la aplicación para smartphone 90
Figura 65. Pantalla de monitoreo del servidor de televisión 93
Figura 66. Archivo de configuración del servidor de tv-digital 94
Figura 67. Diseño Organizacional 120
Figura 68. Pantalla de identificación del administrador 137

INDICE DE TABLAS

Tabla 1 . Simbología del Sensor LM35 31
Tabla 2 . Características Sistema Operativo Android. 39
Tabla 3 . Porcentajes de uso de las versiones de Android. 40
Tabla 4 . Operacionalización de la variable independiente..... 50
Tabla 5 . Operacionalización de la Variable dependiente. 51
Tabla 6 . Pines disponibles de la tarjeta distribuidos por puertos 60
Tabla 7. Pines asignados para comunicación 60
Tabla 8. Configuración del Brick TS FILE IN 76
Tabla 9. Tabla de base de datos alarmas UNACH. 82
Tabla 10. Tabla de base de datos registro UNACH. 83
Tabla 11. Comprobación del tiempo de activación del sensor de temperatura y activación del equipo de enfriamiento. 98
Tabla 12. Resumen tiempo de activación sensor de temperatura y activación de equipo de enfriamiento..... 98
Tabla 13. Análisis de Varianza tiempo de activación sensor de temperatura y activación de equipo de enfriamiento 99
Tabla 14. Comprobación del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles. 101
Tabla 15. Resumen del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles. 101
Tabla 16. Análisis de varianza del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles. 102
Tabla 17. Comprobación del tiempo de activación de apertura de la puerta de forma remota. 104
Tabla 18. Resumen de la activación de apertura de la puerta de forma remota. 104
Tabla 19. Análisis de varianza de la activación de la puerta de forma remota. .. 105
Tabla 20. Comprobación del tiempo de adquisición de señales del servidor de televisión digital. 107
Tabla 21. Resumen adquisición de señales del servidor de televisión digital. ... 107
Tabla 22. Análisis de varianza adquisición de señales del servidor de televisión digital. 108
Tabla 23 Muestras del tiempo de activación del equipo de enfriamiento. 110
Tabla 24. Muestras del tiempo de respuesta para la llegada de notificaciones de los eventos..... 110
Tabla 25 Muestras del tiempo de activación de la chapa eléctrica 111
Tabla 26. Muestras del tiempo de respuesta para las consultas al servidor de televisión digital. 111
Tabla 27. Parámetros del servidor de televisión a ser monitoreados. 112
Tabla 28. Muestras del tiempo de respuesta para las consultas al servidor de televisión digital. 113
Tabla 29. Muestras cumplen trabajo en tiempo real 114

RESUMEN

Se diseñó e implemento un sistema de telemetría y telemando utilizando tecnología machine to machine (M2M) para control y monitoreo del laboratorio de televisión digital de la Universidad Nacional de Chimborazo. Con el objetivo de obtener datos en tiempo real del laboratorio así como del servidor de televisión para verificar su estado y funcionamiento de forma remota.

El prototipo se diseñó enteramente en lenguaje de programación JAVA, bajo diferentes plataformas como CooCox, Eclipse y Android Studio los cuales son software de código abierto. Y bajo parámetros básicos de comunicaciones maquina a máquina.

El proyecto de graduación final se lo desarrollo utilizando el método investigativo, el cual ayudo a encontrar una forma viable de intercambiar datos entre el emisor del sistema hacia el receptor y viceversa, además de hallar una manera estable para realizar el monitoreo y control de los parámetros de funcionamiento del servidor de televisión digital. Para cada etapa del desarrollo, se recabó toda la información posible sobre los distintos elementos como: los sensores, actuadores, controlador y demás dispositivos que intervinieron en la realización del proyecto.

El sistema funciona dentro de los parámetros establecidos para ser considerado un sistema de tiempo real; dando como resultado un tiempo de respuesta menor a un segundo gracias a la utilización del internet móvil como canal de comunicación.

Se concluye que al combinar nuevas tecnologías con otras ya existentes se pueden crear sistemas de monitoreo y control muy versátiles y económicos.

Recomendamos a los estudiantes y usuarios del sistema continuar con la investigación para ampliar nuevas funciones en el sistema de telemetría.

SUMMARY

INTRODUCCIÓN

Durante más de 100 años se ha ido innovando en la comunicación entre personas, en la actualidad la apuesta para los próximos años consiste en desarrollar las comunicaciones entre objetos inanimados. Las comunicaciones M2M están transformando sectores, desde los servicios básicos hasta el transporte, al ofrecer formas más eficientes, baratas y sostenibles de desarrollar los modelos de negocio y hacer posibles nuevos negocios. (Labarca, 2013).

M2M es un sistema informático que permite el intercambio de datos entre dos máquinas remotas, el objetivo es el control y la supervisión de diferentes procesos automatizados. Además según enuncia en el diario Expreso se estima que para el 2020 habrán en el mundo más de 50.000 millones de dispositivos conectados, 1.500 millones de automóviles con tecnología M2M y 3.000 millones de dispositivos M2M al interior de los hogares.

De cumplirse estas predicciones, además de facilitar las tareas a los habitantes, el uso inteligente de dicha conectividad permitirá logros importantes en la salud, medio ambiente, transporte, minería, pesca, energía, educación, y hasta en deporte. (Espinoza, 2013).

Por lo cual nace la necesidad de estudiar e implementar la tecnología M2M abarcando los dos mayores campos para los que fue creada como son la telemetría y el telecontrol de automatismos.

El problema consiste en lograr la comunicación de máquinas de manera autónoma, utilizando como canal de comunicación el internet móvil de la red de telefonía celular dado que el proyecto pretende lograr desarrollar un sistema de telemetría y telemando utilizando tecnología M2M combinando características o servicios fundamentales que ofertan la operadoras como: los mensajes de texto y el acceso a internet; se ha optado por utilizar como canal de comunicación a la red de telefonía celular ya que es la más óptima por costo y rendimiento.

Se justifica debido a que se intenta crear un sistema de telemetría capaz de obtener datos en tiempo real en cualquier lugar donde se encuentre el administrador del laboratorio; dado que el alcance de la red celular está presente en casi todas las zonas geográficas urbanas y que la tecnología M2M permite utilizar las diferentes tecnologías celulares desde la 2G hasta 4G se lograra crear un sistema robusto.

Entonces se obtendrán datos en tiempo real del servidor de televisión digital lo que permitirá monitorear parámetros de configuración y trasmisión del servidor de televisión. Además se implementara un control de medio ambiente del laboratorio que se regulara automáticamente, para lo cual se creara un sistema con su debida interfaz que permita el control automático del medio ambiente del laboratorio; creyendo pertinente agregar además un control de acceso al laboratorio mediante acceso remoto y también control de acceso mediante huella dactilar; lo que permitirá llevar un control del ingreso del personal esta información se almacenara en una base de datos para tener un registro exacto de accesos al laboratorio y de las alarmas preventivas que se lleguen a generar.

Se diseñó una aplicación para sistemas android que monitorea en tiempo real los datos que se generen en el sistema de telemetría acerca del funcionamiento de los sensores y actuadores como el estado del servidor de televisión de manera remota; logrando visualizar dicha información desde cualquier lugar a través de cualquier dispositivo móvil con sistema android.

Lo que se va a comprobar es si el diseño e implementación de un sistema de telemetría y telemando utilizando tecnología machine to machine (M2M) permitiría el intercambio de información en formato de datos para el control y monitoreo del laboratorio de televisión digital de la UNACH en tiempo real.

En el capítulo 1 se presenta la información recopilada acerca de las tecnologías y equipos a utilizarse en el proyecto, en el capítulo 2 se describen los procesos y diseños utilizados en la realización del proyecto y obtención de datos estadísticos, en el capítulo 3 se analiza los resultados obtenidos en las pruebas; mientras que en el capítulo 4 se realiza la discusión de dichos resultados y en el capítulo 5 se exponen las conclusiones y recomendaciones obtenidas.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. Antecedentes del tema

Con la aparición de dispositivos móviles ha cambiado la manera de cómo acceder a internet se ha multiplicado la cantidad de aparatos electrónicos con la cual podemos accesar a la red, pero la conexión a internet ya no es solo posible a través de estos dispositivos; todos los objetos que nos rodean pueden estar conectados siendo capaces de recolectar información, procesarla y compartirla, lo cual ofrece un mundo nuevo de posibilidades.

Diariamente crece el número de objetos y dispositivos que usan internet lo que contribuye a la construcción de un planeta inteligente; se estima que para el año 2020 habrá más de 50000 millones de aparatos conectados a internet un promedio de 6 dispositivos por cada habitante del planeta. (Caceres, 2014)

M2M es la próxima revolución de la tecnología que conecta a las personas, los sistemas inteligentes y dispositivos en nuevas y transformadoras maneras, proporcionando además valor sustancial en términos de utilidad.

En el país existen tesis referentes a telemetría y telecontrol que utilizan las redes celulares, como ejemplo en la FIE de la ESPOCH reposan las tesis "Mantenimiento y Diagnóstico Remoto de un Equipo Mecatrónico Usando un Módulo GSM" de los autores Viera Mesa, Andrea Enriqueta y "Diseño e implementación de un módulo de comunicación para el control y monitoreo de un MPS vía GSM" de los autores Quito Pérez, Diego Javier; y en otras universidades del país como en la UTA existe el tema de tesis "Implementación del Sistema M2M aplicado a la Telemetría en el Laboratorio OMRON"; con la particularidad que dichos proyectos solo utilizan los mensajes cortos SMS y códigos AT&T para el intercambio de información.

1.2. Enfoque teórico.

1.2.1. La telemetría.

Consiste en la transmisión de datos desde un sensor a un servidor donde los datos son almacenados para su administración y visualización, usando modernos sistemas de gestión diseñados a medida de las necesidades de cada cliente; que permiten visualización on-line de eventos y alarmas, generación de reportes estadísticos, almacenamiento de datos históricos, etc. (pc-mobil, 2014)

La telemetría es una técnica automatizada de las telecomunicaciones con la ayuda de las mediciones y recopilación de datos que se obtienen en lugares remotos y de transmisión para la vigilancia; se utiliza comúnmente transmisión inalámbrica, aunque fue creada originalmente para sistemas de transmisión por cable. Los usos más importantes de telemetría incluyen la recopilación de datos del clima, supervisión de plantas de generación de energía, etc.

1.2.2. El telecontrol

Es el control de un equipo operativo a cierta distancia utilizando la transmisión de la información mediante técnicas de telecomunicación. Consiste en el uso de actuadores con capacidad de ejecutar acciones tales como interruptores, bloquear/desbloquear dispositivos de seguridad, abrir/cerrar válvulas, aumentar/disminuir flujos de fluidos, intensidad de velocidad, voltaje, etc.

El sistema de gestión Telemático provee de las variables necesarias para que el sistema de Telecontrol realice las acciones, convirtiendo el sistema en autónomo y fácilmente gestionable. (pc-mobil, 2014).

1.2.3. M2M

Es el intercambio de información en formato de datos, entre dos máquinas remotas, sin intervención humana, por supuesto, en un espacio urbano con infraestructuras, redes y plataformas inteligentes, con millones de sensores, dentro de los que se incluyen también a los habitantes y sus dispositivos móviles. (Espinoza, 2013).

M2M es una forma de comunicación entre máquinas (Machine To Machine) que permite agilizar los procesos, realizarlos de forma más eficiente y liberar a las personas de tareas tediosas o peligrosas.

La comunicación autónoma entre máquinas es ya una realidad y cada día se desarrollan nuevas aplicaciones en todos los campos: Desde domótica y seguridad en el hogar, a gestión de flotas y logística, etc.

El concepto M2M (máquina a máquina) es, en síntesis, la capacidad de intercambiar datos entre dos máquinas remotas, de forma que mediante este intercambio, es posible controlar y supervisar de forma automática procesos en los que intervienen máquinas como se muestra en la figura 1.

Figura 1. Objetivo de la comunicación M2M

Fuente: http://www2.alcatel-lucent.com/techzine/es/la-internet-de-las-cosas-la-hora-de-la-estandarizacion/

1.2.3.1. Arquitectura de M2M

La arquitectura M2M se puede definir como un terminal de comunicaciones que no depende de la interacción con las personas y que se comunica con una red troncal o con otro terminal para funciones de automatización de servicios, como se describe en la Figura 2.

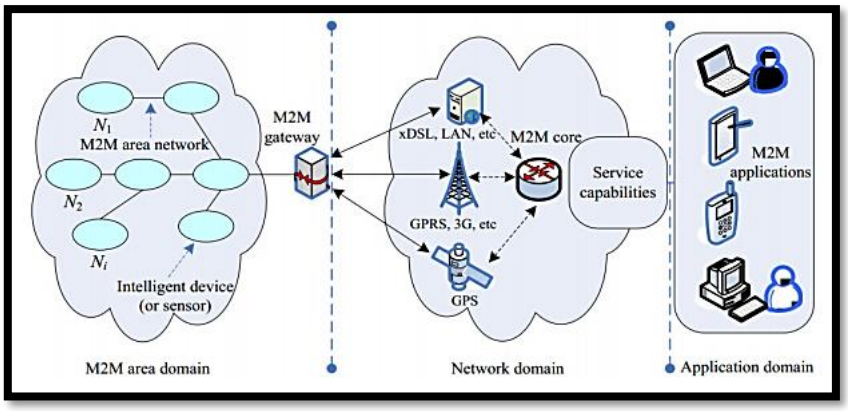


Figura 2. Arquitectura básica de M2M

Fuente: http://blog.altran.es/telecomunicaciones-media/estandarizacion-m2m-de-la-etsi/

Para definir la arquitectura de red se enunciaran los elementos que la forman y los dominios de red que crean estos elementos. Existen etapas que describen las operaciones llevadas a cabo, siendo éstas iguales independientemente de la función que deba cubrir el sistema.

1.2.3.2. Elementos de la arquitectura M2M

1.2.3.2.1. Dominios de M2M

Dentro de la estandarización de telecomunicaciones, el grupo **ETSI** (European Telecommunication Stardars Institute)*;* se encarga de definir la estandarización M2M cuyo planteamiento básico se define en los siguientes 3 dominios que se muestran en la figura 3.

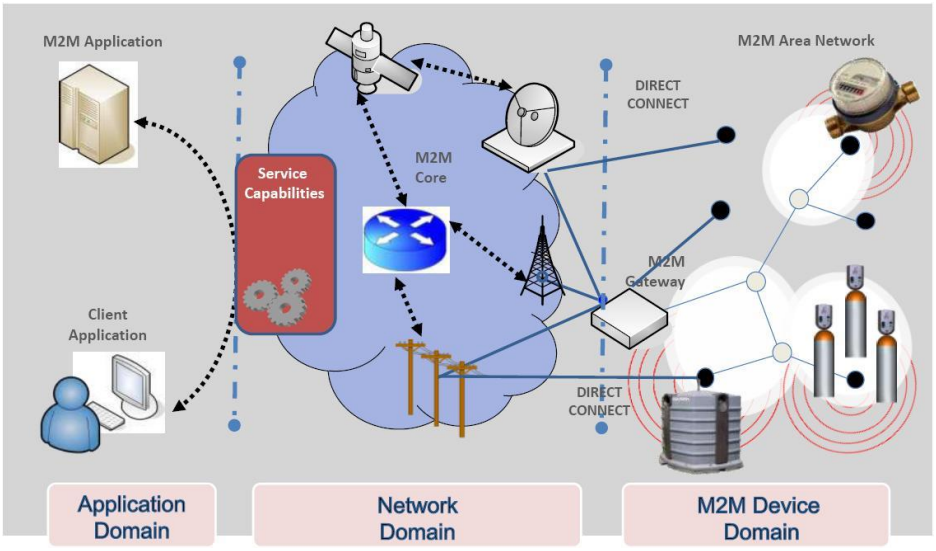


Figura 3 Arquitectura y elementos de la comunicación M2M

Fuente: http://blog.altran.es/telecomunicaciones-media/estandarizacion-m2m-de-la-etsi

La arquitectura y elementos de la comunicación machine to machine se muestran en la figura 3 divididos en sus respectivos dominios.

1.- M2M device domain: Dispositivos M2M capaces de responder ante una petición o de transmitir datos de forma automática.

2.- Network domain (red de dominio)

Dada la diferente naturaleza de los dispositivos M2M, este dominio hace referencia a los niveles de interconexión de redes para llevar a cabo dicha transmisión de información. En este punto se encuentran los gateways y redes de acceso (entre los dispositivos y los gateways, y entre los gateways y el dominio de aplicación).

La red de dominio está compuesto por el Access Network y el Core Networks que a continuación se detallan:

El Access Network conecta dispositivos M2M y Gateway con el Core Network; puede usarse diferentes tecnologías.

Core Networks proporciona conectividad IP, funciones de servicio y control de red, interconexión con otras redes y roaming.

3.- Application domain: Es la interfaz por el que circulan los datos a través de aplicaciones de servicio y donde se haya el motor de los procesos propios del servicio. Se habilitan aplicaciones y comunicaciones genéricas.

En la figura 4 se indican las tecnologías de los 3 dominios, que se aplican actualmente sobre cada uno de los dominios:

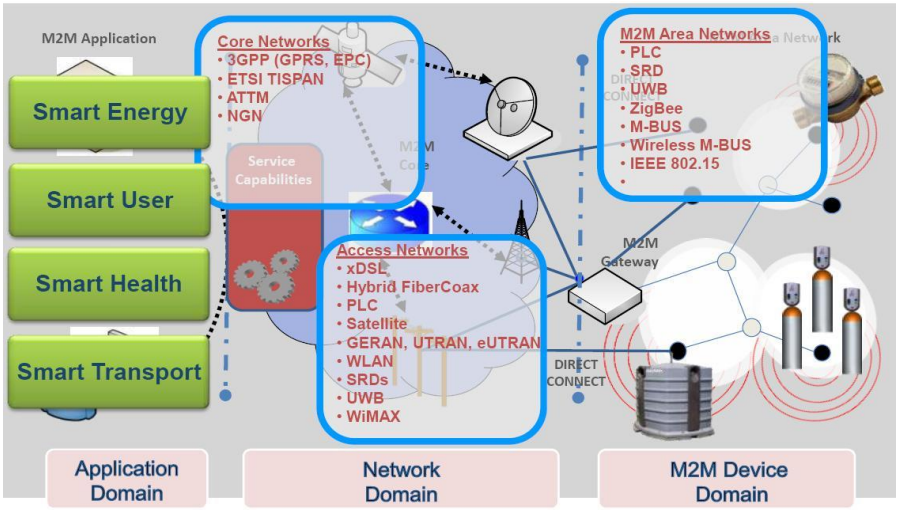


Figura 4. Tecnologías usadas de los dominios M2M.

Fuente: http://blog.altran.es/telecomunicaciones-media/estandarizacion-m2m-de-la-etsi/

1.2.3.2.2. Elementos de M2M

Basándose en los elementos que aparecen en las comunicaciones M2M, el European Telecommunications Standards Institute (ETSI) ha definido la arquitectura de éste tipo de redes, en los siguientes elementos que se muestran en la figura 5 en diagrama de bloques de la arquitectura M2M con sus respectivos elementos y dominios.

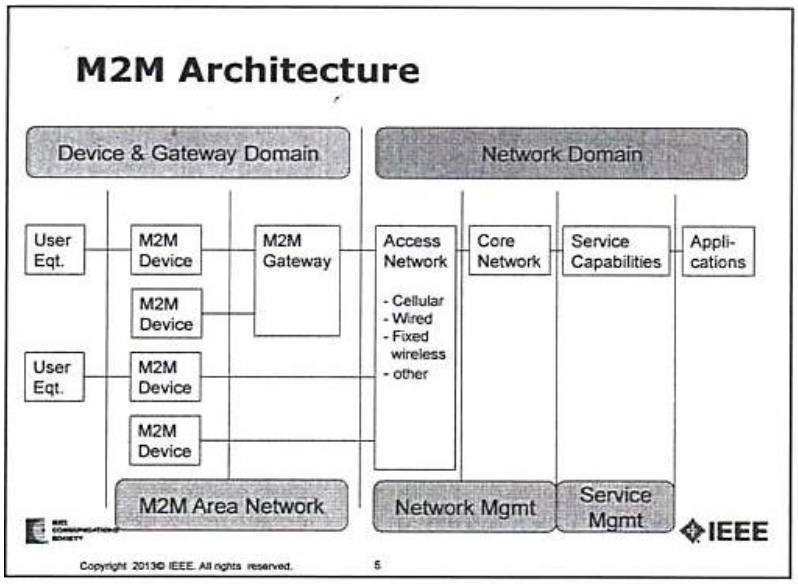


Figura 5. Diagrama de bloques de arquitectura M2M

Fuente: http://ieee802.org/16/m2m/

A continuación se describen los 5 elementos de la arquitectura:

1.- User_equipment

El equipo de usuario es cualquier dispositivo utilizado directamente por el usuario final para comunicarse. Puede ser un teléfono de mano, un ordenador portátil equipado con una banda ancha móvil adaptador, o cualquier otro dispositivo.

2.- M2M device (Dispositivo M2M**)**

Ejecuta aplicaciones M2M usando capacidades de servicio M2M. Dispositivos con conectividad directa incluyen las funciones de:

- Registro
- Autentificación
- Autorización
- Administración

3.- M2M gateway

Gateway actúa como un proxy de red; es decir sirve de intermediario entre un navegador Web e Internet. El Proxy contribuye a la seguridad de la red. En la figura 6 se muestran algunos de los módems M2M disponibles en el mercado.

Figura 6. Módems M2M

Fuente: http://www.powertec.com.au/netcomm-ntc-4908-m2m-serial-modem

Los dispositivos M2M pueden estar conectados a la red de dominio vía múltiples Gateways M2M

4.- M2M area network.

Esta red está formada por los dispositivos inteligentes y los Gateways. Se observan ejemplos de este tipo de redes en tecnologías como Zigbee, Bluetooth, WiFi, etc.

5.- M2M service capabilities

- Habilitación de aplicaciones
- Comunicación genérica
- Selección de comunicación
- Seguridad

1.2.3.3. Tarjetas SIM para M2M

Una SIM M2M son parte esencial del equipo para soluciones de comunicación de máquina es la tarjeta SIM. Los dispositivos móviles no se pueden comunicar sin una tarjeta SIM, y M2M ofrece una gama de tarjetas SIM que incluye la SIM de alta duración industrial para los dispositivos M2M para completar una solución.

Las tarjetas SIM Industrial son más robustas y vienen en el factor de forma estándar con una mayor resistencia a la corrosión para resistir las condiciones ambientales más extremas para las exigentes aplicaciones M2M. El aumento de la esperanza de la fiabilidad y la vida útil de las tarjetas SIM Industrial reduce el riesgo de posibles fallos y el tiempo y el costo de reemplazo.

En la figura 7 se muestran una tarjeta SIM y sus aplicaciones.

Figura 7. Sim card M2M

Fuente: http://www.maquina-a-maquina-m2m.es/sim-global.html

Muchos dispositivos M2M incorporan tarjetas SIM normales que permiten una conexión independiente. La tarjeta SIM crea la conexión a la red al tratarse de una unidad central para seguridad y control de accesos.

1.2.3.4. Etapas de trabajo M2M

Usualmente cada aplicación M2M tendrá sus propias etapas de funcionamiento, existen cuatro etapas básicas que suelen ser comunes.

- Etapa de recolección de datos. lleva a cabo la recolección de información, que puede tener distintos orígenes dependiendo de la aplicación y servicio. Por ejemplo, puede recoger información del medio donde se encuentra, información del estado del propio dispositivo, información de las operaciones que lleva a cabo, etc.

- Etapa de transmisión de datos a través de la red. Atendiendo a la arquitectura de la red existen dos redes visiblemente diferenciadas y separadas por el Gateway, la etapa de transmisión de datos engloba el tráfico de las ambas. Desde que la información es transmitida por el dispositivo final hasta que es recibida por la aplicación; desde que la respuesta es emitida por la aplicación hasta que llega al dispositivo.

- Etapa de valoración de los datos. Los datos recibidos por el centro de control, son evaluados y almacenados.

- Etapa de respuesta a la información disponible. Se generan respuestas en consecuencia a los datos recibidos en el centro de control. Puede no existir respuesta a los datos recibidos en caso de no ser necesario o puede ser una respuesta dirigida a un elemento externo de la red M2M, en forma de alarma.

1.2.3.5. Funcionamiento M2M

Tal vez la forma más básica para describir M2M se muestra en la Figura 8 (la esencia de M2M). El papel de M2M es establecer las condiciones que permiten a un dispositivo bidireccional el intercambio de información con una aplicación a través de una red de comunicación, de modo que el dispositivo o aplicación puede actuar como la base para este intercambio de información.

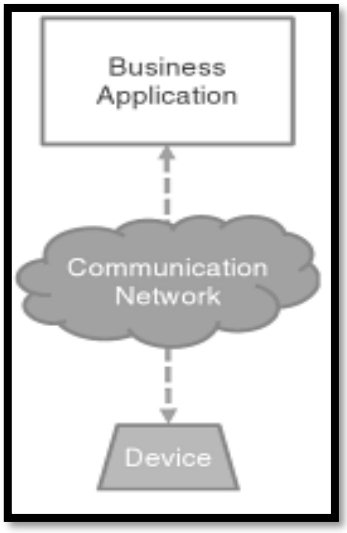


Figura 8. La esencia de M2M

Fuente: http://ieee802.org/16/m2m/

M2M implica a un grupo de dispositivos similares que interactúan con una sola aplicación, como se muestra en la Figura 9. Gestión de flota es un ejemplo de una aplicación de este tipo, donde los dispositivos son, por ejemplo, camiones, y la red de comunicaciones es una red móvil.

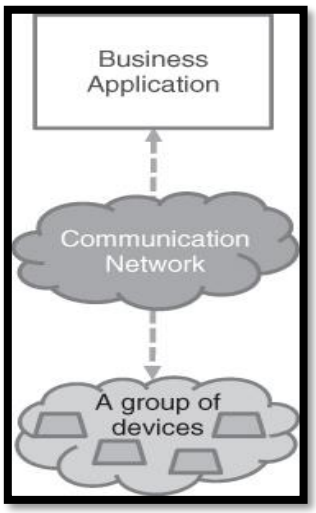


Figura 9. Gestión de flota

Fuente: http://ieee802.org/16/m2m/

En algunos casos, como se muestra en la Figura 10, los dispositivos en el grupo pueden no interactuar directamente con la aplicación debido a que tiene sólo una capacidad limitada. La medición inteligente es un ejemplo de una aplicación de este tipo.

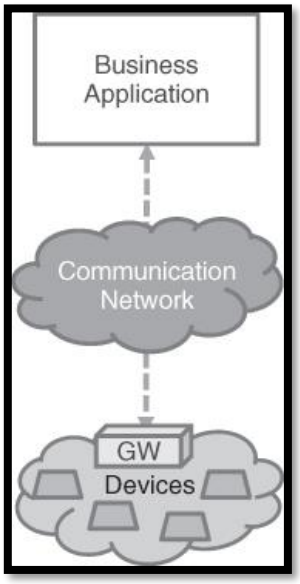


Figura 10. Gestión mediada por dispositivos inteligentes

Fuente: http://ieee802.org/16/m2m/

1.2.3.6. Características de los dispositivos finales.

Multitud.- Este es el cambio más defendido provocada por M2M. El número de "dispositivos" conectados en relación a M2M superará pronto en gran medida la suma de todos aquellos que interactúan directamente con los seres humanos.

Variedad.- Ya hay un número particularmente elevado de casos de posible uso documentado de M2M que se aplican a una variedad de contextos y ámbitos empresariales. Las implementaciones iniciales de las aplicaciones M2M ya han dado lugar a la aparición de una gran variedad de dispositivos con muy diversas necesidades en términos de tasa de intercambio de datos, factor de forma, la informática, o la capacidad de comunicación.

Invisibilidad.- Esta es una fuerte exigencia en muchas aplicaciones M2M: los dispositivos tienen que entregar de forma rutinaria su servicio con muy poco o ningún control humano. En particular, esto es la prevención de los seres humanos a partir de la corrección de errores. Como resultado, la administración de dispositivos, más que nunca se convierte en una parte clave del servicio y la gestión de la red y tiene que integrarse a la perfección.

Criticidad.- Algunos dispositivos son salvavidas, como en el campo de la salud en línea (captores de sangre, detectores, etc.). Algunos son elementos clave de las infraestructuras críticas para la vida, tales como detectores de tensión o de fase, interruptores, etc. en la red inteligente. Su uso impone requisitos estrictos sobre la latencia o la fiabilidad, que pueden desafiar o exceder las capacidades de las redes de hoy en día.

1.2.3.7. Normas M2M.

A diferencia de varios otros segmentos TIC donde puede ser posible desplegar sistemas operativos a pesar de la falta de normas, varios segmentos del mercado M2M exigen un estándar definido para garantizar la protección de la inversión a largo plazo. Durante varias aplicaciones M2M, incluyendo la medición inteligente o Smart Grids, existe la expectativa de que el equipo instalado se desplegará durante más de 20 años.

Si bien un periodo tan largo de vida para un proyecto como este puede parecer poco realista para implementaciones tradicionales, las infraestructuras desplegadas por empresas de servicios públicos tienen ciclos de desarrollos muy largos que pueden influir drásticamente su diseño y en adelante las normas relacionadas.

Aunque existe un acuerdo común de que el mercado aún carece de normas para M2M, la situación ha ido evolucionando, aunque el nivel de madurez de los estándares M2M aún varía, dependiendo del segmento de las normas. Ahora es

cada vez relativamente claro lo que hay que hacer y en qué áreas técnicas y geográficas.

