



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES.**

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

Título del proyecto:

**DISEÑO E IMPLEMENTACION DE UN SISTEMA DE COMUNICACION
INTEGRANDO TECNOLOGIAS PARA EL MEJORAMIENTO DE VOZ IP
INTERNA DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD
NACIONAL DE CHIMBORAZO**

AUTORES:

Diego Roberto Obando Flores
Sebastián José Escobar Martínez

TUTOR:

Ing. Deysi Inca Balseca

Riobamba – Ecuador

**AÑO
2015**

Los miembros del Tribunal de Graduación del proyecto de investigación de título:
**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIÓN
INTEGRANDO TECNOLOGÍAS PARA EL MEJORAMIENTO DE VOZ IP
INTERNA DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD
NACIONAL DE CHIMBORAZO**

Presentado por:

Diego Roberto Obando Flores

Sebastián José Escobar Martínez

Y dirigida por:

Ing. Deysi Inca Balseca

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:

Ing. Carlos Peñafiel

Presidente del Tribunal



Firma

Ing. Deysi Inca Balseca

Directora de Tesis



Firma

Ing. Fabián Gunsha

Miembro del Tribunal



Firma

ASPECTOS GENERALES

TÍTULO DEL TRABAJO DE GRADUACIÓN

Diseño e Implementación de un Sistema de Comunicación integrando Tecnologías para el mejoramiento de voz IP interna de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo

AUTORES:

Diego Roberto Obando Flores
Sebastián José Escobar Martínez

ASESOR:

Ing. Deysi Inca Balseca

COLABORADOR(ES):

Ing. Carlos Peñafiel
Ing. Fabián Gunsha

LUGAR DE REALIZACIÓN

Facultad de Ingeniería de la Universidad Nacional de Chimborazo.

TIEMPO ESTIMADO DE ESTUDIO

6 MESES

BENEFICIARIOS

Facultad de Ingeniería

COSTO ESTIMADO

3000 Dólares

FINANCIAMIENTO

AUTOFINANCIADO

CERTIFICACIÓN DEL TUTOR

CERTIFICACIÓN

Certifico que el presente trabajo de investigación previo a la obtención del Grado de Ingeniero en ELECTRÓNICA Y TELECOMUNICACIONES, con el tema:

“DISEÑO E IMPLEMENTACION DE UN SISTEMA DE COMUNICACION INTEGRANDO TECNOLOGIAS PARA EL MEJORAMIENTO DE VOZ IP INTERNA DE LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO.” ha sido elaborado por **DIEGO ROBERTO OBANDO FLORES** y **SEBASTIÁN JOSÉ ESCOBAR MARTÍNEZ**, el mismo que ha sido revisado y analizado en un cien por ciento con el asesoramiento permanente de mi persona en calidad de Tutor, por lo que se encuentra apto para su presentación y defensa respectiva.

Es todo cuanto puedo informar en honor a la verdad.


Firma del Director de Tesis

AUTORÍA

Yo **DIEGO ROBERTO OBANDO FLORES** con cédula de identidad N° **060353139-3** y **SEBASTIÁN JOSÉ ESCOBAR MARTÍNEZ** con cédula de identidad N° **060380365-1**, somos responsables de las ideas, doctrinas, resultados y propuesta realizadas en la presente investigación y el patrimonio intelectual del trabajo investigativo pertenece a la Universidad Nacional de Chimborazo.



DIEGO ROBERTO
OBANDO FLORES

060353139-3



SEBASTIÁN JOSÉ
ESCOBAR MARTÍNEZ

060380365-1

Dedicatoria

Dedico el presente trabajo en primer lugar a Dios que me permite compartir este momento tan importante en mi vida con los seres que yo amo y han estado junto a mí.

A mis queridos Padres que me dieron la vida y que gracias a su esfuerzo, trabajo y apoyo incondicional e llegado a culminar una etapa muy importante en mi vida, me faltan palabras para dedicarles este logro muy grande, que sin su ayuda no habría podido alcanzarlo, gracias por todo papá y mamá, gracias por brindarme su ejemplo y hacerme una persona de bien y poder ayudar a la sociedad.

A todos mis hermanos William, Byron, a mi hija amada Brithany Daniela y la madre de mi hija Esthela que por medio de ellos he podido sobresalir que siempre estuvieron conmigo y que de una u otra forma nunca me abandonaron.

A Dios por haberme permitido cumplir con una parte fundamental de mi formación profesional, por la fortaleza y sabiduría que me ha concedido en este camino que emprendí para alcanzar mi meta.

A mis padres por la educación y los buenos principios inculcados formándome para ser una persona digna, por el amor y cariño que me demuestran sobre todo en los momentos más difíciles, que han hecho de mi una persona fuerte y emprendedora, sin ellos no hubiera llegado tan lejos.

A mi esposa e hijo quienes han sido mis principales motivadores para alcanzar este logro y por ser un apoyo fundamental en el desarrollo de este trabajo.

Agradecimientos

En este corto párrafo queremos agradecer a Dios que nos ilumino durante los años de estudio para poder alcanzar nuestras metas.

A la Universidad Nacional de Chimborazo que nos abrió las puertas para poder seguir la carrera y cumplir nuestros sueños como una personas profesional.

A nuestra tutora de Tesis Ing. Deysi Inca que siempre estuvo apoyándonos en cualquier duda e inquietud que teníamos para poder realizar nuestra tesis.

Índice

1. MARCO REFERENCIAL	10
1.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA	11
1.2. ANÁLISIS CRÍTICO	11
1.3. PROGNOSIS	11
1.4. DELIMITACIÓN	11
1.5. OBJETIVOS	12
1.6. JUSTIFICACIÓN	13
2. MARCO TEÓRICO	15
2.1. ANTECEDENTES	15
2.2. REDES INALÁMBRICAS	16
2.3. REDES CELULAR	18
2.4. ALMACENAMIENTO DE INFORMACIÓN	20
2.5. SISTEMAS DE BASES DE DATOS	22
2.6. SERVIDOR WEB	23
2.7. PANTALLAS LED	23
2.8. ARDUINO	28
2.9. HIPÓTESIS	60
2.10. IDENTIFICACIÓN DE VARIABLES	60
3. ESTUDIO DE COBERTURA	61
3.1. Distribución de las Plantas del Edificio de la Facultad de Ingeniería	61
3.2. Software De Medición De La Cobertura De La Red Inalámbrica	63
3.3. Gráficas obtenidas mediante la utilización del software	66
3.4. Adquisición de datos de señal de cobertura	68
3.5. Cobertura De La Señal Inalámbrica De Cada Punto De Medición	70
4. MEJORAMIENTO DE LA RED INALÁMBRICA	78
5. METODOLOGÍA	82

5.1.	MÉTODOS Y TÉCNICAS	82
5.2.	OPERACIONALIZACIÓN DE LAS VARIABLES	82
5.1.	TIPO DE ESTUDIO	83
5.2.	POBLACIÓN BENEFICIARIA	83
5.3.	PROCEDIMIENTOS.....	84
6.	RESULTADOS	113
6.1.	RESULTADOS OBTENIDOS	113
6.2.	COMPROBACIÓN DE LA HIPÓTESIS.....	116
6.3.	DEMOSTRACIÓN DE LA HIPÓTESIS.....	117
7.	DISCUSIÓN	119
8.	CONCLUSIONES Y RECOMENDACIONES.....	120
8.1.	CONCLUSIONES	120
8.2.	RECOMENDACIONES.....	121
9.	PROPUESTA.....	122
9.1.	TÍTULO DE LA PROPUESTA	122
9.2.	INTRODUCCIÓN.....	122
9.3.	OBJETIVOS.....	122
9.4.	FUNDAMENTACIÓN CIENTÍFICO – TÉCNICA	123
9.5.	DESCRIPCIÓN DE LA PROPUESTA	124
9.6.	DISEÑO ORGANIZACIONAL	126
9.7.	MONITOREO Y EVALUACIÓN DE LA PROPUESTA.....	126
10.	BIBLIOGRAFÍA.....	129
	ANEXOS.....	131
	ANEXO A: Módulo Arduino Leonardo.....	132
	ANEXO B: Módulo GSM/GPRS SIM 900	133

Índice de figuras

Figura 1. Red de celdas.	18
Figura 2. Áreas de cobertura de una red celular.....	20
Figura 3. Pantalla Tipo LED.	24
Figura 4. Píxeles en una Pantalla.	25
Figura 5. Tipos de Tamaño de Punto.	26
Figura 6. Angulo de Visión en Pantallas LED SMD.	26
Figura 7. (a) Conector DVI. (b) Conector RCA. (c) Conector HDMI. (d) Conector VGA. (e) Conector para la energía eléctrica.	27
Figura 8. Microcontrolador Atmega32u4.	49
Figura 9. Arduino Leonardo.....	52
Figura 10. Chip SIM900 de SIMCOM.	52
Figura 11. Pines del módulo SIM900.	53
Figura 12. Diagrama funcional del SHIELD MODEM GSM/GPRS.	53
Figura 13. Diagrama esquemático del Shield Modem Gsm Gprs Sms Sim900.	59
Figura 14. Shield Modem Gsm Gprs Sms Sim900.....	59
Figura 15. Inicialización del software ACRYLIC WIFI-HEATMAPS.....	64
Figura 16. Aplicaciones con Acrylic WI-FI HeadMaps.	65
Figura 17. Icono que detalla las redes presentes.....	65
Figura 18. Canales utilizados por la red.....	67
Figura 19. Variación de la señal en el transcurso del tiempo.....	68
Figura 20. Medición Área 1	72
Figura 21. Medición Área 2	73
Figura 22. Medición Área 3	73
Figura 23. Medición Área 4	74
Figura 24. Medición Área 5	75
Figura 25. Medición Área 6	75
Figura 26. Medición Área 7	76

Figura 27. Medición Área 8	76
Figura 28. Medición Área 9	77
Figura 29. Rango de colores.	78
Figura 30. Medición De Señal De La Planta Baja	79
Figura 31. Medición de señal de la segunda planta.	79
Figura 32. Medición De Señal De La Tercera Planta.	80
Figura 33. Diagrama de procedimientos.	85
Figura 34. Pantalla inicio Arduino Software.....	86
Figura 35. Selección del módulo Arduino Leonardo.	87
Figura 36. Selección del puerto virtual para conexión a la placa Arduino.	87
Figura 37. Extracto código Arduino. Configuración inicial variables.	88
Figura 38. Extracto código Arduino. Configuración módulo Arduino.....	88
Figura 39. Extracto código Arduino. Lectura de mensajes de la placa GSM/GPRS. 89	
Figura 40. Extracto código Arduino. Comparación de la clave.	90
Figura 41. Extracto código Arduino. Envío mensaje de respuesta al remitente.	90
Figura 42. Funcionamiento servidor.	91
Figura 43. Conector video Vga a Vga.....	92
Figura 44. Librerías Processing.....	93
Figura 45. Librerías Processing.....	94
Figura 46. Librerías Processing.....	94
Figura 47. Librerías Processing.....	94
Figura 48. Variables para los menús y submenús	94
Figura 49. Menús y submenús principales de la aplicación.....	94
Figura 50. Variables para el manejo de fechas	94
Figura 51. Variables color y tipo de letra.....	95
Figura 52. Colocación imagen de la UNACH	95
Figura 53. Variable para almacenar los datos provenientes del puerto serial.....	95
Figura 54. Variables para el manejo de la rotación de las noticias	95
Figura 55. Variables para el manejo del puerto serial.....	95
Figura 56. Variables para el manejo de la conexión y consultas a la base de datos ..	95

Figura 57. Declaración de Variables para los cuadros de texto y botones.....	96
Figura 58. Pantalla principal para La Lista de Noticias	96
Figura 59. Código para definir el lugar, en el cual se ubicaran los cuadros de texto y botones	97
Figura 60. Declaración de Variables para los cuadros de texto y botones.....	97
Figura 61. Código para definir el lugar, en el cual se ubicaran los cuadros de texto y botones	97
Figura 62. Pantalla Principal para el ingreso de una nueva noticia	98
Figura 63. Declaración de Variables para los cuadros de texto y botones.....	98
Figura 64. Código para definir el lugar, en el cual se ubicaran los cuadros de texto y botones	98
Figura 65. Pantalla principal para el ingreso de los emails.....	99
Figura 66. Código declarar variables y definir el lugar, en el cual se ubicaran los cuadros de texto y botones	99
Figura 67. Pantalla principal para modificar el intervalo.....	99
Figura 68. Variables conexión a la base de datos	100
Figura 69. Conexión a la base de datos.....	100
Figura 70. Definir el tamaño de la pantalla.....	100
Figura 71. Código para recepción del mensaje recibido a través del puerto serial ..	101
Figura 72. Código manejo de menús.....	101
Figura 73. Código para recepción del mensaje recibido a través del puerto serial ..	102
Figura 74. Código para cerrar la aplicación	102
Figura 75. Código para consultas query a la base de datos.....	103
Figura 76. Código para consultas query a la base de datos.....	103
Figura 77. Código para consultas query a la base de datos.....	104
Figura 78. Código para consultas envió de mensajes vía email.....	104
Figura 79. Código para consultas envió de mensajes vía email.....	105
Figura 80. Pantalla inicial Aplicación.....	105
Figura 81. Menús principales.....	106
Figura 82. Menú enlistar noticias.....	106

Figura 83. Menú modificar noticia.....	107
Figura 84. Menú ingresar nueva noticia.....	107
Figura 85. Menú Datos usuario fuente.....	108
Figura 86. Menú datos usuario destino.....	108
Figura 87. Intervalos.....	109
Figura 88. Rotación de noticias en la pantalla.....	109
Figura 89. Pantalla vacía, luego de accionar Parar.....	110
Figura 90. Módulos en funcionamiento.....	110
Figura 91. Envío y recepción del mensaje SMS.....	111
Figura 92. Recepción de la respuesta en el correo electrónico.....	111
Figura 93. Adición de la información en la pantalla, al momento de la recepción...	112
Figura 94. Módulos en funcionamiento.....	113
Figura 95. Envío y recepción del mensaje SMS.....	114
Figura 96. Recepción de la respuesta en el correo electrónico.....	114
Figura 97. Adición de la información en la pantalla, al momento de la recepción...	115
Figura 98. Colocación de la pantalla.....	116
Figura 99. Conectores VGA / UTP utilizados.....	116
Figura 100. Pantalla Tipo LED.....	124
Figura 101. Diagrama de procedimientos.....	126
Figura 102. Módulos en funcionamiento.....	126
Figura 103. Envío y recepción del mensaje SMS.....	127
Figura 104. Recepción de la respuesta en el correo electrónico.....	127
Figura 105. Adición de la información en la pantalla, al momento de la recepción.	128

Índice de Tablas

Tabla 1. Modelos de Arduino.	28
Tabla 2. Arreglos.....	37
Tabla 3. Bucles.....	37
Tabla 4. Impresión de cadenas	38
Tabla 5. Pines de la interfaz serial	55
Tabla 6. Bandas de frecuencia utilizadas por redes inalámbricas	62
Tabla 7. Medición del nivel de calidad de señal	69
Tabla 8. Rangos de valor de señal tomado en dBm	71
Tabla 9. Resumen cobertura de las distintas áreas.....	77
Tabla 10. Operación de Variables.....	83
Tabla 11. Datos de la variable.....	116
Tabla 12. Valores calculados	117
Tabla 13. Comprobación de Hipótesis	118

Resumen

Las redes inalámbricas están reglamentadas por los estándares 802.11 a/b/g, las cuales deben estar consideradas en el momento de su instalación y configuración. La Universidad Nacional de Chimborazo ha mejorado notablemente el acceso a los servicios como mensajería instantánea, navegación por internet, mensajería electrónica para los estudiantes en cualquier espacio abierto del campus universitario, garantizando movilidad, cobertura y la disponibilidad en forma permanente.