Las diferentes áreas en las que se necesitan normas para M2M se clasifican a continuación.

1.2.3.7.1. Modelos de Datos

Los modelos de datos determinan explícitamente la estructura de los datos intercambiados principalmente entre las aplicaciones M2M, sino también con otras entidades dentro de un sistema M2M.

1.2.3.7.2. Redes de área M2M

La red de área M2M término fue utilizado por primera vez en la ETSI TS 102 690 Especificaciones técnicas [TS 102 690]. Una red de área de M2M es un término genérico que se refiere a cualquier tecnología de red que proporciona conectividad física y la capa MAC entre diferentes dispositivos M2M conectados a la misma red de área de M2M o permitir que un dispositivo M2M para obtener acceso a una red pública a través de un router o una puerta de enlace.

1.2.3.7.3. Plataformas horizontales.

En la próxima fase del desarrollo de M2M debe de estar apoyada por plataformas que faciliten el desarrollo de aplicaciones para crear nuevos modelos de negocio. Éstas deberán reunir conjuntos de capacidades que potencien y reúnan las funcionalidades que permitan el desarrollo, prueba y despliegue de aplicaciones de alto nivel en diversos sectores. Estas plataformas pondrán a disposición de las aplicaciones un conjunto de APIs que permitan la integración de los servicios que ofrecen en distintas soluciones verticales.

1.2.3.8. Aplicaciones de M2M.

Las principales aplicaciones donde M2M juega un papel importante son la sanidad, el transporte, la energía, la seguridad y la vigilancia, la gestión de los servicios públicos, punto de venta, el control de la construcción, gestión, automatización, control industrial, domótica y la agricultura. A continuación se explica con más detalle.

1.2.3.8.1. Transporte

- Gestión de la flota
- Determinar la ubicación de los vehículos.
- Mantenimiento de vehículos
- Obtener diferentes parámetros de funcionamiento del vehículo con el fin de diagnosticar proactivamente problemas mecánicos.
- Control remoto de la ubicación y uso de datos y datos de comportamiento de los conductores para proporcionar diferentes primas de seguros a fin de reflejar los diferentes perfiles de conductor.

Casi todas las aplicaciones de transporte implican el uso de redes celulares comerciales debido al requisito de movilidad de las comunicaciones vehiculares.

1.2.3.8.2. Energía

- La medición inteligente de recogida automática de consumo, y los datos de diagnóstico de la información del estado de los dispositivos de medición de energía.
- Aplicaciones de consumo gestionados.
- Supervisar el uso de la electricidad para los aparatos personales e informar a una entidad central que proporciona servicios de gestión.
- Fuente de energía alternativa (paneles solares, viento, etc.)

Aplicaciones de energía utilizan una mezcla de redes de área doméstica, tecnologías de comunicación locales, como las comunicaciones de línea eléctrica o de malla de radiofrecuencia (RF) redes a la subestación más cercana, y las redes de telefonía fija a la ubicación central donde se procesan los datos. Algunos medidores inteligentes también se basan en las redes inalámbricas celulares.

1.2.3.8.3. Telemetría inteligente

Redes inteligentes y aplicaciones de medición inteligentes están siendo desplegadas por el sector de la energía para lograr un mayor grado de eficiencia y fiabilidad del uso de la energía.

Los medidores inteligentes y sensores Smart Grid se conectarán a través de una variedad de tecnologías alámbricas e inalámbricas, como PLC, malla RF, acceso celular, etc. Estos proporcionan conectividad de red entre medidores inteligentes, sensores o dispositivos M2M y aplicaciones de back-end de la utilidad, las bases de datos y sistemas de gestión.

En la figura 11 se puede observar una aplicación típica de M2M para la medición inteligente del uso de energía eléctrica.

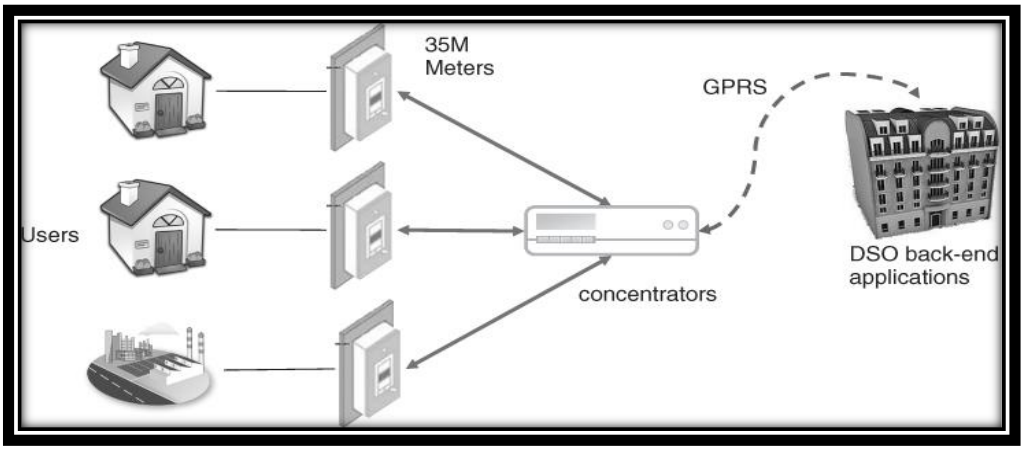


Figura 11. Escenario inteligente de implementación de medición típica PLC/celular

Fuente: http://ieee802.org/16/m2m/

1.2.3.8.4. Vigilancia y Seguridad

Dispositivos de vigilancia y seguridad se han desplegado principalmente en instalaciones residenciales y pequeñas empresas para proporcionar la imagen y video vigilancia información para aplicaciones de alarma de seguridad. Los dispositivos de seguridad en general, utilizar las redes celulares como el acceso principal o de respaldo para proporcionar conectividad a las aplicaciones de monitoreo de seguridad o para los propietarios de los locales.

1.2.3.8.5. Punto de Venta (POS)

Terminales de puntos de venta (incluyendo cajeros automáticos) proporcionan servicios tales como dispensación de efectivo, pagos, y todas las tareas de los contadores de tienda - de pago y envío, como el pago por transferencia bancaria o tarjetas de crédito, verificaciones de transacciones, informes de ventas, etc.

1.2.3.8.6. Salud

Algunas aplicaciones de salud en línea son las siguientes:

- RPM: permite a los proveedores de salud para monitorear y diagnosticar las condiciones de salud mediante la recopilación de forma remota, almacenamiento, recuperación y análisis relacionados con la salud - paciente información. Dispositivos RPM permiten a los proveedores de atención médica para tratar a los pacientes antes de que sus condiciones se vuelven más agudas, evitando así desplazamientos innecesarios a los servicios de urgencias y reingresos al hospital.
- Sistema de respuesta de emergencia personal (PERS): proporciona al paciente el acceso a un centro de atención telefónica 24/7 para obtener ayuda. Dispositivos desplegados se llevan en un colgante del cuello o pulsera y permiten el acceso inmediato a los servicios de asistencia a través de la pulsación de un solo botón.

1.2.3.8.7. Escenario de implementación

- Figura 12 proporciona un escenario de implementación equipamiento típico de un sistema de medición inteligente. Este escenario muestra los dispositivos inteligentes de medición (por ejemplo, válvulas, medidor eléctrico, medidor de gas y medidor de agua) que están conectados a los centros de datos a través de una pasarela de comunicaciones.

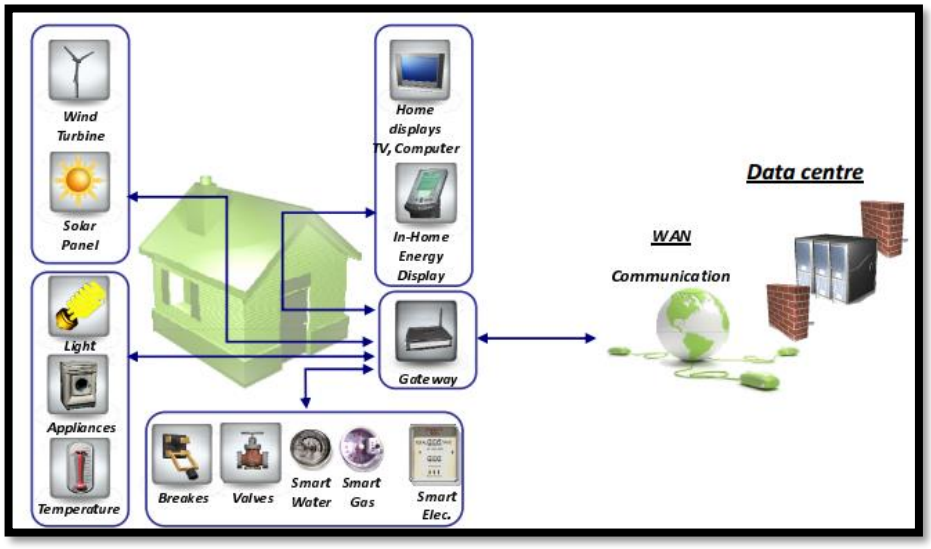


Figura 12. Escenario de implementación típica de los sistemas de medición inteligente
Fuente: http://ieee802.org/16/m2m/docs/80216p-11_0029r1.pdf

- El centro de datos recoge los datos de los dispositivos de medición inteligente y es capaz de controlar los dispositivos de metro inteligente pertinentes de forma remota a través de la puerta de enlace de comunicaciones.
- En este escenario, la puerta de enlace proporciona una interfaz para los dispositivos de automatización del hogar, como los sensores (por ejemplo, sensores de temperatura), las pantallas y los aparatos y de la electricidad al micro- generadores instalados en entornos residenciales. (Boswarthick, Elloumi, & Hersent, 2012)

En la figura 13 se observa el uso de la tecnología M2M para aplicaciones en dispositivos móviles como tablets y smartphones.

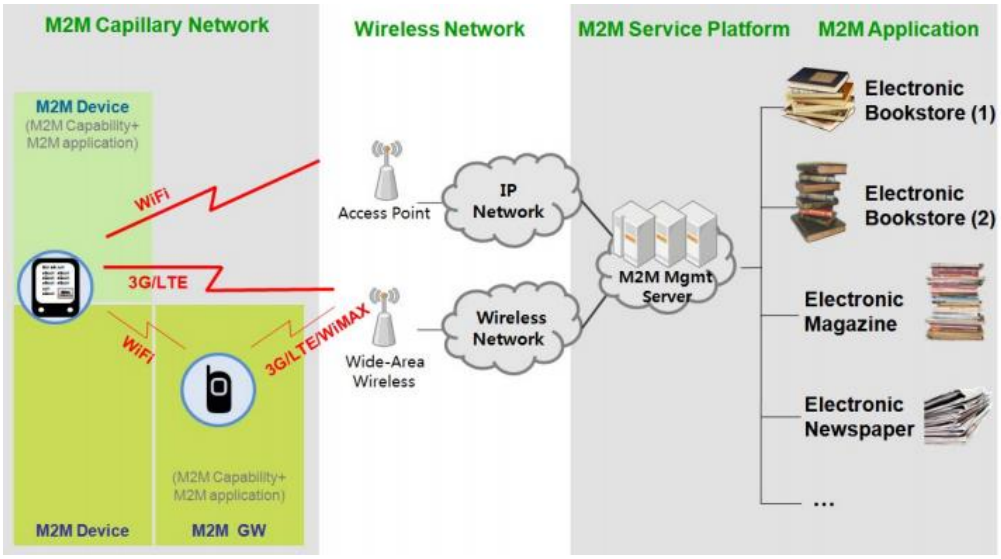


Figura 13. Ejemplo de una aplicación M2M

Fuente: http://ieee802.org/16/m2m/docs/80216p-11_0029r1.pdf

1.2.3.9. M2M en la industria.

La importancia de las comunicaciones entre máquinas que pueden llegar a tener en el mundo real es muy extensa. Existen una gran cantidad de áreas donde éstas podrían aportar grandes beneficios, proporcionando nuevos servicios y mejorando la calidad de los servicios ya existentes, además de ahorrar recursos conllevando a un mundo más sustentable entre sí.

En la actualidad, ya existen sectores donde se utiliza la tecnología M2M, que facilita la gestión de multitud de dispositivos, agiliza operaciones y ahorra costes de operación y despliegue. Sin embargo, también hay sectores donde aún no es posible su implementación, principalmente debido a la rigidez de ciertos sistemas.

En Ecuador la operadora movistar oferta el servicio M2M para empresas y demás corporaciones o entidades públicas; en la ciudad de Quito ha trabajado con la

comunidad para la implementación de alarmas comunitarias. Ofrece todas las posibilidades de comunicación disponibles, en territorio nacional: UMTS, GPRS y SMS.

Entre los beneficios que publicita la operadora Movistar están los siguientes:

Incremento en la productividad, ya que automatiza:

• Los procesos de recogida de datos.
• La monitorización.
• La vigilancia.
• El control remoto de sistemas.

Comunicación en tiempo real.

• Mejora los procesos operativos de la empresa.
• Reducción de costes
• Nuevas oportunidades de negocio

1.2.3.10. Nuevos requerimientos potenciales.

- Un dispositivo M2M debe ser capaz de registrar su información de capacidad (por ejemplo, tecnología de acceso, número de serie, dirección de acceso, la lista de usuarios autorizados, etc.) para el Sistema de M2M.
- Dispositivos M2M y gateways M2M deberían ser capaces de manejar la programación de múltiples accesos que múltiples partes remotas tratan de acceder a un dispositivo M2M o una puerta de enlace M2M simultáneamente permitiendo soluciones horizontales.
- Reducción de la inversión inicial.
- SCL en el servidor de red, gateways y dispositivos.
- Define un "Service Capability Layer" (SCL) en la parte superior de las capas de conectividad.

En la figura 14 se muestra un escenario en el que se puede aplicar la tecnología M2M; para sensar y controlar variables físicas del medio ambiente además se muestra la estructura de comunicación de dicha tecnología.

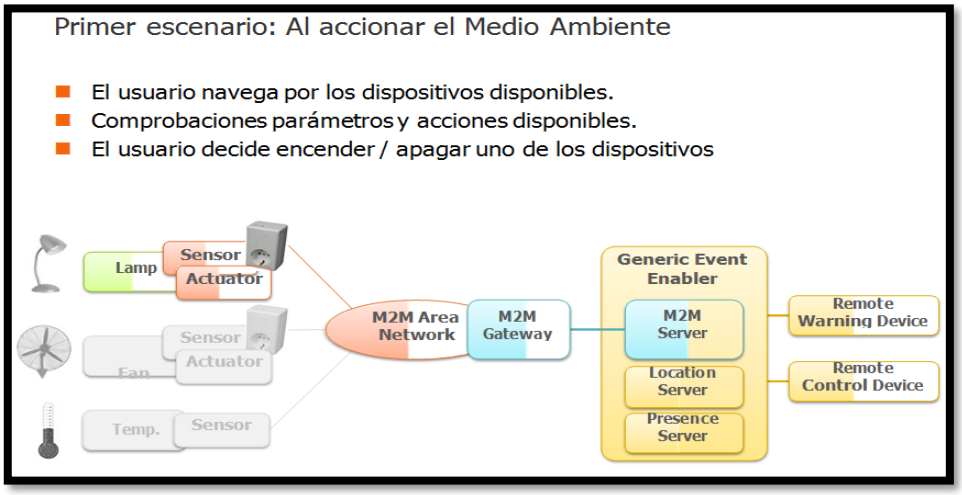


Figura 14. Ejemplo de un primer escenario: Al accionar el Medio Ambiente

Fuente: http://ieee802.org/16/m2m/docs/80216p-11_0029r1.pdf

1.2.4. Televisión digital.

La Televisión Digital Terrestre (TDT) es la nueva señal de aire que implica un cambio tecnológico de la televisión a nivel mundial. En Ecuador nace como un proyecto cultural y social que está en proceso de implementación. La TDT busca mejorar la Televisión para todos, pasando desde el simple entretenimiento, hacia la participación ciudadana, educación entretenida, etc.

También permite disfrutar de todo el conjunto de ventajas que ofrece la nueva televisión digital, como pueden ser: Acceso condicional, televisión interactiva (MHP) o la televisión en alta definición.

El video, el audio y los datos se transforman en señales digitales, de esta manera los posibles errores son corregibles, lo cual se percibe al no tener interferencias ni distorsiones en el televisor.

1.2.4.1. Transmisión y recepción de sistema ISDB-TB.

Para medir la calidad de transmisión de una señal analógica, prácticamente se medía la calidad de la Señal de prueba enviada en el intervalo Vertical y el nivel de distorsión. Pero en redes de sistema digital, estas posibilidades no existen.

Es importante saber que la señal deseada este en buen estado en diferentes puntos de la red desde la codificación de vídeo, audio y datos, hasta la recepción y presentación de la imagen y sonido en el televisor.

Las normas brasileñas de televisión digital consiguen aumentar la oferta de programas y permiten ofrecer servicios interactivos a través de los cuales se puede interactuar con la programación televisiva, acceder a información acerca de los contenidos transmitidos con múltiples señales de audio, teletexto, guía electrónica de programas EPG (Electronic Program Guide), canales de radio, etc.

Para este proyecto se utilizó Village Flow que es un software, herramienta versátil para la transmisión de señales TDT, desarrolla un flujo de trabajo automático desde el proceso de adquisición de contenido en tiempo real (cámaras de video, teléfonos, PC, etc) o desde el servidor a partir de archivos de video multiplexados conjuntamente con datos de interactividad, EPG(Guía de Programación) hasta la generación del flujo de datos Transport Stream en diferentes formatos de video como TV Móvil, Alta Definición, cumpliendo con el estándar ISDB-T, incluidos los módulos RF para modulación y demodulación de la señal de TV Digital.

1.2.4.2. Medición y monitoreo de ISDB-TB.

Las redes de televisión digital usan complicados componentes y procesos para crear la señal y mandarlos a los transmisores. La televisión digital usa las más modernas tecnologías para recibir la señal y para presentar la imagen. Pero aun los mejores equipos son sujetos a fallas debido a edad del equipo o de otras influencias. Esto generalmente resulta en una degradación de la señal o en casos extremos, una falla del sistema de transmisión y recepción.

Para garantizar una alta calidad y tener confianza en los servicios de televisión digital, una variedad de parámetros tienen que ser medidos y analizados durante el monitoreo y la medición de las señales de televisión digital.

- Frecuencia
- Potencia de RF
- Retardo de grupo. Cada señal debe consumir 6 Mhz de ancho de banda sin armónicas ni aberraciones que puedan afectar canales vecinos.

1.2.5. Internet of things “IoT” (internet de la cosas).

La evolución de Internet ha dado paso a la interconexión de muchos dispositivos, tales como sensores de control de tráfico, sensores de humedad y temperatura, cámaras de seguridad, etc. hasta el punto de que actualmente existen una mayor cantidad de objetos conectados a internet que usuarios.

Para llegar a la culminación de la idea de IoT, será necesaria una masiva distribución de datos y la convergencia de servicios, para ofrecer servicios avanzados a empresas y ciudadanos.

El concepto IoT es sencillo de explicar; se trata de la interconexión de todo tipo de elementos, desde los más simples y cotidianos hasta los más complejos, en una red global para ofrecer más servicios y de mayor calidad.

Como se muestra en la Figura 15, se puede observar la evolución de la colaboración de dispositivos y aplicaciones que utilizan tecnologías facilitadas por la arquitectura de M2M. Se observa una progresión en la que la gente colabora más en las redes sociales en la Web o utiliza Internet, con tecnologías de la Web 2.0 para desarrollar sus actividades.

Figura 15. Evolución Internet de las Cosas

Fuente: http://www2.alcatel-lucent.com/techzine/es/

La relación entre Internet of Things y M2M es muy estrecha ya que, en el fondo, IoT no es más que la intercomunicación entre máquinas de manera masiva.

En la figura 16 se observa el funcionamiento de la tarjeta Sim y las aplicaciones en las que puede ser utilizada.

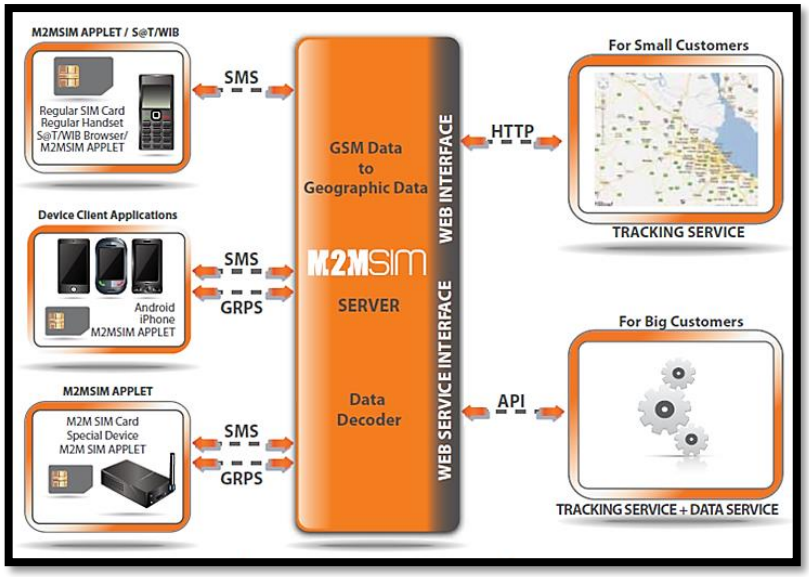


Figura 16. Funcionamiento de la SIM.

Fuente: http://movasim.com/m2msim/

La Solución M2M-SIM permite al MNO gestionar diferentes dispositivos, desde teléfonos de baja gama hasta equipos GPS de última generación, mediante diferentes canales *(SMS y Datos)*, usando múltiples mecanismos complementarios entre sí para obtener la mayor precisión posible en la información entregada.

1.2.6. Equipos

1.2.6.1. STM32F4 Discovery

Basada en el microcontrolador STM32F407VGT6, incluye la herramienta de depuración ST-LINK/V2, dos MEMS ST, acelerómetro digital, micrófono digital, un DAC de audio con driver de altavoz de audio clase D, LEDs, botones pulsadores y un conector USB OTG micro-AB.

La placa STM32F4DISCOVERY permite descubrir las características del STM32F4 y desarrollar aplicaciones.

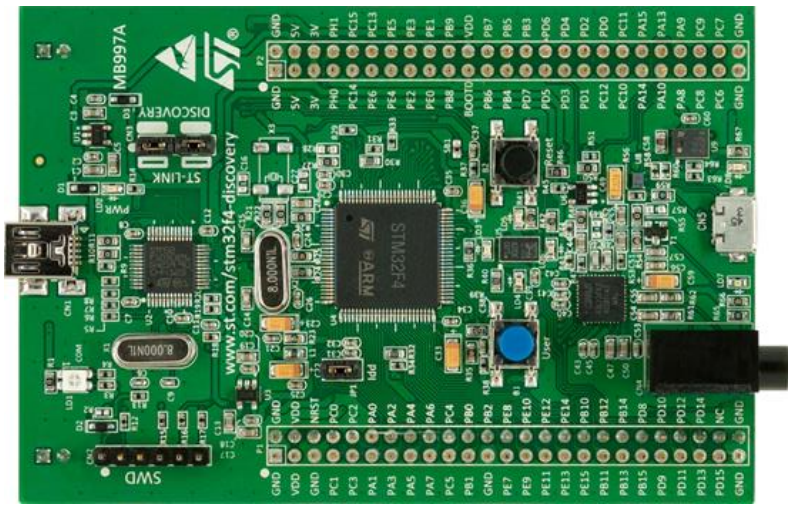


Figura 17. Placa STM32F4
Fuente: http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/LN1848/PF252419

1.2.6.1.1. Características

La placa STM32F4DISCOVERY ofrece las siguientes características:

- Microcontrolador STM32F407VGT6, núcleo 32-bit ARM Cortex-M4F, 1 MB Flash, 192 KB RAM en empaque LQFP100.
- ST-LINK/V2 incorporado con selector usar el kit como un ST-LINK/V2 independiente (con conector SWD para programación y depuración).

- Fuente de alimentación: a través del bus USB o desde una fuente de alimentación externa de 5V. Fuente de poder para aplicaciones externas de: 3V y 5V.
- Sensor de movimiento ST MEMS LIS302DL, acelerómetro con salida digital de 3 ejes.
- Sensor de audio ST MEMS MP45DT02, micrófono digital omnidireccional.
- Audio DAC CS43L22 con controlador integrado de altavoz clase D.
- Ocho LEDs.
- Dos pulsadores (user and reset).
- USB OTG con conector micro-AB.

1.2.6.2. KAISSEN PRO 8000

Como se aprecia en la figura 18 se trata de una Tablet con pantalla de 8 " con 4GB de almacenamiento con sistema operativo Android 4.2; posee un procesador de doble núcleo; además tiene la característica de GSM desbloqueado 3G y conectividad mediante Wi-Fi.

Figura 18. Tablet KAISSEN PRO 8000

Fuente: http://www.kaissen.com/index_eng.html#!home

1.2.6.2.1. Características

- 8 "pantalla TFT táctil capacitiva (Resolución 1024 X 768)
- ARM Cortex-A9 1.5GHz de doble núcleo del procesador
- 4GB flash de almacenamiento
- 1 GB de RAM
- Android 4.2.1 Sistema Operativo
- Lector de tarjetas MicroSD (Soporta tarjeta hasta 32 GB)
- Wi-Fi Redes (802.11 b / g / n)
- Bluetooth Activado
- Soporte 3G GSM desbloqueado
- Altavoz incorporado y micrófono

1.2.6.3. Sensor de temperatura LM 35

El sensor que mide la temperatura del producto es el dispositivo LM35, es un sensor de precisión con una escala de medición de 1ºC. Su rango de medición abarca desde -55°C hasta 150°C. La salida es lineal y cada grado centígrado equivale a 10mV. En la figura 19 se muestra el sensor LM35

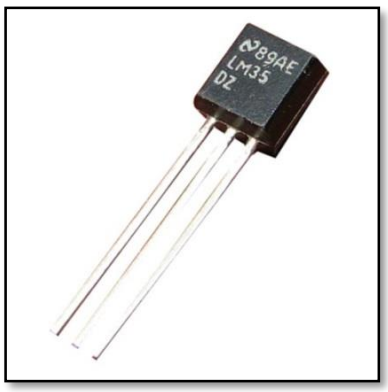


Figura 19. Sensor LM35

Fuente: http://www.trastejant.es/tutoriales/electronica

Este dispositivo no requiere de circuitos adicionales para calibrarlo externamente. La baja impedancia de salida, su salida lineal y su precisa calibración hace posible que el integrado sea instalado fácilmente en un circuito de control. Debido a su baja corriente de alimentación se produce un efecto de auto calentamiento muy reducido.

1.2.6.3.1. Características:

- Está calibrado directamente en grados Celsius.
- La tensión de salida es proporcional a la temperatura.
- Tiene una precisión garantizada de 0.5°C a 25°C.
- Opera entre 4 y 30 voltios de alimentación.
- Baja impedancia de salida.
- Baja corriente de alimentación (60uA).

1.2.6.3.2. Características Eléctricas

En la figura 20, se representa el modo de conexión para la comunicación del sensor:

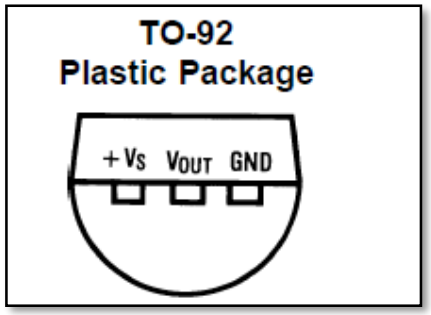


Figura 20. Diagrama eléctrico del sensor LM35

Fuente: Autor

Como indica la tabla 1, la conexión y comunicación de datos es la siguiente:

Tabla 1 . Simbología del Sensor LM35

Símbolo	Señal
+Vs	Señal de alimentación de 24 Voltios
GND	Señal de Común
Vout	Salida de temperatura del sensor.

Fuente: Autor

1.2.6.4. Sensor Biométrico

La técnica consiste en colocar el dedo sobre una superficie de cristal o un prisma que está iluminado por un diodo LED. Cuando las crestas de las huellas del dedo tocan la superficie, la luz es absorbida, mientras que entre dichas crestas se produce una reflexión total. La luz resultante y las zonas de oscuridad son registradas en un sensor de imagen.

Figura 21. ZFM 20 lector de huellas

Fuente: http://www.electronicaestudio.com/sensores.htm

Funciona con un micro DSP que hace la representación y cálculo de la imagen. Se puede conectar a un serial TTL y mandar la información para detectar, comprobar las huellas dactilares. Puede guardar hasta 162 huellas diferentes en la memoria flash y cuenta con un led que indica cuando está funcionando.

Hay básicamente dos requisitos para usar el sensor óptico de huellas dactilares.
El primero es que debe significa asignar ID # a cada huella dactilar, para consultarlos posteriormente. Una vez que haya registrado todos los ID # se pueden "buscar" y "comparar".

1.2.6.4.1. Características

- Alimentación 3.6 - 6.0VDC
- Interfaz: TTL serial.
- Corriente: 120mA max.
- Velocidad en baudios: 9600, 19200, 28800, 38400, 57600 (default 57600).
- Tasa de aceptación de ID falsas: <0.001% (Nivel de seguridad 3).
- Tiempo de adquisición menor a 1 segundo

1.2.6.5. Sensor de humo

Un detector de humo es una alarma que detecta la presencia de humo en el aire y emite una señal acústica avisando del peligro de incendio. Atendiendo al método de detección que usan pueden ser de varios tipos: detectores iónicos y detectores ópticos. El sensor de humo utilizado en este proyecto se muestra en la figura 22.

Figura 22. Detector de humo Fotoeléctrico.
Fuente: http://www.agm.com.co/images/products/259/SD-4WP.jpg

Los detectores de humo fotoeléctrico de la Serie SD son fabricados en estricto cumplimiento de los códigos UL para protección de vidas y responden efectivamente a una variedad de espectros de fuego.

La Serie SD de detectores de humo están diseñados para ser usados en comercios, industrias, instituciones y residencias. Su diseño de bajo perfil incrementa su fácil instalación y decoración con el lugar.

1.2.6.6. Sensor de movimiento microondas

Estos sensores registran y comparan la frecuencia de las señales transmitidas y reflejadas. Las personas y objetos causan las diferencias entre las frecuencias emitidas y recibidas mientras se mueven a través de la zona de detección, activando de este modo el relé del sensor.

En la figura 23 se puede observar el sensor magnético que se utilizó en el proyecto y que se encuentra instalado en el laboratorio de televisión digital de la UNACH el cual detecta la presencia o irrupción al laboratorio.

Figura 23. Sensor de movimiento MC420s

Fuente: http://www.sz-sensor.com/en/products_xq.asp?id=146

Sensor de microondas: También llamado Mircrowave radar. Sólo respuesta a objeto en movimiento. La velocidad de respuesta es muy rápido. Adecuado para la

condición que persona que se mueve en velocidad normal. Cuando la persona deja de moverse, el sensor se detendrá respuesta también.

1.2.6.6.1. Características:

- Tecnología: Mircowave y microprocesador
- Principio de detección: Doppler procesador
- Max. altura de montaje: 4 m
- Ángulo de radiación: 120 grados en horizontal, 80 grados en vertical.
- Área de detección: 4 m (ancho) x 3 m (largo)
- Modo de detección: movimiento
- Energía: dc/ac 12 ~ 24 V
- Frecuencia: 50 ~ 60 Hz

1.2.6.7. Actuadores

1.2.6.7.1. Chapa eléctrica

Una chapa eléctrica estándar es muy utilizada porque puede abrirse a distancia con solo pulsar un interruptor, funciona con 12 voltios por lo que se hace necesario un transformador. La chapa utilizada en el proyecto se muestra en la figura 24; y está instalada en el laboratorio de televisión digital de la UNACH.

Figura 24. Chapa eléctrica marca DURAL

Fuente: Autor

Opera del siguiente modo, cuando se pulsa el interruptor, la corriente pasa por el primario del transformador a 110 voltios, el secundario del mismo a 12 voltios conectado en serie con el solenoide de la chapa hace que esta se magnetice desactivando el enclavamiento que asegura el pestillo más ancho de la chapa, el mismo que recorre a la posición de abierto gracias a la acción de un resorte que quedo tensado el momento que se cerró la chapa, toda chapa eléctrica tiene dos pestillos, uno más angosto conectado al resorte del que se está hablando y el más grande que asegura la puerta

1.2.6.7.2. Ventilador

Se utiliza para la ventilación o para aumentar la velocidad del aire en un espacio habitado, básicamente para refrescar. Por esta razón, es un elemento muy utilizado en climas cálidos. Como elemento de refrigeración es además económico puesto que tienen potencias de entre 20 y 40 W, lo que significa que su consumo de energía y gasto económico son muy reducidos.

Figura 25. Ventilador

Fuente: Autor

1.2.7. Software

1.2.7.1. CooCox 1.7.8.

CooCox es un entorno de desarrollo ARM gratuito y abierto para ARM Cortex-M microcontroladores. Ofrece gratuitamente una poderosa herramienta de desarrollo de software para microcontroladores basados en ARM Cortex-M. Es también una plataforma de red social para desarrolladores de sistemas integrados y proporciona una plataforma de conocimiento de desarrollo integrado para los usuarios.

Figura 26. Programa CooCox

Fuente: Autor

Herramienta de desarrollo Coocox constan de varios módulos como CoIDE, CoFlash, CoLinkEx, Coos, CoCenter y coadjutor.

CoIDE incluye todas las herramientas necesarias para desarrollar soluciones de software de alta calidad en un menor tiempo y de manera rentable.

CoIDE consiste en una plataforma de red basada en componentes y un entorno de desarrollo basado en Eclipse. La componente es un conjunto de códigos relativamente independientes, programas controladores de sus periféricos, algunos algoritmos y sistema operativo. A diferencia de otros tradicionales IDE de desarrollo integrado, enfoque de programación CooCox se basa en sus componentes, rutinas y ejemplos.

1.2.7.2. Android Studio 1.0.1

Android Studio es un entorno de desarrollo integrado (IDE) para la plataforma Android como indica la figura 27. Está disponible para desarrolladores para probarlo gratuitamente. Basado en IntelliJ IDEA de JetBrains, está diseñado específicamente para desarrollo en plataforma Android.

Figura 27. Programa Android Studio 1.0.1

Fuente: autor

1.2.7.2.1. Características

- Generación de imagen y video en tiempo real
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en la herramienta Gradle.
- Refactorización especifica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear

1.2.7.3. Eclipse 4.2.

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse. Sin embargo, también se puede usar para otros tipos de aplicaciones cliente.

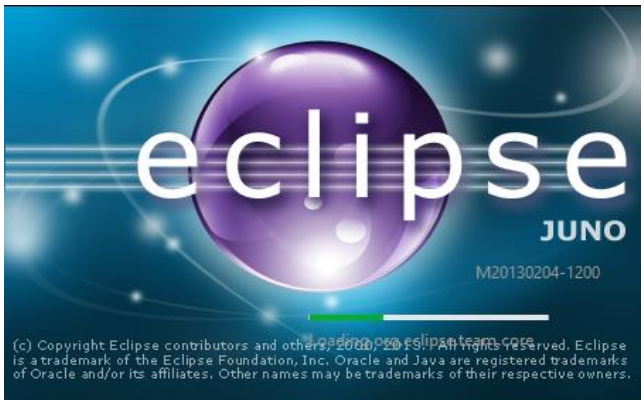


Figura 28. Programa eclipse JUNO 4.2.

Fuente: autor

Eclipse permite crear aplicaciones para celulares con sistema Android mediante el uso de herramientas y códigos en lenguaje java, además de proporcionar modelos gráficos preexistentes, permitiendo utilizar los diferentes API y compiladores ya existentes para las diversas versiones de Android; permite la creación de pantallas y personalización de pantallas.

1.2.7.4. Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tablets; y también para relojes inteligentes, televisores y automóviles. Típicamente representado por su logotipo de la figura 29.

Figura 29. Logotipo Android

Fuente: www.tecnologia.net

A continuación en la Tabla 2 se enuncian algunas de las características más importantes del sistema operativo Android.

Tabla 2 . Características Sistema Operativo Android.

Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D y diseño de teléfonos tradicionales.
Almacenamiento	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
Conectividad	Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, CDMA, UMTS, Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX, GPRS,UMTS.
Mensajería	SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud
Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome.
Soporte de Java	Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma.
Soporte multimedia	Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), AAC, HE-AAC , MP3, MIDI,Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.

La figura 30 muestra de manera gráfica cual es la versión de Android más utilizada actualmente en la mayoría de dispositivos y en la tabla 2 tenemos expresados los porcentajes de forma numérica estos porcentajes fueron extraídos de la página oficial de desarrolladores de Android (developer.android.com).

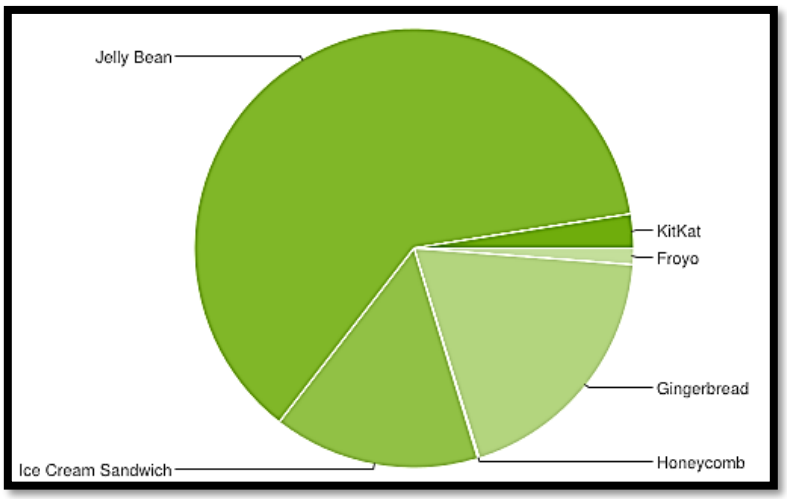


Figura 30. Gráfico pastel distribución de versiones Android.

Fuente: http://developer.android.com/

El 62% de dispositivos Android llevan alguna de las versiones de Jelly Bean, por lo cual se ha optado trabajar bajo esta versión de Android para obtener la mejor opción de compatibilidad con la mayoría de dispositivos en el mercado.

Tabla 3 . Porcentajes de uso de las versiones de Android.

Versión	Nombre	API	Distribución
2.2	Froyo	8	1.20%
2.3.3-2.3.7	Gingerbread	10	19.00%
3.2	Honeycomb	13	0.10%
4.0.3-4.0.4	Ice Cream Sandwich	15	15.20%
4.1	jelly Bean	16	35.30%
4.2		17	17.10%
4.3		18	9.60%
4.4	KitKat	19	2.50%

Fuente: Autor

1.2.8. Servicio web

Un Web Service o servicio web en español, es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

1.2.8.1. Ventajas de los servicios web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro.

Un servicio web es un método de comunicación entre dos dispositivos a través de una red. Es una función de software suministrado en una dirección de red a través de Internet.

Los diferentes sistemas de software a menudo necesitan intercambiar datos entre sí, y un servicio Web es un método de comunicación que permite a dos sistemas de software el intercambio de datos en internet en nuestro caso los datos serán los de nuestro sistema de telemetría y control del laboratorio; además de datos provenientes del servidor de televisión digital de la UNACH.

1.2.9. PHP

PHP es un script del lado del servidor lenguaje creado en 1995 y diseñado para el desarrollo web, pero también se utiliza como un lenguaje de programación de propósito general.

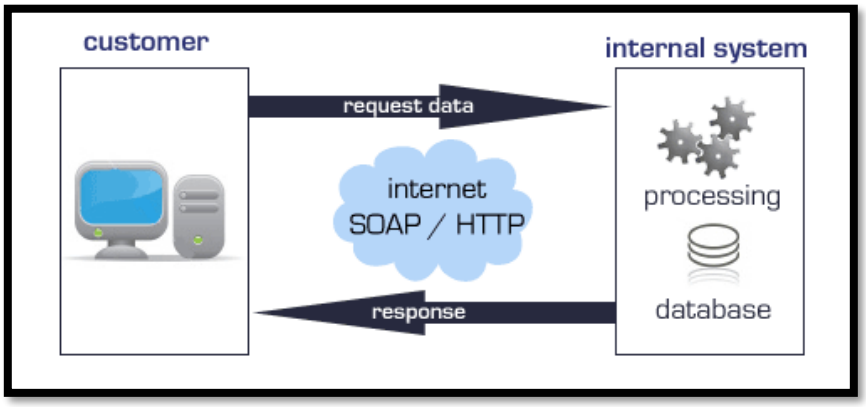


Figura 31. Esquema de intercambio de datos

Fuente: http://www.aztecmedia.co.uk/webservices-xml-api.aspx

Código PHP puede ser simplemente mezclado con HTML código, o se puede utilizar en combinación con varios motores de plantillas y marcos web; ha sido ampliamente portado y puede ser desplegado en la mayoría de los servidores web en casi cada sistema operativo y plataforma, de forma gratuita.

PHP es un lenguaje de programación de propósito general que es especialmente adecuado para el lado del servidor de desarrollo web, en cuyo caso PHP generalmente se ejecuta en un servidor web PHP puede ser desplegado en la mayoría de servidores web.

PHP ahora se centra principalmente en secuencias de comandos del lado del servidor, y es similar a otros lenguajes de script del lado del servidor que proporcionan contenido dinámico desde un servidor web a un cliente, serán en estos archivos almacenados en el servidor donde se programara el código para realizar consultas y respuestas tanto de nuestra base de datos de registro de

ingreso y alarmas; como también de los datos y parámetros de configuración de nuestro servidor de televisión digital. Tal como se muestra en la figura 32.

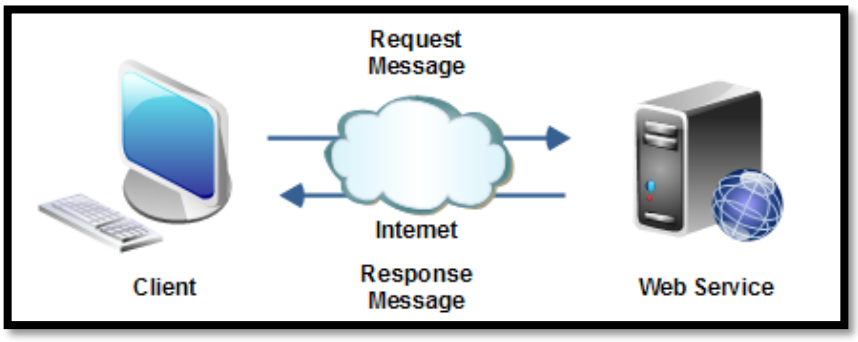


Figura 32. Cliente intercambiando información con el servidor web

Fuente: http://tutorials.jenkov.com/web-services/message-formats.html

El diagrama de la figura 33 es la representación de un modelo basado en el servicio. En este modelo, el servicio web reside en el servidor web. Un equipo cliente puede solicitar y consumir este servicio web y dar por terminado el servicio como y cuando se desee. El programa cliente necesita ser proporcionado sólo con la url del servicio web. Las aplicaciones de cliente pueden estar en ejecución en diferentes plataformas o desarrollados usando diferentes herramientas.

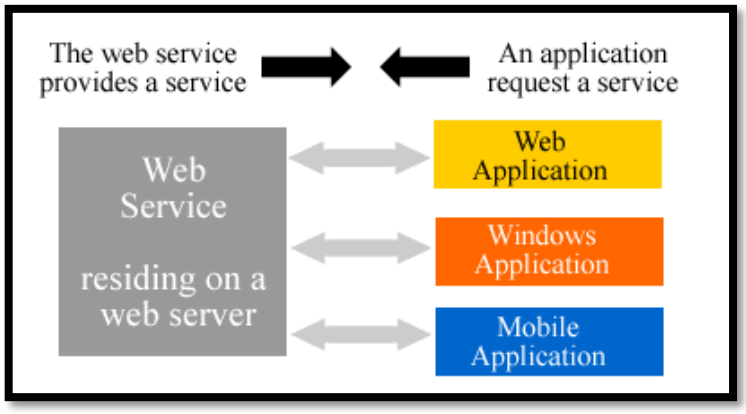


Figura 33. Medios para acceso a un web service

Fuente: http://www.corba.com.mx/desarrollo-de-software/

Los servicios Web transportan sus mensajes a través de HTTP, lo que significa que estos mensajes se transmiten a través del puerto 80, un puerto abierto para servidores de seguridad de servidor web. Un Web service suele ser un archivo PHP que recibe peticiones de nuestra aplicación y se las envía al servidor, posteriormente el servidor responde a nuestro web service, y este, a su vez, responde a nuestra aplicación. Esto se muestra de forma gráfica en la figura 34.

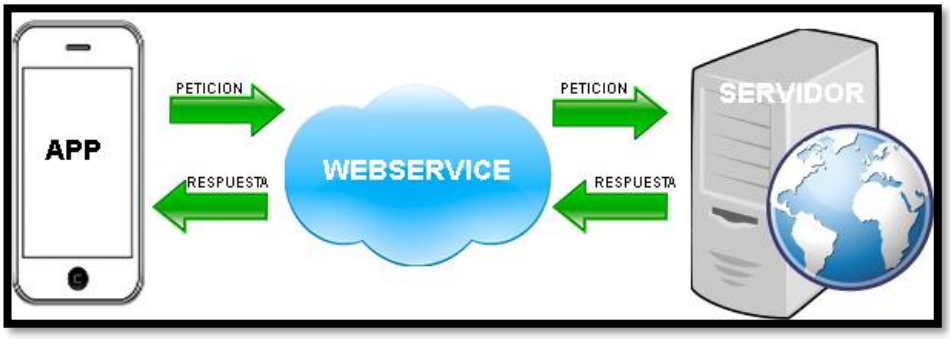


Figura 34. Consulta y respuesta desde un smartphone

Fuente: http://picarcodigo.blogspot.com/2014/05/webservice-conexiones.html

Los archivos PHP pueden contener texto, HTML, CSS, JavaScript, lenguaje XML y el código PHP. Entonces al ingresar desde cualquier dispositivo vía internet a nuestro servidor se accede a dichos archivos para realizar consultas y peticiones y acceder desde cualquier dispositivo como se indica en la figura 35.

Figura 35. Acceso desde cualquier dispositivo

Fuente: http://jqueryasp.net/

Como se indica en la figura 36 el código PHP se ejecuta en el servidor, y el resultado se devuelve al navegador como HTML plano. Los Archivos PHP tienen extensión ".php".

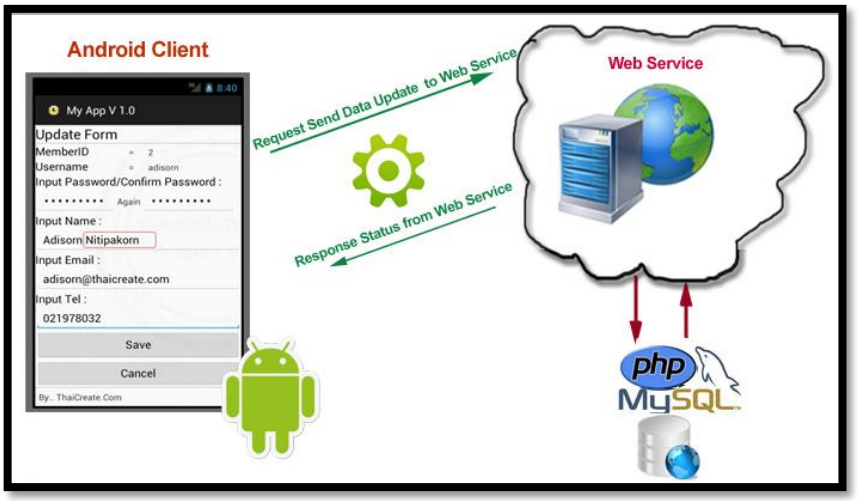


Figura 36. Interacción Cliente Android y Web Service mediante PHP

Fuente: https://norfipc.com

Lenguaje php solicita conexión a web service → web → web service desarrollado responde y envía la respuesta → php recibe la respuesta.

1.2.9.1. Aplicaciones de lenguaje PHP

- generar páginas con contenidos dinámicos
- puede crear, abrir, leer, escribir, borrar y cerrar archivos en el servidor
- recopilar datos de formularios
- puede añadir, borrar, modificar los datos de su base de datos
- puede ser usado para controlar el acceso de usuario

1.2.9.2. Ventajas de lenguaje PHP

- Se ejecuta en varias plataformas (Windows, Linux, Unix, Mac , etc.)
- Es compatible con casi todos los servidores utilizados en la actualidad
- PHP soporta una amplia gama de bases de datos
- PHP es gratuito.

De momento el API de android no provee ningún método que permita conectarse a través de internet directamente a una Base de Datos Remota y ejecutar una consulta dentro de ella. Para poder realizar esto se puede utilizar un web service al cuál se pueda acceder a el pasando diversos parámetros nos devuelve ya sea en formato XML o JSON, nuestro trabajo será obtener esa información devuelta y utilizarla en nuestra aplicación.

Para hacer uso basta con crear algunos archivos .php, colocarlos en su directorio web, y el servidor analizará automáticamente estos archivos. Por lo tanto no hay necesidad de compilar nada ni instalar ninguna herramienta adicional. Dado que PHP es libre, la mayoría de los servidores web ofrecen soporte para PHP.

1.2.10. Servicio web ligero o peso pluma

Al decir ligero se refiere a un servicio web simplista. Uno donde tanto más poderoso y complejo, los protocolos se dispensan. Peso pluma se basa estrictamente en el protocolo más ligero posible, el protocolo GET de peticiones web. Entonces se utiliza este protocolo para acceder al contenido generado por los archivos PhP almacenados en el servidor web el proceso se asemeja el mostrado en la figura 37.

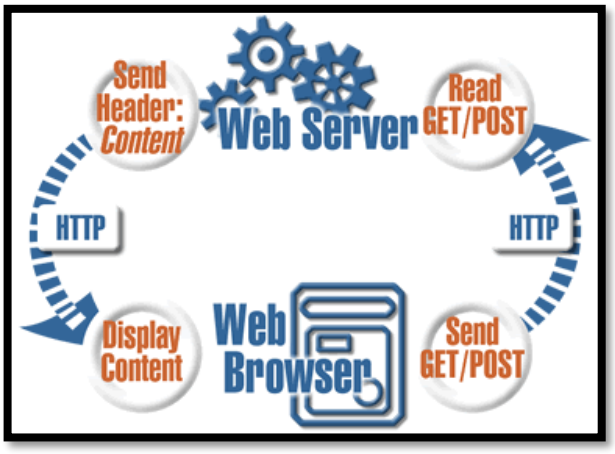


Figura 37. Uso del protocolo GET/POST

Fuente: http://feather.elektrum.org/book/webservices.html

Peso pluma es el adecuado para los problemas que involucran configurar una vez, leer a menudo. Situaciones donde se desea algo de código y la manipulación de los datos sin la sobrecarga de idiomas y protocolos no estándar; es decir, no todos los escritorios en red en el mundo está usando XML todavía, pero todos hablan de HTTP.

1.2.11. Base de datos MYSQL

Mediante PHP, un usuario puede conectarse y manipular bases de datos. MySQL es el sistema de base de datos más popular utilizado con PHP.

1.2.11.1. Características MySQL?

- Es un sistema de base de datos utilizada en la web.
- Es un sistema de base de datos que se ejecuta en el servidor.
- Es ideal tanto para aplicaciones pequeñas y grandes.
- Es muy rápido, fiable y fácil de usar.
- Usa SQL estándar.
- Es libre para descargar y utilizar.

1.2.11.2. Aplicaciones de lenguaje SQL

- Puede ejecutar consultas contra una base de datos
- Recuperar datos de una base de datos
- Insertar registros en una base de datos
- Actualizar registros en una base de datos
- Eliminar registros de una base de datos
- Crear nuevas bases de datos
- Crear nuevas tablas en una base de datos

Los datos en una base de datos MySQL se almacenan en tablas. Una tabla es una colección de datos relacionados, y se compone de columnas y filas. Las bases de datos son útiles para almacenar información de forma categórica.

CAPÍTULO II

2. METODOLOGÍA

2.1. Tipo de Estudio.

2.1.1. Investigación aplicada.- Se realizara una investigación aplicada; ya que se busca la utilización de los conocimientos que se adquieren, depende de los descubrimientos, avances y resultados de la investigación básica, lo que interesa al investigador son las consecuencias prácticas.

2.1.2. Investigación Analítica.- Con esta investigación se pretende verificar los datos obtenidos por el prototipo de sistema, con el fin de comprobar una hipótesis sobre uno o varios hechos siendo capaz de predecir fallas o acontecimientos. Este método consiste en la desmembración de un todo, descomponiéndolo en sus partes o elementos para observar las causas, la naturaleza y los efectos.

2.2 Población Muestra.

2.2.1. Población.

Para el trabajo de investigación se considera como población al conjunto de pruebas que se realizarán en el sistema de telemetría y telemando, teniendo planeado tomar 5 mediciones de cada factor cada 2 horas por día, durante 7 días dando como resultado una población de 35 mediciones realizadas y enviadas por el sistema; se considerará el tiempo para activación de los siguientes factores:

- El sensor de temperatura y activación del equipo de enfriamiento de forma autónoma y remota.

- La activación de eventos, el registro en base de datos y envió a dispositivos móviles.
- El activador de apertura de puerta de forma remota.
- Adquisición de señales del servidor de televisión digital.

2.2.2. Muestra.

Para determinar el tamaño de la muestra se optó por tomar la población de 35 mediciones debido a que el número de mediciones de los factores que intervienen no es muy grande y para un mejor análisis del tiempo de respuesta en los 4 factores se cree conveniente el uso de todas las mediciones.
Por lo cual la muestra será de 35 mediciones por cada uno de los 4 factores estudiados para la comprobación del sistema y su trabajo en tiempo real.

2.2.3. Hipótesis.

El diseño e implementación de un sistema de telemetría y telemando utilizando tecnología machine to machine (M2M) permitiría el intercambio de información en formato de datos para el control y monitoreo del laboratorio de televisión digital de la UNACH en tiempo real.

2.3 Operacionalización de Variables

2.3.1. Variable independiente

El diseño e implementación de un sistema de telemetría y telemando utilizando tecnología machine to machine.

2.3.2. Variable dependiente

El intercambio de información en formato de datos para el control y monitoreo del laboratorio de televisión digital de la UNACH en tiempo real.

Tabla 4 . Operacionalización de la variable independiente.

VARIABLES	DIMENSIONES	INDICADORES	ITEM	Técnicas e Instrumentos
Variable independiente:	**Estudio de las necesidades y requerimientos precisos para el correcto diseño del sistema de telemetría del laboratorio.**	-Estudio comparativo de las tecnologías M2M para determinar los parámetros, protocolos y arquitecturas a utilizar determinando cual es el más adecuado para el diseño del sistema de telemetría que permita censar y controlar en tiempo real al laboratorio de televisión digital.	-Tipos de arquitecturas -Menor costo -Mayor flexibilidad -Mayor escalabilidad	-Análisis comparativo de las diferentes arquitecturas y tecnologías M2M. -Realizar un estudio sobre las caracterizas, ventajas y desventajas de las tecnologías usadas en telemetría.
DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE TELEMETRÍA Y TELEMANDO CON TECNOLOGÍA M2M	**Implementación de los equipos de control y monitoreo interconectados al sistemaM2M para el intercambio de información**	-Implementación de un sistema que permita el monitoreo y control del laboratorio y del servidor de televisión mediante la creación y el uso de una interfaz que administre los datos y que permita la visualización y toma de decisiones de manera autónoma haciendo uso de la tecnología M2M.	-Red de área local constituida por los sensores y actuadores. -Alerta de fallos, caída de velocidad e intrusión al laboratorio de televisión digital.	-Medición de potencia de Tx, Ts RATE, tamaño de paquete, ancho de banda, frecuencia del servidor de televisión. -Verificación de la conectividad física en la red de área local

Fuente: Autor.

Tabla 5 . Operacionalización de la Variable dependiente.

VARIABLES	DIMENSIONES	INDICADORES	ITEM	Técnicas e Instrumentos
Variable dependiente: **INTERCAMBIO DE INFORMACION PARA EL CONTROL Y MONITOREO EN TIEMPO REAL**	**Control de Acceso al laboratorio.**	-Control acceso al laboratorio de forma remota o con huella dactilar mediante sensor biométrico. -Registro de ingreso al laboratorio del personal autorizado usando una base de datos.	• Apertura de puertas • Control del ingreso.	-Activador de apertura de puertas. -Base de datos de la entrada del personal autorizado y envió automático de esta información a los dispositivos móviles.
	Control de temperatura del laboratorio.	-Diseño de un sistema que permita controlar de forma autónoma la temperatura del laboratorio de televisión digital	• Temperatura optima • Activación de equipo de enfriamiento.	-Activador de equipo de enfriamiento de forma autónoma. -Verificación y modificación de la temperatura de forma remota.
	Verificación, monitoreo y control remoto mediante una aplicación para Smartphone	-Creación de una aplicación para Smartphone que permita acceder a la interfaz de monitoreo de forma remota. -Utilizar la red de telefonía celular para el envío y recepción de datos desde el área de red local hacia el dispositivo móvil.	• Conectividad • Innovación • Monitoreo del ancho de banda y parámetros del servidor. • Envió y recepción de información en tiempo real	• Smartphone • Placa electrónica para el manejo de señales. • Adquisición de información del servidor en formato XML.

Fuente: Autor

2.4. Procedimientos

Se aplica el método hipotético-deductivo debido a que a partir de un problema detectado se formulará una hipótesis que se espera confirmar con la experiencia.

El método descriptivo que consiste en realizar una exposición narrativa, numérica y/o gráfica, lo más detallada y exhaustiva posible de la realidad que se investiga. Realizar un análisis de la zona a investigar y del mejor diseño a implementar para llevar detalles de las mismas.

Se aplica un método analítico ya que se debe tener un conocimiento claro de cada uno de los elementos y dispositivos que forman parte de las etapas del sistema a estudiar. Se presenta todos los factores del estudio.

Se hará uso del método investigativo experimental; que es un tipo de investigación que utiliza experimentos y los principios encontrados en el método científico. Los experimentos pueden ser llevados a cabo en el laboratorio o en la vida real (ALfredo, 2013). y realizar las correspondientes pruebas para determinar el mejor diseño y correcto funcionamiento del mismo.

2.5. Procesamiento y Análisis

2.5.1. Sistemas de telemetría.

El servicio de telemetría o sistema de telemetría permite hacer transmisión de datos de sensores y contactos en forma remota a través de la tecnología celular en tiempo real.

2.5.1.1. Tiempo real.

El trabajar en tiempo real (Real Time) se refiere a la capacidad del ordenador para gestionar los programas de procesamiento de datos para que siempre esté listo

para proporcionar los resultados dentro de un tiempo especificado. En este contexto "estrictamente en tiempo real" significa que un sistema reacciona a los eventos externos dentro de un tiempo especificado en un 100% de los casos.