El presente artículo detalla el trabajo realizado para la medición de cobertura de la red inalámbrica de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo, usando el software Acrylic Wi-Fi Heat Maps para la toma de medidas de potencia en cada punto de acceso. Para facilitar la visualización se contará con planos de los distintos puntos de medición y de la información recogida se realiza un mapa de niveles de potencia para posteriormente sugerir la implementación de access-point en puntos estratégicos con miras a mejorar la cobertura.

Mejorar el desempeño de la red inalámbrica y por medio del uso de pantallas informativas ofrecer de forma óptima el acceso a la información de los docentes, estudiantes y ciudadanía en general. Implementando un prototipo que permita emitir una noticia con urgencia en una pantalla LED, enviando información a través de mensajes de texto desde un teléfono celular.



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERIA
CENTRO DE IDIOMAS



Lic. Eduardo Herrera

06 de enero de 2016

SUMMARY

INTRODUCCIÓN

Desde el origen del hombre, la comunicación ha sido un factor determinante para la evolución constante del ser humano sin importar distancia, idioma o cualquier otro tipo de barrera. Hoy en día, gracias a la evolución de la tecnología permite cómodamente del “poder” de comunicarse con solo mover un dedo, el acceso a noticias y por qué no, publicar contenido en la nueva era del conocimiento. Sin duda el desarrollo e investigación realizado por equipos especialistas permite a los seres humanos la oportunidad de adquirir a un precio “razonable” comunicación e interacción instantánea en cuestión de segundos.

En la actualidad, con la demanda cada vez más constante de acceso a la información, ha sido necesario el desarrollo de nuevas herramientas para procesar y presentar los datos a determinados usuarios.

El uso de estas herramientas por si solas no lleva a ninguna parte, si no va acompañada de una utilización inteligente de la información que las tecnologías permitan gestionar.

Uno de los medios para brindar acceso a la información es el uso de tecnologías de comunicación inalámbricas. Con el aumento de usuarios que requieren estos servicios, aparecen limitantes en este tipo de tecnología, razón por la cual es necesaria una reestructuración de estas redes o buscar otras alternativas para brindar nuevos servicios.

A través del uso de pantallas informativas se puede proyectar noticias, eventos, ubicación, etc. En el mercado existen varias opciones comerciales, pero debido a su alto costo, se puede reemplazar por otros mecanismos para publicar información a través de pantallas LED, con costos inferiores.

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

El Campus de la Universidad Nacional de Chimborazo cubre una extensión importante de territorio, en el cual se encuentran distribuidas varias edificaciones, entre ellas la de la facultad de ingeniería.

Ante la necesidad de publicar notas informativas como el calendario académico, ubicación de los docentes, información de parte de las autoridades hacia los docentes, estudiantes y ciudadanía en general; esta actividad, en la mayoría de ocasiones, se ha realizado utilizando carteles, pancartas e incluso hojas volantes. Por ello, es imperiosa la necesidad de implementar un mecanismo que permita emitir esta información de mejor manera. Este mecanismo puede mejorarse ostensiblemente a través del mejoramiento de la red inalámbrica y por medio del uso de pantallas informativas, optimizando así el tiempo y recursos.

1.2. ANÁLISIS CRÍTICO

Ante la necesidad de brindar mejores accesos a la información tanto de docentes, estudiantes y ciudadanía en general, se pretende optimizar la red inalámbrica e implementar el uso de pantallas informativas.

1.3. PROGNOSIS

A través del mejoramiento de la red inalámbrica y por medio del uso de pantallas informativas permitirá ofrecer de mejor forma el acceso de la información a los docentes, estudiantes y ciudadanía en general.

1.4. DELIMITACIÓN

El presente proyecto se llevará a cabo en la ciudad de Riobamba en la Universidad Nacional de Chimborazo en la Facultad de Ingeniería, debido a que existen muchas dificultades al momento de hacer público algún evento. Además de dar a conocer las actividades académicas por efectuarse y la ubicación de los docentes. Mediante la ejecución de este proyecto será factible comunicar de mejor manera la información.

La información a proyectar se almacenará en una base datos, que posteriormente se emite en una pantalla LED¹. Si se necesita proyectar una noticia con urgencia se implementa un prototipo que permitirá enviar información a través de mensajes de texto desde un teléfono celular, a un sistema, el cual se encargará de recibir y proyectar los datos en una pantalla LED. El sistema permite comunicarse bidireccionalmente con el celular.

Al momento de realizar el envío de un SMS², enviará un reporte vía email, del que se ha realizado el envío.

Para la implementación del sistema se necesita un computador, el cual realiza las funciones de servidor; además de un celular y dos módulos de arduino: Un Shield Modem Gsm Gprs Sms Sim900 y un módulo de Arduino Leonardo también se necesitamos cable vga y conectores para la transmisión utilizamos cable UTP categoría 6 para poder visualizar la información que se grabara en el servidor y posterior en la pantalla led.

1.5. OBJETIVOS

¹ Light Emitting Diode, que en español significa Diodo Emisor de Luz

² Short Message Service, que en español significa servicio de mensajes cortos

1.5.1. OBJETIVO GENERAL

- Mejorar el desempeño de la red inalámbrica y por medio del uso de pantallas informativas ofrecer de forma óptima el acceso a la información de los docentes, estudiantes y ciudadanía en general.

1.5.2. OBJETIVOS ESPECÍFICOS

- Mejorar el desempeño de la red inalámbrica ofreciendo recomendaciones y sugerencias, resultado de un estudio de cobertura de la red.
- Mediante el uso de pantallas informativas ofrecer de mejor forma el acceso a la información a los docentes, estudiantes y ciudadanía en general.
- Implementar un prototipo que permita emitir una noticia con urgencia en una pantalla LED, enviando información a través de mensajes de texto desde un teléfono celular.
- Enviar un reporte vía email, al emitir una noticia a través de un SMS.

1.6. JUSTIFICACIÓN

La época actual está ligada al uso cada vez más amplio de las tecnologías en toda actividad humana. Por otra parte, la información tanto al interior de cualquier entidad es muy importante. El acceso a la información, ayuda a que todos los actores de una entidad, se comprometan a participar activamente en el desarrollo de la misma.

En la UNACH, se encuentra un sin número de formas para transmitir información hacia todos los actores del quehacer educativo. Se pueden ver carteles, y hasta pequeñas hojas volantes que contienen información. Si bien estas herramientas son útiles, con el avance de las tecnologías actuales que pueden facilitar esta tarea.

Sin dejar de lado estas herramientas informativas tradicionales, es muy importante aprovechar otras alternativas tecnológicas, que permitan un flujo más oportuno de la información. Por otra parte, la implementación de las TIC's, puede significar ahorros importantes en tiempo de transmisión, pero también el ahorro de material como el papel, lo cual además contribuye a una mejor protección del ambiente.

Realizando un estudio de cobertura se delimitará la eficiencia de la red inalámbrica. Resultado de este estudio se brindará recomendaciones y sugerencias para el mejoramiento de la red.

Teniendo la necesidad de publicar notas informativas como el calendario académico, ubicación de los docentes, información de parte de las autoridades hacia los docentes, estudiantes y ciudadanía en general; se implementarán mecanismos que permitan proyectar esta información en una pantalla informativa LED, con costos inferiores.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. ANTECEDENTES

El acceso a la información es un conjunto de técnicas para buscar, categorizar, modificar y acceder a la información que se encuentra en un sistema, como: bases de datos, bibliotecas, archivos, Internet, etc. (Ley Ciencia y Tecnología, 2002)

Al procesar esta información por el entendimiento humano o por algún tipo de sistema de procesamiento automático, no solamente se limita a clasificarla y archivarla; sino encontrar la mejor manera de obtener los datos deseados utilizando para ello el menor número de recursos. Permitiendo de esta forma el acceso a la información a la ciudadanía y público en general que lo requiera. (Derecho acceso información - CIDH, 2011)

Los datos o información son almacenados en un medio físico como discos duros, memorias flash o DVD. A su vez es necesario un sistema lógico para administrarlo. En sus inicios las primeras empresas y organizaciones empezaron a usar sistemas informáticos se trabajaba con sistemas de ficheros, pero con el incremento de la cantidad de información, surgieron problemas (integridad y duplicidad de información, seguridad etc.). Finalmente se mejoró estos procesos mediante la adopción de un sistema más ordenado y manejable basado en la centralización de la gestión y la organización de los datos en forma de bases de datos. (Hueso Ibañez Galindo Luis. 2012)

Generalmente para acceder a los datos está delimitada mediante una red cableada Ethernet, fibra óptica o por acceso inalámbrico WiFi, siendo estos métodos los más utilizados.

Las redes inalámbricas han provocado un gran impacto en todos los ámbitos sociales y económicos. Tanto la comunicación por voz como la transferencia de datos, han pasado de ser herramientas ancladas a un lugar y conectadas con cables a elementos que pueden ser transportados y utilizados brindando movilidad, en cualquier momento y en cualquier lugar. Las redes inalámbricas cubren un amplio espectro de tecnologías que se utilizan actualmente para la comunicación sin cables. Así, se tiene las redes Wi-Fi, Bluetooth o 3G, es decir sobre todos los aspectos necesarios para maximizar el rendimiento de la conexión, haciendo hincapié en la instalación de los sistemas, la puesta en producción y la seguridad. (Redes inalámbricas en los países en desarrollo, 2013).

2.2. REDES INALÁMBRICAS

IEEE³ en febrero de 1980 desarrolla el proyecto 802, con el fin de crear estándares para que diferentes tipos de tecnologías pudieran integrarse y trabajar juntas. El proyecto 802 define aspectos relacionados con el cableado físico y la transmisión de datos.

En 1997 se publica el estándar IEEE 802.11 (ISO/IEC 8802-11), especifica dos velocidades de transmisión “teóricas” de 1 y 2 megabits por segundo (Mbit/s) que se transmiten por señales infrarrojas (IR). Define el protocolo CSMA/CA⁴ como método de acceso.

El estándar 802.11 es un estándar internacional que define las características de una red de área local inalámbrica (WLAN). Wi-Fi (que significa "Fidelidad inalámbrica" es el nombre de la certificación otorgada por la Wi-Fi Alliance, anteriormente WECA

³ Institute of Electrical and Electronics Engineers, en español Instituto de Ingenieros Eléctricos y Electrónicos

⁴ Carrier sense multiple access with collision avoidance, en español múltiple acceso por detección de portadora evitando colisiones

(Wireless Ethernet Compatibility Alliance), grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar 802.11. Una red Wi-Fi es en realidad una red que cumple con el estándar 802.11.

Se desarrollaron varias revisiones detalladas a continuación:

- IEEE 802.11a

La revisión 802.11a fue aprobada en 1999. Opera en la banda de 5 GHz y utiliza 52 subportadoras OFDM⁵, con una velocidad máxima de 54 Mbit/s, para redes inalámbricas las velocidades reales son de aproximadamente 20 Mbit/s.

No puede interoperar con equipos del estándar 802.11b, excepto si se dispone de equipos que implementen ambos estándares.

- IEEE 802.11b

La revisión 802.11b del estándar original fue ratificada en 1999. Tiene una velocidad máxima de transmisión de 11 Mbps, el método de acceso es CSMA/CA. Funciona en la banda de 2,4 GHz. Debido al espacio ocupado por la codificación del protocolo CSMA/CA, en la práctica, la velocidad máxima de transmisión con este estándar es de aproximadamente 5,9 Mbit/s sobre TCP y 7,1 Mbit/s sobre UDP.

- IEEE 802.11g

En junio de 2003, se ratificó un tercer estándar de modulación: 802.11g, que es la evolución de 802.11b. Utiliza la banda de 2,4 Ghz, pero opera a una velocidad teórica máxima de 54 Mbit/s, que en promedio es de 22,0 Mbit/s de velocidad real de transferencia.

- IEEE 802.11n

⁵ Orthogonal Frequency-Division Multiplexing, en español acceso múltiple por división de frecuencias ortogonales

Es una propuesta de modificación al estándar IEEE 802.11-2007 para mejorar significativamente el rendimiento de la red más allá de los estándares anteriores, tales como 802.11b y 802.11g, con un incremento significativo en la velocidad máxima de transmisión de 54 Mbps a un máximo de 600 Mbps.

2.3. REDES CELULAR

La telefonía móvil es la comunicación a través de dispositivos que no están conectados mediante cables. El medio de transmisión es el aire y funciona mediante la comunicación por una red de celdas.

La red celular o red de celdas consiste en un sistema, en el cual mediante la combinación de un conjunto de estaciones transmisoras-receptoras de radio (estaciones base) y una serie de centrales telefónicas de conmutación, se posibilita la comunicación entre terminales telefónicos portátiles (teléfonos móviles) o entre terminales portátiles y teléfonos de la red fija tradicional.

Se puede dividir un área (como una ciudad) en células. Cada célula es típicamente de un tamaño de 10 millas cuadradas (unos 26Km²), formando hexágonos en un campo grande, como se muestra en la Figura 1.

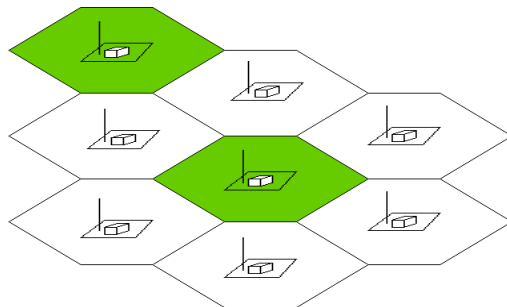


Figura 1. Red de celdas.

Fuente: <http://www.monografias.com/trabajos34/telefonía-celular/telefonía-celular.shtml>

Sin embargo, el tamaño de las células puede variar mucho dependiendo del lugar en que se encuentre. Las estaciones de base se separan entre 1 a 3 Km. en zonas urbanas, llegando a separarse por más de 35Km en zonas rurales.