2.5.2. Desarrollo del prototipo.

El diseño del prototipo se realizó siguiendo un orden establecido por las siguientes etapas:

1. Selección e instalación de sensores y actuadores.
2. Programación de la placa STM32f4
3. Conexión de la placa con la Tablet
4. Programación de la Tablet
5. Monitoreo del servidor de televisión digital
6. Conexión de la Tablet con el servidor
7. Almacenamiento y lectura de datos
8. Creación de la aplicación de monitoreo para Smartphone.

El proyecto busca cumplir con los siguientes objetivos:

- El diseñar e implementar un sistema de telemetría utilizando tecnología M2M para monitoreo y control del laboratorio de televisión digital de la UNACH usando como canal de comunicación la red de telefonía celular.

- Diseñar un sistema que permita sensar y controlar las variables ambientales del laboratorio como temperatura, presencia de humo.

- Permitir controlar el acceso al laboratorio de forma remota, desde un dispositivo móvil o mediante reconocimiento de huella con lector biométrico.

- Monitorear y enviar en tiempo real la información del servidor de televisión digital para monitorear parámetros de transmisión del servidor de televisión como potencia de transmisión, latencia, Ts-rate, frecuencia.

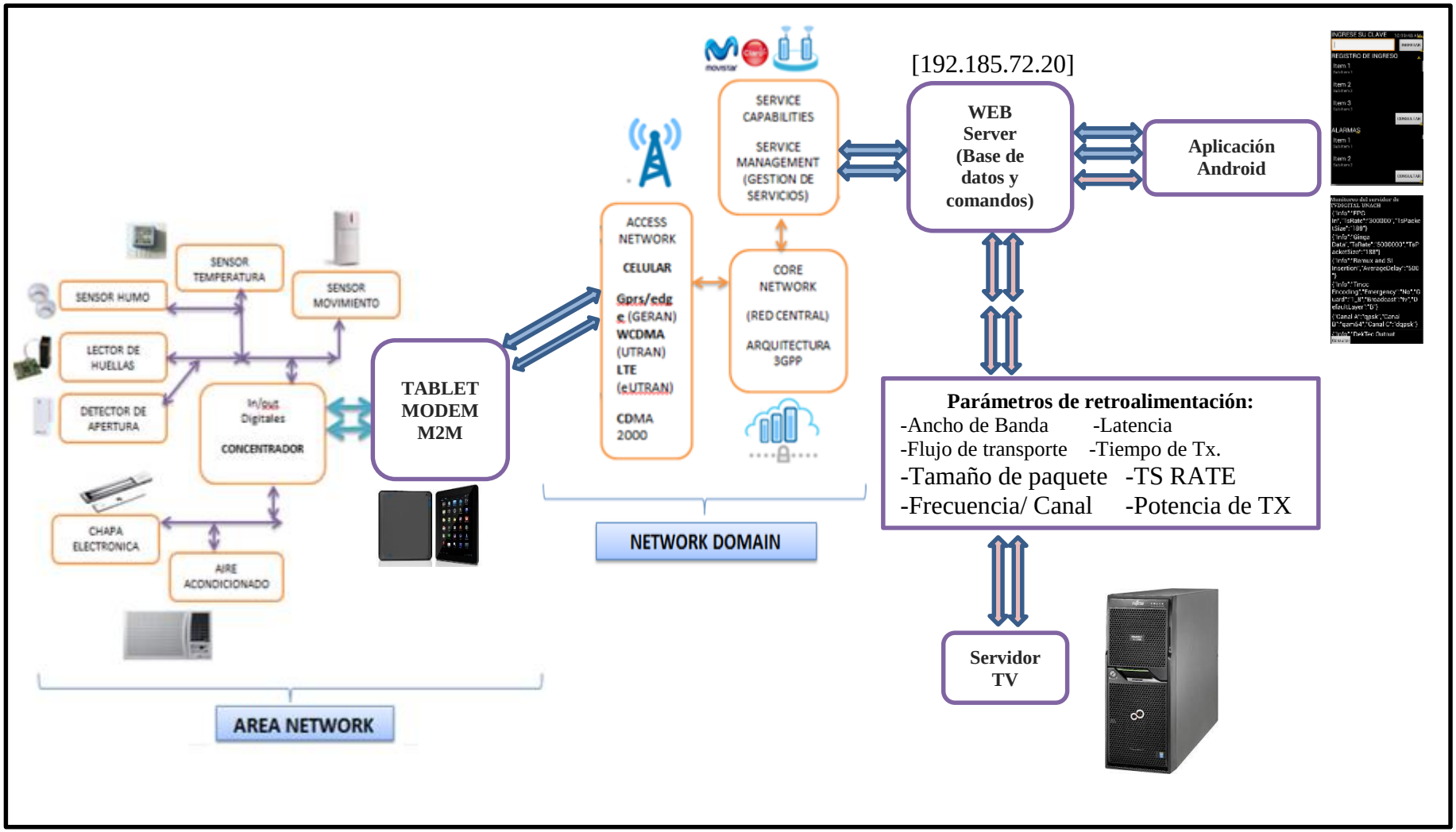


Figura 38. Diagrama de bloques del sistema

Fuente: Autor

2.5.2.1. Selección e instalación de sensores y actuadores.

En la figura 39 se muestra un diagrama de los sensores y actuadores que se utilizaron en este proyecto, los cuales son conectados a un concentrador, la función de dicho concentrador es la de interpretar la información generada por los sensores y de activar los actuadores de ser necesario.

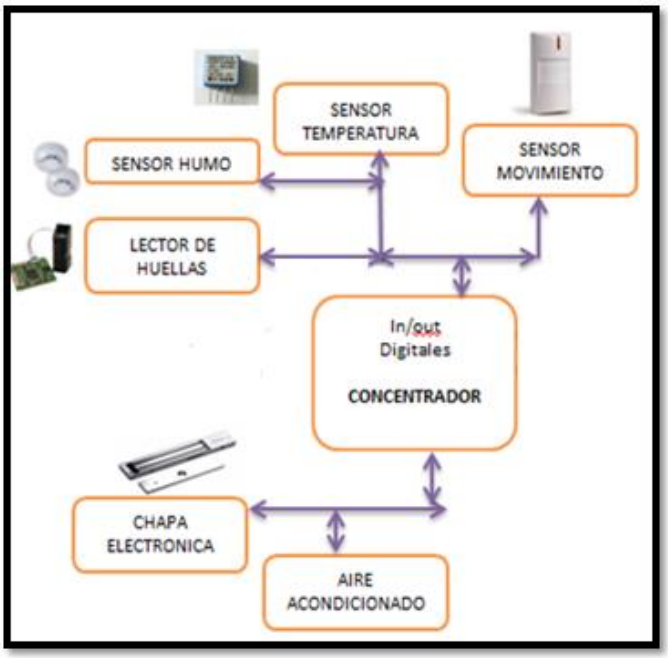


Figura 39. Sensores y actuadores conectados al concentrador

Fuente: Autor

Los sensores utilizados son los siguientes:

- Sensor de temperatura LM35
- Sensor biométrico ZFM 20
- Sensor de humo
- Sensor de movimiento

Los actuadores utilizados son los siguientes:

- Una chapa eléctrica de 12 VCD
- Un ventilador de 12 VCD

Para este proyecto se utilizó la placa de desarrollo STM32F4 como concentrador, el cual recibe e interpreta la información recibida por los sensores tales como el de temperatura para sensar el medio ambiente, el sensor de movimiento para detectar la presencia o irrupción al laboratorio de personal no autorizado, el sensor de humo en caso de incendio, y el lector de huellas que permitirá la apertura de la puerta de ingreso del laboratorio de televisión digital de la UNACH. Este dispositivo también permite controlar los diferentes actuadores tales como la chapa eléctrica y el ventilador.

La distribución de los pines y las partes de la tarjeta STM32f4 se indican en la figura 40.

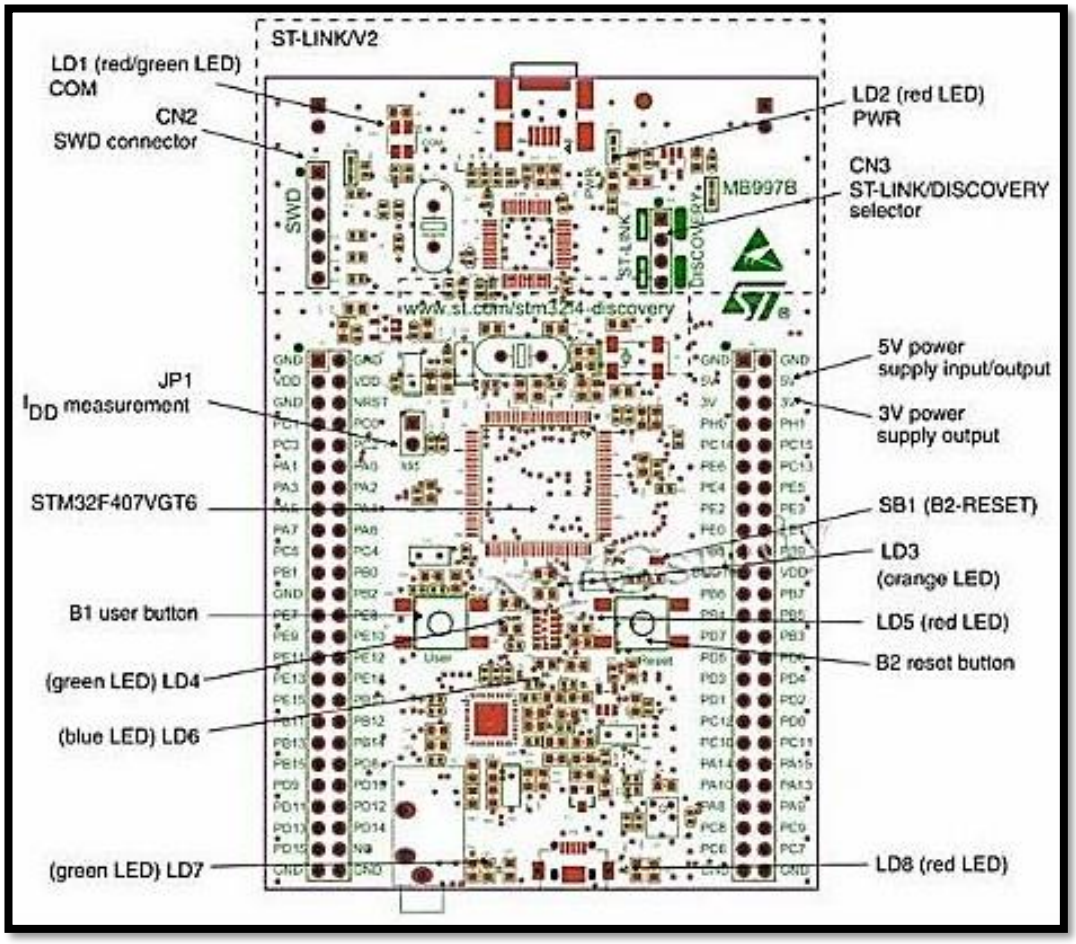


Figura 40. Tarjeta STM32f4

Fuente: http://3.bp.blogspot.com/-OnJ4SFsg5as/s1600/descarga+(2).jpg

2.5.2.2. Programación y funcionamiento de la placa stm32f4

Como concentrador para los sensores y actuadores se utiliza la placa electrónica STM32F407, la cual cumple la función de interactuar con el HMI de la Tablet.

El programa creado mediante el entorno de programación CooCox-CoIDE es el encargado de interactuar o intercambiar información desde los sensores hacia la plataforma HMI creada bajo sistema Android y viceversa; el programa toma los datos en bruto y los transforma a datos legibles para el usuario mediante fórmulas matemáticas y librerías existentes; de manera que se puedan entender los datos obtenidos y decidir que se quiere hacer con estos.

Además el programa será el encargado de interactuar con los diferentes actuadores del sistema, como la chapa eléctrica y un ventilador que controlara la temperatura en el laboratorio; además cumple con la función de medir constantemente las variables existentes en el laboratorio mediante los sensores.

La programación se lo realiza en el entorno CoIDE la cual maneja el lenguaje de programación java. Este entorno ofrece al usuario la gran ventaja de ser un software libre por lo cual existen una gran cantidad de ejemplos y librerías disponibles para los programadores. Por lo tanto se ha hecho uso de estas librerías que facilitan algunas funciones propias de la placa que a continuación se describen las principales utilizadas y la función que cumplen cada una de estas:

- `#include "stm32_ub_uart.h"`

Esta librería activa la comunicación serial; permite comunicar la placa STM con el Puerto USB de la tablet mediante puertos de comunicación seriales.

- `#include "stm32_ub_led.h"`

Esta librería permite controlar los led internos de la placa además de los puertos que controlan dichos leds los cuales serán utilizados como salidas para la activación de los actuadores.

- `#include "stm32_av_zfm20x.h"`

Esta librería es usada para controlar y obtener la información proveniente del lector de huellas ZFM20; contiene diferentes funciones tales como enrolar, buscar, borrar, entre otras; las cuales son usadas para identificar y registras nuevas huellas.

- `#include "stm32_ub_adc1_single.h"`

Es una librería que accede al ADC interno de la placa que permite la lectura de sensores análogos en este caso para leer la temperatura desde el sensor LM35

- `#include "stm32_ub_tim2.h"`

Habilita puertos y temporizadores incluyendo un reloj interno de la placa

- `#include "stm32_ub_dig_in.h"`

Habilita puertos digitales de entrada de la placa.

Al inicio del programa se declaran las librerías y las variables globales a utilizar, tal como se muestra en el siguiente código:

```
#include "main.h"
#include "stm32_ub_uart.h"
#include "stm32_ub_led.h"
#include "stm32_av_zfm20x.h"
#include "stm32_ub_adc1_single.h"
#include "stm32_ub_tim2.h"
#include "stm32_ub_dig_in.h"
int ident = 0;
uint16_t temp;
uint16_t SPtemp=25;
int alarma1=0;
int alarma2=0;
int aux_a1=0;
int aux_a2=0;
```

Al incluir estas librerías permite utilizar dentro del programa directamente los diferentes pines y puertos que posee la tarjeta con funciones específicas, como

ADC, comunicaciones seriales, activación de puertos como estradas/salida, lectura de datos desde el lector de huella, reloj interno, etc.

Esta tarjeta posee tres ADCs de 12 bits, se utiliza el primero de esto para medir la temperatura desde el sensor LM35 ya que es un valor análogo de voltaje.

También cuentan con interfaces de comunicación estándar y avanzadas; cuatro puertos de comunicación USARTs más dos de comunicación UART, estos son de gran importancia en el prototipo ya que permite comunicar estos puertos a los sensores como el lector de huella dactilar, además de permitir conectar esta tarjeta con la Tablet mediante cables USB para la transferencia de información desde un dispositivo a otro y viceversa, y permite conectar cualquier dispositivo a su puerto mini USB para depuración y programación de manera simultánea sin tener que desconectar ningún puerto de comunicación.

En el lazo principal del programa hay que inicializar las librerías a utilizar y declarar las variables locales que se utilizaran tal como se muestra en el siguiente código.

```
int main(void) {
	UART_RXSTATUS_t check;
	SystemInit();
	UB_Uart_Init();
	UB_Led_Init();
	AV_Zfm20x_Delay(0xFFFFFF);
	AV_Zfm20x_Init();
	UB_ADC1_SINGLE_Init(); // Init vom ADC1
	UB_DigIn_Init();
	UB_TIMER2_Init(8399,9999);
	char buff[50];
	int bus = 0;
	int var = 0;
	uint16_t dato=0;
	int newDir=0;
	UB_TIMER2_Start();
```

En la tabla 6 se indican los diferentes puertos existentes en la tarjeta los cuales están disponibles para su utilización distribuidos de acuerdo a cada puerto al que pertenecen.

Tabla 6 . Pines disponibles de la tarjeta distribuidos por puertos

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	113	14	15
Port A	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7	PA8	PA9	PA10			PA13	PA14	PA15
Port B	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7	PB8	PB9	PB10	PB11	PB12	PB13	PB14	PB15
Port C	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15
Port D	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7	PD8	PD9	PD10	PD11	PD12	PD13	PD14	PD15
Port E	PE0	PE1	PE2	PE3	PE4	PE5	PE6	PE7	PE8	PE9	PE10	PE11	PE12	PE13	PE14	PE15

Fuente: http://www.sciencezero.org/index.php?title=STM32F407_Microcontroller

En la tabla 7 se indican cuáles son los pines disponibles en la tarjeta que permiten realizar una comunicación serial.

Tabla 7. Pines asignados para comunicación

Periférico	Función	Pin
USART 1	TX	PA9
USART 1	RX	PA10
USART 2	TX	PA2
USART 2	RX	PA3
USART 3	TX	PB10
USART 3	RX	PB11
UART 4	TX	PA0
UART 4	RX	PA10
UART 5	TX	PC12
UART 5	RX	PD2

Fuente: http://www.sciencezero.org/index.php?title=STM32F407_Microcontroller

Para la comunicación serial se utilizó el puerto serial USART 3, que corresponden a los pines PB10 para la transmisión y PB11 para la recepción.

En el siguiente código se muestra el almacenamiento de la información enviada por la Tablet hacia la placa mediante puerto serial en una variable check; se realiza la lectura del puerto serial COM3 y la información es almacenada en una variable local llamada buff de tipo char de 50 caracteres.

```
check = UB_Uart_ReceiveString(COM3, buff);
if (check == RX_READY) {
ident = buff[0];
for (var = 0; var < 50; ++var) {
      buff[var]=buff[var+1];
      }
dato=UB_String_DezStringToInt(buff);
```

El siguiente código se utiliza para la lectura de la temperatura y los sensores de movimiento y de humo instalados en el laboratorio. Se introduce dentro de una función denominada **(UB_TIMER2_ISR_CallBack)** la cual será la encargada realizar el proceso de lectura de datos cada determinado tiempo.

```
void UB_TIMER2_ISR_CallBack(void)
{

      temp=UB_ADC1_SINGLE_Read_MW(ADC_PC4)*0.072;

      if (alarma1==1 && aux_a1==0) {
            UB_Uart_SendString(COM3, "l1", LFCR);
            aux_a1=1;
      }

      if (alarma2==1 && aux_a2==0) {
            UB_Uart_SendString(COM3, "l2", LFCR);
            aux_a2=1;
      }
}
```

La variable temp corresponde a la temperatura ambiente registrada por el sensor lm35 que es leída mediante el puerto PC4 correspondiente al ADC de la placa, la variable alarma1 corresponde al sensor de movimiento si este se activa envía el código “l1” y la variable alarma2 corresponde al sensor de humo si este se activa se envía el código “l2” mediante el puerto serial hacia la tablet.

2.5.2.3. Conexión de la placa electrónica con la Tablet

La trasferencia de datos rápida y eficiente es de vital importancia; la placa de desarrollo STM32F4 es la encargada de conectarse y obtener información de los sensores y decidir lo que se va a hacer con dicha información según lo programado por el administrador. Pero estos datos deben de ser legibles e informativos para cualquier usuario o para el administrador mismo, por lo cual estos datos serán enviados hacia un HMI, el cual viene a ser la Tablet con sistema Android, por lo cual hay que comunicar a la placa electrónica con la Tablet.

La comunicación físicamente se lo realiza de la siguiente manera: se utiliza el puerto de comunicación serial de la placa y se lo conecta al puerto mini USB disponible en la Tablet para realizar una comunicación serial. Para lo cual se hace uso de dos adaptadores el primero un convertidor de TTL serial a USB, y luego un conector USB hembra a macho mini USB tipo B como se indica en la figura 41.

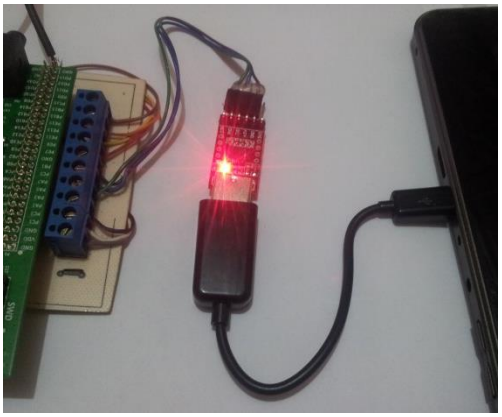


Figura 41. Conexión placa-tablet.

Fuente: autor

Para activar la comunicación serial en la placa se utiliza el siguiente comando en la programación de CoIDE la cual hace referencia al COM3, es decir que se utiliza el puerto UART3- (TX: pin [PD8] y RX: pin [PD9])

```
check = UB_Uart_ReceiveString(COM3, buff);
```

En la siguiente secuencia de comandos se declara la función para el uso de puerto USB de la Tablet para comunicación serial.

```
private static UsbSerialDriver sDriver = null;

private SerialInputOutputManager mSerialIoManager;
private final SerialInputOutputManager.Listener mListener =
        new SerialInputOutputManager.Listener() {
    public void onRunError(Exception e) {
        Log.d(TAG, "Runner stopped.");
    }
    public void onNewData(final byte[] data) {
        SerialConsoleActivity.this.runOnUiThread(new Runnable() {
            public void run() {
                SerialConsoleActivity.this.updateReceivedData(data);}
        });
    }};
```

Sin embargo se acompaña también de las funciones MESSAGE_REFRESH y REFRESH_TIMEOUT_MILLIS que permiten la obtención continua de datos que se ingresen en la tablet dentro de un tiempo muy corto establecido en 5000 milisegundos.

```
private static final int MESSAGE_REFRESH = 101;
private static final long REFRESH_TIMEOUT_MILLIS = 5000;
```

La placa electrónica STM32F407 se conecta mediante el puerto de comunicación serial al puerto USB de una Tablet Kaissen PRO con sistema android, en la cual se encuentra instalada una aplicación creada por los tesistas que cumple con la función de HMI (interfaz maquina humano) para interactuar con el usuario mediante su pantalla táctil además de intercambiar datos con la placa; que será la encargada de administrar y obtener información de la base de datos almacenada en el servidor web de forma inalámbrica haciendo uso de la red de telefonía móvil mediante la conexión de la Tablet con una tarjeta SIM de cualquier operadora logrando conectar todo el sistema usando la internet como medio de comunicación para intercambiar la información entre el sistema y el servidor web.

2.5.2.4. Programación de la tablet

La interfaz hombre maquina es la encargada de interactuar con el usuario permite visualizar los datos de los sensores así como registros desde la base de datos, permite administrar el sistema, enrolar nuevos usuarios, controlar los parámetros del sistema; es decir permite la administración y control del sistema, además cumple la función de puente para conectar todo el sistema desde las partes más sencillas como los sensores hasta las más complejas como el intercambio de información mediante servicios web que utilizan bases de datos para el registro y almacenamiento de información.

Al hacer uso de una Tablet para el HMI se ahorra recursos y espacios ya que la Tablet cumple la función de una computadora, además de permitir visualizar la información en su pantalla; también ofrece la utilidad de poder ser usada como teclado para ingresar información gracias a que posee una pantalla touch.

Por lo cual se creó un software específico para cumplir con los requerimientos del sistema para lo cual se creó diferentes pantallas haciendo uso de la interfaz de programación Android Studio la cual se muestra en la figura 42.

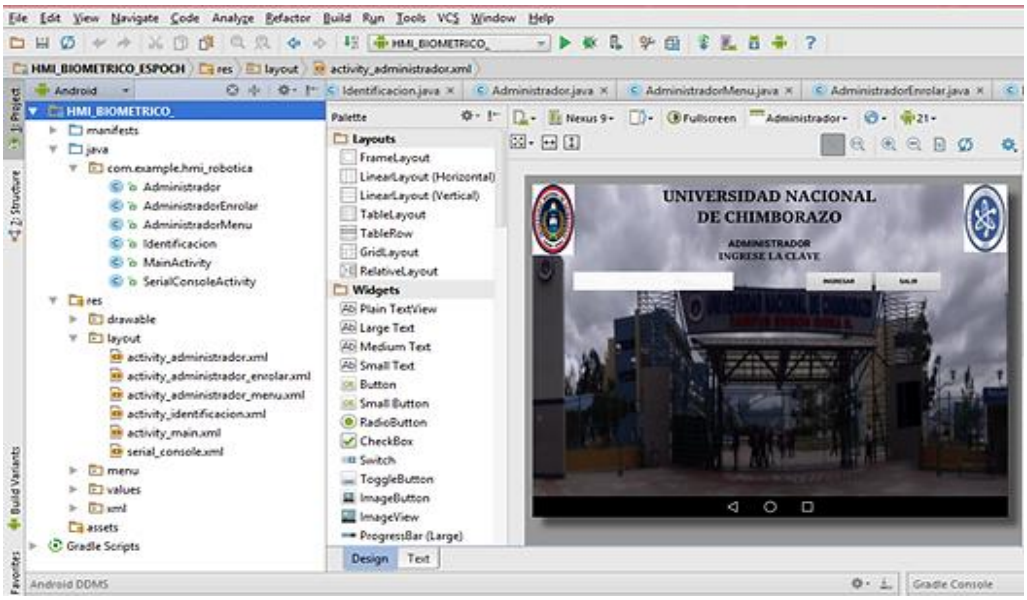


Figura 42. IDE Android Studio
Fuente: Auto

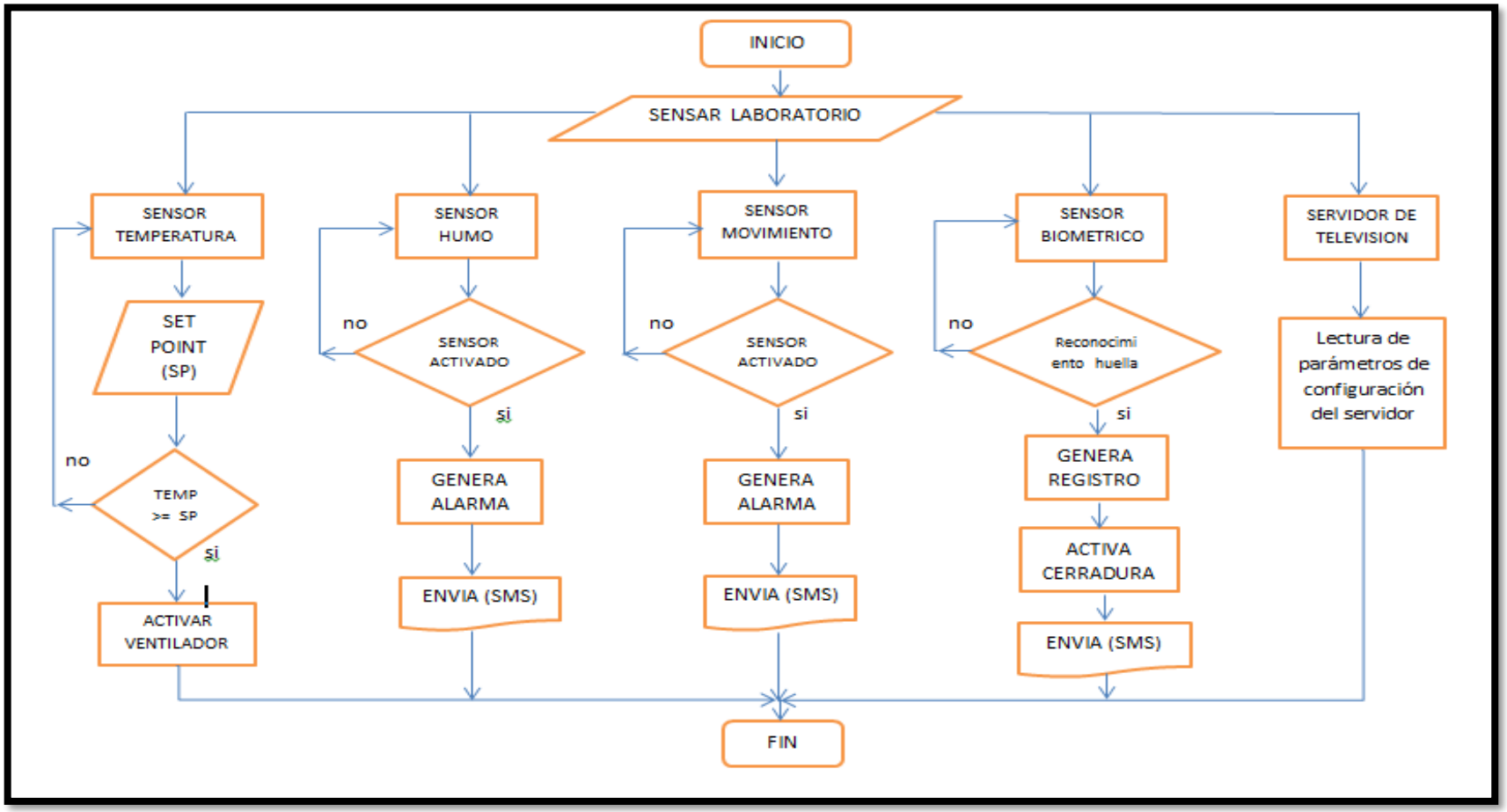


Figura 43. Flujograma de programación
Fuente: Autor

En la figura anterior se puede observar el flujograma que se utilizó para la programación del HMI.

Para crear una nueva actividad, es decir que para cada pantalla de presentación se crea un archivo de programación; como se observa en la figura 43 las pantallas de presentación se almacenan en la carpeta layout, la programación de cada una de estas se almacenan en la carpeta java. A continuación se explica la función y programación de cada una de las actividades del programa.

En la figura 44 se indica la pantalla principal de presentación del HMI, esta es una aplicación creada para sistema operativo Android, en esta primero pantalla se observa dos iconos el de identificación y administrador.

Figura 44. Pantalla principal
Fuente: autor

Al presionar el icono de identificación la aplicación nos envía a otra pantalla donde se solicita al usuario colocar su dedo para identificación, si se pulsa sobre el icono de administrador se accede a otra pantalla donde el administrador debe identificarse para acceder a las opciones de control del sistema. En la programación se muestra la declaración de los botones.

```
bIdent = (Button) findViewById(R.id.IDENT);
bAdmin = (Button) findViewById(R.id.ADMIN);
```

En las siguientes sentencias se muestra lo que sucede cuando se presiona uno de los botones de la pantalla; si se presiona el botón identificación visualiza en pantalla el mensaje de bienvenido usuario y asigna el valor de 1 para la variable de la estructura case, si se presiona el botón de administrador se visualiza en pantalla el mensaje de bienvenido administrador y se asigna el valor de 3 para la variable case.

```
bIdent.setOnClickListener(new OnClickListener() {
   public void onClick(View v) {
      timer.cancel();
      showConsoleActivity(1,"BIENVENIDO USUARIO");
   }
});

bAdmin.setOnClickListener(new OnClickListener() {
   public void onClick(View v) {
      timer.cancel();
      showConsoleActivity(3,"BIENVENIDO ADMINISTRADOR");
   }
});
```

En las siguientes sentencias se muestra el código de la estructura case, el cual se utiliza para el cambio de actividades y de pantallas de presentación en el programa. Es decir si se presiona el botón de identificación se da un valor de 1 a la variable case, lo cual envía al programa a la actividad Identificación.

```
switch (pag){
        case 0:
           pag=0;
           SerialConsoleActivity.show(this, driver);
           break;
        case 1:
           pag=0;
           Identificacion.show(this, driver);
           break;
        case 2:
           pag=0;
           break;
        case 3:
           pag=0;
           Administrador.show(this, driver);
           break;
```

```
case 4:
   pag=0;
   AdministradorMenu.show(this, driver);
   break;
case 5:
   pag=0;
   AdministradorEnrolar.show(this, driver);
   break;
default:
   pag=0;
   SerialConsoleActivity.show(this, driver);
   break;
```

En la figura 45 está la pantalla de identificación del usuario, aquí se solicita que coloque su dedo sobre el lector de huellas para identificación, aquí se accede mediante programación al lector de huellas y se compara el número de identificación dado por el lector con la base de datos si existe ese código en la base es un usuario registrado y tiene acceso al laboratorio,

Figura 45. Pantalla de identificación de huella

Fuente: autor

Si la identificación es correcta se activa la chapa eléctrica y además se guarda la fecha y hora de acceso al laboratorio junto con la identificación de la persona que accedió al laboratorio, además por seguridad se envía un sms al celular de la persona administradora del laboratorio.

Para cumplir con estos objetivos se utiliza el siguiente código de programación en esta actividad donde el sensor biométrico envía el valor de 1 en caso de detectar una coincidencia es decir un usuario registrado y 0 si no encontró coincidencia.

```
switch (data[1]){
        case '0':
           showConsoleActivity(0,"USUARIO NO ADMITIDO");
           break;
        case 't':
           showConsoleActivity(0,"TIEMPO AGOTADO");
           break;
        case '1':
           byte[] optt;
           String see="A1\r";
           optt = see.getBytes();
          }
```

De ser el caso que se haya detectado una coincidencia el programa envía el código “A1” hacia la placa STM32F4 lo que permite que esta accione la cerradura eléctrica.

En las siguientes sentencias se muestra los comandos para almacenar generar código asignado a cada usuario, la fecha y la hora a las cuales se registró el ingreso al laboratorio; además el comando para enviar por mensaje de texto dicha información al número de celular asignado.

```
String codigo=new String(data);
codigo=codigo.substring(1,data.length-2);

SimpleDateFormat fecha = new SimpleDateFormat("dd/MM/yyyy");
String sFecha = fecha.format(new Date());

SimpleDateFormat hora = new SimpleDateFormat("HH:mm:ss");
String sHora = hora.format(new Date());

 sendSMS("0983499915"," Ingreso Usuario:"+codigo+" "+sFecha+" "+sHora);
```

En la figura 46 se observa la pantalla de identificación del administrador, se solicita una clave la cual solo debe poseer la persona encargada del laboratorio la cual será la encargada de enrolar nuevos usuarios y verificar el funcionamiento.

Figura 46. Pantalla de identificación del administrador

Fuente: autor

En esta actividad se declara las variables locales a utilizar como clave, el cuadro de texto donde se escribe la contraseña y los botones.

```
String clave="a1234";
EditText ePass;
Button bIngresarAdmin, bSalirAdmin;

    ePass = (EditText) findViewById(R.id.PASS_AD);
    bIngresarAdmin = (Button) findViewById(R.id.INGRESAR_AD);
    bSalirAdmin = (Button) findViewById(R.id.SALIR_AD);
```

Es en esta función donde se lee los caracteres introducidos en el cuadro de texto de nombre EditText y los compara con los de la variable clave; de ser correcta se visualiza en pantalla el mensaje “clave correcta” y envía al programa a la actividad o pantalla 4.

```
    bIngresarAdmin.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            if(ePass.getText().toString().equals(clave)){
                showConsoleActivity(4,"CLAVE CORRECTA");
```

```
    }else{
        showConsoleActivity(0,"CLAVE INCORRECTA");
    }
});
```

En la figura 47 se observa la pantalla a la que se accede después de ingresar a la opción de administrador, en esta pantalla se observan las opciones de enrolar usuario, set point temperatura, borrar.

Figura 47. Pantalla de administración

Fuente: autor

Enrolar usuario se refiere a la opción de ingresar nuevos usuarios al sistema de acceso al laboratorio esta nos envía a otra pantalla donde se registra los nombres y la huella de la persona, el sp (set point) temperatura permite calibrar la temperatura ambiente a la que se requiere mantener el laboratorio, y en borrar se eliminar de registro a usuarios registrados.

En la figura 48 se observa la pantalla de registro para agregar a un nuevo usuario, en esta pantalla se registran los datos del nuevo usuario además de su huella dactilar mediante el sensor biométrico para que pueda tener acceso al laboratorio, estos datos de usuario son almacenados en la base de datos del servidor web, el programa asigna un código a cada usuario dicho código es comparado con la base de datos para verificar la existencia del usuario y de esa manera permitir el acceso.

Figura 48. Pantalla para agregar usuario

Fuente: autor

En las siguientes sentencias se muestra el código para guardar el nombre de un nuevo usuario y activar el sensor biométrico para registrar una nueva huella.

```
bGuardarEnrolar.setOnClickListener(new View.OnClickListener() {
      public void onClick(View view) {
        tCodigo.setText("");
        if(eNombre.getText().toString().equals("")){

        }else{
          byte[] opt;
          String se="E0\r";
          opt = se.getBytes();
          try {
            sDriver.write(opt, 15);
          } catch (IOException e) {
            e.printStackTrace();
          }

          se="D15\r";
          opt = se.getBytes();
          try {
            sDriver.write(opt, 15);
          } catch (IOException e) {
            e.printStackTrace();
          }
        }
```

El HMI trabaja en conjunto con la placa electrónica y la base de datos del servidor web, para comparar y enviar datos en tiempo real.

El sistema genera alarmas que permiten advertir al administrador del laboratorio de cualquier irregularidad o emergencia que se pueda presentar en el mismo. El siguiente código describe como se realiza el proceso y registro de dichos eventos.

La placa electrónica STM32f4 envía la información de los sensores a la tablet esta información es leída en la variable data, entonces si la variable data es igual a 1 representa que se activó el sensor de movimiento y si además su activación fue entre las 19:00 horas y las 6:00 horas entonces se genera la alarma de detección de intruso.

```
        if (data[1]=='1'&&
((Integer.parseInt(sHora.substring(0,2))>19)||(Integer.parseInt(sHora.substring(0,2))<6)))
{
            sendSMS("0983499915", "1 Detección INTRUSO "+sFecha+" "+sHora);
```

Además esta información es enviada mediante sms al celular del administrador y registrada en una base de datos que se encuentra en un dominio externo como se muestra en el siguiente código:

```
httpGetData("http://www.alluc.com.ec/trabajo/registrarAlarmasUnach.php?codigo=1&no
mbre= Intruso&fecha="+sFecha+"&hora="+sHora);
              Toast.makeText(getApplicationContext(), "El dato ha sido enviado
correctamente", Toast.LENGTH_LONG).show();
            }catch(Exception e){
              Toast.makeText(getApplicationContext(), "Error en el envio de la
informacion, verifique su conexion a internet y vuelva a intentarlo.",
Toast.LENGTH_LONG).show();
            }
        }
```

Si la información enviada por la tarjeta es igual a 2, entonces se genera una alarma de incendio que sigue el mismo proceso descrito en el código anterior.

2.5.2.5. Monitoreo del servidor de televisión digital

Este archivo contiene todos los parámetros de la configuración de nuestro servidor, dado que se encuentran en formato XML es relativamente fácil hacer consultas a cada bloque (Brick). El servidor trabaja con el sistema VILLAGEFLOW el cual trabaja con tres bloques fundamentales que son las entradas, el procesamiento y las salidas representadas en la figura 49.

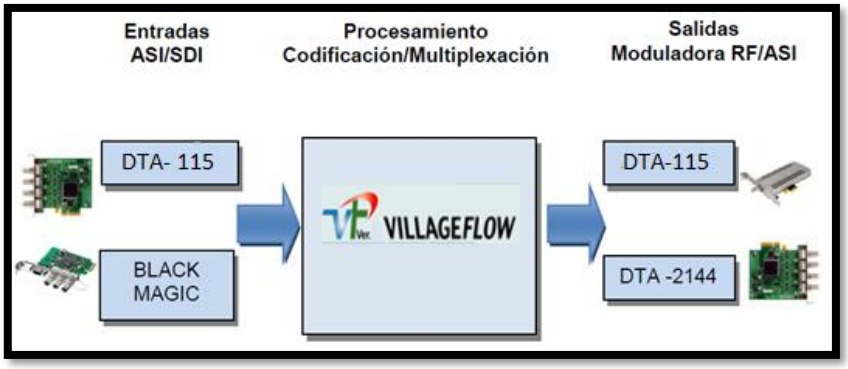


Figura 49. Funcionamiento básico del servidor VILLAGEFLOW

Fuente: Autor

1.- Bloques de entrada

Estos son los encargados de producir la conversión de una o más señales analógicas del dominio de frecuencia digital, codificando el video, audio y datos por separado.

2.- Bloques de proceso

Aquí se realiza la multiplexación del contenido codificado proveniente de los bloques de entrada, así como también la configuración de los parámetros TMCC

3.- Bloques de salida

Permite la configuración de los diferentes tipos de salida tales como: RF, IP, ASÍ y creación de archivos TS (TRANSPORT STREAM).

4.- TS es el protocolo de comunicación para el estándar MPEG-2 ISO/IEC 13818-1, que permite la multiplexación de video, audio y datos en un único flujo de bits de transmisión síncrona. Cada TS está formado por paquetes de transportes con una longitud fija de 188 bytes. El TS está formado de flujos elementales de señal (video, audio, datos, etc.)

Como se muestra en la figura 50 la configuración del servidor se lo realiza mediante bloques, para transmitir en un solo canal contenido, los bloques de entrada pueden ser diversos servicios como ONE SEG, interactividad (Ginga), guía de programación EPG, EWBS. Los bloques de proceso son Remux y Tmcc Encoding y los bloques de salida tenemos Dektec output card y Asi Out.

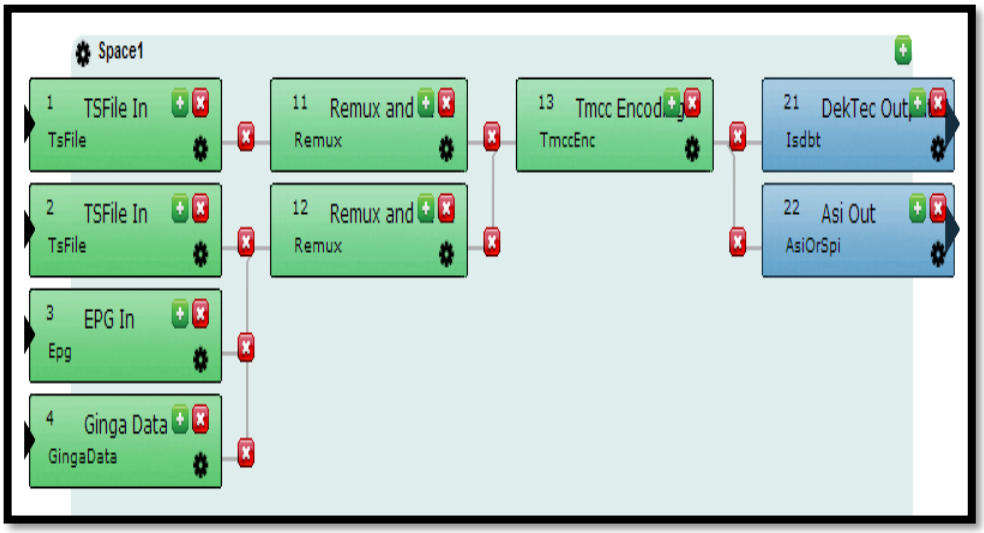


Figura 50. Espacio de VillageFlow configurado

Fuente: Autor

La información de todos los bloques de configuración son almacenados en un archivo denominado (VF_CONF) con formato XML. Dicho archivo de configuración es al que se debe ingresar para realizar consultas de los parámetros de configuración de cada bloque.

Se puede observar gráficamente como está formado este archivo de configuración en la figura 51 de la siguiente página.

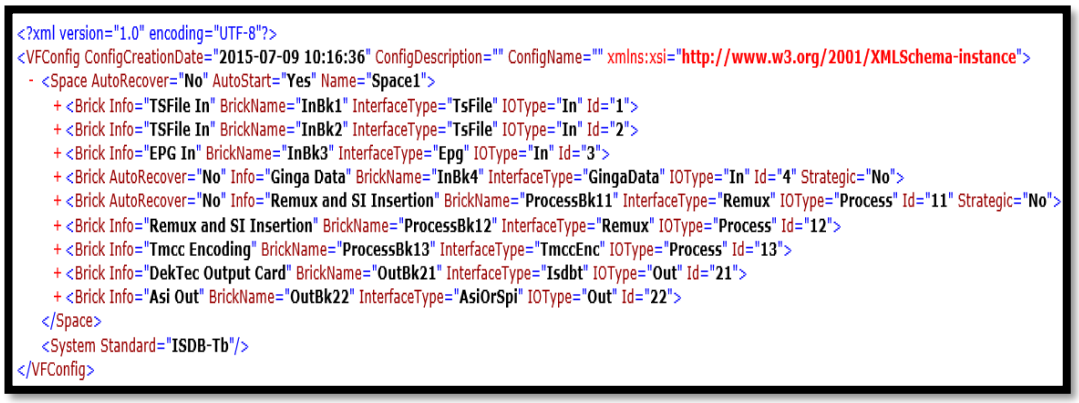


Figura 51. Archivo VF_CONF.XML

Fuente: Autor

2.5.2.5.1. Parámetros de configuración del servidor

2.5.2.5.1.1. Brick TS FILE IN

Se generan los archivos de entrada de video .TS, se debe configurar de acuerdo a la tabla 8 que muestra a continuación:

Tabla 8. Configuración del Brick TS FILE IN

PARAMETRO	VALOR	OBSERVACIONES
Brick Info		Solo para parámetros informativos, puede ser modificado pero no afecta el proceso del VF
Ts Rate	0	Se deja en 0 la mayoría de los casos ya que el VF es capaz de evaluar y ajustar automáticamente este valor
Ts Packet Size	0	Valor por defecto debido a que el Village Flow encuentra el valor automáticamente
Loop	0	Valor por defecto debido a que el Village Flow encuentra el valor automáticamente.
File Path		Ruta del archivo
Continuity Counter Restamp	NO	Permite que el parámetro loop sea infinito

Fuente: autor.

Todos los parámetros son cero o nulos por defecto así que no es necesario monitorear este bloque del servidor ya que prácticamente no necesita ser configurado.

2.5.2.5.1.2. Brick EPG (guía de programación electrónica)

Bloque dedicado a la generación de la Guía de Programación Electrónica EPG.

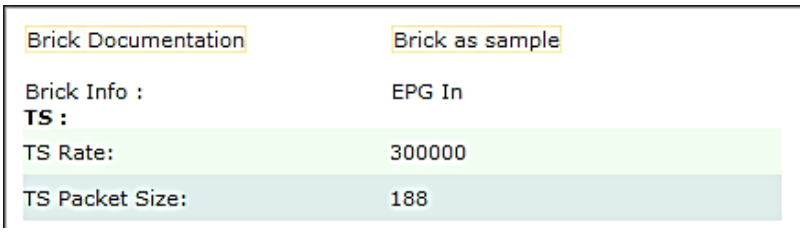


Figura 52. Parámetros de configuración del Brick EPG

Fuente: Autor

En la figura 52 se muestran los parámetros configurados en el bloque EPG y que serán monitoreados; TS Rate (la tasa de salida de datos) y TS Packet Size.

2.5.2.5.1.3. Brick GINGA DATA

Este bloque de entrada codifica aplicaciones GINGA de lenguaje NCL o JAVA en un TS encapsulado en un carrusel de objetos dando como resultado el flujo de datos MPEG. Se establece los parámetros de configuración en el bloque de entrada GINGA de la interfaz de VillageFlow.

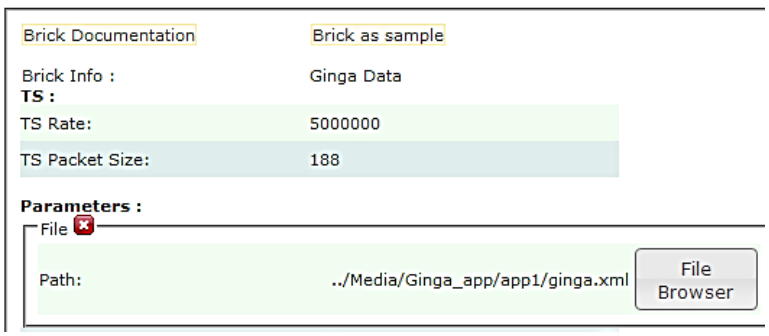


Figura 53. Parámetros de configuración del Brick EPG

Fuente: Autor

2.5.2.5.1.4. Brick REMUX

Este bloque de proceso se encarga de la multiplexación del contenido audio, video y datos de las entradas con las Tablas PSI/SI correspondientes a cada uno de los servicios.

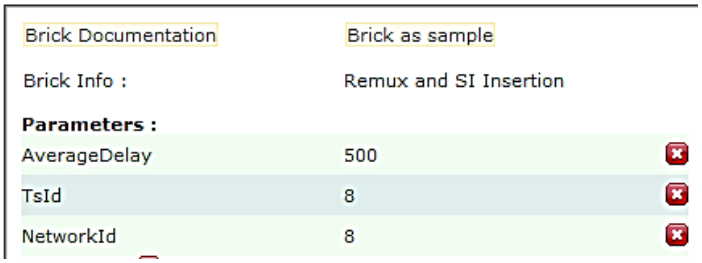


Figura 54. Parámetros de configuración del Brick REMUX SIMPLE

Fuente: Autor

El parámetro a monitorear en este bloque es el de AverageDelay que es el retardo promedio en milisegundos; este debe estar en un rango entre 500 a 1000 mseg.

2.5.2.5.1.5. Brick TMCC ENCODING

El bloque de proceso TMCC (Transmisión Configuration Control) realiza el control de transmisión, configuración del segmento del canal y parámetros de transmisión en la figura 55 se observa los posibles valores a configurar.

NOMBRE	DESCRIPCIÓN	POSIBLES VALORES
Broadcast	Tipo de transmisión para ISDBT	tv, rad1, rad3
Default Layer	Capa en la cual se asignan por defecto los PID entrantes.	A, B, C
Emergency	Activación / Desactivación de la bandera de alerta de emergencia (EWBS)	*yes, no*
Guard	Selección del intervalo de guarda	1_4, 1_8, 1_16, 1_32

Figura 55. Parámetros de configuración del TMCC ENCODING

Fuente: Autor

En los parámetros de configuración se tiene que configurar los tres canales de los que dispone el servidor para el caso de nuestro se está transmitiendo en las 3

capas pero cada una con diferente modulación. El usuario puede seleccionar si desea cambia de capa por defecto para transmitir con otra modulación disponible.

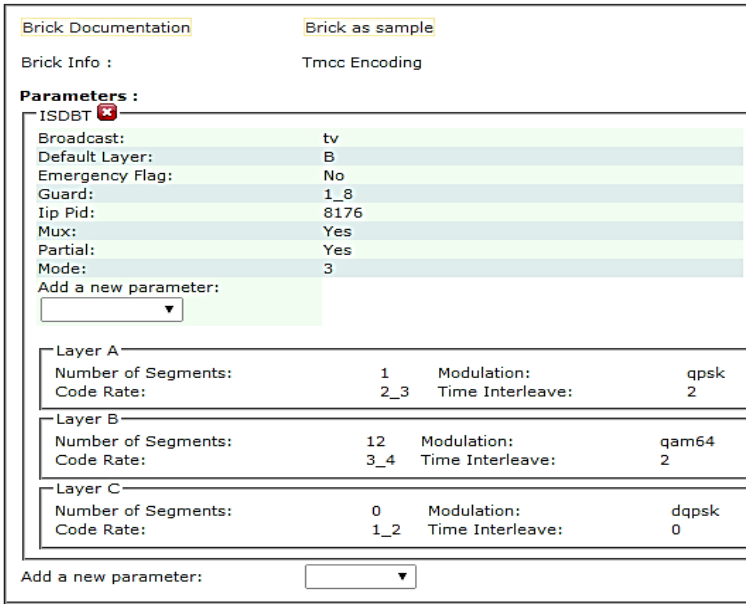


Figura 56. Parámetros de configuración del Brick TMCC ENCODING

Fuente: Autor

2.5.2.5.1.6. Brick DEKTEC OUTPUT CARD

Este bloque de salida establece los parámetros de transmisión. Se establece también la tasa de datos para un ancho de banda de 6 MHz, el tamaño de los paquetes, el nivel RF a la salida, el canal de salida y su frecuencia tal como indica la figura 57.

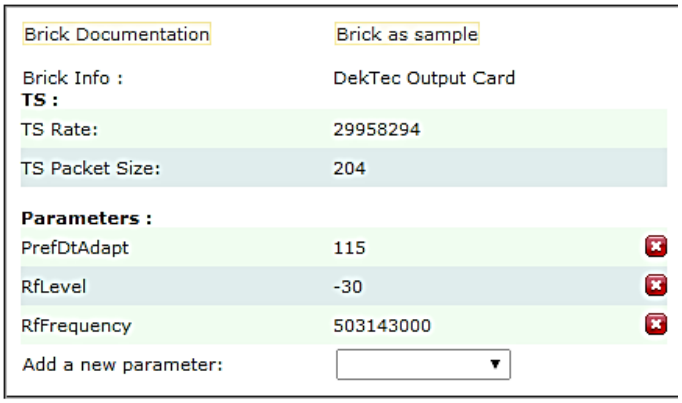


Figura 57. Parámetros del Brick DEKTEC OUTPUT CARD

Fuente: Autor

2.5.2.6. Conexión de la Tablet con el servidor

La comunicación entre la aplicación Android instalada en la Tablet y el servidor web se la realiza de forma inalámbrica utilizando como canal de comunicación el internet móvil de cualquier operadora para el intercambio de información entre el HMI y la base de datos.

El objetivo de crear un servidor web es almacenar lo datos y registros de ingresos y alarmas de forma segura; además permite al administrador acceder a esta información de manera remota desde cualquier dispositivo que tenga instalada la aplicación Android de control HMI o la aplicación para monitoreo lo que permite estar conectado en tiempo real con el sistema; además se puede acceder al sistema desde cualquier parte del mundo ya que solo requiere acceso a internet.

2.5.2.6.1. Diseño y funcionamiento del Servidor web

A continuación se explica cómo se diseñó y el funcionamiento del servidor web, como se indica en la figura 58. el proceso de comunicación va desde el dispositivo con sistema Android el cual realiza él envió y petición de datos hacia el servidor es decir una conexión punto a punto utilizando la nube de internet como medio de comunicación.

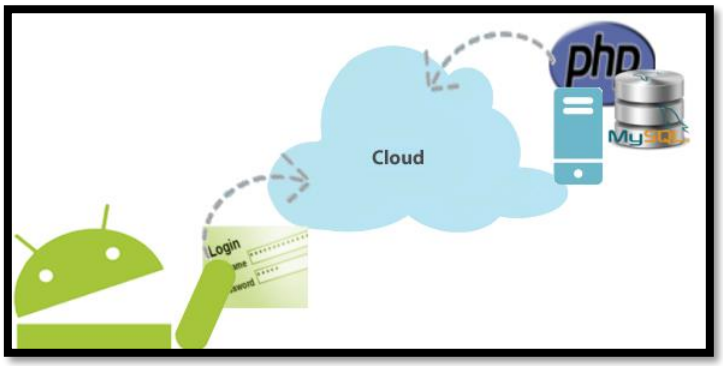


Figura 58. Funcionamiento y comunicación del servidor web

Fuente: http://androideity.com/

La aplicación creada para la Tablet con sistema operativo Android se conecta a la nube de internet a través de la cual busca comunicarse a una IP pública o dominio de internet en la cual va a acceder mediante archivos PHP cargados previamente en el servidor web a la base de datos creada bajo MySQL.

El dominio utilizado en el proyecto es el siguiente: **http://www.alluc.com.ec**

Se parte desde el servidor web debido a la imperativa necesidad de conocer el dominio en el cual se almacenara la base de datos y al cual se comunican los dispositivos móviles para acceder a los datos.

Todo servidor web contiene por defecto el software MySQL que se utiliza para crear base de datos utilizada en la web, este sistema de base de datos se ejecuta directamente en el servidor es muy eficaz para aplicaciones pequeñas como las de este proyecto.

Las bases de datos se crean en el entorno phpMyAdmin que se muestra en la figura. Para ingresar a este entorno hay que abrir cualquier navegador web y escribir la siguiente dirección IP (127.0.0.1); la cual es la dirección por defecto del servidor. Una vez ingresado al servidor web hay que ingresar en la opción herramientas y abrir la aplicación phpMyAdmin.

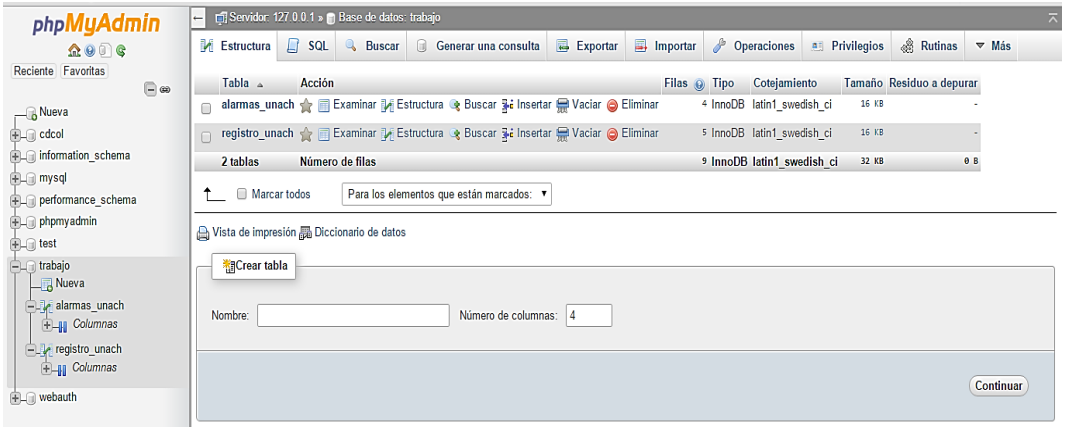


Figura 59. Entorno de MySQL

Fuente: Autor

Como se muestra en la figura anterior es aquí donde se crea la tabla alarmas y la tabla registro, que conforman la base de datos. Entonces basta con crear estas dos tablas dentro de una carpeta creada por el administrador, para este proyecto se creó una carpeta denominada trabajo la cal contiene a las dos tablas, y es a esta carpeta a la que hay que accesar remotamente para realizar las actualización y consultas de las bases de datos.

Al crear la base de datos directamente en el servido permite ejecutar consultas, ingresar datos o registros en la base, actualizar o eliminar registros, entre otras funciones. Los datos son almacenados en tablas como las que se muestran a continuación en la tabla 9.

Tabla 9. Tabla de base de datos alarmas UNACH.

codigo	nombre	fecha	hora
1	Intruso	1	1
1	Intruso	1	2
1	Intruso	07/05/2015	14:50:22
1	Intruso	07/05/2015	14:50:24
2	Fuego	07/05/2015	14:51:19
2	Fuego	07/05/2015	14:51:24
2	Fuego	07/05/2015	14:53:46
2	Fuego	07/05/2015	14:54:03
2	Fuego	07/05/2015	14:55:50
2	Fuego	07/05/2015	14:55:51
1	Intruso	07/05/2015	14:56:44
1	Intruso	07/05/2015	14:56:47
1	Intruso	07/05/2015	14:56:49
1	Intruso	07/05/2015	14:57:56
2	Fuego	07/05/2015	16:37:00
1	Intruso	01/06/2015	15:28:25
1	Intruso	01/06/2015	15:28:26
2	Fuego	01/06/2015	15:28:30
2	Fuego	01/06/2015	15:28:30

Fuente: Autor

Las tablas se pueden crear mediante uso de comando o utilizando métodos gráficos. A continuación se muestra la declaración para crear la tabla alarmas.

```
CREATE TABLE IF NOT EXISTS `alarmas_unach` (
  `codigo` bigint(2) unsigned NOT NULL,
  `nombre` text COLLATE utf8_bin NOT NULL,
  `fecha` text COLLATE utf8_bin NOT NULL,
```

```
  `hora` text COLLATE utf8_bin NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

A continuación se muestra el código para generar la tabla de registro unach:

```
CREATE TABLE IF NOT EXISTS `registro_unach` (
  `codigo` bigint(2) unsigned NOT NULL,
  `nombre` text COLLATE utf8_bin NOT NULL,
  `accion` text COLLATE utf8_bin NOT NULL,
  `fecha` text COLLATE utf8_bin NOT NULL,
  `hora` text COLLATE utf8_bin NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

La tabla 10 muestra cómo se ordenan y almacenan los datos cada vez que se ingresa al laboratorio o se enrola a un nuevo usuario los datos de cada campo son enviados desde el HMI instalada en la Tablet hacia al servidor mediante internet y utilizando los archivos php para acceder y actualizar los nuevos datos en la base de datos.