En zonas muy densamente pobladas o áreas con muchos obstáculos (como puede ser edificios altos), las células pueden concentrarse en distancias cada vez menores. Algunas tecnologías, como los PCS⁶, requieren células muy cercanas unas de otras debido a su alta frecuencia y bajo poder en el que operan.

Los edificios pueden, a su vez, interferir con el envío de las señales entre las células que se encuentren más lejanas, por lo que algunos edificios tienen su propia "microcélula." Los subterráneos son típicos escenarios donde una microcélula se hace necesaria.

Microcélulas pueden ser usadas para incrementar la capacidad general de la red en zonas densamente pobladas como ser los centros capitalinos.

Debido a que los teléfonos celulares y las estaciones de base utilizan transmisores de bajo poder, las mismas frecuencias pueden ser reutilizadas en células no adyacentes. Cada celda en un sistema análogo utiliza un séptimo de los canales de voz disponibles. Eso es, una celda, más las seis celdas que la rodean en un arreglo hexagonal, cada una utilizando un séptimo de los canales disponibles para que cada celda tenga un grupo único de frecuencias y no haya colisiones entre células adyacentes.

Esta configuración puede verse en forma gráfica en la Figura 2:

⁶ Personal Communication Services, en español Servicio de comunicación personal

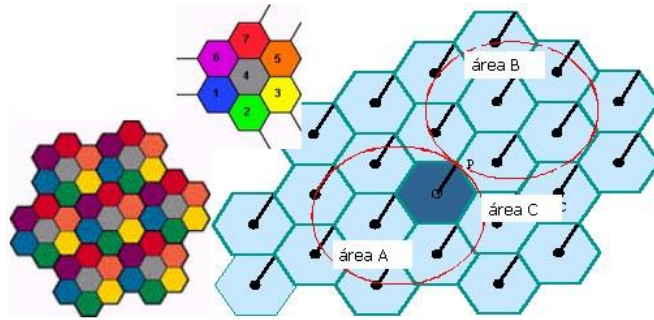


Figura 2. Áreas de cobertura de una red celular.

Fuente: <http://www.monografias.com/trabajos34/telefonía-celular/telefonía-celular.shtml>

De esta forma, en un sistema analógico, en cualquier celda pueden hablar 59 personas en sus teléfonos celulares al mismo tiempo. Con la transmisión digital, el número de canales disponibles aumenta. Por ejemplo el sistema digital TDMA⁷ puede acarrear el triple de llamadas en cada celda, alrededor de 168 canales disponibles simultáneamente.

Cada célula tiene una estación base que consta de una torre y un pequeño edificio en donde se tiene el equipo de radio. Cada célula utiliza un séptimo de los 416 canales duales de voz. Dejando entonces a cada célula aproximadamente los 59 canales disponibles nombrados anteriormente.

2.4. ALMACENAMIENTO DE INFORMACIÓN

Todas las aplicaciones informáticas trabajan en última instancia con datos o información que deben ser almacenados en un medio físico como discos duros,

⁷ Time Division Multiple Access, en español Acceso múltiple por división de tiempo

memorias flash o DVD. Estos medios forman una jerarquía que distingue entre tres niveles de almacenamiento, primario, secundario e intermedio.

2.4.1. Almacenamiento primario

Se refiere a aquellos medios sobre los que la CPU del ordenador puede acceder directamente, y por tanto más rápidamente. Son la memoria principal o memoria RAM y las memorias caché de primer y segundo nivel, más pequeñas pero más rápidas.

2.4.2. Almacenamiento secundario

Se refiere a dispositivos más lentos pero de mayor capacidad como los discos ópticos y magnéticos o las cintas. Para acceder a los datos la CPU debe copiarlos previamente en el almacenamiento primario.

El almacenamiento secundario de más amplio uso es el disco aunque las cintas se usan sobre todo para copias de seguridad por su estabilidad, capacidad y durabilidad.

En un disco duro normalmente agrupa varios discos ópticos cada uno de los cuales se divide en pistas o círculos concéntricos, en ellas se almacena la información. La agrupación de pistas de todos los discos se denomina cilindro.

Cada pista, al contener gran cantidad de información, se subdivide en bloques o sectores de tamaño fijo determinado por el sistema operativo al inicializarlo con un sistema de archivos dado. Los bloques, también llamados páginas, suelen tener un tamaño entre 512 y 4096 bytes.

La transferencia de información entre memoria y disco tiene lugar en unidades de bloque. Cuando produce una orden de lectura se copia uno o varios bloques en el llamado buffer de la memoria (área reservada de la memoria principal), si se necesita efectuar una escritura se copia el bloque correspondiente al bloque del disco.

2.4.3. Almacenamiento intermedio

Cuando se necesita transferir varios bloques de disco a memoria principal y se conocen todas las direcciones de bloque es posible reservar varias áreas de almacenamiento intermedio o buffers dentro de la memoria principal para agilizar la transferencia. De este modo mientras la CPU procesa datos de un buffer puede leer o escribir en otro. El uso de este tipo de almacenamiento es muy común en sistemas de bases de datos.

2.5. SISTEMAS DE BASES DE DATOS

Inicialmente, cuando las primeras empresas y organizaciones empezaron a usar sistemas informáticos se trabajaba con sistemas de ficheros. Es decir, se trabajaba con programas que manejaban información almacenada en ficheros. Cada equipo trabajaba con sus propios datos y programas y se encargaba de su mantenimiento y gestión. Al principio el sistema funcionó, pero con el tiempo y, sobre todo, con el incremento de la cantidad de información, así como de los usuarios que la manejaban, surgieron problemas (integridad y duplicidad de información, seguridad etc.). Estos llevaron finalmente a la organización de la información mediante un sistema más ordenado y manejable basado en la centralización de la gestión y la organización de los datos en forma de bases de datos. Un sinnúmero de sistemas de base de datos se desarrollaron a partir de ahí, entre opciones comerciales y libres, la mejor opción es MYSQL.

2.5.1. MYSQL

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

MySQL es la base de datos de código abierto de mayor aceptación mundial y permite la administración de datos fiables, de alto rendimiento y fácilmente ampliables basadas en la web.

2.6. SERVIDOR WEB

Un servidor web o es un software que administra y publica sitios web. Mediante un lenguaje de programación o aplicación permite a un cliente acceder a la información proyectada. HTML es el lenguaje más utilizado para la creación de páginas web, las instrucciones y comandos son interpretadas por el programa navegador (Google Chrome, Mozilla) utilizados por el cliente, HTML tiene las siguientes características:

- Sintaxis sencilla
- No hay variables
- Las instrucciones no se compilan
- Permite escribir hipertexto
- Se denominan etiquetas a las instrucciones

La página puede ser escrita por completo utilizando cualquier editor de texto, guardando el archivo con extensión .html.

2.6.1. SERVIDOR APACHE

Apache es un servidor web HTTP de código abierto para plataformas Unix-like (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Es el servidor web más usado en todo el mundo, superando en 2009 los 100 millones de sitios web, el 70% del total. Está desarrollado y mantenido por una comunidad de usuarios en torno a la Apache Software Foundation.

2.7. PANTALLAS LED

Gracias al avance tecnológico de las pantallas a color, ahora contamos con pantallas electrónicas tipo LED. Las mismas que han evolucionado con el objetivo de ofrecer mayores prestaciones como son una gran brillantez, mejores niveles de contraste y

resolución lo que ofrece una calidad única, además que consumen menos energía, es menos nociva con el medio ambiente y su vida útil es mayor. El ahorro energético es una ventaja en la implementación de estas pantallas por el mismo hecho de estar construidas con diodos LED que presentan dichas características de ahorro energético.

El uso de pantallas gigantes para exteriores es muy difundido actualmente, siendo la limitante sus altos costos de instalación y puesta a punto. La estructura de este tipo de pantallas publicitarias es generalmente de tipo modular, lo que permite construir pantallas de diferentes dimensiones y agiliza el montaje de la misma. Para la implementación de una pantalla comercial se unen hasta dieciséis gabinetes que son estructuras donde se añaden varios módulos pequeños cada uno con su propio controlador y en comunicación entre sí.



Figura 3. Pantalla Tipo LED.

Fuente: (Jácome, Montenegro, 2012)

Las pantallas son habitualmente utilizadas en televisores, monitores para los ordenadores, cámaras fotográficas e incluso a gran escala como son las pantallas gigantes. El color en las pantallas ha ido evolucionando, lo que da como resultado que se puede crear imágenes más nítidas e inimaginables, como se muestra en la Figura 3.

Pero para que las imágenes presentadas en las pantallas tengan esa perfección deseada se deben tomar en cuenta algunas características que son las siguientes: pixel, tamaño de punto, ángulo de visión, luminancia, contraste y resolución. Las pantallas son habitualmente utilizadas en televisores, monitores para los ordenadores, cámaras fotográficas e incluso a gran escala como son las pantallas gigantes. Todos estos parámetros antes mencionados serán descritos con mayor profundidad a continuación:

Pixel: El pixel es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico que se encuentra en la pantalla.

(Wikipedia, Pixel, 2015), como se muestra en la Figura 4.



Figura 4. Píxeles en una Pantalla.

Fuente: (Jácome, Montenegro, 2012)

Tamaño de punto: El tamaño de punto se lo conoce como el espacio que existe entre dos fósforos de un mismo color. Se debe tomar en cuenta que no siempre es el mismo valor de tamaño de punto en la distribución al medir en vertical que en horizontal. Los puntos de color se pueden encontrar en diferentes disposiciones en la pantalla ya sean tipo de rejilla o máscara de sombra como se muestran en la Figura 5.

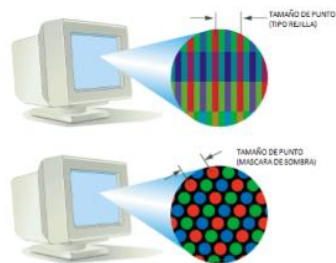


Figura 5. Tipos de Tamaño de Punto.
Fuente: (Jácome, Montenegro, 2012)

Ángulo de Visión: El ángulo de visión en las pantallas que se ofrecen en el mercado son de 160 grados para pantallas LED con dispositivo de montaje superficial SMD “Surface Mounted Device” como se muestra en la Figura 6.

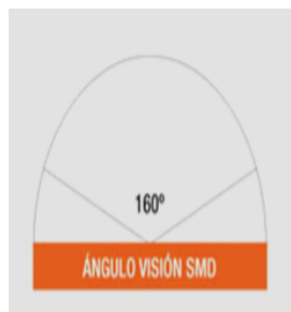


Figura 6. Angulo de Visión en Pantallas LED SMD.
Fuente: (Jácome, Montenegro, 2012)

Luminancia: La luminancia es un factor directamente proporcional con el brillo de la imagen que se presenta en las pantallas.

2.7.1. PANTALLAS Y CONTRASTES.

El contraste en pantallas LED y diseño está definido como la oposición simultánea entre luz y oscuridad, por lo que en pantallas LED, esta variable determina cuál de los 2 factores tendrá prioridad para que la imagen en pantalla sea más perceptible por el ojo humano. Se manejan dos tipos de contraste, el dinámico y el estático.

Al contraste estático, comercialmente se le denomina real y está relacionado con la diferencia de tonos en un momento determinado entre dos píxeles opuestos (oscuro e iluminado), que indica lo ideal de la pantalla para imágenes con poco movimiento, la forma de especificarlo es de la siguiente manera: CE X:YYY, por ello se encuentra pantallas con CE 800:1, 1000:1, etc.



Figura 7. (a) Conector DVI. (b) Conector RCA. (c) Conector HDMI. (d) Conector VGA. (e) Conector para la energía eléctrica.

Fuente: Los Autores.

Los tipos de conectores de la pantalla LED, se puede apreciar en la Figura 7. Cuenta con un conector de 3 patas para la alimentación eléctrica, mientras que para los datos tiene un conector VGA de 15 pines, DVI, RCA y HDMI, mientras que de audio cuentan con los básicos Jack 3.5 y RCA audio, entre otros.

El contraste dinámico es una acción electrónica que realiza la pantalla aumentando y reduciendo la potencia de la iluminación para que sobresalgan los tonos en la pantalla en un lapso de momentos, esto es si la película tiene predominante el color negro, reduce la potencia de la iluminación para que prevalezca tal color, este para determinar que tan ideal es la pantalla con imágenes en movimiento. La forma de especificar el contraste por los fabricantes es el siguiente: CD X:YYY, encontrando valores muy altos con CE como 1:1,000, 1:5,000, 1:50,000.

2.8. ARDUINO

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. (Arduino-Wikipedia, 2015).

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores usualmente utilizados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños.

El software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa. Se programa en el ordenador para que la placa controle los componentes electrónicos.

Tabla 1. Modelos de Arduino.

Placas:	Escudos (Shields):	Kits:
---------	--------------------	-------

Arduino Uno	Arduino GSM Shield	The Arduino Starter Kit
Arduino Leonardo	Arduino Ethernet Shield	Arduino Materia 101
Arduino Due	Arduino WiFi Shield	
Arduino Yún	Arduino Wireless SD Shield	Accesorios:
Arduino Tre (En Desarrollo)	Arduino USB Host Shield	TFT LCD Screen
Arduino Zero (EEUU)	Arduino Motor Shield	USB/Serial Light Adapter
Arduino Micro	Arduino Wireless Proto Shield	Arduino ISP
Arduino Esplora	Arduino Proto Shield	Mini USB/Serial Adapter
Arduino Mega ADK		
Arduino Ethernet		
Arduino Mega 2560		
Arduino Robot		
Arduino Mini		
Arduino Nano		
LilyPad Arduino Simple		
LilyPad Arduino SimpleSnap		
LilyPad Arduino		
LilyPad Arduino USB		
Arduino Pro Mini		
Arduino Fio		
Arduino Pro		

Fuente: (<https://es.wikipedia.org/wiki/Arduino>)

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente. También cuenta con su propio software que se

puede descargar de su página oficial que ya incluye los drivers de todas las tarjetas disponibles lo que hace más fácil la carga de códigos desde el computador.

Los modelos en venta de Arduino se categorizan en 4 diferentes productos, como se puede apreciar en la Tabla 1 y a continuación:

- placas

- escudos
- kits
- accesorios

2.8.1. Aplicaciones

La plataforma Arduino se utiliza como base en diversas aplicaciones electrónicas:

- Xoscillo: Osciloscopio de código abierto.
- Equipo científico para investigaciones.
- Arduinome: Un dispositivo controlador MIDI.
- OBDuino: un económetro que usa una interfaz de diagnóstico a bordo que se halla en los automóviles modernos.
- Humane Reader: dispositivo electrónico de bajo coste con salida de señal de TV que puede manejar una biblioteca de 5000 títulos en una tarjeta microSD.
- The Humane PC: equipo que usa un módulo Arduino para emular un computador personal, con un monitor de televisión y un teclado para computadora.
- Ardupilot: software y hardware de aeronaves no tripuladas.
- ArduinoPhone: un teléfono móvil construido sobre un módulo Arduino.
- Impresoras 3D.