Tabla 10. Tabla de base de datos registro UNACH.

codigo	nombre	accion	fecha	hora
1	Alexis Velastegui	ENROLAR	04/05/2015	17:17:36
1	VERIFICAR	INGRESO	04/05/2015	17:18:11
1	VERIFICAR	INGRESO	04/05/2015	17:18:35
3	Fernando Asd	ENROLAR	04/05/2015	17:25:45
4	Awd	ENROLAR	04/05/2015	17:28:59
1	VERIFICAR	INGRESO	06/05/2015	16:45:06
1	VERIFICAR	INGRESO	05/05/2015	17:05:19
1	Verificar	Ingreso	06/05/2015	17:19:04
1	Verificar	Ingreso	06/05/2015	17:22:40
1	Verificar	Ingreso	07/05/2015	14:49:16
1	Verificar	Ingreso	07/05/2015	16:36:11
1	Verificar	Ingreso	06/05/2015	18:15:15
1	Verificar	Ingreso	06/05/2015	18:31:47
1	Verificar	Borrar	06/05/2015	18:33:45
1	alexis	Enrolar	06/05/2015	18:37:45
1	Verificar	Ingreso	06/05/2015	18:38:17
2	fernando	Enrolar	06/05/2015	18:38:52
2	Verificar	Ingreso	06/05/2015	18:39:07
1	Verificar	Borrar	06/05/2015	18:40:04

Fuente: autor

Mediante archivos PHP almacenados en el servidor web, un usuario puede conectarse y manipular bases de datos. Por lo cual se diseñaron 8 archivos PHP que van a cumplir con dichas funciones. Estos archivos se almacenan en una carpeta en el dominio (www.alluc.com.ec) tal como se muestra en la figura 60, y se pueden accesar de manera remota lo que permite el monitoreo.

Figura 60. Archivos php en el dominio.

Fuente: Autor

Para subir los archivos php al dominio hay que crear una carpeta que contenga a dichos archivos y guardarla en la carpeta principal del servidor, tal como se muestra en la figura 61

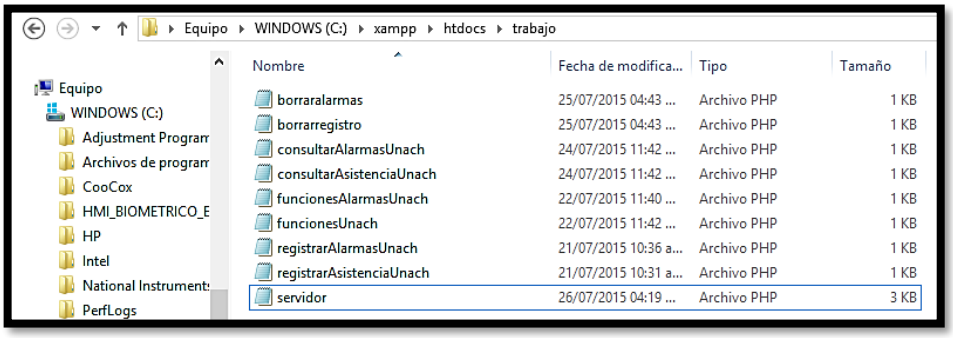


Figura 61. Archivos php en el servidor.

Fuente: Autor

A continuación se describe la estructura y funcionamiento de los archivos php.
Un archivo php contiene dos partes importantes:

1. El encabezado que indica el formato de codificación de caracteres o tipo de datos que se utilizan.

-header ('Content-Type: text/html;charset=utf-8');

2. La declaración de la ubicación de la carpeta de trabajo en la que se encuentra creada la base de datos.

-$mysqli=newmysqli("localhost","alluccom_jmln1","trabajo", "trabajo");

Una consulta es una pregunta o una petición. Podemos consultar una base de datos para obtener información específica, tal como se muestra a continuación en el código utilizado para el archivo PHP consultarAlarmasUnach

```
$codigo=$_GET['codigo'];
if ($resultset = getSQLResultSet("SELECT codigo,nombre,fecha,hora
FROM  `alarmas_unach` where codigo='$codigo'")) {
        while ($row = $resultset->fetch_array(MYSQLI_NUM)) {
        echo json_encode($row);       }
    }
```

Para ingresar nueva información en la base de datos se utilizan la siguiente secuencia en un archivo PHP de nombre registrarAlarmasUnach.

```
ejecutarSQLCommand("INSERT INTO  `alarmas_espoch` (`codigo` ,`nombre` ,
`fecha`,`hora`) VALUES ( '$codigo' , '$nombre' , '$fecha', '$hora') ");
```

2.5.2.7. Almacenamiento y lectura de datos

La información generada por el sistema de telemetría es almacenada en la base de datos que se creó en el servidor de dominio (www.alluc.com.ec), dicha información es clasificada en dos tablas; la tabla de registro es donde se guarda los datos del usuario que ha ingresado al laboratorio haciendo uso del sensor biométrico y la segunda tabla de alarmas que es donde se guarda los datos generados por los sensores instalados en el laboratorio.

Entonces para almacenar nueva información hay que actualizar las tablas cada vez que ocurra un nuevo evento, para lo cual se hace uso de lenguaje SQL dentro de la programación del HMI en la Tablet, ya que será esta la encargada de comunicarse con el servidor vía internet y actualizar dichas tablas.

Para el envió de la información desde la Tablet hacia el servidor se utiliza el siguiente código de programación java en la Tablet:

```
httpGetData("http://www.alluc.com.ec/trabajo/registrarAsistenciaUnach.php?codigo="
+codigo+"&nombre= Verificar &accion= Ingreso &fecha="+sFecha+"&hora="+sHora);

Toast.makeText(getApplicationContext(), "El dato ha sido enviado correctamente",
Toast.LENGTH_LONG).show();

        }catch(Exception e){
Toast.makeText(getApplicationContext(), "Error en el envio de la informacion, verifique
su conexion a internet y vuelva a intentarlo.", Toast.LENGTH_LONG).show();
        }
```

El código anterior logra la comunicación con el servidor y además accede al archivo (registrarAsistenciaUnach.php) y envía la información de las variables código, nombre, acción, fecha y hora; que se utilizan para actualizar los campos de la tabla registroUnach.

Dicha información es almacena en tablas como las que se muestran en la siguiente figura 62.

codigo	nombre	accion	fecha	hora
1	alexis	Enrolar	06/05/2015	18:37:45
1	Verificar	Ingreso	06/05/2015	18:38:17
2	fernando	Enrolar	06/05/2015	18:38:52
2	Verificar	Ingreso	06/05/2015	18:39:07
1	Verificar	Borrar	06/05/2015	18:40:04

Figura 62. Almacenamiento de información.

Fuente: Autor

Para la lectura de los datos se debe realizar una consulta o petición a las tablas de las bases de datos para obtener información específica. Antes de poder acceder a la información en la base de datos MySQL, hay que comunicarse con el servidor lo cual se hace mediante los archivos PHP utilizando la siguiente sentencia:

```
$mysqli=new mysqli("localhost", "root", " ", "trabajo");
```

El primer campo corresponde a la dirección o dominio, ya que los archivos se almacenan el servidor se utiliza localhost, el segundo campo corresponde al usuario como se trata de un servidor el usuario es root, el tercero es la contraseña en caso de existir y el cuarto el nombre de la carpeta donde están almacenadas las tablas de bases de datos con las que se trabaja.

Una vez hecha la hecha la conexión con la base de datos se ingresa los comandos SQL que realizaran las diferentes consultas, ingresos y borrado de datos por ejemplo se muestra a continuación el código SQL para hacer consultas:

```
if ($resultset = getSQLResultSet("SELECT codigo,nombre,fecha,hora
FROM  `alarmas_unach` where codigo='$codigo' ORDER BY fecha DESC")
```

Hay 3 principales interfaces Cliente, servidor, base de datos.

- El cliente puede ser cualquier cosa móvil, navegador, script. Resumido cualquiera que puede hacer una petición al servidor es un cliente.
- Servidor puede ser cualquier máquina que tenga facilidad para ejecutar el servidor; todos los archivos de servicios web se almacenan en el servidor.
- Base de datos almacena los datos proporcionados por el cliente o ya completados.

Por ejemplo, si se dispone de una base de datos de clientes y se quiere obtener una lista con ellos, habría que acceder a la URL del recurso "clientes", http://servidor.com/clientes. Una petición GET a este recurso que devolvería un array con todos los clientes que tenemos en nuestra BB.DD.

Si lo que se desea es obtener la información de un cliente en particular, por ejemplo el cliente con identificador igual a 01, se haría una petición GET a la url http://servidor.com/clientes/01. Para crear un cliente nuevo se debería enviar por POST una petición a http://servidor.com/clientes con los datos del nuevo cliente.

Normalmente los datos que devuelve el servidor estarán en formato JSON o XML. Para nuestro ejemplo se utilizó JSON, ideal para trabajar con aplicaciones javascript. Mediante el uso la siguiente instrucción (**new JSONArray)** se obtiene los datos del servidor en nuestra aplicación Android.

En la figura 63 se muestra paso a paso la estructura del sistema básico para el intercambio de información entre la aplicación Android y el servidor web realizando consultas y peticiones desde la aplicación Android hacia la base de datos del servidor.

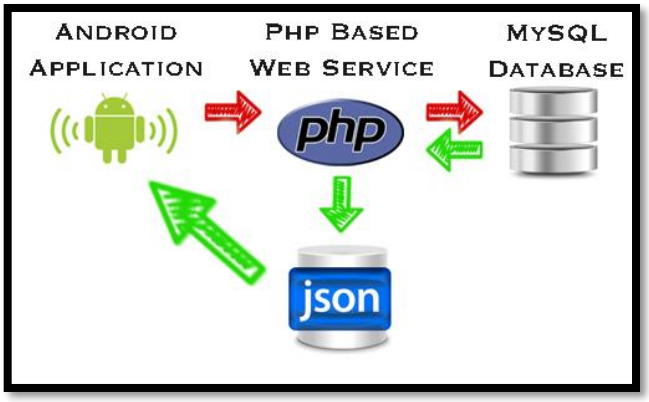


Figura 63. Intercambio de información.

Fuente: http://stackoverflow.com/

Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías. Para codificar datos en formato JSON basta con usar la siguiente instrucción dentro de nuestros archivos PHP: **json_encode(variable)**

2.5.2.8. Creación de la aplicación de monitoreo para Smartphone.

La aplicación de monitoreo para teléfonos inteligentes con sistema Android permite monitorear los datos de ingresos y alarmas, además de poder controlar de manera remota el acceso al laboratorio es decir abrir la puerta de ingreso al laboratorio con solo tocar un botón de la aplicación, las consultas y envió de información se la realiza de igual manera que la aplicación HMI es decir utilizando el internet móvil de cualquier operadora, además ya que es un dispositivo móvil puede accesar de otras formas al internet lo que permite al administrador comunicarse con el sistema desde cualquier parte del mundo.

Esta aplicación fue programada en eclipse que utiliza programación java. La aplicación puede ser instalada en cualquier teléfono inteligente que posea sistema

Android 4.1 Jelly Bean o superior, la pantalla de inicio de esta aplicación se muestra en la figura 64.

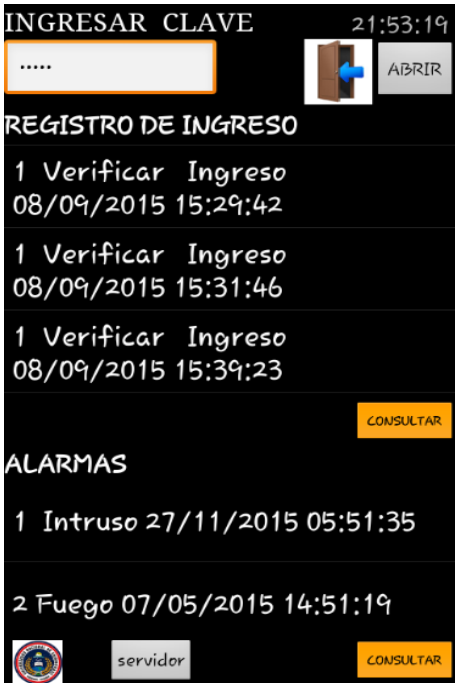


Figura 64. Pantalla de la aplicación para smartphone

Fuente: autor

El administrador tendrá que ingresar su clave maestra (a1234) para abrir remotamente la puerta de acceso al laboratorio. Además hay dos listas donde se visualizan el registro de ingreso y alarmas que se son procesos que se actualizan automáticamente en caso de haber actividad, basta con seleccionar la opción de consultar.

En la lista de REGISTRO se mostrara los campos de código asignado a cada usuario, y la fecha y hora específica en la que se accedió al laboratorio mediante el lector de huella dactilar. En la lista de ALARMAS se visualiza el código asignado a cada caso de alerta y la fecha y hora específica en la que se generó dicha alarma.

Para realizar las consultas del registro de ingreso se utiliza las siguientes sentencias en la programación de la aplicación para Smartphone en las cuales se describe la creación de una función llamada bConsultar que corresponde al primer botón de consulta:

```
bConsultar.setOnClickListener(new View.OnClickListener() {

   public void onClick(View view) {
    if(ePass.getText().toString().equals(clave)){
    JSONArray ja=null;
    try {
    String data;
    for(int j=0; j<7;j++){
data=httpGetData("http://www.alluc.com.ec/trabajo/consultarAsistenciaUnach.php?codigo="+Integer.toString(j));
    if(data.length()>1){
    for (int i = -1; (i = data.indexOf("[", i + 1)) != -1; ) {
      ja=new JSONArray(data.substring(i, data.length()));
      onAddItem(ja.getString(0)+" "+ja.getString(1)+" "
      +ja.getString(2)+" "+ja.getString(3)+" "+ja.getString(4));}
      }
     }
```

Mediante estas sentencias primero se compara si la clave ingresada en correcta y luego se accede al archivo *php almacenados previamente en el servidor; y se obtienen todos los datos de la tabla registro ordenados por código. A continuación se muestra el código utilizado dentro del archivo consultaAsistenciaUnach.php donde se ejecutan sentencias en lenguaje SQL que permiten realizar consultas de la tabla registro_unach y visualizar su contenido en orden descendente.

```
<?php
include('funcionesUnach.php');
$codigo=$_GET['codigo'];
if ($resultset = getSQLResultSet("SELECT codigo,nombre,accion,fecha,hora FROM `registro_unach` where codigo='$codigo' ORDER BY fecha DESC,hora DESC")) {

        while ($row = $resultset->fetch_array(MYSQLI_NUM)) {
        echo json_encode($row); }
  }  ?>
```

Para realizar la consulta del registro de alarmas se creó una función similar a la descrita anteriormente con la única diferencia que para realizar las consultas del registro de alarmas hay que acceder al archivo consultarAlarmasUnach.php tal como se indica en el siguiente código:

data=httpGetData("http://www.alluc.com.ec/trabajo/consultarAlarmasUnach.php?codigo="+Integer.*toString*(j));

Al estar vinculada directamente la aplicación con el dispositivo Android lo único que necesita es tener acceso a internet para realizar las consultas y envíos.

Para activar la cerradura eléctrica se creó la siguiente función cuya trabajo es enviar el dato codigo=1, el cual es leído por la Tablet y envía a la placa la orden de poner un pulso en alto durante 500 ms.

```
bIngresarAdmin.setOnClickListener(new View.OnClickListener() {

  public void onClick(View view) {
  if(ePass.getText().toString().equals(clave)){
      try{

      httpGetData("http://www.alluc.com.ec/trabajo/registrarUsu
      ario.php?codigo=1");                              }
             }
     });
```

2.5.2.8.1. Conexión y monitoreo del servidor de televisión digital

Para esta segunda actividad como se denomina las pantallas de presentación en desarrollo de aplicaciones Android se visualizan los datos y parámetros de configuración y funcionamiento del servidor de televisión digital de la UNACH para lo cual se realizan consultar a dicho servidor mediante el uso de los ya mencionados archivos PHP, con la variación que para este caso se realizan consultas directamente al archivo de configuración de nuestro servidor VILLAGE FLOW, denominado **VF_CONF.XML**.

En la aplicación se creó una segunda actividad o pantalla en donde se realizan consultas y se muestran los datos obtenidos del servidor de televisión digital tal como se muestra en la siguiente figura 65.

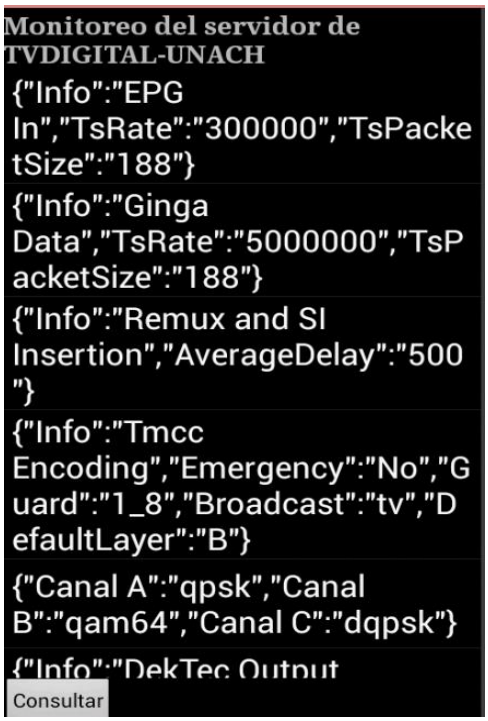


Figura 65. Pantalla de monitoreo del servidor de televisión

Fuente: autor

2.5.2.8.2. Comunicación de la aplicación con el servidor

VillageFlow es una plataforma cuya interfaz ha sido desarrollada en lenguaje de programación WEB (php) y lenguaje XML de marcadores para la comunicación y configuración de sus bloques de proceso dentro de un espacio. La programación XML se expresa de forma estructurada compuesta de elementos señalados mediante etiquetas. Un elemento puede tener contenido y atributos que añade propiedades a los elementos.

Al igual que en procesos anteriores se utilizara un archivo PhP para realizar las consultas de los parámetros de configuración del servidor de televisión digital.

A continuación se muestra el código en java que se utiliza para llamar al archivo servidor.php, el cual es el encargado de realizar las consultas del servidor de televisión digital.

```
bservidor.setOnClickListener(new View.OnClickListener() {

    public void onClick(View view) {
      JSONArray ja=null;
      try {
      String data;
      for(int j=0; j<1;j++){

      data=httpGetData("http://192.168.43.104/trabajo/servidor.php
?codigo="+Integer.toString(j));
      if(data.length()>1){
      for (int i = 0; (i = data.indexOf("[", i + 1)) != -1; ) {
            ja=new JSONArray(data.substring(i, data.length()));
            onAddItem(ja.getString(0));

            }}}
```

Los archivos XML son archivos de texto que utilizan un lenguaje de marcas para almacenar datos en forma legible por lo que hay que acceder al contenido de cada elemento y luego consultar sus atributos y los valores que estos atributos puedan tener esto se muestra en la siguiente figura 66.

```
- <Space AutoRecover="No" AutoStart="Yes" Name="Space1">
   + <Brick Info="TSFile In" BrickName="InBk1" InterfaceType="TsFile" IOType="In" Id="1">
   + <Brick Info="TSFile In" BrickName="InBk2" InterfaceType="TsFile" IOType="In" Id="2">
   - <Brick Info="EPG In" BrickName="InBk3" InterfaceType="Epg" IOType="In" Id="3">
       <Ts TsPacketSize="188" TsRate="300000"/>
     + <Parameters xsi:type="In_Epg">
     + <Connectors>
     </Brick>
```

Figura 66. Archivo de configuración del servidor de tv-digital
Fuente. Autor

Para consultar la información de este archivo se utiliza el siguiente código que se encuentra en el archivo servidor.php.

```
<?php
header( 'Content-Type: text/html;charset=utf-8' );
$info = simplexml_load_file('http://localhost/tesis/VF_CONF.xml');
$a= $info->Space->Brick[2] ['Info'];
$b= $info->Space->Brick[2]->Ts['TsRate'];
$c= $info->Space->Brick[2]->Ts['TsPacketSize']; ?>
```

Lo primero es declarar el tipo de formato de los datos y el tipo de comunicación en la instrucción (header). Luego se declara una variable que va a contener al archivo XML que va ser analizado mediante la instrucción (simplexml_load_file) y por ultimo almacenar en variables los valores de los atributos a consultar.

2.6 Comprobación de la hipótesis

Para la comprobación de la hipótesis se requiere realizar un Análisis de Varianza (ANOVA) de un factor. Se usa este método estadístico porque permite comparar una varianza de factores con una constante que para este caso es de 1 segundo como tiempo de activación mínimo, se comparará las varianzas del tiempo de respuesta de los siguientes factores:

- El sensor de temperatura y activación del equipo de enfriamiento de forma autónoma y remota.
- La activación de eventos, el registro en base de datos y dispositivos móviles.
- El activador de apertura de puerta de forma remota.
- Adquisición de señales del servidor de televisión digital.

Con lo cual se pretende demostrar que el trabajo de estos factores en tiempo real es factible para comprobar la hipótesis.

El ANOVA se basa en la descomposición de la variación total de los datos con respecto a la media global (SCT), que bajo el supuesto de que H0 es cierta es una estimación de σ^2 obtenida a partir de toda la información muestral, en dos partes:

- Variación dentro de las muestras (SCD) o Intra-grupos, cuantifica la dispersión de los valores de cada muestra con respecto a sus correspondientes medias.
- Variación entre muestras (SCE) o Inter-grupos, cuantifica la dispersión de las medias de las muestras con respecto a la media global.

Las expresiones para el cálculo de los elementos que intervienen en el ANOVA son las siguientes:

Media Global: $\overline{X} = \frac{\sum_{j=1}^{K} \sum_{i=1}^{nj} x_{ij}}{n}$.

Variación Total: $\mathrm{SCT} = \sum_{j=1}^{K} \sum_{i=1}^{nj} (x_{ij} - \overline{X})^2$

Variación Intra-grupos: $\mathrm{SCD} = \sum_{j=1}^{K} \sum_{i=1}^{nj} (x_{ij} - \overline{X_j})^2$

Variación Inter-grupos: $\mathrm{SCE} = \sum_{j=1}^{K} (\overline{X_j} - \overline{X})^2 n_j$

Siendo *xij* el i-ésimo valor de la muestra j-ésima; *nj* el tamaño de dicha muestra y $\overline{X_j}$ su media.

Cuando la hipótesis nula es cierta SCE/K-1 y SCD/n-K son dos estimadores insesgados de la varianza poblacional y el cociente entre ambos se distribuye según una F de Snedecor con K-1 grados de libertad en el numerador y N-K grados de libertad en el denominador. Por lo tanto, si H0 es cierta es de esperar que el cociente entre ambas estimaciones será aproximadamente igual a 1, de forma que se rechazará H0 si dicho cociente difiere significativamente de 1. Esto se comprueba en las tablas de análisis de varianza de los 4 aspectos a investigar.

2.6.1 Comprobación del tiempo de activación del sensor de temperatura y activación del equipo de enfriamiento.

a. Planteamiento de la hipótesis

- H_0: El tiempo promedio en que se demora el sensor de temperatura y la activación del equipo de enfriamiento es menor o igual al tiempo de activación mínimo. Ecuación (1).
- H_1: El tiempo promedio en que se demora el sensor de temperatura y la activación del equipo de enfriamiento es mayor al tiempo de activación mínimo. Ecuación (2).

$$H_0: \mu_{T1} \leq \mu_{T2} \leq \mu_{T3} \leq \mu_{T4} \leq \mu_{T5} \leq Tiempo\ de\ Activacion\ Minimo \quad (1)$$

$$H_1: \mu_{T1} \geq \mu_{T2} \geq \mu_{T3} \geq \mu_{T4} \geq \mu_{T5} \geq Tiempo\ de\ Activacion\ Minimo \quad (2)$$

b. Nivel de Significancia

La prueba se realizó con un 95% de confiabilidad, y con un nivel de significancia de $\alpha = 0{,}05$.

c. Estadístico de la Prueba.

Se usara la temperatura superior a 25º C que activa el equipo de enfriamiento y se buscaran como resultado el tiempo de activación mínimo que para el estudio se estableció un tiempo mínimo menor o igual a 1 segundo.

Para el estudio se usa el tiempo mínimo $\leq 1\ segundo$, el tiempo censado se considera de la activación del equipo de enfriamiento durante 7 días cada 2 horas iniciando a las 8 horas hasta las 16 horas; en intervalos de 2 horas dando 35 muestras totales, los valores se muestran recopilados se muestran en la tabla 11.

Tabla 11. Comprobación del tiempo de activación del sensor de temperatura y activación del equipo de enfriamiento.

Comprobación del tiempo de activación del sensor de temperatura y activación del equipo de enfriamiento.					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1 día	0,5 seg.	1,2 seg.	1 seg.	0,5 seg.	0,3 seg.
2 día	1 seg.	1 seg.	1 seg.	1 seg.	1 seg.
3 día	0,3 seg.	3 seg.	0,3 seg.	0,3 seg.	0,8 seg.
4 día	1 seg.	1 seg.	1 seg.	1 seg.	1 seg.
5 día	0,5 seg.	2 seg.	0,8 seg.	0,5 seg.	1,1 seg.
6 día	0,7 seg.	1 seg.	1,2 seg.	0,8 seg.	0,6 seg.
7 día	1 seg.	1 seg.	0,9 seg.	1,1 seg.	1 seg.

Fuente: Autor

La tabla 12. Muestra el resumen de las pruebas realizadas como: la cuenta, suma, promedio y varianza del tiempo de activación del sensor de temperatura y activación del equipo de enfriamiento. Teniendo en cuenta cada 2 horas; estos datos son obtenidos atreves del uso de ANOVA como herramienta para el proceso de los datos a evaluar.

Tabla 12. Resumen tiempo de activación sensor de temperatura y activación de equipo de enfriamiento

RESUMEN				
Grupos	*Cuenta*	*Suma*	*Promedio*	*Varianza*
8 horas	7,00	5,00	0,71	0,08
10 horas	7,00	3,60	0,51	0,15
12 horas	7,00	4,00	0,57	0,11
14 horas	7,00	4,70	0,67	0,09
16 horas	7,00	5,30	0,76	0,10

Fuente: Autor

Dentro de la tabla 13. El proceso de ANOVA se obtienen los datos de análisis de Varianza para comprobar el tiempo de activación sensor de temperatura y activación de equipo de enfriamiento y poder demostrar el valor crítico al cual se someten estos datos como se puede ver.

Tabla 13. Análisis de Varianza tiempo de activación sensor de temperatura y activación de equipo de enfriamiento

ANÁLISIS DE VARIANZA						
Origen de las variaciones	*Suma de cuadrados*	*Grados de libertad*	*Promedio de los cuadrados*	*F*	*Probabilidad*	*Valor crítico para F (Fc)*
Entre grupos	0,28	4,00	0,07	0,67	0,62	2,69
Dentro de los grupos	3,18	30,00	0,11			
Total	3,47	34,00				

Fuente: Autor

a. Regla de decisión.

Para aceptar la hipótesis (H_0) de la ecuación (3), el valor de F o valor estadístico de prueba debe ser menor que el valor crítico para F de la tabla 12, realizada en Excel:

$$F < F_c \, (3)$$

Donde

$$0{,}67 < 2{,}69$$

Demostrando que la hipótesis es factible ya que el valor estadístico de prueba es menor al valor crítico del estadístico de prueba con una probabilidad de H0 de éxito del 0,62. Con lo cual se demuestra que el análisis de varianza de un factor H0 es aproximadamente cerca al valor de 1.

2.6.2 Comprobación del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles.

a. Planteamiento de la hipótesis

H_0: El tiempo promedio en que se demora activación de eventos, el registro en base de datos y dispositivos móviles es menor o igual al tiempo de activación mínimo. Ecuación (5).

- H_1: El tiempo promedio en que se demora activación de eventos, el registro en base de datos y dispositivos móviles es mayor al tiempo de activación mínimo. Como se representa en la ecuación (6).

$$H_0: \mu_{T1} \leq \mu_{T2} \leq \mu_{T3} \leq \mu_{T4} \leq \mu_{T5} \leq Tiempo\ de\ Activacion\ Minimo \quad (5)$$

$$H_1: \mu_{T1} \geq \mu_{T2} \geq \mu_{T3} \geq \mu_{T4} \geq \mu_{T5} \geq Tiempo\ de\ Activacion\ Minimo \quad (6)$$

Nivel de Significancia

La prueba se realizó con un 95% de confiabilidad, y con un nivel de significancia de $\alpha = 0{,}05$.

b. Estadístico de la Prueba.

Para comprobar el funcionamiento e intercambio de información en formato de datos para el control y monitoreo del laboratorio de televisión digital de la UNACH en tiempo real.