2.8.2. LENGUAJE DE PROGRAMACIÓN

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing que es similar a C++.

El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing).

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el popular lenguaje de programación de alto nivel Processing, basado en java (Arduino, 2015).

Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino. Algunos ejemplos son:

- JavaFlash (mediante ActionScript)
- ProcessingPure
- DataMaxMSP (entorno gráfico de programación para aplicaciones musicales, de audio y multimedia)
- VVVV (síntesis de vídeo en tiempo real)
- Adobe Director
- Python
- Ruby
- C
- C++ (mediante libSerial o en Windows)
- C#
- Cocoa/Objective-C (para Mac OS X)
- Linux TTY (terminales de Linux)
- 3DVIA Virtools (aplicaciones interactivas y de tiempo real)
- SuperCollider (síntesis de audio en tiempo real)
- Instant Reality (X3D)
- Liberlab (software de medición y experimentación)
- BlitzMax (con acceso restringido)
- Squeak (implementación libre de Smalltalk)
- Mathematica
- Matlab

- Minibloq (Entorno gráfico de programación, corre también en OLPC)
- Isadora (Interactividad audiovisual en tiempo real)
- PERL
- Visual Basic
- .NETVBScript

Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie, soportado por la mayoría de los lenguajes anteriormente citados. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida.

2.8.2.1. Funciones básicas y operadores

Arduino está basado en C y soporta todas las funciones del estándar C y algunas de C++. A continuación se muestra un resumen con la estructura y sintaxis del lenguaje Arduino:

- Sintaxis básica
 - Delimitadores:;, { }
 - Comentarios: //, /* */
 - Cabeceras: #define, #include
 - Operadores aritméticos: +, -, *, /, %
 - Asignación: =
 - Operadores de comparación: ==, !=, <, >, <=, >=
 - Operadores Booleanos: &&, ||, !
 - Operadores de acceso a punteros: *, &
 - Operadores de bits: &, |, ^, ~, <<, >>
 - Operadores compuestos:
 - Incremento y decremento de variables: ++, --
 - Asignación y operación: +=, -=, *=, /=, &=, |=

- Estructuras de control
 - Condicionales: if, if...else, switch case
 - Bucles: for, while, do. while
 - Bifurcaciones y saltos: break, continue, return, goto

- Variables

En cuanto al tratamiento de las variables también comparte un gran parecido con el lenguaje C.
- Constantes
 - HIGH/LOW: representan los niveles alto y bajo de las señales de entrada y salida. Los niveles altos son aquellos de 3 voltios o más.
 - INPUT/OUTPUT: entrada o salida.
 - false (falso): Señal que representa al cero lógico. A diferencia de las señales HIGH/LOW, su nombre se escribe en letra minúscula.
 - true (verdadero): Señal cuya definición es más amplia que la de false. Cualquier número entero diferente de cero es "verdadero", según el álgebra de Boole, como en el caso de -200, -1 o 1. Si es cero, es "falso".

- Tipos de datos

void, boolean, char, unsigned char, byte, int, unsigned int, word, long, unsigned long, float, double, string, array.

- Conversión entre tipos

Estas funciones reciben como argumento una variable de cualquier tipo y devuelven una variable convertida en el tipo deseado.

char(), byte(), int(), word(), long(), float()

- Cualificadores y ámbito de las variables

static, volatile, const

- Utilidades
 - sizeof()

- Funciones básicas
 - E/S digital
 - pinMode(pin, modo)
 - digitalWrite(pin, valor)
 - int digitalRead(pin)
 - E/S analógica
 - analogReference(tipo)
 - int analogRead(pin)
 - analogWrite(pin, valor)
 - E/S avanzada[editar]
 - shiftOut(dataPin, clockPin, bitOrder, valor)
 - unsigned long pulseIn(pin, valor)

- Tiempo
 - unsigned long millis()
 - unsigned long micros()
 - delay(ms)
 - delayMicroseconds(microsegundos)

- Matemáticas
 - min(x, y), max(x, y), abs(x), constrain(x, a, b), map(valor, fromLow, fromHigh, toLow, toHigh), pow(base, exponente), sqrt(x)

- Trigonometría
 - sin(rad), cos(rad), tan(rad)
 - Números aleatorios

randomSeed(semilla), long random(máx), long random(mín, máx)

- Bits y Bytes

lowByte(), highByte(), bitRead(), bitWrite(), bitSet(), bitClear(), bit()

- Interrupciones externas

attachInterrupt(interruptión, función, modo)

detachInterrupt(interruptión)

- Interrupciones

interrupts(), noInterrupts()

- Comunicación por puerto serie

Las funciones de manejo del puerto serie deben ir precedidas de la palabra "Serial" aunque no necesitan ninguna declaración en la cabecera del programa. Por esto se consideran funciones base del lenguaje. Estas son las funciones para transmisión serial:

begin(), available(), read(), flush(), print(), println(), write()

- Manipulación de puertos

Los registros de puertos permiten la manipulación a más bajo nivel y de forma más rápida de los contactos de entrada/salida del microcontrolador de las placas Arduino. Los contactos eléctricos de las placas Arduino están repartidos entre los registros B(0-7), C (analógicos) y D(8-13). Mediante estas variables se observado y modificado su estado:

- DDR[B/C/D]: Data Direction Register (o dirección del registro de datos) del puerto B, C ó D. Es una variable de Lectura/Escritura que

sirve para especificar cuales contactos serán usados como entrada y salida.

- PORT[B/C/D]: Data Register (o registro de datos) del puerto B, C ó D. Es una variable de Lectura/Escritura.
- PIN[B/C/D]: Input Pins Register (o registro de pines de entrada) del puerto B, C ó D. Variable de sólo lectura.

- AVR Libc

Los programas compilados con Arduino (salvo en las placas con CorteX M3) se enlazan contra AVR Libc por lo que tienen acceso a algunas de sus funciones. AVR Libc es un proyecto de software libre con el objetivo de proporcionar una biblioteca C de alta calidad para utilizarse con el compilador GCC sobre microcontroladores Atmel AVR. Se compone de 3 partes: avr-binutils, avr-gcc, avr-libc

La mayoría del lenguaje de programación Arduino está escrita con constantes y funciones de AVR y ciertas funcionalidades sólo se pueden obtener haciendo uso de AVR.

- Interrupciones

Las señales de interrupción son las siguientes:

- cli(): desactiva las interrupciones globales
- sei(): activa las interrupciones

- Temporizadores

La función `delayMicroseconds()` crea el menor retardo posible del lenguaje Arduino que ronda los 2µs. Para retardos más pequeños se debe utilizar la llamada de ensamblador 'nop' (no operación). Cada sentencia 'nop' se ejecutará en un ciclo de máquina (16 MHz) de aproximadamente 62,5ns.

- Manipulación de puertos

La manipulación de puertos con código AVR es más rápida que utilizar la función `digitalWrite()` de Arduino.

- Establecer Bits en variables

`cbi` y `sbi` son mecanismos estándar (AVR) para establecer o limpiar bits en PORT y otras variables.

2.8.3. Diferencias entre lenguajes principales de programación

La sintaxis del lenguaje de programación Arduino es una versión simplificada de C/C++ y tiene algunas diferencias respecto de Processing. Debido a que Arduino está basado en C/C++ mientras que Processing se basa en Java, existen varias diferencias en cuanto a la sintaxis de ambos lenguajes y el modo en que se programa, como se puede apreciar en la Tabla 2, Tabla 3 y Tabla 4.

Tabla 2. Arreglos

Arduino	Processing
<code>int bar[8];</code>	<code>int[] bar = new int[8];</code>
<code>bar[0] = 1;</code>	<code>bar[0] = 1;</code>
<code>int foo[] = { 0, 1, 2 };</code>	<code>int foo[] = { 0, 1, 2 };</code>
	o bien
	<code>int[] foo = { 0, 1, 2 };</code>

Fuente: (<https://es.wikipedia.org/wiki/Arduino>)

Tabla 3. Bucles

Arduino	Processing
int i; for (i = 0; i < 5; i++) { ... }	for (int i = 0; i < 5; i++) { ... }

Fuente: (<https://es.wikipedia.org/wiki/Arduino>)

Tabla 4. Impresión de cadenas

Arduino	Processing
Serial.println("hello world");	println("hello world");
int i = 5; Serial.println(i);	int i = 5; println(i);
int i = 5; Serial.print("i="); Serial.print(i); Serial.println();	int i = 5; println("i = " + i);

Fuente: (<https://es.wikipedia.org/wiki/Arduino>)

2.8.4. BIBLIOTECAS EN ARDUINO

Para hacer uso de una biblioteca, se añade el comando `#include` seguida del nombre de la biblioteca correspondiente. Las bibliotecas estándar que ofrece Arduino son las siguientes:

- Serial
Lectura y escritura por el puerto serie.
- EEPROM
Lectura y escritura en el almacenamiento permanente.
 - read(), write()
- Ethernet

Conexión a Internet mediante “Arduino Ethernet Shield“. Puede funcionar como servidor que acepta peticiones remotas o como cliente. Se permiten hasta cuatro conexiones simultáneas. Los comandos usados son los siguientes:

Servidor: `Server()`, `begin()`, `available()`, `write()`, `print()`, `println()`

Cliente: `Client()`, `connected()`, `connect()`, `write()`, `print()`, `println()`, `available()`, `read()`, `flush()`, `stop()`

- Firmata

Es una biblioteca de comunicación con aplicaciones informáticas utilizando el protocolo estándar del puerto serie.

- LiquidCrystal

Control de LCDs con chipset Hitachi HD44780 o compatibles. La biblioteca soporta los modos de 4 y 8 bits.

- Servo

Biblioteca para el control de servo-motores. A partir de la versión 0017 de Arduino la biblioteca soporta hasta 12 motores en la mayoría de las placas Arduino y 48 en la Arduino Mega. Estos son los comandos usados:
`attach()`, `write()`, `writeMicroseconds()`, `read()`, `attached()`, `detach()`

- SoftwareSerial

Comunicación serie en contactos digitales. Por defecto Arduino incluye comunicación sólo en los contactos 0 y 1 pero gracias a esta biblioteca puede realizarse esta comunicación con los restantes.

- Stepper

Control de motores paso a paso unipolares o bipolares.

Stepper (steps, pin1, pin2), Stepper(steps, pin1, pin2, pin3, pin4),
setSpeed(rpm), step(steps)

- Wire

Envío y recepción de datos sobre una red de dispositivos o sensores mediante
Two Wire Interface (TWI/I2C).

- Creación de bibliotecas

Los usuarios de Arduino tienen la posibilidad de escribir sus propias bibliotecas. Ello permite disponer de código que puede reutilizarse en otros proyectos, mantener el código fuente principal separado de las bibliotecas y la organización de los programas construidos es más clara.

2.8.5. ESTRUCTURA DE UN PROGRAMA EN ARDUINO

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

```
void setup() //Primera Parte
{
  //estamentos;
}
void loop() //Segunda Parte
{
  //estamentos;
```

```
}
```

En donde `setup()` es la parte encargada de recoger la configuración y `loop()` es la que contiene el programa que se ejecutará cíclicamente (de ahí el término `loop –bucle-`). Ambas funciones son necesarias para que el programa trabaje.

La función de configuración (`setup`) debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta sólo una vez, y se utiliza para configurar o inicializar `pinMode` (modo de trabajo de las E/S), configuración de la comunicación en serie y otras.

La función bucle (`loop`) siguiente contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc) Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

2.8.5.1. Void setup()

La función `setup()` se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. Así mismo se puede utilizar para establecer el estado inicial de las salidas de la placa.

Se indica cómo va la programación:

```
void setup()
{
  pinMode(pin, OUTPUT); // configura el 'pin' como salida
  digitalWrite(pin, HIGH); // pone el 'pin' en estado HIGH
}
```

2.8.5.2. Void loop()

Después de llamar a setup(), la función loop() hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan en la placa.

```
void loop()
{
  digitalWrite(pin, HIGH); // pone en uno (on, 5v) el 'pin'
  delay(1000);           // espera un segundo (1000 ms)
  digitalWrite(pin, LOW); // pone en cero (off, 0v.) el 'pin'
  delay(1000);
}
```

2.8.5.3. Funciones

Una función es un bloque de código que tiene un nombre y un conjunto de instrucciones que son ejecutadas cuando se llama a la función. Son funciones setup() y loop() de las que ya se ha hablado. Las funciones de usuario pueden ser escritas para realizar tareas repetitivas y para reducir el tamaño de un programa. Las funciones se declaran asociadas a un tipo de valor “type”. Este valor será el que devolverá la función, por ejemplo 'int' se utilizará cuando la función devuelve un dato numérico de tipo entero. Si la función no devuelve ningún valor entonces se colocará delante la palabra “void”, que significa “función vacía”. Después de declarar el tipo de dato que devuelve la función se debe escribir el nombre de la función y entre paréntesis se escribirán, si es necesario, los parámetros que se deben pasar a la función para que se ejecute.

```
type nombreFunción(parámetros)
{
  instrucción;
}
```

La función siguiente devuelve un número entero, delayVal() se utiliza para poner un valor de retraso en un programa que lee una variable analógica de un potenciómetro conectado a una entrada de Arduino. Al principio se declara como una variable local,

'v' recoge el valor leído del potenciómetro que estará comprendido entre 0 y 1023, luego se divide el valor por 4 para ajustarlo a un margen comprendido entre 0 y 255, finalmente se devuelve el valor 'v' y se retornaría al programa principal. Esta función cuando se ejecuta devuelve el valor de tipo entero 'v'.

```
int delayVal()
{
int v;          // crea una variable temporal 'v'
v= analogRead(pot); // lee el valor del potenciómetro
v /= 4;        // convierte 0-1023 a 0-255
return v;      // devuelve el valor final

}
```

2.8.5.4. Entre llaves {}

Las llaves sirven para definir el principio y el final de un bloque de instrucciones. Se utilizan para los bloques de programación setup(), loop(), if., etc.

```
type funcion()
{
instrucciones;
}
```

Una llave de apertura “{” siempre debe ir seguida de una llave de cierre “}”, si no es así el programa dará errores.

El entorno de programación de Arduino incluye una herramienta de gran utilidad para comprobar el total de llaves. Sólo tienes que hacer click en el punto de inserción de una llave abierta e inmediatamente se marca el correspondiente cierre de ese bloque (llave cerrada).

2.8.5.5. Punto y Coma;

El punto y coma “;” se utiliza para separar instrucciones en el lenguaje de programación de Arduino. También se utiliza para separar elementos en una instrucción de tipo “bucle for”.

```
int x = 13; /* declara la variable 'x' como tipo entero de valor 13 */
```

Nota: Olvidaos de poner fin a una línea con un punto y coma o se producirá en un error de compilación. El texto de error puede ser obvio, y se referirá a la falta de una coma, o puede que no. Si se produce un error raro y de difícil detección lo primero que debemos hacer es comprobar que los puntos y comas están colocados al final de las instrucciones.

2.8.5.6. Bloque de comentarios /*... */

Los bloques de comentarios, o comentarios multi-línea son áreas de texto ignorados por el programa que se utilizan para las descripciones del código o comentarios que ayudan a comprender el programa. Comienzan con /* y terminan con */ y pueden abarcar varias líneas.

```
/* esto es un bloque de comentario no se debe olvidar cerrar los comentarios estos  
deben estar equilibrados */
```

Debido a que los comentarios son ignorados por el compilador y no ocupan espacio en la memoria de Arduino pueden ser utilizados con generosidad. También pueden utilizarse para "comentar" bloques de código con el propósito de anotar informaciones para depuración y hacerlo mas comprensible para cualquiera.

Nota: Dentro de una misma línea de un bloque de comentarios NO se puede escribir otro bloque de comentarios (usando /*..*/).

```
// línea de comentarios
```

Una línea de comentario empieza con // y terminan con la siguiente línea de código. Al igual que los comentarios de bloque, los de línea son ignoradas por el programa y no ocupan espacio en la memoria.

```
// esto es un comentario
```

Una línea de comentario se utiliza a menudo después de una instrucción, para proporcionar más información acerca de lo que hace ésta o para recordarla más adelante

2.8.6. ESTRUCTURA DE UN PROGRAMA EN PROCESSING

El código ejecutado en Processing es similar a la de Arduino, pero la sintaxis esta basada en el lenguaje de programación JAVA. Las siguientes son las funciones principales:

2.8.6.1. Void setup()

Esta función es llamada una sola vez cuando el programa inicia. Se usa para definir las propiedades iniciales del entorno. Un programa sólo puede contener una función.

setup().

```
void setup(){  
    // funciones, variables, medios  
}
```

- Funciones en setup()

Algunas funciones que suelen incluirse en setup():

- size(). Define las dimensiones de ancho y alto de la ventana donde se ejecutará el programa (en píxeles).
- background(). Color o imagen de fondo de la ventana
- fill(). Color de relleno de las formas dibujadas por el programa
- stroke(). Color del contorno de las formas
- strokeWeight(). Grosor de ese contorno

Por ejemplo:

```
void setup(){
  //dimensiones
  size(650, 100);
  //color del fondo
  background(#339966);
  //color del trazo en hexadecimal
  stroke(#ffffff);
  //grosor del trazo en píxeles
  strokeWeight(5);
  //declara la variable que contiene las coordenadas y
  float y = height/2;
  //y dibuja tres puntos
  point(width*0.25, y);
  point(width*0.5, y);

  point(width*0.75, y);
}
```

2.8.6.2. Void draw()

Inicialmente cuando el programa arranca hace el llamado una sola vez a la función **setup()** y se ejecuta el código contenido dentro de ella. Para crear una animación es necesario que al menos una parte de ese código se ejecute muchas veces, y que se produzca alguna variación en los datos cada vez que se ejecute.

Para esto se utiliza **draw()**, una función que se ejecuta constantemente (60 veces por segundo).

Si se requiere que los puntos se muevan sobre el fondo, es necesario además que al menos una de las coordenadas varíe a lo largo del tiempo. Por ejemplo, al conseguir que las coordenadas **y** tengan un valor cada vez mayor, se observa los puntos moverse hacia abajo.

```
void setup(){
  //dimensiones
  size(650, 100);
  //color del fondo
  background(#339966);
  //color del trazo en hexadecimal
  stroke(#ffffff);
  //grosor del trazo en píxeles
  strokeWeight(5);
}
float y = height/2; //declara la variable y

void draw(){
  //actualiza el fondo
  background(#339966);
```

```

//dibuja tres puntos
point(width*0.25, y);
point(width*0.5, y);
point(width*0.75, y);
//y ahora viene la magia: se añade 1 a la variable <strong>y</strong>
y=y+1;
//por último se reiniciar la animación cuando los puntos alcancen el borde
inferior
if (y > height){
    y=height/2;
}
}

```

2.8.7. ARDUINO LEONARDO.

Arduino Leonardo es el nuevo modelo de Arduino, Figura 9 . El Arduino Leonardo basa su funcionamiento en el chip ATmega32u4,

Figura 8, contiene todo lo necesario para empezar a usar el micro controlador, tan solo se conecta a un ordenador con un cable USB, alimentarlo con un adaptador AC-DC o conectarle una batería para empezar.

El microcontrolador Atmega32u4 cuenta con 20 pines de entradas/salidas digitales (de los cuales 7 se pueden utilizar como salidas PWM y 12 como entradas analógicas), un oscilador de cristal de 16 MHz, una conexión micro USB, un conector de alimentación, un puerto ICSP, y un botón de reset.

El puerto de comunicación USB es emulado, por tanto, deja el puerto serial hardware libre para la programación.

De esta forma ya no ocurren conflictos de programación mientras se tiene periféricos con puertos serial, conectados a la placa.

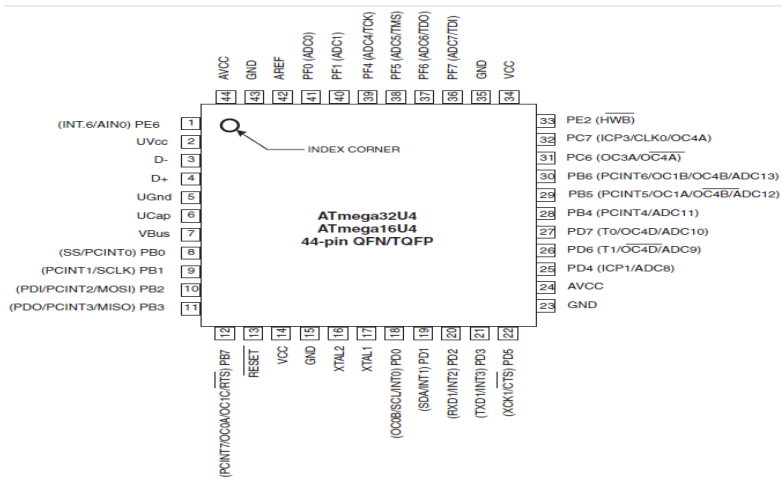


Figura 8. Microcontrolador Atmega32u4.

Fuente: <http://webdelcire.com/wordpress/archives/2905>

La placa de Arduino Leonardo posee las siguientes funcionalidades:

- Alimentación: Puede ser alimentado a través de la conexión micro USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La placa puede funcionar con una alimentación externa de 6 a 20 voltios. Si se suministra con menos de 7V, el pin de 5V puede suministrar menos de cinco voltios y la conexión puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- VIN. Voltaje de entrada, cuando se utiliza una fuente de alimentación externa se puede suministrar tensión a través de este pin.
 - 5V. La fuente de alimentación regulada utiliza para alimentar el microcontrolador y otros componentes de la placa. Esto puede venir de VIN a través de un regulador de la placa, o ser suministrada por USB o otra fuente de 5V regulada.
 - 3V3. Alimentación de 3,3 voltios generada por el regulador de la placa. La corriente máxima es de 50mA.
 - GND. Pin de tierra.
- Entrada y salida

Cada uno de los pines E / S digitales en la placa de Arduino Leonardo se puede utilizar como una entrada o salida, usando `pinMode ()`, `digitalWrite ()`, y funciones `digitalRead ()`. Operan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

- Serial: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) TTL, son datos en comunicación en serie utilizando la capacidad de serie del hardware Atmega32U4. Hay que tener en cuenta que en Leonardo, la clase de serie se refiere a USB (CDC) de comunicación; de la serie TTL en los pines 0 y 1, utilizando la función `Serial1`.
- TWI: 2 (SDA) y 3 (SCL). Comunicación TWI, utilizando la librería `Wire`.
- PWM: 3, 5, 6, 9, 10, 11, y 13. Proporcionar una salida PWM de 8 bits con la función `analogWrite ()`.

- LED: 13. Hay un LED incorporado conectado al pin digital 13. Al existir un valor lógico en, el LED está encendido, cuando el valor es bajo, está apagado.
- Entradas analógicas: A0-A5, A6 – A11 (en pines digitales de 4, 6, 8, 9, 10, y 12). El Arduino Leonardo tiene 12 entradas analógicas, a través de la etiqueta A0 A11, todos los cuales también se pueden utilizar como E / S digitales. Los pulsadores A0-A5 aparece en los mismos lugares que en el Arduino Uno; Entradas A6-A11 están en E / S digital pines 4, 6, 8, 9, 10 y 12 respectivamente. Cada entrada analógica proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto, la medida de las entradas analógicas de la tierra es a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el pin AREF y la función analogReference ().

En resumen la placa de Arduino Leonardo tiene las siguientes características:

- Micro controlador Atmega32u4
- 5V de funcionamiento
- 7-12V de entrada (recomendado)
- 6-20V de entrada (limite)
- 20 Pines E/S Digitales
- 7 Canales PWM
- 12 Entradas Analógicas
- 40mA de corriente máxima pines E/S
- 50mA de corriente en pin 3.3V



Figura 9. Arduino Leonardo.

Fuente: <https://www.arduino.cc/en/Main/ArduinoBoardLeonardo>

2.8.8. SHIELD MODEM GSM GPRS SMS SIM900

La placa GSM / GPRS es una placa basada en el módulo SIM900 de SIMCOM y compatible con Arduino, Figura 10, Figura 11.



Figura 10. Chip SIM900 de SIMCOM.

Fuente: SIM900_Hardware Design_V2.00

La Figura 12, muestra el diagrama funcional del SHIELD MODEM GSM/GPRS, con los siguientes elementos principales:

- El motor de banda base de GSM
- Flash
- La parte de radiofrecuencia de GSM
- La interfaz de la 54ravs54
- Otras interfaces

2.8.8.1. Fuente de alimentación

La alimentación del módulo está constituida con un regulador de voltaje de baja caída lineal, que permite conectar una gran variedad de fuentes de alimentación no regulada que oscila en el rango de 4,2V a 14V, siendo de 5 Voltios el estándar.

2.8.8.2. Interfaz de antena

El SHIELD MODEM GSM/GPRS posee una interfaz de antena de RF (Radio Frecuencia), que permite amplificar la señal para la comunicación con la red celular.

2.8.8.3. Interfaz serial

El módulo está diseñado con un circuito RS232 convertidor de nivel, que permite conectarse directamente a un puerto serial, la velocidad de transmisión se puede configurar desde los 1200 hasta 115200 baudios, siendo los más comunes: 1200, 2400, 4800, 9600, 38400, 57600, 115200 baudios.

El puerto serial contiene líneas de datos TXD y RXD, líneas de estado RTS y CTS, líneas de control DTR, DCD, DSR y RI, Tabla 5.

Tabla 5. Pines de la interfaz serial

Nombre	PIN	Función
DTR	3	Terminal de Datos Listo
RI	4	Ring Indicador
DCD	5	Datos de Detección de portadora
DSR	6	Datos Listos
CTS	7	Borrar para enviar
RTS	8	Solicitud de envío
TXD	9	Transmisión de Datos
RXD	10	Recepción de Datos

Fuente: Los Autores

El puerto serial puede ocuparse para CDS FAX, servicio GPRS, función de multiplexado, enviar comandos AT.

Para la comunicación el modulo solo necesita 3 hilos de conexión (TX, RX, GND). Usando este modem se podrá enviar y recibir mensajes de texto (SMS), utilizando comandos AT. Los comandos AT para configurar el puerto serial son: “AT+IPR” y “AT+IFC”.

2.8.8.4. Comandos AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal MODEM.

Los comandos AT surgen en el año de 1977, Dennis Hayes los desarrollo como una interfaz de comunicación con un modem para su configuración a través de instrucciones, como marcar un número de teléfono. Las compañías Microcomm y US Robotics expandieron su utilización hasta universalizarlo. Los comandos AT se denominan así por la abreviatura de attention.

Aunque la finalidad principal de los comandos AT es la comunicación de módems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico, que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar SMS, además de muchas otras opciones de configuración del terminal.

La configuración entre módems GSM, para pruebas o conexión, se realiza mediante un ordenador o microcontrolador por lo que no dispone de teclado y pantalla como se indicó anteriormente, para dicho fin los módems GSM pueden tener puertos RS232 o USB.

Los comandos AT están compuestos por cadenas de caracteres ASCII que para su ejecución se debe anteponer la palabra "AT" a excepción de los comandos de pause y de repetición de comando anterior, en los que no se requiere.

Para enviar comandos AT (configuración) a un modem GSM se debe seguir la siguiente estructura:

$$\begin{array}{ccc} AT & CMGF = 1 & < CR > \\ PREFIJO & COMANDO & SUFIJO \end{array}$$

La respuesta del modem ante un comando tiene la siguiente estructura:

$$\begin{array}{ccc} < CR >< LF > & OK & < CR >< LF > \\ PREFIJO & CÓDIGO & SUFIJO \end{array}$$

Los comandos que hacen que el módulo realice alguna acción se escriben de la siguiente manera:

$$\begin{array}{cc} AT & < X > \\ SUFIJO & ACCIÓN \end{array}$$

A continuación se detallan algunos comandos AT.

- Acción realizada por el modulo
 - ATH: Descuelga el teléfono
 - ATI : Revisa la memoria ROM del modulo
 - ATM: Conexión/Desconexión del altavoz
- Comandos para información del equipo
 - AT+CGMI: Identificación del fabricante.
 - AT+CGSN: Obtener número de serie.
 - AT+CIMI: Obtener el IMSI (Identificación de la Estación Móvil Internacional).
 - AT+CPAS: Leer estado del modem.
- Comandos del servicio de red
 - AT+CSQ: Obtener calidad de la señal.
 - AT+COPS: Selección de un operador.
 - AT+CREG: Registrarse en una red.
 - AT+WOPN: Leer nombre del operador.
- Comandos de seguridad
 - AT+CPIN: Introducir el PIN.
 - AT+CPINC: Obtener el número de reintentos que quedan.
 - AT+CPWD: Cambiar contraseña.
- Comandos para la agenda de teléfonos
 - AT+CPBR: Leer todas las entradas.
 - AT+CPBF: Encontrar una entrada.
 - AT+CPBW: Almacenar una entrada.
 - AT+CPBS: Buscar una entrada.
- Comandos para SMS
 - AT+CPMS: Seleccionar lugar de almacenamiento de los SMS.
 - AT+CMGF: Seleccionar formato de los SMS.
 - AT+CMGR: Leer un SMS almacenado.
 - AT+CMGL: Listar los mensajes almacenados.

- AT+CMGS: Enviar SMS.
- AT+CMGW: Almacenar mensaje en memoria.
- AT+CMSS: Enviar mensaje almacenado.
- AT+CSCA: Establecer el Centro de mensajes a usar.
- AT+ WMSC: Modificar el estado de un mensaje.
- Comandos utilizados:
 - ATE: Eliminar el ECO, retornan los datos sin en el comando recibido.
 - AT: Atención
 - ATH: Cuelga Llamada
 - AT+CMGL: Lista los Mensajes Almacenados
 - AT+CMGS: Enviar mensaje SMS.
 - AT+CMGD: Borra mensaje SMS.