Se usaran los tiempos para la activación de eventos, el registro en base de datos y dispositivos móviles, se buscaran como resultado el tiempo de activación mínimo que para el estudio se estableció un tiempo mínimo menor o igual a 1 segundo.

Para el estudio se usa el tiempo mínimo $\leq 1\ segundo$, el tiempo censado con el que se activó el equipo de enfriamiento durante 7 días cada 2 horas iniciando a las 8 horas hasta las 16 horas; en intervalos de 2 horas dando 35 muestras totales, los valores se muestran recopilados se muestran en la tabla 14.

Tabla 14. Comprobación del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles.

Comprobación del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles.					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1er día	3,5 seg.	2,7 seg.	2,5 seg.	4,5 seg.	2,9 seg.
2do día	1,8 seg.	1 seg.	4 seg.	2 seg.	4 seg.
3er día	3 seg.	5,3 seg.	3,3 seg.	4,3 seg.	2,3 seg.
4to día	4 seg.	2 seg.	1 seg.	5 seg.	5,1 seg.
5to día	5 seg.	4,5 seg.	2,5 seg.	2,5 seg.	4,5 seg.
6to día	7 seg.	3,7 seg.	4,7 seg.	4,2 seg.	2,7 seg.
7mo día	4,1 seg.	2 seg.	3 seg.	1,5 seg.	3,1 seg.

Fuente: Autor

La tabla 15. Muestra el resumen de las pruebas realizadas, tanto como la cuenta, suma, promedio y varianza de los tiempos de respuesta ante la comprobación del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles.; estos datos son obtenidos atreves del uso de ANOVA como herramienta para el proceso de los datos a evaluar.

Tabla 15. Resumen del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles.

RESUMEN				
Grupos	*Cuenta*	*Suma*	*Promedio*	*Varianza*
8 horas	7,00	4,80	0,69	0,10
10 horas	7,00	5,10	0,73	0,10
12 horas	7,00	4,10	0,59	0,08
14 horas	7,00	4,70	0,67	0,09
16 horas	7,00	4,40	0,63	0,06

Fuente: Autor

En la tabla 16. Dentro del proceso de ANOVA se obtienen los datos para el análisis de varianza del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles. Se puede demostrar el valor crítico al cual se someten estos datos como se puede ver:

Tabla 16. Análisis de varianza del tiempo de activación de eventos, el registro en base de datos y dispositivos móviles.

ANÁLISIS DE VARIANZA						
Origen de las variaciones	*Suma de cuadrados*	*Grados de libertad*	*Promedio de los cuadrados*	*F*	*Probabilidad*	*Valor crítico para F*
Entre grupos	0,08	4,00	0,02	0,24	0,91	2,69
Dentro de los grupos	2,58	30,00	0,09			
Total	2,66	34,00				

Fuente: Autor

c. Regla de decisión.

Para aceptar la hipótesis nula (H_0) de la ecuación (7), el valor de F o valor estadístico de prueba, debe ser menor que el valor crítico para F de la tabla 12, realizada en Excel:

$$F < F_c \quad (7)$$

Donde

$$0{,}24 < 2{,}69$$

Demostrando que la hipótesis nula es factible ya que el valor estadístico de prueba es menor al valor crítico del estadístico de prueba con una probabilidad de H0 de éxito del 0,91. Con lo cual se demuestra que el análisis de varianza de un factor H0 es aproximadamente cerca al valor de 1.

2.6.3 Comprobación del tiempo de activación de apertura de la puerta de forma remota.

a. Planteamiento de la hipótesis

H_0: El tiempo promedio en que se demora activación de apertura de la puerta de forma remota es menor o igual al tiempo de activación mínimo. Ecuación (9).

H_1: El tiempo promedio en que se demora activación de apertura de la puerta de forma remota es mayor al tiempo de activación mínimo. Como se representa en la ecuación (10).

$$H_0: \mu_{T1} \leq \mu_{T2} \leq \mu_{T3} \leq \mu_{T4} \leq \mu_{T5} \leq Tiempo\ de\ Activacion\ Minimo \quad (9)$$

$$H_1: \mu_{T1} \geq \mu_{T2} \geq \mu_{T3} \geq \mu_{T4} \geq \mu_{T5} \geq Tiempo\ de\ Activacion\ Minimo \quad (10)$$

b. Nivel de Significancia.

La prueba se realizó con un 95% de confiabilidad, y con un nivel de significancia de $\alpha = 0{,}05$.

c. Estadístico de la Prueba.

Para comprobar el funcionamiento e intercambio de información en formato de datos para el control y monitoreo del laboratorio de televisión digital de la UNACH en tiempo real.

Para el estudio se usa el tiempo mínimo $\leq 1\ segundo$, las pruebas denotan el tiempo transcurrido para la activación de apertura de la puerta de forma remota con: 7 días cada 2 horas iniciando a las 8 horas hasta las 16 horas; en intervalos de 2 horas dando 35 muestras totales, los valores se muestran recopilados se muestran en la tabla 17.

Tabla 17. Comprobación del tiempo de activación de apertura de la puerta de forma remota.

Comprobación del tiempo de activación de apertura de la puerta de forma remota.					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1er día	1,2 seg.	1 seg.	2,2 seg.	1,9 seg.	1,6 seg.
2do día	2,5 seg.	1,8 seg.	3,3 seg.	2,4 seg.	2,3 seg.
3er día	2,3 seg.	0,7 seg.	1,3 seg.	1 seg.	2,8 seg.
4to día	1,8 seg.	1,2 seg.	2,5 seg.	2,3 seg.	2,5 seg.
5to día	2,8 seg.	2,7 seg.	1,8 seg.	2,3 seg.	1,8 seg.
6to día	0,7 seg.	3,3 seg.	1,5 seg.	1 seg.	3,2 seg.
7mo día	1 seg.	1,2 seg.	2,3 seg.	1,5 seg.	2,5 seg.

Fuente: Autor

La tabla 18. Muestra el resumen de las pruebas realizadas, tanto como la cuenta, suma, promedio y varianza de los tiempos de respuesta de la activación de apertura de la puerta de forma remota; estos datos son obtenidos atreves del uso de ANOVA como herramienta para el proceso de los datos a evaluar.

Tabla 18. Resumen de la activación de apertura de la puerta de forma remota.

RESUMEN				
Grupos	*Cuenta*	*Suma*	*Promedio*	*Varianza*
8 horas	7,00	9,70	1,39	0,85
10 horas	7,00	8,90	1,27	0,35
12 horas	7,00	8,90	1,27	0,62
14 horas	7,00	10,40	1,49	0,59
16 horas	7,00	9,40	1,34	0,72

Fuente: Autor

En la tabla 19. Se obtienen los datos para el análisis de varianza de la activación de apertura de la puerta de forma remota. Se puede demostrar el valor crítico al cual se someten estos datos como se puede ver:

Tabla 19. Análisis de varianza de la activación de la puerta de forma remota.

ANÁLISIS DE VARIANZA						
Origen de las variaciones	*Suma de cuadrados*	*Grados de libertad*	*Promedio de los cuadrados*	*F*	*Probabilidad*	*Valor crítico para F*
Entre grupos	0,22	4,00	0,06	0,09	0,98	2,69
Dentro de los grupos	18,76	30,00	0,63			
Total	18,99	34,00				

Fuente: Autor

d. Regla de decisión.

Para aceptar la hipótesis nula (H_0) de la ecuación (11), el valor de F debe ser menor que el valor crítico para F de la tabla 12, realizada en Excel:

$$F < F_c \quad (11)$$

Donde

$$0{,}09 < 2{,}69$$

Demostrando que la hipótesis nula es factible ya que el valor estadístico de prueba es menor al valor crítico del estadístico de prueba con una probabilidad de H0 éxito del 0,98. Con lo cual se demuestra que el análisis de varianza de un factor H0 es aproximadamente cerca al valor de 1.

2.6.4 Comprobación del tiempo de adquisición de señales del servidor de televisión digital.

a. Planteamiento de la hipótesis

H_0: El tiempo promedio en que se demora el de adquisición de señales del servidor de televisión digital es menor o igual al tiempo de activación mínimo. Ecuación (13).

H_1: El tiempo promedio en que se demora el de adquisición de señales del servidor de televisión digital es mayor al tiempo de activación mínimo. Como se representa en la ecuación (14).

$$H_0: \mu_{T1} \leq \mu_{T2} \leq \mu_{T3} \leq \mu_{T4} \leq \mu_{T5} \leq Tiempo\ de\ Activacion\ Minimo \quad (13)$$

$$H_1: \mu_{T1} \geq \mu_{T2} \geq \mu_{T3} \geq \mu_{T4} \geq \mu_{T5} \geq Tiempo\ de\ Activacion\ Minimo \quad (14)$$

b. Nivel de Significancia

La prueba se realizó con un 95% de confiabilidad, y con un nivel de significancia de $\alpha = 0{,}05$.

c. Estadístico de la Prueba.

Para comprobar el funcionamiento e intercambio de información en formato de datos para el control y monitoreo del laboratorio de televisión digital de la UNACH en tiempo real.

En la tabla 20. Se usan los tiempos de la prueba de la adquisición de señales del servidor de televisión digital, desde que se genera la petición del usuario como consulta de los datos del servidor de televisión digital y la respectiva respuesta dentro de la aplicación destinada al usuario administrador; por lo cual se demuestran las siguientes pruebas y datos obtenidos cada 2 horas durante 7 días.

Tabla 20. Comprobación del tiempo de adquisición de señales del servidor de televisión digital.

Comprobación del tiempo de adquisición de señales del servidor de televisión digital.					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1er día	1,2 seg.	1,5 seg.	2,1 seg.	1,4 seg.	1 seg.
2do día	1,4 seg.	1,2 seg.	1,8 seg.	2,4 seg.	1,3 seg.
3er día	1,5 seg.	1,7 seg.	1,9 seg.	1,2 seg.	1,8 seg.
4to día	1,6 seg.	1,5 seg.	1,7 seg.	1,3 seg.	1,7 seg.
5to día	2,1 seg.	1,1 seg.	1,4 seg.	1,9 seg.	1,5 seg.
6to día	1,8 seg.	1,6 seg.	1,8 seg.	1 seg.	1,5 seg.
7mo día	1 seg.	0,9 seg.	2,3 seg.	1,5 seg.	1,8 seg.

Fuente: Autor

La tabla 21. Muestra el resumen de las pruebas realizadas, tanto como la cuenta, suma, promedio y varianza de los tiempos de respuesta adquisición de señales del servidor de televisión digital; estos datos son obtenidos atreves del uso de ANOVA como herramienta para el proceso de los datos a evaluar.

Tabla 21. Resumen adquisición de señales del servidor de televisión digital.

RESUMEN				
Grupos	*Cuenta*	*Suma*	*Promedio*	*Varianza*
8 horas	7,00	3,66	0,52	0,10
10 horas	7,00	3,50	0,50	0,08
12 horas	7,00	5,70	0,81	0,06
14 horas	7,00	5,50	0,79	0,06
16 horas	7,00	4,60	0,66	0,06

Fuente: Autor

En la tabla 22. El proceso de ANOVA se obtienen los datos de análisis de Varianza para comprobar el tiempo de activación sensor de temperatura y activación de equipo de enfriamiento y poder demostrar el valor crítico al cual se someten estos datos como se puede ver.

Tabla 22. Análisis de varianza adquisición de señales del servidor de televisión digital.

ANÁLISIS DE VARIANZA						
Origen de las variaciones	*Suma de cuadrados*	*Grados de libertad*	*Promedio de los cuadrados*	*F*	*Probabilidad*	*Valor crítico para F*
Entre grupos	0,59	4,00	0,15	2,02	0,58	2,69
Dentro de los grupos	2,18	30,00	0,07			
Total	2,77	34,00				

Fuente: Autor

d. Regla de decisión.

Para aceptar la hipótesis nula (H_0) de la ecuación (15), el valor de F debe ser menor que el valor crítico para F de la tabla 12, realizada en Excel:

$$F < F_c \quad (15)$$

Donde

$$2{,}02 < 2{,}69$$

Demostrando que la hipótesis nula es factible ya que el valor estadístico de prueba es menor al valor crítico del estadístico de prueba con una probabilidad de H0 de éxito del 0,58. Con lo cual se demuestra que el análisis de varianza de un factor H0 es aproximadamente cerca al valor de 1.

CAPÍTULO III

3. RESULTADOS

Con el desarrollo e implementación del sistema de telemetría y telecontrol para el laboratorio de televisión digital de la UNACH se logró construir un sistema autónomo y robusto que permite monitorear y controlar parámetros físicos y ambientales del laboratorio, además monitorear el servidor de televisión digital permitiendo consultar el estado y funcionamiento del mismo.

La implementación de este prototipo entrega los siguientes datos en la toma de muestras de tiempo de respuesta de los parámetros:

- El sensor de temperatura y activación del equipo de enfriamiento de forma autónoma y remota.
- La activación de eventos, el registro en base de datos y dispositivos móviles.
- El activador de apertura de puerta de forma remota.
- Adquisición de señales del servidor de televisión digital.

3.1. El sensor de temperatura y activación del equipo de enfriamiento de forma autónoma y remota.

Los resultados que se dieron al tomar los tiempos de respuesta para la activación del equipo de enfriamiento al tener una subida de temperatura mayor a la establecida de 25º C.

En la codificación de la placa se establece un tiempo de activación que es ampliamente superado por el tiempo que se mide en la configuración eléctrica de la placa y el rele.

Las pruebas brindan los siguientes datos de tiempo que se expresa en la tabla 23.

Tabla 23 Muestras del tiempo de activación del equipo de enfriamiento.

Muestras del tiempo de activación del equipo de enfriamiento.					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1 día	0,5 seg.	1,2 seg.	1 seg.	0,5 seg.	0,3 seg.
2 día	1 seg.	1 seg.	1 seg.	1 seg.	1 seg.
3 día	0,3 seg.	3 seg.	0,3 seg.	0,3 seg.	0,8 seg.
4 día	1 seg.	1 seg.	1 seg.	1 seg.	1 seg.
5 día	0,5 seg.	2 seg.	0,8 seg.	0,5 seg.	1,1 seg.
6 día	0,7 seg.	1 seg.	1,2 seg.	0,8 seg.	0,6 seg.
7 día	1 seg.	1 seg.	0,9 seg.	1,1 seg.	1 seg.

Fuente: Autor

3.2. La activación de eventos, el registro en base de datos y dispositivos móviles.

Los resultados que se dieron al tomar los tiempos de respuesta para la llegada de notificaciones de los eventos producidos dentro del laboratorio de televisión digital se muestran en la tabla 24.

Tabla 24. Muestras del tiempo de respuesta para la llegada de notificaciones de los eventos

Muestras del tiempo de respuesta para la llegada de notificaciones de los eventos					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1 día	3,5 seg.	2,7 seg.	2,5 seg.	4,5 seg.	2,9 seg.
2 día	1,8 seg.	1 seg.	4 seg.	2 seg.	4 seg.
3 día	3 seg.	5,3 seg.	3,3 seg.	4,3 seg.	2,3 seg.
4 día	4 seg.	2 seg.	1 seg.	5 seg.	5,1 seg.
5 día	5 seg.	4,5 seg.	2,5 seg.	2,5 seg.	4,5 seg.
6 día	7 seg.	3,7 seg.	4,7 seg.	4,2 seg.	2,7 seg.
7 día	4,1 seg.	2 seg.	3 seg.	1,5 seg.	3,1 seg.

Fuente: Autor

3.3. El activador de apertura de puerta de forma remota.

Los resultados que se dieron al tomar los tiempos de respuesta para la activación de la chapa eléctrica se muestran a continuación:

Tabla 25 Muestras del tiempo de activación de la chapa eléctrica

Muestras del tiempo de activación de la chapa eléctrica					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1 día	1,2 seg.	1 seg.	2,2 seg.	1,9 seg.	1,6 seg.
2 día	2,5 seg.	1,8 seg.	3,3 seg.	2,4 seg.	2,3 seg.
3 día	2,3 seg.	0,7 seg.	1,3 seg.	1 seg.	2,8 seg.
4 día	1,8 seg.	1,2 seg.	2,5 seg.	2,3 seg.	2,5 seg.
5 día	2,8 seg.	2,7 seg.	1,8 seg.	2,3 seg.	1,8 seg.
6 día	0,7 seg.	3,3 seg.	1,5 seg.	1 seg.	3,2 seg.
7 día	1 seg.	1,2 seg.	2,3 seg.	1,5 seg.	2,5 seg.

Fuente: Autor

3.4. Adquisición de señales del servidor de televisión digital.

Los resultados que se dieron al tomar los tiempos de respuesta para las consultas al servidor de televisión digital, se demuestran en la siguiente tabla 26.

Tabla 26. Muestras del tiempo de respuesta para las consultas al servidor de televisión digital.

Muestras del tiempo de respuesta para las consultas al servidor de televisión digital.					
Hora Vs Días	8 horas	10 horas	12 horas	14 horas	16 horas
1 día	1,2 seg.	1,5 seg.	2,1 seg.	1,4 seg.	1 seg.
2 día	1,4 seg.	1,2 seg.	1,8 seg.	2,4 seg.	1,3 seg.
3 día	1,5 seg.	1,7 seg.	1,9 seg.	1,2 seg.	1,8 seg.
4 día	1,6 seg.	1,5 seg.	1,7 seg.	1,3 seg.	1,7 seg.
5 día	2,1 seg.	1,1 seg.	1,4 seg.	1,9 seg.	1,5 seg.
6 día	1,8 seg.	1,6 seg.	1,8 seg.	1 seg.	1,5 seg.
7 día	1 seg.	0,9 seg.	2,3 seg.	1,5 seg.	1,8 seg.

Fuente: Autor

El monitoreo del servidor de televisión digital se lo realiza mediante consultas a los archivos de configuración del programa VILLAGEFLOW, a partir de dicho archivo se puede consultar y monitorear todos los parámetros de funcionamiento de nuestro servidor de televisión digital los cuales se pueden visualizar de forma remota en la aplicación Android para Smartphone creada por los tesistas para el desarrollo de este proyecto.

Los parámetros del servidor de televisión digital más importantes a monitorear y tener en cuenta para un correcto funcionamiento del mismo se muestran en la siguiente tabla 27:

Tabla 27. Parámetros del servidor de televisión a ser monitoreados.

BLOQUES	**PARÁMETROS A MONITOREAR**		
EPG In	TsRate	TsPacketSize	
Ginga Data	TsRate	TsPacketSize	
Remux and SI Insertion	AverageDelay		
Tmcc Encoding	Broadcast	Emergency	
	DefaultLayer	Guard	
MODULACIÓN DE CANALES	Canal A	Canal B	Canal C
DekTec Output Card	TsRate	TsPacketSize	
	Rf-Level	Rf-Frequency	

Fuente: Autor

La aplicación para Smartphone accede a las bases de datos para consultar los registros y visualizarlos, además de monitorear en tiempo real al servidor de televisión digital ingresando al archivo de configuración del programa VillageFlow, esta comunicación es inalámbrica utilizando principalmente como canal de comunicación al internet móvil de las redes de telefonía celular, además se utilizan los servicios de mensajes cortos como servicios complementarios para generar un sistema de telemetría robusto y eficiente.

CAPÍTULO IV

4. DISCUSIÓN

Para el desarrollo del sistema hubo que tener en cuenta los tres elementos que intervienen en todo sistema de comunicaciones; el emisor, el canal y el receptor. Como emisor y receptor se tiene a los dispositivos móviles tales como la Tablet y un teléfono inteligente y para resolver el problema de la comunicación entre máquinas remotas, se optó por utilizar como canal de comunicación el internet móvil de la red de telefonía celular dotando al sistema de total independencia para su funcionamiento e intercambio de datos.

Para la comprobación hipótesis se usan los valores del factor de varianza y del factor crítico de varianza, para cumplir la regla el primero debe ser menor que el segundo.

$$F < F_c$$

Se resume las varianzas, indicadores de todos los aspectos analizados en la siguiente tabla 28.

Tabla 28. Muestras del tiempo de respuesta para las consultas al servidor de televisión digital.

Varianza	F	Fc	Resultado Ho
Tiempo de activación sensor de temperatura y activación de equipo de enfriamiento	0,67	2,69	Aprueba
Tiempo de activación de eventos, el registro en base de datos y dispositivos móviles	0,24	2,69	Aprueba
Activación de apertura de la puerta de forma remota	0,09	2,69	Aprueba
Adquisición de señales del servidor de televisión digital.	2,02	2,69	Aprueba

Fuente: Autor

Otro punto importante con el que debe cumplir el sistema es que debe trabajar en tiempo real para lo cual se toma en consideración que el sistema debe tener un tiempo de respuesta igual o menor a 5 segundo ($t < ó \approx 5$ seg).

De las pruebas realizadas a los factores que intervienen en la tesis se constata que los peores tiempos posibles redondeados, asumiendo las peores respuestas del sistema para que pueda trabajar en tiempo real se obtiene la siguiente tabla 29.

Tabla 29. Muestras cumplen trabajo en tiempo real

Factor	Indicador	Tiempo respuesta	Resultado
Temperatura	Supera los 25º C	1 seg	Aprueba
Apertura de puerta	Apertura remota	3 seg	Aprueba
Llegada de notificaciones	Notificación intruso	5 seg	Aprueba
Consulta al servidor de TV digital	Consulta TsRate	2 seg	Aprueba

Fuente: Autor

El intercambio de datos se lo realiza mediante un servidor web el cual contiene las bases de datos que almacenan la información de los eventos que ocurren en el laboratorio; además que también alberga el archivo de configuración del sistema VillgeFlow del servidor de televisión digital de la UNACH, del cual se obtienen la información de los parámetros del mismo; los cuales son monitoreados para conocer el estado y funcionamiento de dicho servidor de televisión.

Para crear un sistema confiable y robusto se optó por mantener comunicado a los dispositivos encargados de generar los datos y controlar los sensores y actuadores, en nuestro caso la tableta a la nube de internet de dos maneras utilizando la red de telefonía como red principal y la red de Wi-Fi de la UNACH como red secundaria, de esta manera se asegura que el sistema esté siempre conectado.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES.

5.1 Conclusiones

- Se diseñó e implementó un sistema de telemetría y telemando para monitorear y controlar el estado del laboratorio de televisión digital de la Universidad Nacional de Chimborazo utilizando principios de la tecnología M2M.

- Mediante la utilización del internet móvil se logró mantener la comunicación entre los diferentes bloques del sistema de telemetría en tiempo real.

- Gracias a la utilización de un servidor web se pudo administrar la información generada por el sistema de telemetría al cargar esta información en bases de datos dentro del servidor web, además permite monitorear el funcionamiento del servidor TDT siendo capaz de acceder a los archivos de configuración propios de dicho servidor.

- La creación de una interfaz hombre maquina (HMI) permite interactuar al usuario con el sistema para ingreso al laboratorio mediante identificación de huella, también permite enrolar a nuevos usuarios; el HMI además controla los sensores y actuadores del sistema de forma precisa.

- El sistema permite censar y controlar las variables ambientales del laboratorio tales como temperatura, presencia de humo, generando alarmas que adviertan cualquier peligro existente en el laboratorio de

forma que se puedan tomar las medidas necesarias para preservar la integridad del mismo.

- Como otro aporte se desarrolló aplicaciones en código abierto java bajo el sistema operativo Android que cumplan con las funciones de HMI en el caso de la Tablet; y como interfaz de monitoreo y control del sistema mediante un teléfono inteligente.

5.2 Recomendaciones.

- Continuar con la investigación de sistemas de telemetría y control de forma que se puedan agregar nuevas funciones a este prototipo y poder crear un sistema más complejo y de mayores alcances.

- Incrementar la funciones de las aplicaciones Android para hacerlas más versátiles de forma que puedan ser adaptativas y auto configurables sin la necesidad de cambiar el código fuente de programación.

- Se recomienda vaciar el contenido de las bases de datos cada mes para que el sistema de monitoreo no se haga lento con el tiempo ya que este realiza consultas de cada una de las filas existentes en dichas bases hasta encontrar el dato o coincidencia deseada por lo cual es recomendable borrar datos demasiado antiguos.

CAPÍTULO VI

6. PROPUESTA

6.1 Título de la Propuesta.

Implementación de un sistema de telemetría y telecontrol para el laboratorio de televisión digital de la UNACH.

6.2 Introducción

El prototipo del sistema debe ser capaz de monitorear y controlar el laboratorio de televisión digital. Monitorear parámetros de funcionamiento del servidor de televisión digital y controlar ciertos parámetros ambientales del laboratorio, para lo cual requiere que el sistema se mantenga siempre conectado en todo momento.

En el prototipo del sistema de telemetría ofrece varias funciones como las de poder controlar ciertas funciones de forma remota desde una aplicación. Por lo que surge la necesidad de la implementación de un sistema que regule y controle las variables ambientales como la temperatura del laboratorio para preservar la integridad y buen funcionamiento del servidor de televisión.

6.3 Objetivos

6.3.1 General

- Implementar un sistema de telemetría y telecontrol en el laboratorio de televisión digital de la UNACH

6.3.2 Específicos:

- Instalar sensores y actuadores que monitoreen y controlen el funcionamiento del laboratorio.
- Desarrolla un sistema de telemetría que permita conocer en tiempo real el estado y funcionamiento del laboratorio y del servidor de televisión.
- Realizar pruebas de funcionamiento del sistema.

6.4 Fundamentación Científica – Técnica.

Para el diseño y posteriormente la implementación del sistema de telemetría y control hay que tener en cuenta los elementos que intervendrán en el desarrollo del mismo, los cuales deberán ser detallados específicamente como se describe a continuación:

Controlador

El controlador se seleccionara de acuerdo a las variables que haga que sensar y a los elementos que se deban controlar, además de tener en cuenta que la comunicación entre los sensores y actuadores en conjunto con la interfaz HMI deben ser en el menor tiempo posible y sin errores.

Sensor

El prototipo de sistema constara de varios sensores que ayudaran en la recolección de información del estado del laboratorio de manera eficaz y rápida, además que deben ser correctamente distribuidos en el espacio físico del laboratorio.

Actuador.

Los actuadores deben accionarse de forma rápida y oportuna ya sea por orden manual del administrador o de forma automática según lo programado en el controlador.

Transmisor.

En el sistema debe transmitir los datos obtenidos en tiempo real es decir en un tiempo igual o menor a un segundo.

Receptor.

El sistema al ser de telemetría debe obtener las consultas de datos sin perder ningún tipo de información y en el menor tiempo posible y presentar los datos de una forma clara y legible para el ser humano.

6.5. Descripción de la Propuesta

El objetivo es desarrollar e implementar un sistema de telemetría y telecontrol para el laboratorio de televisión digital de la Universidad Nacional de Chimborazo, para cumplir este objetivo principal se debe seguir el siguiente orden establecido por las siguientes etapas:

1. Selección e instalación de sensores y actuadores.
2. Programación de la placa STM32f4
3. Conexión de la placa con la Tablet
4. Programación de la Tablet
5. Monitoreo del servidor de televisión digital
6. Conexión de la Tablet con el servidor
7. Almacenamiento y lectura de datos
8. Creación de la aplicación de monitoreo para Smartphone.

El proyecto busca cumplir con los siguientes objetivos:

- El diseñar e implementar un sistema de telemetría utilizando tecnología M2M para monitoreo y control del laboratorio de televisión digital de la UNACH usando como canal de comunicación la red de telefonía celular.
- Diseñar un sistema que permita sensar y controlar las variables ambientales del laboratorio como temperatura, presencia de humo.
- Permitir controlar el acceso al laboratorio de forma remota, desde un dispositivo móvil o mediante reconocimiento de huella con lector biométrico.

- Monitorear y enviar en tiempo real la información del servidor de televisión digital para monitorear parámetros de transmisión del servidor de televisión como potencia de transmisión, latencia, Ts-rate, frecuencia.

6.6. Diseño Organizacional

El diseño organizacional de las personas que se involucraran y ejecutaran la propuesta se describe a continuación en la siguiente figura 67.

Figura 67. Diseño Organizacional

Fuente: Autor

Como director del proyecto el Ing. Diego Barahona director del ICITS, como tutor de tesis el Ing. Daniel Santillán y tesistas los alumnos Galo Santillán y Daniel Haro.

6.7 Monitoreo y Evaluación de la propuesta.

El monitoreo y evaluación del sistema de telemetría y telecontrol para el laboratorio de televisión digital de la UNACH se lo realizará en las pruebas de funcionamiento; para lo cual se realizaran mediciones diarias durante el periodo de una semana.

CAPÍTULO VII

7. BIBLIOGRAFÍA

[1] **ALFREDO**. *academia*.,
http://www.academia.edu/6180549/Investigacion_cientifica
19-03-2014

[2] **BOSWARTHICK, D**., Elloumi, O., & Hersent, O. (2012). *Machine-to-machine communications.* Francia: A John Wiley & Sons, Ltd., Publication.

[3] **CACERES, E**. *redesbiomedica*.
http://redesbiomedica.blogspot.com/2014_03_01_archive.html
03-04-2014

[4] **DOUPOVEC, M.** *metodologia02.*
http://metodologia02.blogspot.com/p/tecnicas-de-la-investigacion.html
24-03-2014,

[5] **ESPINOZA, G.** *expreso*.
http://expreso.ec/expreso/plantillas/nota.aspx?idart=5156916&idcat=23668&tipo=2
02-04-2014

[6] **LABARCA, A.** (6 de 09 de 2013). *smartcitieschile*.
http://www.smartcitieschile.cl/m2m-telefonica/
22-04-2014

[7] **PC-MOBIL**.*pcmobilt*.

http://pcmobilt.blogspot.com/2011/03/la-telemetria-en-la-transmision-de.html
12-04-2014

[8] **RACHELLGR.***buenastareas.*,
http://www.buenastareas.com/ensayos/Filosofia/6534546.html
19-03-2014

[9] **WIKIPEDIA.**
http://es.wikipedia.org/wiki/SMS
12-04-2014

[10] **ALONSO, J**. Tecnologías de la Información y de la Comunicación, 8a Ed., Barcelona-España., Ra-Ma, 2010.,pp 20-30

[11] **DORF, R. C. y BISHOP, R. H.** Sistemas de control moderno. , Traducido del ingles por Sebastian Dormido Canto y Raquel Dormido Canto. ,10a ed., Madrid-España, Pearson Educación., 2005., pp 1-24.