En resumen el módulo Shield Modem Gsm Gprs Sms SIM900 tiene las siguientes características, Figura 13, Figura 14:

- Dimensiones: 24mm x 24mm x 3mm
- Cuatri Banda 850/ 900/ 1800/ 1900 MHz
- Peso: 3.4gramos
- Control vía comandos AT (GSM 07.07 ,07.05 y comandos AT SIMCOM)
- Rango de Alimentación: 4.2... 14VDC.
- Bajo consumo de Energía: 1.5mA (sleep mode)
- Temperatura de operación: -40°C a +85 °C
- Interfaz a SIM externa de 3V/1.8V
- Interfaz de Audio Analógico.
- RTC backup
- Interfaz SPI (opcional)
- Interfaz Serial
- Pad de Antena
- Interfaz Celular de comandos AT

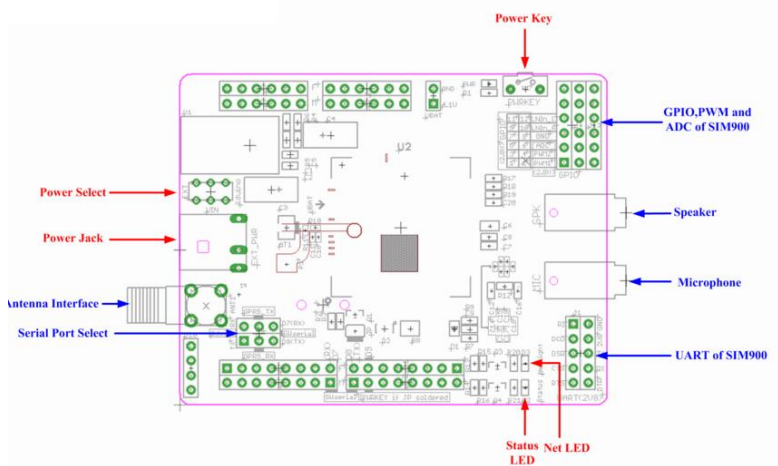


Figura 13. Diagrama esquemático del Shield Modem Gsm Gprs Sms Sim900.

Fuente: <https://developer.mbed.org/users/lawliet/notebook/gprs/>

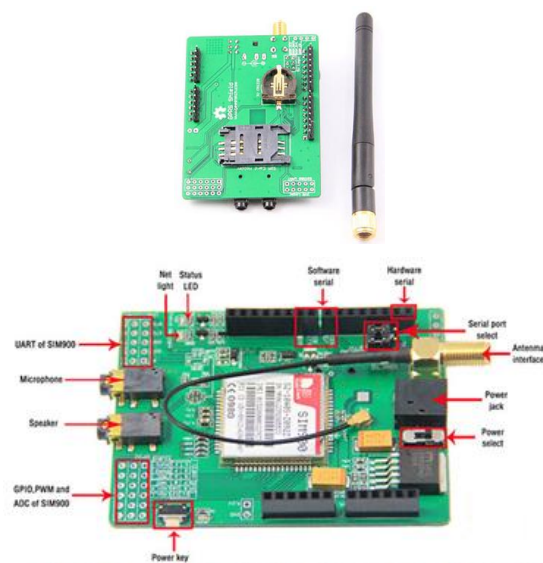


Figura 14. Shield Modem Gsm Gprs Sms Sim900.

Fuente: http://articulo.mercadolibre.com.ec/MEC-406959540-shield-modem-gsm-gprs-sms-sim900-ideal-arduino-uno-mega-_JM

2.9. HIPÓTESIS

El Diseño e Implementación de un Sistema de Comunicación Integrando Tecnologías para el Mejoramiento de Voz IP permitirá acceder a la comunicación de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo.

2.10. IDENTIFICACIÓN DE VARIABLES

- Diseño E Implementación De Un Sistema De Comunicación Integrando Tecnologías Para El Mejoramiento De Voz IP Interna De La Facultad De Ingeniería De La Universidad Nacional De Chimborazo
- Ofrecer un mejor servicio para el acceso de la información a los docentes, estudiantes y ciudadanía en general.

CAPÍTULO III

3. ESTUDIO DE COBERTURA

Previamente antes de realizar el proyecto, se procedió a analizar la cobertura actual de la red inalámbrica y revisar si la ubicación de las diferentes estaciones base de esta red están bien ubicadas, además saber si la cobertura de la señal inalámbrica en sitios abiertos como son: pasillos y aulas cumplen con el objetivo de la Universidad al instalar este tipo de red.

Se describe el diagnóstico de la cobertura, y a su vez se proponen recomendaciones para mejorar el rendimiento de la red inalámbrica como por ejemplo; distribución de las frecuencias de los diferentes puntos de acceso, reubicación de los puntos de acceso o si las estaciones base utilizadas son óptimas para satisfacer las necesidades y así optimizar la cobertura de la red inalámbrica.

3.1. Distribución de las Plantas del Edificio de la Facultad de Ingeniería

Actualmente el edificio consta de tres plantas, dentro de las cuales se encuentran distribuidos los puntos de enlace de la siguiente manera tal como se muestra en la Tabla 6.

En un determinado espacio está ubicado un equipo que permite el acceso a la red de manera inalámbrica. Cada uno de estos equipos tiene sus propias características y configuraciones.

Según la marca y el modelo que posee se define el rango de cobertura.

Tabla 6. Bandas de frecuencia utilizadas por redes inalámbricas

ESTACION BASE	UBICACIÓN	NOMBRE A.P	802.11	MARCA	REFERENCIA	MAC
1	1 PISO LADO A	Aula 3 de Ingles, baños, Centro de Idiomas, laboratorio de Idiomas y Aula de 7 semestre de Civil.	b.g	CISCO SYSTEMS.INC	MGT- (TKIP 1 CCMP)	00:24:9 7:83:2: 30
2	1 PISO LADO B	Aula 2 Centro de Idiomas, aula de Civli Primer año, civil segundo año, Aula 1 de Idiomas	b.g	CISCO SYSTEMS.INC	MGT- (TKIP 1 CCMP)	00:24:9 7:B9:3 E:91
3	2 PISO LADO A	Sala multimedia,baños, sala multimedia, auditorio, sala multimedia	b.g.n	TP-LINK TECHNOLOGIES CO.LTD	PSK-TKIP	00:23: CD:CO :D5:2B
4	2 PISO LADO B	Direcciones de escuela, Decanato, Vicedecanato	b.g	CISCO SYSTEMS.INC	MGT- (TKIP 1 CCMP)	00:24:9 7:B9:4 0:A3
5	3 PISO LADO A	AULAS Y LABORATORIOS DE ELECTRONICA Y SISTEMAS	b.g.n	CISCO	MGT- (TKIP 1 CCMP)	CC:D5: 39:9F: CD:B1
6	3 PISO LADO B	AULAS Y LABORATORIOS DE ELETRONICA Y SISTEMAS	b.g.n	CISCO	MGT- (TKIP 1 CCMP)	CC:D5: 39:9F: AD:FO

Comentado [D1]: Hagan más pequeña la letra en esta fila, el ing. Peñafiel dijo que debe estar completa la palabra....

Fuente: Los Autores

3.2. Software De Medición De La Cobertura De La Red Inalámbrica

A continuación se describirán las características más importantes del software Acrylic WI-FI HeadMaps utilizado para estas mediciones. Este software catalogado como free-ware, es decir de uso libre no requiere de la compra de la licencia alguna para su utilización.

- ¿Qué es Acrylic WI-FI HeadMaps?
 - Es un software para sistema operativo Windows XP que facilita la detección de redes inalámbricas WLAN dentro de su área de recepción, a través de tarjetas de red que operen con el estándar 802.11 (WI-FI) y trabaje con los protocolos 802.11a, 802.11b y 802.11g.

- Las aplicaciones que se puede desarrollar a través de este software son:
 - Comprobar las redes WLAN presentes
 - Analizar la cobertura de cada red y la intensidad de señal.
 - Determinar la ubicación de puntos de acceso así como también su marca y configuración.
 - Permite conocer que estaciones base se encuentran abiertas o con seguridad WEP, WAP, WAP2.

- Descripción de características de Acrylic WI-FI HeadMaps
 - Se ejecuta el software y a continuación se procede a realizar las mediciones de los ambientes mencionados, la imagen que se muestra es como la Figura 15 de inicialización del programa.



Figura 15. Inicialización del software ACRYLIC WIFI-HEATMAPS.

Fuente: Los Autores

Como siguiente paso se procede a buscar cuales son las redes disponibles en el ambiente para poder determinar el nivel de recepción de internet de cada red presente.

La pantalla principal del software es la mostrada en la Figura 16 la misma que muestra cada una de las aplicaciones a desarrollarse con la ayuda del mismo.

Este software catalogado como free-ware, es decir de uso libre no requiere de la compra de la licencia alguna para su utilización.

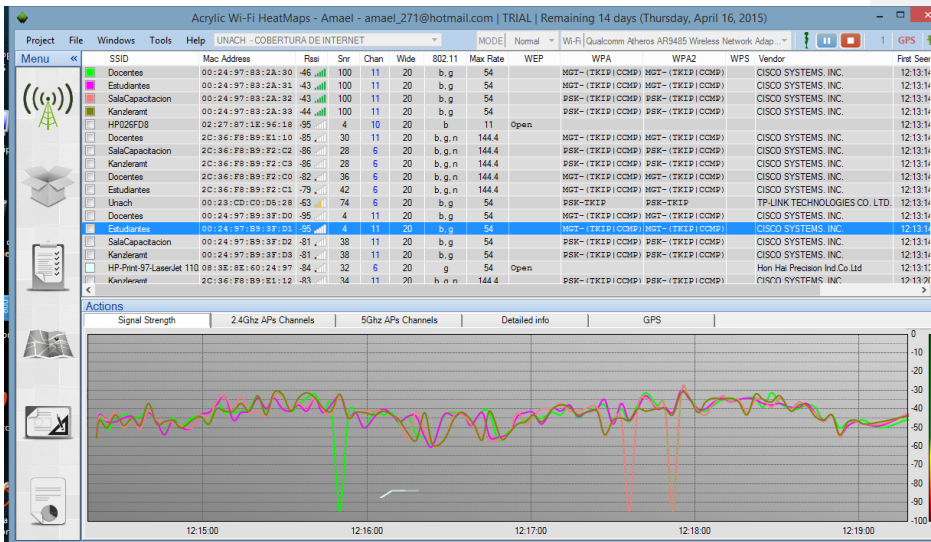


Figura 16. Aplicaciones con Acrylic WI-FI HeadMaps.

Fuente: Los Autores

Al observar la ventana principal del programa, a medida que se va escaneando el entorno genera un listado de las redes inalámbricas que va encontrando y se van completando espacios correspondientes a mediciones de parámetros que detallan la calidad de la señal.

ICONO: El cual aparece en la columna izquierda, indica detalladamente el análisis de cada una de las redes disponibles en ese momento, Figura 17.



Figura 17. Icono que detalla las redes presentes.

Fuente: Los Autores

A continuación se detalla cada uno de los parámetros encontrados en esta ventana.

- SSID: Identificación de la red, mediante el nombre de la misma.
- MAC: Dirección MAC física de la estación base.
- RSSI: Nivel de recepción de la señal en dbm
- SNR: Signal Noise Ratio. Indica la relación actual entre el nivel de la señal y ruidos de cada red.
- CHA: Indica el canal o canales utilizados por la red.
- Wide: Ancho de banda
- 802.11: Protocolo utilizado dentro del 66raves66a 802.11
- Max Rate: Velocidad máxima
- WPA: Tipo de seguridad
- FIRST SEEN: Indica la hora en la cual la red fue encontrada por primera vez.
- LAST SEEN: Indica la hora en la cual la red fue encontrada por última vez.

3.3. Gráficas obtenidas mediante la utilización del software

En la parte inferior de la pantalla principal del programa se puede visualizar cada una de las gráficas correspondientes a las señales presentes en las cuales se va graficando el nivel de la señal y como va variando en el transcurso del tiempo, así como también los canales que ocupan estas señales de la misma manera a medida que se va escaneando el entorno genera un listado de las redes inalámbricas que va encontrando y se van completando espacios correspondientes a mediciones de parámetros que detallan la calidad de la señal. Por cada piso se realizaron diferentes medidas de acuerdo al área de los espacios y a su infraestructura, la toma de muestras se la realizó mediante el software Acrylic Wi-Fi Heatmaps descrito anteriormente, el mismo que fue instalado en el equipo portátil como se muestran en las Figura 18, Figura 19.

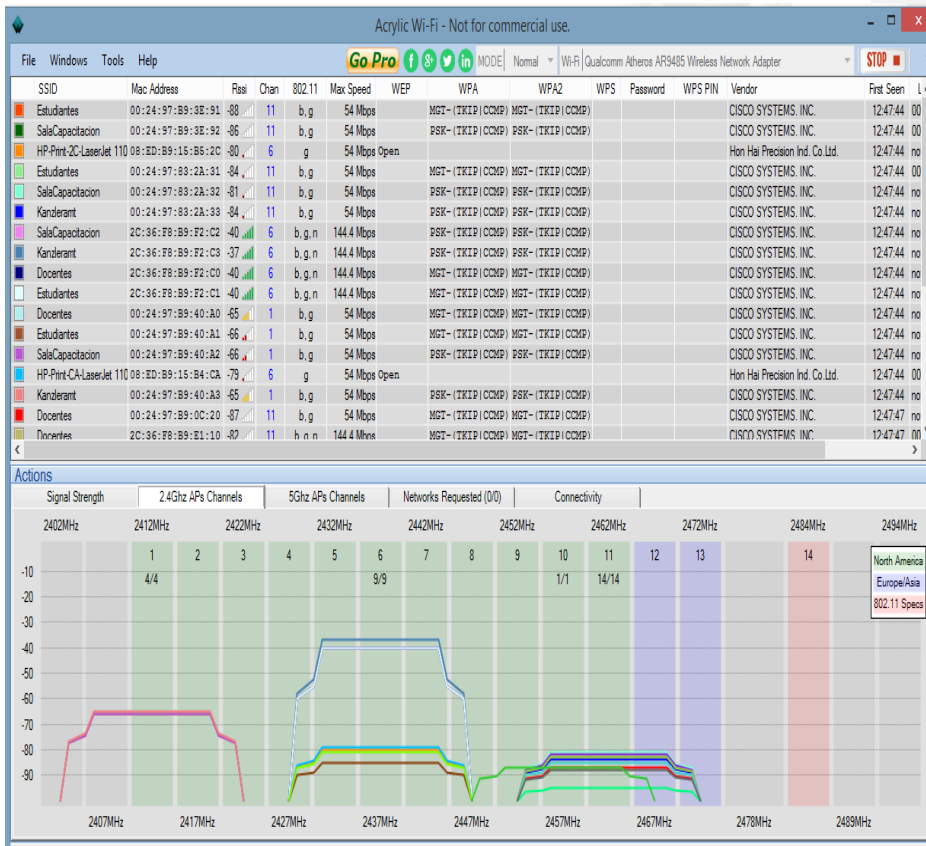


Figura 18. Canales utilizados por la red.