[12] **KALPAKJIAN, S. y SCHMID S. R**. Manufactura, ingeniería y tecnología. , traducido por Jaime Espinosa Limón., 5a ed., México-México, Pearson Educación., 2008., pp 1085-1236.

CAPÍTULO VIII

8. ANEXOS

Anexo 1

Manual del sensor LM35

National Semiconductor

November 2000

LM35
Precision Centigrade Temperature Sensors

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only ±¼°C typical
- Low impedance output, 0.1 Ω for 1 mA load

Typical Applications

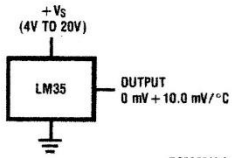
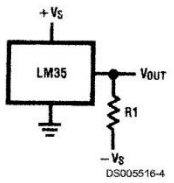


FIGURE 1. Basic Centigrade Temperature Sensor (+2°C to +150°C)



Choose $R_1 = -V_S/50\ \mu A$
V_{OUT} = +1,500 mV at +150°C
= +250 mV at +25°C
= −550 mV at −55°C

FIGURE 2. Full-Range Centigrade Temperature Sensor

© 2000 National Semiconductor Corporation

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	+35V to −0.2V
Output Voltage	+6V to −1.0V
Output Current	10 mA
Storage Temp.;	
TO-46 Package,	−60°C to +180°C
TO-92 Package,	−60°C to +150°C
SO-8 Package,	−65°C to +150°C
TO-220 Package,	−65°C to +150°C
Lead Temp.:	
TO-46 Package, (Soldering, 10 seconds)	300°C
TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: T_{MIN} to T_{MAX} (Note 2)	
LM35, LM35A	−55°C to +150°C
LM35C, LM35CA	−40°C to +110°C
LM35D	0°C to +100°C

Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy	T_A=+25°C	±0.2	±0.5		±0.2	±0.5		°C
(Note 7)	T_A=−10°C	±0.3			±0.3		±1.0	°C
	T_A=T_{MAX}	±0.4	±1.0		±0.4	±1.0		°C
	T_A=T_{MIN}	±0.4	±1.0		±0.4		±1.5	°C
Nonlinearity (Note 8)	$T_{MIN} \le T_A \le T_{MAX}$	**±0.18**		**±0.35**	**±0.15**		**±0.3**	°C
Sensor Gain (Average Slope)	$T_{MIN} \le T_A \le T_{MAX}$	**+10.0**	**+9.9, +10.1**		**+10.0**		**+9.9, +10.1**	mV/°C
Load Regulation	T_A=+25°C	±0.4	±1.0		±0.4	±1.0		mV/mA
(Note 3) $0 \le I_L \le 1$ mA	$T_{MIN} \le T_A \le T_{MAX}$	**±0.5**		**±3.0**	**±0.5**		**±3.0**	mV/mA
Line Regulation	T_A=+25°C	±0.01	±0.05		±0.01	±0.05		mV/V
(Note 3)	$4V \le V_S \le 30V$	**±0.02**		**±0.1**	**±0.02**		**±0.1**	mV/V
Quiescent Current	V_S=+5V, +25°C	56	67		56	67		µA
(Note 9)	V_S=+5V	**105**		**131**	**91**		**114**	µA
	V_S=+30V, +25°C	56.2	68		56.2	68		µA
	V_S=+30V	**105.5**		**133**	**91.5**		**116**	µA
Change of Quiescent Current (Note 3)	$4V \le V_S \le 30V$, +25°C	0.2	1.0		0.2	1.0		µA
	$4V \le V_S \le 30V$	**0.5**		**2.0**	**0.5**		**2.0**	µA
Temperature Coefficient of Quiescent Current		**+0.39**		**+0.5**	**+0.39**		**+0.5**	µA/°C
Minimum Temperature for Rated Accuracy	In circuit of *Figure 1*, I_L=0	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	T_J=T_{MAX}, for 1000 hours	±0.08			±0.08			°C

Anexo 2

CÓDIGO DE PROGAMACIÓN TARJETA STM32F407 EN CoIDE

```
#include "main.h"
#include "stm32_ub_uart.h"
#include "stm32_ub_led.h"
#include "stm32_av_zfm20x.h"
#include "stm32_ub_adc1_single.h"
#include "stm32_ub_tim2.h"
#include "stm32_ub_dig_in.h"
int ident = 0;
uint16_t temp;
uint16_t SPtemp=25;
int alarma1=0;
int alarma2=0;
int aux_a1=0;
int aux_a2=0;

int main(void) {
	UART_RXSTATUS_t check;
	SystemInit();
	UB_Uart_Init();
	UB_Led_Init();
	AV_Zfm20x_Delay(0xFFFFFF);
	AV_Zfm20x_Init();
	UB_ADC1_SINGLE_Init(); // Init vom ADC1
	UB_DigIn_Init();
	UB_TIMER2_Init(8399,9999);
	char buff[50];
	int bus = 0;
	int var = 0;
	uint16_t dato=0;
	int newDir=0;
	UB_TIMER2_Start();

	while (1) {
		check = UB_Uart_ReceiveString(COM3, buff);
		if (check == RX_READY) {
		ident = buff[0];
		for (var = 0; var < 50; ++var) {
			buff[var]=buff[var+1];
			}
		dato=UB_String_DezStringToInt(buff);

		if (ident == 'E') {
			AV_Zfm20x_Delay(0x2FFFFF);
			while (1) {
			check = UB_Uart_ReceiveString(COM3, buff);
				if (check == RX_READY) {
				ident = buff[0];
				for (var = 0; var < 50; ++var) {
					buff[var]=buff[var+1];
				}
				dato = AV_Zfm20x_Enrolar(newDir);
				while (dato == 0){
					newDir++;
					dato = AV_Zfm20x_Enrolar(newDir); }
```

```
        if (dato==1) {
              AV_Zfm20x_Delay(0xFFFFF);
              sprintf(buff, "e%d", newDir);
              UB_Uart_SendString(COM3, buff, LFCR);
        }else{
              AV_Zfm20x_Delay(0xFFFFF);
              UB_Uart_SendString(COM3, "e0", LFCR);
        }
        break;
        }
        }
}


if (ident == 'B') {
        bus = AV_Zfm20x_Buscar();
        if (bus == -1) {
              UB_Uart_SendString(COM3, "b0", LFCR);
        } else if (bus == -2) {
              UB_Uart_SendString(COM3, "bt", LFCR);
        } else {
              sprintf(buff, "b%d", bus);
              UB_Uart_SendString(COM3, buff, LFCR);
        }
}
if (ident == 'U') {
        UB_Uart_SendString(COM3, "Dirección", LFCR);
        while (1) {
        check = UB_Uart_ReceiveString(COM3, buff);
              if (check == RX_READY) {
                    ident = buff[0];
                    for (var = 0; var < 50; ++var) {
                          buff[var]=buff[var+1];
                    }


                    dato=UB_String_DezStringToInt(buff);
                    AV_Zfm20x_BorrarD(dato);
                    break;
                    }
              }
        }

        if (ident == 'T') {
              if(dato==111){
                    AV_Zfm20x_BorrarT();
              }else{
                    AV_Zfm20x_BorrarD(dato);
              }
        }
        if (ident == 'A') {
              UB_Led_On(LED_GREEN);
              AV_Zfm20x_Delay(0x2FFFFFF);
              UB_Led_Off(LED_GREEN);
              ident == 0;
        }
        if (ident == 'S') {
              SPtemp=dato;
              ident == 0;
```

```
                }
            }


            if (SPtemp>=temp) {
                UB_Led_Off(LED_RED);
            }else {
                UB_Led_On(LED_RED);
            }

            if(UB_DigIn_Read(DIN_PB11)==Bit_RESET){
                UB_Led_On(LED_ORANGE);

                if (alarma1==0) {
                    alarma1=1;
                }
            }else{
                UB_Led_Off(LED_ORANGE);
                if (aux_a1==1) {
                    aux_a1=0;
                    alarma1=0;
                }
            }

            if(UB_DigIn_Read(DIN_PB13)==Bit_RESET){
                UB_Led_On(LED_BLUE);
                if (alarma2==0) {
                    alarma2=1;
                }

            }else{
                UB_Led_Off(LED_BLUE);
                if (aux_a2==1) {
                    aux_a2=0;
                    alarma2=0;
                }
            }
        }
}


// ISR von Timer2

void UB_TIMER2_ISR_CallBack(void)
{

        temp=UB_ADC1_SINGLE_Read_MW(ADC_PC4)*0.072;
        if (alarma1==1 && aux_a1==0) {
            UB_Uart_SendString(COM3, "l1", LFCR);
            aux_a1=1;
        }

        if (alarma2==1 && aux_a2==0) {
            UB_Uart_SendString(COM3, "l2", LFCR);
            aux_a2=1;
        }
}
```

Anexo 3

CÓDIGO DE PROGAMACIÓN PARA SMARTPHONE ANDROID

```
package com.example.administrador;

import java.io.IOException;

import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.BasicResponseHandler;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONArray;
import org.json.JSONException;

import android.app.Activity;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends Activity {

    String clave="a1234";
    EditText ePass;
    Button bIngresarAdmin, bConsultar, bConsultarAlarmas;
    ListView lRegistro, lAlarmas;
    private ArrayAdapter<String> mListData, mListAlarmas;


       @Override
       public void onCreate(Bundle savedInstanceState) {
       StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);
             super.onCreate(savedInstanceState);
             setContentView(R.layout.activity_main);
             ePass = (EditText) findViewById(R.id.editText1);
        bIngresarAdmin = (Button) findViewById(R.id.button1);
        bConsultar = (Button) findViewById(R.id.consultar);
        bConsultarAlarmas = (Button) findViewById(R.id.consultarAlarmas);
        lRegistro = (ListView) findViewById(R.id.listView1);
        lAlarmas = (ListView) findViewById(R.id.ListView2);
        mListData = new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1);
        mListAlarmas = new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1);
        lRegistro.setAdapter(mListData);
        lAlarmas.setAdapter(mListAlarmas);



        bIngresarAdmin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(ePass.getText().toString().equals(clave)){
                          try{
```

```
        httpGetData("http://www.alluc.com.ec/trabajo/registrarUsuario.php?
cc=1&nombre=34&apellido=1&direccion=1&telefono=1");
        Toast.makeText(getApplicationContext(), "El dato ha sido enviado
correctamente", 1000).show();
        }catch(Exception e){
        Toast.makeText(getApplicationContext(), "Error en el envio de la
informacion, verifique su conexion a internet y vuelva a intentarlo.",
1000).show();
                        }
                    }
                 }
                });

                 bConsultar.setOnClickListener(new View.OnClickListener() {
                     @Override
                     public void onClick(View view) {
                      if(ePass.getText().toString().equals(clave)){
                             JSONArray ja=null;
                             try {
                                    String data;
                                    for(int j=0; j<15;j++){

        data=httpGetData("http://www.alluc.com.ec/trabajo/consultarAsisten
ciaUnach.php?codigo="+Integer.toString(j));
        if(data.length()>1){
        for (int i = -1; (i = data.indexOf("[", i + 1)) != -1; ) {
        ja=new JSONArray(data.substring(i, data.length()));
        onAddItem(ja.getString(0)+" "+ja.getString(1)+"
"+ja.getString(2)+" "+ja.getString(3)+" "+ja.getString(4));
                                    }
                                }
                             }


                             } catch (JSONException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        Toast.makeText(getApplicationContext(), e.toString().show();

                      }
                     }
                    }
                 });

                 bConsultarAlarmas.setOnClickListener(new View.OnClickListener() {
                     @Override
                     public void onClick(View view) {
                      if(ePass.getText().toString().equals(clave)){
                                    JSONArray ja=null;
                                    try {
                                           String data;
                                           for(int j=1; j<3;j++){

        data=httpGetData("http://www.alluc.com.ec/trabajo/consultarAlarmas
Unach.php?codigo="+Integer.toString(j));
                      if(data.length()>1){
                      for (int i = -1; (i = data.indexOf("[", i + 1)) != -1; ) {
                      ja=new JSONArray(data.substring(i, data.length()));
                      onAddItemAlarmas(ja.getString(0)+" "+ja.getString(1)+" "
                      +ja.getString(2)+" "+ja.getString(3));
                                    }
                      }
        }
```

```
                        } catch (JSONException e) {
                                // TODO Auto-generated catch block
                e.printStackTrace();
                Toast.makeText(getApplicationContext(), e.toString(),
1000).show();

                    }
                }
              }
          });

        }

        protected void onAddItem(String value){
                mListData.add(value);
                mListData.notifyDataSetChanged();
    }

        protected void onAddItemAlarmas(String value){
                mListAlarmas.add(value);
                mListAlarmas.notifyDataSetChanged();
    }


    public String httpGetData(String mURL) {
        String response="";
        mURL=mURL.replace(" ", "%20");
         Log.i("LocAndroid Response HTTP Threas","Ejecutando get 0:
"+mURL);
          HttpClient httpclient = new DefaultHttpClient();

          Log.i("LocAndroid Response HTTP Thread","Ejecutando get 1");
       HttpGet httppost = new HttpGet(mURL);
          Log.i("LocAndroid Response HTTP Thread","Ejecutando get 2");
       try {


          Log.i("LocAndroid Response HTTP","Ejecutando get");
         // Execute HTTP Post Request
      ResponseHandler<String> responseHandler=new BasicResponseHandler();


                response = httpclient.execute(httppost,responseHandler);
                //onAddItem(response);
          Log.i("LocAndroid Response HTTP",response);
       } catch (ClientProtocolException e) {
         Log.i("LocAndroid Response HTTP ERROR 1",e.getMessage());
         // TODO Auto-generated catch block
    } catch (IOException e) {

        Log.i("LocAndroid Response HTTP ERROR 2",e.getMessage());
        // TODO Auto-generated catch block
    }
    return response;

    }
}
```

Anexo 4

CÓDIGO DE PROGAMACIÓN PARA TABLET ANDROID

```
public class MainActivity extends Activity {

    private static int pag=0;
    private final String TAG = MainActivity.class.getSimpleName();
    private UsbManager mUsbManager;
    private ListView mListView;
    private static final int MESSAGE_REFRESH = 101;
    private static final long REFRESH_TIMEOUT_MILLIS = 5000;

       private final Handler mHandler = new Handler() {
         @Override
         public void handleMessage(Message msg) {
             switch (msg.what) {
                 case MESSAGE_REFRESH:
                     refreshDeviceList();
                     mHandler.sendEmptyMessageDelayed(MESSAGE_REFRESH,
REFRESH_TIMEOUT_MILLIS);
                     break;
                 default:
                     super.handleMessage(msg);
                     break;
             }
         }
    };

    /** Simple container for a UsbDevice and its driver. */
    private static class DeviceEntry {
        public UsbDevice device;
        public UsbSerialDriver driver;

        DeviceEntry(UsbDevice device, UsbSerialDriver driver) {
            this.device = device;
            this.driver = driver;
        }
    }

    private List<DeviceEntry> mEntries = new ArrayList<DeviceEntry>();
    private ArrayAdapter<DeviceEntry> mAdapter;
    private static String mensaje="";
    TextView tMain;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mUsbManager = (UsbManager) getSystemService(Context.USB_SERVICE);
        mListView = (ListView) findViewById(R.id.deviceList);
        tMain = (TextView) findViewById(R.id.textView12);


        if(mensaje.equals("")){
            tMain.setText("PROCESANDO");
        }else{
            tMain.setText(mensaje);
        }

        mAdapter = new ArrayAdapter<DeviceEntry>(this,
android.R.layout.simple_expandable_list_item_2, mEntries) {
            @Override
```

```
            public View getView(int position, View convertView, ViewGroup
parent) {
                final TwoLineListItem row;
                if (convertView == null){
                    final LayoutInflater inflater =
                            (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);
                    row = (TwoLineListItem)
inflater.inflate(android.R.layout.simple_list_item_2, null);
                } else {
                    row = (TwoLineListItem) convertView;
                }

                final DeviceEntry entry = mEntries.get(position);
                final String title = String.format("Vendor %s Product
%s",
                        HexDump.toHexString((short)
entry.device.getVendorId()),
                        HexDump.toHexString((short)
entry.device.getProductId()));
                row.getText1().setText(title);
                final String subtitle = entry.driver != null ?
                entry.driver.getClass().getSimpleName() : "No Driver";
                row.getText2().setText(subtitle);
                if(entry.driver!=null){
                    showConsoleActivity(entry.driver);
                }
                return row;
            }
        };

        mListView.setAdapter(mAdapter);
    }

    @Override
    protected void onResume() {
        super.onResume();
        mHandler.sendEmptyMessage(MESSAGE_REFRESH);
    }

    @Override
    protected void onPause() {
        super.onPause();
        mHandler.removeMessages(MESSAGE_REFRESH);
    }
    private void refreshDeviceList() {
        new AsyncTask<Void, Void, List<DeviceEntry>>() {
            @Override
            protected List<DeviceEntry> doInBackground(Void... params) {
                Log.d(TAG, "Refreshing device list ...");
                SystemClock.sleep(1000);
                final List<DeviceEntry> result = new
ArrayList<DeviceEntry>();
                for (final UsbDevice device :
mUsbManager.getDeviceList().values()) {
                    final List<UsbSerialDriver> drivers =

UsbSerialProber.probeSingleDevice(mUsbManager, device);
                    Log.d(TAG, "Found usb device: " + device);
                    if (drivers.isEmpty()) {
                        Log.d(TAG, "  - No UsbSerialDriver available.");
                        result.add(new DeviceEntry(device, null));
                    } else {
                        for (UsbSerialDriver driver : drivers) {
                            Log.d(TAG, "  + " + driver);
```

```
                            result.add(new DeviceEntry(device, driver));
                        }
                    }
                }
                return result;
            }

            @Override
            protected void onPostExecute(List<DeviceEntry> result) {
                mEntries.clear();
                mEntries.addAll(result);
                mAdapter.notifyDataSetChanged();
                Log.d(TAG, "Done refreshing, " + mEntries.size() + "
entries found.");
            }
        }.execute((Void) null);
    }


    private void showConsoleActivity(UsbSerialDriver driver) {

        switch (pag){
            case 0:
                pag=0;
                SerialConsoleActivity.show(this, driver);
                break;
            case 1:
                pag=0;
                Identificacion.show(this, driver);
                break;
            case 2:
                pag=0;
                break;
            case 3:
                pag=0;
                Administrador.show(this, driver);
                break;

            case 4:
                pag=0;
                AdministradorMenu.show(this, driver);
                break;
            case 5:
                pag=0;
                AdministradorEnrolar.show(this, driver);
                break;
            case 6:
                pag=0;
                break;
            default:
                pag=0;
                SerialConsoleActivity.show(this, driver);
                break;
        }
    }

    static void show(Context context, int pagR, String men) {
        pag = pagR;
        mensaje = men;
        final Intent intent = new Intent(context, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP |
Intent.FLAG_ACTIVITY_NO_HISTORY);
        context.startActivity(intent);
    }
}
```

Anexo 5

MANUAL DE USUARIO

INTRODUCCIÓN

El sistema permite sensar y controlar las variables ambientales del laboratorio como temperatura, presencia de humo. Además Permite controlar el acceso al laboratorio de forma remota, desde un dispositivo móvil o mediante reconocimiento de huella con sensor biométrico del personal autorizado mediante registros almacenados en una base de datos.

Monitorea y envía en tiempo real la información del servidor de televisión digital y viceversa para controlar automáticamente parámetros de transmisión del servidor de televisión como potencia de transmisión, latencia, Ts-rate, frecuencia.

Para cumplir con estas funciones el sistema cuenta con una aplicación para Smartphone que permite acceder al sistema de telemetría almacenado en el servidor de televisión digital de manera segura y óptima.

Utiliza la red de telefonía celular para el envío y recepción de datos desde el servidor hacia el usuario utilizando internet móvil de cualquier operadora celular, además de utilizar los servicios de mensajes cortos como servicio complementario para generar un sistema de telemetría robusto.

INSTRUCCIONES DE SEGURIDAD

- Por seguridad para evitar la manipulación del sistema de personal no autorizado el sistema cuenta con una contraseña de ingreso **a1234**
- Se debe verificar que los componentes del sistema estén siempre alimentados a una fuente de energía eléctrica.
- Verificar el estado de los sensores y actuadores.

HMI SISTEMA DE TELEMETRÍA

FUNCIONES DEL HMI

Monitoreo.- se refiere a la habilidad de obtener y mostrar datos del sistema en tiempo real. Esta información o datos se deben visualizar de manera simple y legible para el ser humano se pueden visualizan en forma de números, texto, gráficos o señales que permitan la correcta comprensión de dicha información.

Supervisión.- esta función permite junto con el monitoreo la posibilidad de ajustar las condiciones de trabajo del sistema, es decir la modificación de variables de trabajo.

Alarmas.- debe tener la capacidad de reconocer eventos inusuales dentro de sistema y ser capaces de reportarlos.

Control.- debe tener la capacidad de ajustar automáticamente valores del sistema mediante correcciones o activación de procesos adiciones que regulen o controlen el correcto funcionamiento del sistema o de las variables que se encuentre monitoreando.

Históricos.- tiene que ver con la capacidad de almacenar en archivos datos del sistema registros e información importante de manera ordena y clara que además puede ser llamada y visualizada de ser requerido el caso.

DESCRIPCIÓN DEL HMI

La interfaz maquina hombre se la realiza mediante una tablet cumple la función de una computadora, además de permitir visualizar la información en su pantalla; también ofrece la posibilidad de ser usada como teclado para ingresar información gracias a que posee una pantalla touch.

Se creó un software específico para cumplir con los requerimientos del sistema para lo cual se diseñaron diferentes pantallas haciendo uso de la interfaz de programación Android Studio las cuales se explican y analizan a continuación.

Figura. Pantalla principal

En la figura superior se muestra la pantalla principal que es la presentación del HMI, es una aplicación creada para sistema operativo Android, en esta primera pantalla se observa dos iconos el de identificación y administrador.

Al presionar el icono de identificación la aplicación nos envía a otra pantalla donde se solicitara al usuario colocar su dedo para identificación, si se pulsa sobre el icono de administrador se accede a otra pantalla donde el administrador debe identificarse para acceder a las opciones de control del sistema.

Figura. Pantalla de identificación de huella

En la figura anterior se muestra la pantalla de identificación del usuario, aquí se solicita que coloque su dedo sobre el lector biométrico para identificación, si la identificación es correcta se activa la chapa eléctrica y además se guarda la fecha y hora de acceso al laboratorio junto con la identificación de la persona que accedió al laboratorio, además por seguridad se envía un sms al celular de la persona administradora del laboratorio.

Figura 68. Pantalla de identificación del administrador

En la figura superior se observa la pantalla de identificación del administrador, se solicita una clave de acceso la cual es (a1234) la cual solo debe poseer la persona encargada del laboratorio la cual será la encargada de enrolar nuevos usuarios y verificar que todo funcione correctamente.

Figura. Pantalla de administración

En la figura anterior se observa la pantalla de presentación a la que se accede después de ingresar a la opción de administrador, en esta pantalla se observan las opciones de enrolar usuario, set point temperatura, borrar.

Enrolar usuario se refiere a la opción de ingresar nuevos usuarios al sistema de acceso al laboratorio esta nos envía a otra pantalla donde se registra los nombres y la huella de la persona, el sp (set point) temperatura permite calibrar la temperatura ambiente a la que se requiere mantener el laboratorio, y en borrar se eliminar de registro a usuarios registrados.

Figura. Pantalla para agregar usuario

Fuente: autor

En la figura superior se observa la pantalla de registro para agregar a un nuevo usuario, en esta pantalla se registran los datos del nuevo usuario además de su huella dactilar para que pueda tener acceso al laboratorio, estos datos de usuario son almacenados en la base de datos del servidor web, el programa asigna un código a cada usuario dicho código es comparado con la base de datos para verificar la existencia del usuario y de esa manera permitir el acceso.

El HMI trabaja en conjunto con la placa electrónica y la base de datos del servidor web, para comparar y enviar datos en tiempo real.

DESCRIPCIÓN DE LA APLICACIÓN PARA SMARTPHONE

La aplicación para teléfonos inteligentes con sistema Android permite monitorear los datos de ingresos y alarmas, además de poder controlar de manera remota el acceso al laboratorio es decir abrir la puerta de ingreso al laboratorio con solo tocar un botón de la aplicación, las consultas y envió de información se la realiza de igual manera que la aplicación HMI es decir utilizando el internet móvil de cualquier operadora, además ya que es un dispositivo móvil puede accesar de otras formas al internet lo que hace que el administrador puede comunicarse con el sistema desde cualquier punto del mundo.

Una vez instalada la aplicación aparecerá en el menú de aplicaciones del teléfono con el nombre de ADMINISTRADOR y el logotipo de la UNACH. Para poder utilizar la aplicación hay que ingresar la clave de administrador (a1234) de caso contrario la aplicación no realizara ninguna función.

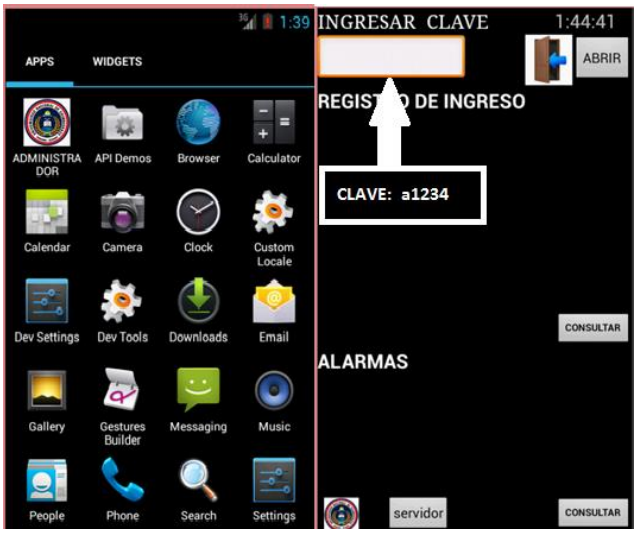
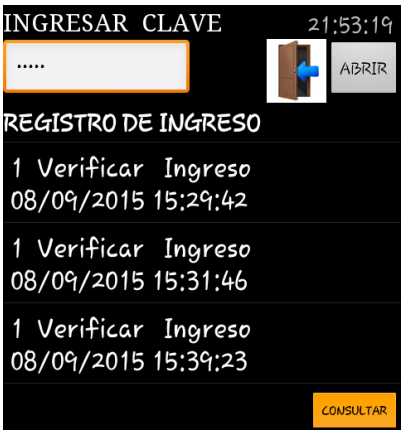


Figura. Entorno de la aplicación

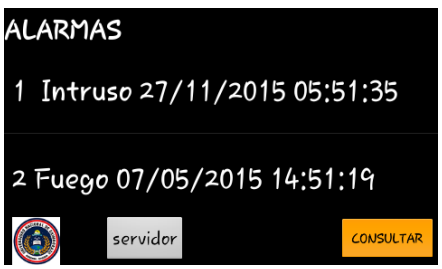
Como se observa en la figura superior la aplicación contiene cuatro botones los cuales se describen a continuación.

En la esquina superior derecha de la aplicación se puede observar un botón de abrir, al presionar este botón la aplicación envía una señal que permite activar remotamente la cerradura eléctrica del laboratorio de televisión digital de la UNACH.

En las siguientes figuras se muestra el funcionamiento de los botones consultar, existen dos botones; el primero para consultar el registro de ingreso. Aquí se visualizan en orden descendente los ingresos del personal autorizado al laboratorio de televisión digital.



El segundo sirve para visualizar el registro de alarmas generadas por los sensores instalados en el laboratorio, tal como se observa en la siguiente figura.



El botón servidor que se encuentra en la parte inferior de la aplicación junto al logotipo de la UNACH, activa una segunda actividad o pantalla.

La segunda pantalla de la aplicación sirve para visualizar los datos de configuración del servidor de televisión digital, esta información se encuentra agrupada por bloques tal como se muestra en la siguiente figura.

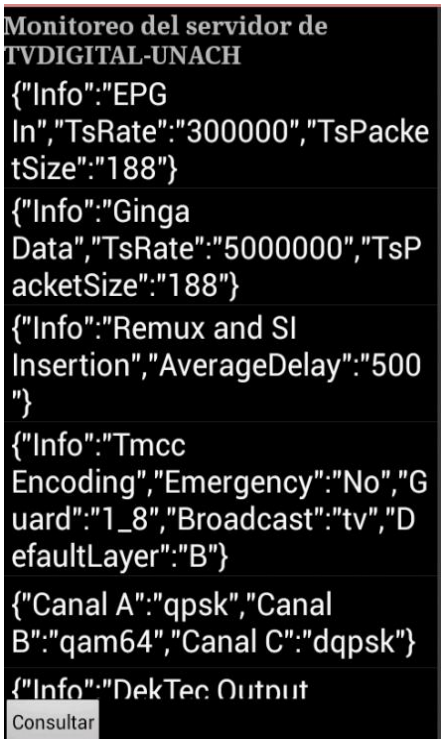


Figura. Monitoreo del servidor de televisión