Fuente: Los Autores

Parámetros encontrados en la figura 18.

- 802.11: Protocolo utilizado dentro del 67ravés67a 802.11
- Max Rate: Velocidad máxima
- WPA: Tipo de seguridad
- FIRST SEEN: Indica la hora en la cual la red fue encontrada por primera vez.
- LAST SEEN: Indica la hora en la cual la red fue encontrada por última vez.

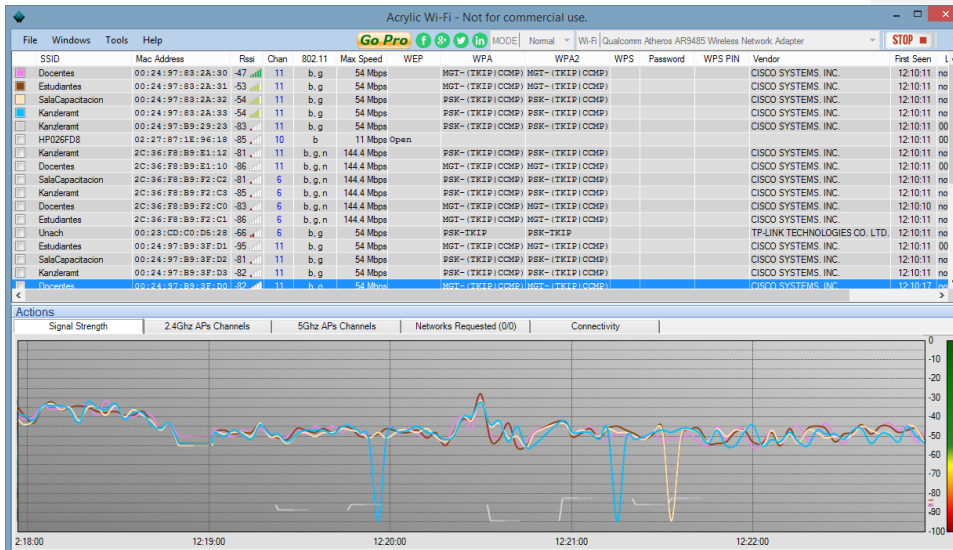


Figura 19. Variación de la señal en el transcurso del tiempo.

Fuente: Los Autores

3.4. Adquisición de datos de señal de cobertura

En este apartado se registrarán los datos tomados de la cobertura de la señal inalámbrica de los diferentes ambientes dentro del edificio de Ingeniería. La toma de muestras se la realizó mediante el software Acrylic Wi-Fi Heatmaps descrito anteriormente, el mismo que fue instalado en el equipo portátil.

3.4.1. Procedimiento De Recolección De Datos

El procedimiento de recolección de datos se llevó a cabo de la siguiente manera:

El punto de partida es la planta baja del edificio de Ingeniería realizando un recorrido por cada uno de los espacios abiertos como pasillos y cerrados como lo son aulas y laboratorios.

Se tomaron 18 muestras las mismas que envían distintas medidas de intensidad de señal de las diferentes plantas que posee el edificio. El primer paso fue adquirir los planos arquitectónicos de las diferentes plantas que conforman el edificio de Ingeniería, posteriormente se ubicó los puntos de acceso y se los gráfico. Por cada piso se realizaron diferentes medidas de acuerdo al área de los espacios y a su infraestructura.

3.4.2. Tabulación De Los Datos Recolectados

A continuación se realiza una descripción de los datos de señal inalámbrica tomados para cada uno de los puntos de medición, especificando la dirección de MAC, la intensidad de la señal inalámbrica dada en dBm, nombre de la red, ancho de banda, velocidad relación señal ruido, entre otras, el resultado se muestra en la Tabla 7.

- AREA 1: Aula 3, Baños, centro de idiomas, Laboratorio A-P1 A, AP1-7, aula de 7 semestre de Ingeniería Civil.(primera planta LADO A)
- AREA 2: Descanso primera planta
- AREA 3: Centro de idiomas aula 2, primer año civil, segundo año civil,Ap1-2 Civil, centro de idiomas aula 1. (primera planta LADO B).
- AREA 4: Sala multimedia, baños, sala multimedia2, auditorio de ingeniería, Sala multimedia 3. (segunda planta LADO A).
- AREA 5: Descanso segunda planta
- AREA 6: Secretarías de escuela y decanato.
- AREA 7: Descanso tercera planta
- AREA 8: Laboratorios de electrónica
- AREA 9: Aulas de electrónica

Tabla 7. Medición del nivel de calidad de señal

ESPACIO	SSID	MAC ADDRESS	RSSI	CHANEL	802.11	Max Rate	EQUIPO
AREA1	DOCENTES	00:24:97:83:2 ^a :30	-46	11	b.g	54	CYSCO SYSTEMS
AREA 1	ESTUDIANTES	00:24:97:83:2 ^a :31	-43	11	b.g	54	CYSCO SYSTEMS
AREA 1	SALA DE CAPACITACION	00:24:97:83:2 ^a :32	-43	11	b.g	54	CYSCO SYSTEMS
AREA 2	SALA DE CAPACITACION	2C:36:F8:B9:F2:3C2	-78	6	b.g.n.	144.4	CYSCO SYSTEMS
AREA 2	DOCENTES	2C:36:F8:B9:F2:3C0	-79	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 2	ESTUDIANTES	2C:36:F8:B9:F2:3C1	-82	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 3	DOCENTES	00:24:97:83:2 ^a :30	-81	11	b.g	54	CYSCO SYSTEMS
AREA 3	ESTUDIANTES	00:24:97:83:2 ^a :31	-79	11	b.g	54	CYSCO SYSTEMS
AREA 3	SALA DE CAPACITACION	00:24:97:83:2 ^a :32	-79	11	b.g	54	CYSCO SYSTEMS
AREA 4	DOCENTES	CC:D5:39:9F:AC:F0	-82	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 4	SALA DE CAPACITACION	CC:D5:39:9F:AC:F0	-86	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 4	ESTUDIANTES	00:24:97:83:2 ^a :31	-74	11	b.g.n	54	CYSCO SYSTEMS
AREA 5	ESTUDIANTES	CC:D5:39:9F:AC:F1	-80	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 5	DOCENTES	CC:D5:39:9F:AC:F0	-81	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 5	UNACH	00:23:CD:CO:D5:28	-73	6	b.g.	54	TP LIINK TECHNOLOGIES
AREA 6	ESTUDIANTES	CC:D5:39:9F:AC:F1	-95	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 6	DOCENTES	CC:D5:39:9F:AC:F0	-80	6	b.g.n	144.4	CYSCO SYSTEMS
AREA 6	UNACH	00:23:CD:CO:D5:28	-83	6	b.g	54	TP LIINK TECHNOLOGI
AREA 7	DOCENTES	00:24:97:97:B9OC:20	-68	11	b.g	54	CYSCO SYSTEMS
AREA 7	ESTUDIANTES	00:24:97:97:B9OC:21	-67	11	b.g	54	cisco
AREA 8	ESTUDIANTES	CC:D5:39:97:AC:F1	-75	6	b.g.n	144.4	cisco
AREA 8	DOCENTES	CC:D5:39:97:AC:F0	-86	6	b.g.n	144.4	cisco
AREA 9	ESTUDIANTES	CC:D5:39:97:AC:F1	-80	6	b.g.n	144.4	Cisco
AREA 9	DOCENTES	CC:D5:39:97:AC:F0	-80	6	b.g.n	144.4	cisco

Fuente: Los Autores

Comentado [D2]: Hagan más pequeña la letra en esta fila, el ing. Peñafiel dijo que debe estar completa la palabra....

3.5. Cobertura De La Señal Inalámbrica De Cada Punto De Medición

A continuación se graficarán en los planos arquitectónicos de la Universidad Nacional de Chimborazo Facultad de Ingeniería, con diferentes colores de los resultados obtenidos de las mediciones de la señal de la cobertura inalámbrica realizada y cuyos resultados quedaron plasmados anteriormente, de cada medición realizada los valores (S/N) arrojados para determinar la cobertura oscilan entre -41 dBm y -95 dBm. Teniendo en cuenta que entre más éste valor se acerque a -100dBm es más débil la

Comentado [D3]: Cambien por dBm

señal y por lo contrario más fuerte el ruido, una de las causas de este efecto es que la distancia que existe entre el dispositivo y el punto de acceso sea muy grande y este fuera del rango ideal alcance para una buena y estable conexión.

Además se graficará la calidad de la señal obtenida de las diferentes muestras de la siguiente manera: Excelente COMUNICACIÓN, buena (B), regular COMUNICACIÓN y mala (M), de los colores verde, amarillo, anaranjado y rojo respectivamente.

Los rangos asignados para determinar la calidad o nivel de señal este determinan por la Tabla 8.

Tabla 8. Rangos de valor de señal tomado en dBm

Valor de la SEÑAL	Rango	Color
-41 dBm a -55 dBm	Excelente	Verde
-56 dBm a -65 dBm	Buena	Amarillo
-66 dBm a -75 dBm	Regular	Anaranjado
-76dBm a -100dBm	Mala	Rojo

Fuente: Los Autores

3.5.1. Análisis de cobertura de la red inalámbrica

El análisis va hacer enfocado en los sitios de más concurrencia por la comunidad universitaria como son las aulas, los laboratorios y pasillos de acceso. De acuerdo a las mediciones y a los resultados obtenidos en estas se logró determinar que la cobertura de la red inalámbrica en la Universidad Nacional de Chimborazo, Facultad de Ingeniería es irregular debido a que existen lugares en los cuales la señal es óptima y en otros en los cuales la señal decae constantemente.

Uno de los principales inconvenientes es que la velocidad de transmisión en los diferentes ambientes es variada ya que en algunos sitios tenemos una velocidad de

54Mbps y en otros hasta 144.4 Mbps, esto provoca que mientras más usuarios estén conectados a la red de 54Mbps la señal decaerá. Adicionalmente el nivel de la señal es alto en los lugares en donde se encuentran conectados los ACCESS POINT; teniendo en cuenta que cada planta se sub divide en tres partes, el lugar más cercano al punto de acceso tendrá mejor recepción de la señal.

El resultado de las mediciones en cada una de las áreas se puede apreciar en la Figura 20 hasta la Figura 28.

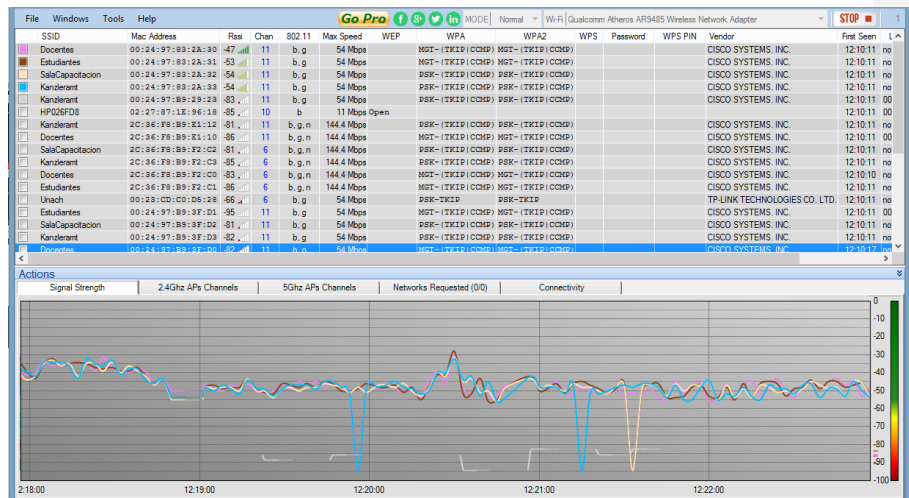


Figura 20. Medición Área 1

Fuente: Los Autores

Adicionalmente el nivel de la señal es alto en los lugares en donde se encuentran conectados los ACCESS POINT; teniendo en cuenta que cada planta se sub divide en tres partes, el lugar más cercano al punto de acceso tendrá mejor recepción de la señal como se muestra en la figura 20.

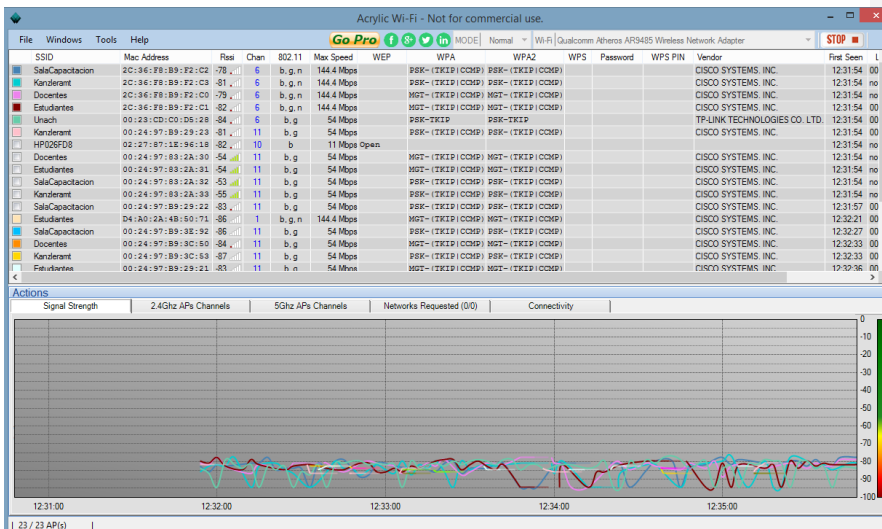


Figura 21. Medición Área 2

Fuente: Los Autores

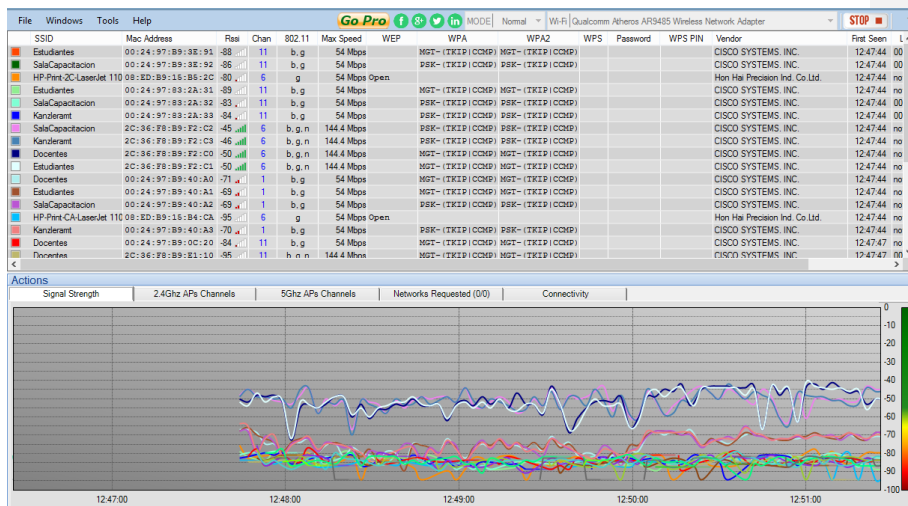


Figura 22. Medición Área 3

Fuente: Los Autores

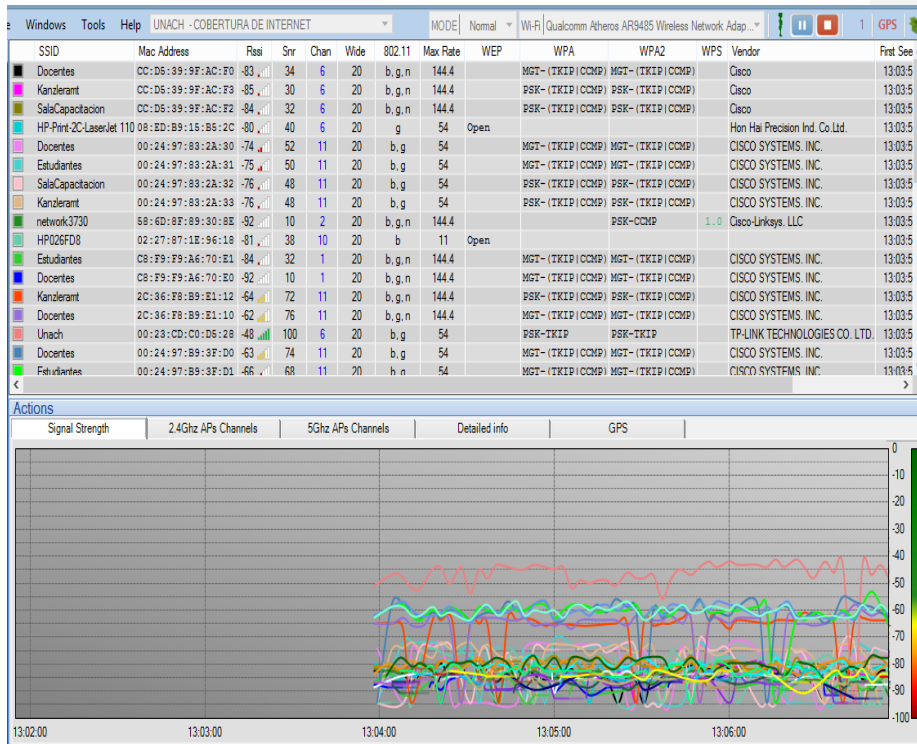


Figura 23. Medición Área 4

Fuente: Los Autores

Adicionalmente el nivel de la señal es alto en los lugares en donde se encuentran conectados los ACCESS POINT; teniendo en cuenta que cada planta se sub divide en tres partes, el lugar más cercano al punto de acceso tendrá mejor recepción de la señal como se muestra en la figura 23.

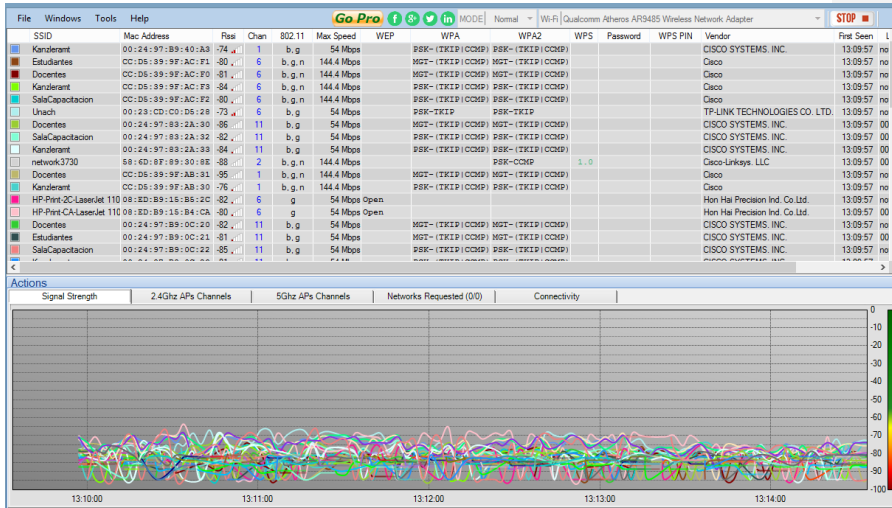


Figura 24. Medición Área 5

Fuente: Los Autores

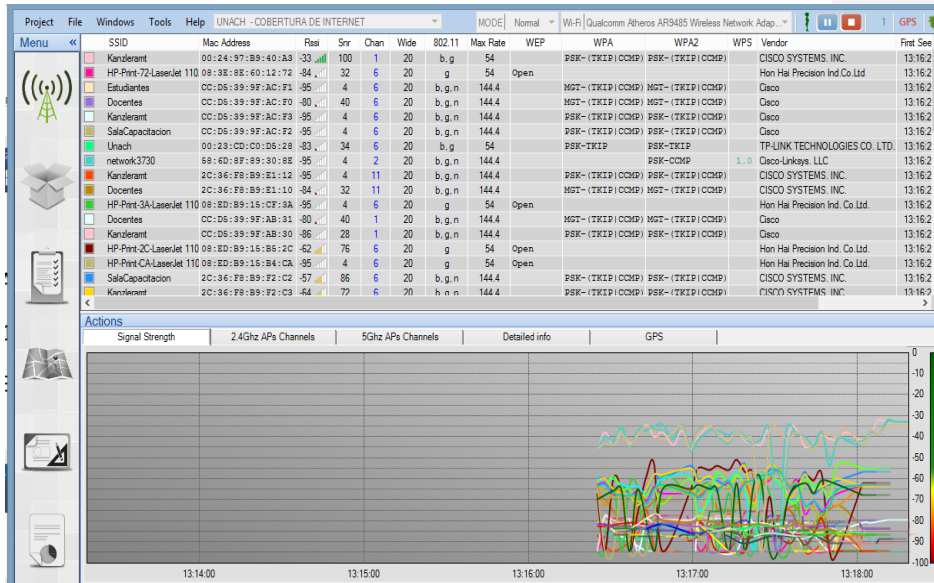


Figura 25. Medición Área 6

Fuente: Los Autores

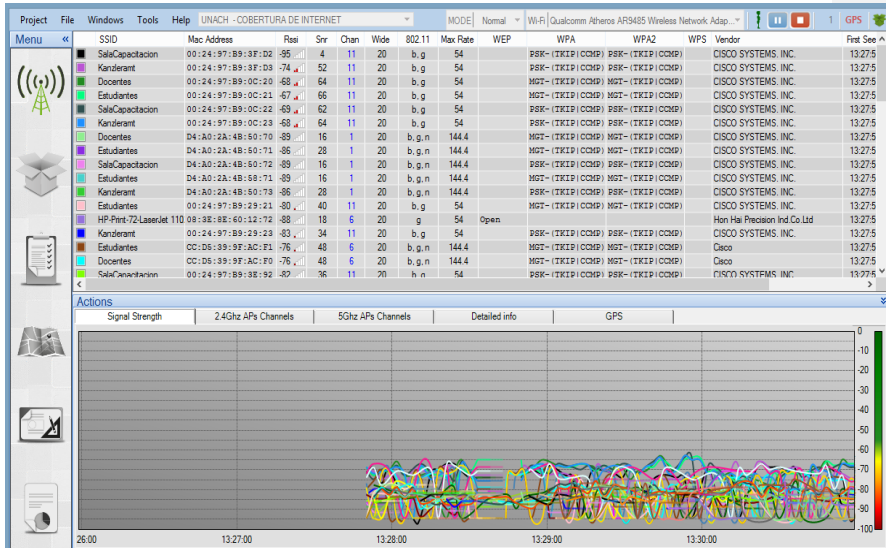


Figura 26. Medición Área 7

Fuente: Los Autores

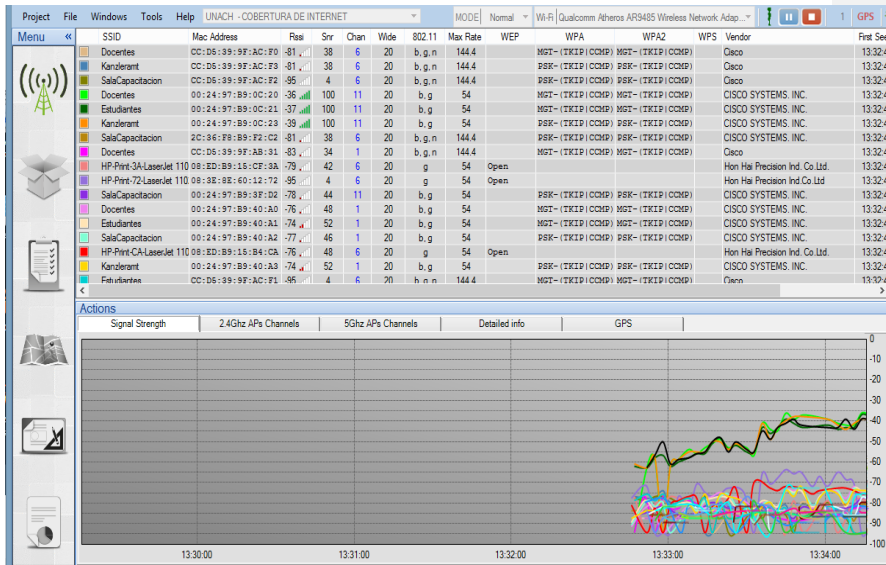


Figura 27. Medición Área 8

Fuente: Los Autores

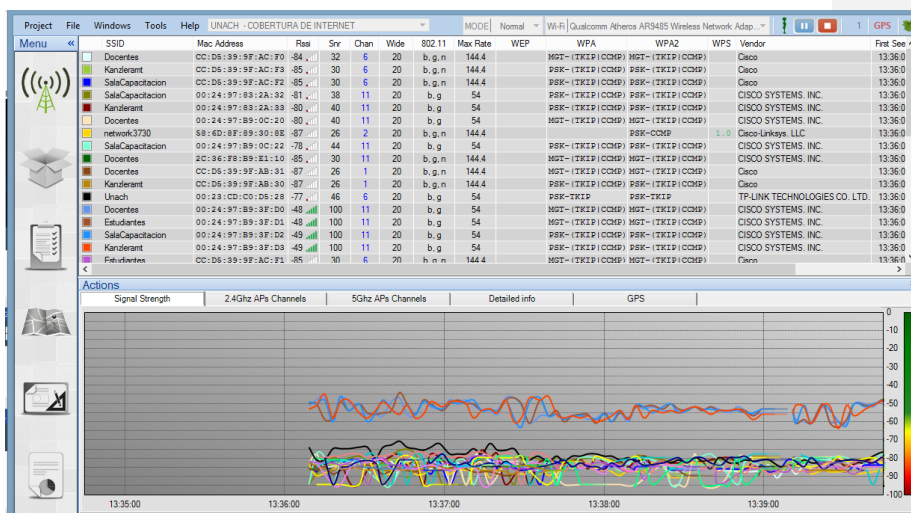


Figura 28. Medición Área 9

Fuente: Los Autores

Para resumir la cobertura de las distintas áreas se encuentra en la Tabla 9.

Tabla 9. Resumen cobertura de las distintas áreas

UBICACIÓN	CALIDAD DE LA SEÑAL	DESCRIPCIÓN
AREA 1	Excelente con -44dBm	La señal es óptima
AREA 2	Mala con -79 dBm	La señal decae constantemente
AREA 3	Mala con -80 dBm	La señal decae constantemente
AREA 4	Regular con -74 dBm	La señal se mantiene estable
AREA 5	Regular con -73 dBm	La señal se mantiene estable
AREA 6	Mala con -83 dBm	La señal decae constantemente
AREA 7	Regular con -67 dBm	La señal se mantiene estable
AREA 8	Regular con -75 dBm	La señal se mantiene estable
AREA 9	Mala con -80 dBm	La señal decae constantemente

Fuente: Los Autores

CAPITULO IV

4. MEJORAMIENTO DE LA RED INALÁMBRICA

Mediante el estudio de análisis de cobertura de la red inalámbrica del edificio de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo, (apartado anterior), se determina si la ubicación de las diferentes estaciones base de esta red están bien ubicadas, además saber si la cobertura de la señal inalámbrica en sitios abiertos como son: pasillos y aulas cumplen con el objetivo de la Universidad al instalar este tipo de red.

El análisis está enfocado en las áreas de más concurrencia por la comunidad universitaria como son las aulas, los laboratorios y pasillos de acceso.

Mediante un rango de colores se indicará la intensidad de la señal presente para una región determinada. El color verde indica un alto nivel de penetración y uso intensivo, mientras que el rojo corresponde a espacios donde el nivel no es óptimo, se puede apreciar en la Figura 29.

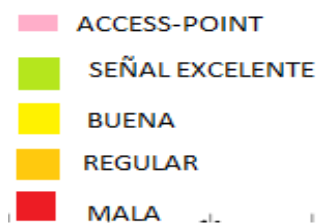


Figura 29. Rango de colores.

Fuente: Los Autores

Como resultado se obtiene el nivel de señal presente en las distintas áreas del edificio, se puede apreciar en la Figura 30, Figura 31, Figura 32. Lo cual permite determinar la calidad de la señal presente en cada parte del edificio de la Facultad de Ingeniería.

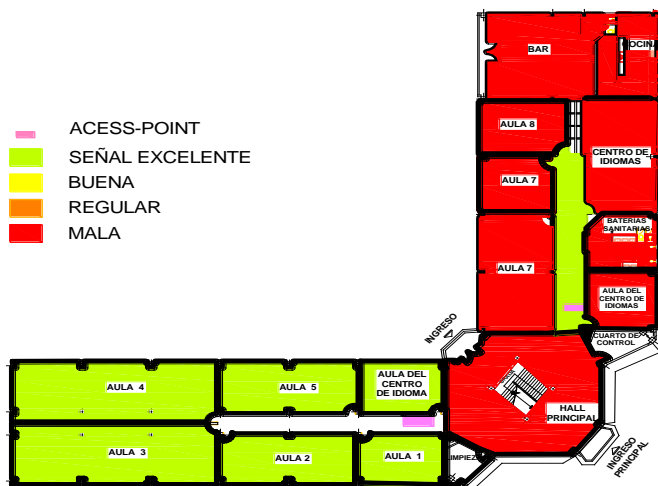


Figura 30. Medición De Señal De La Planta Baja

Fuente: Los Autores

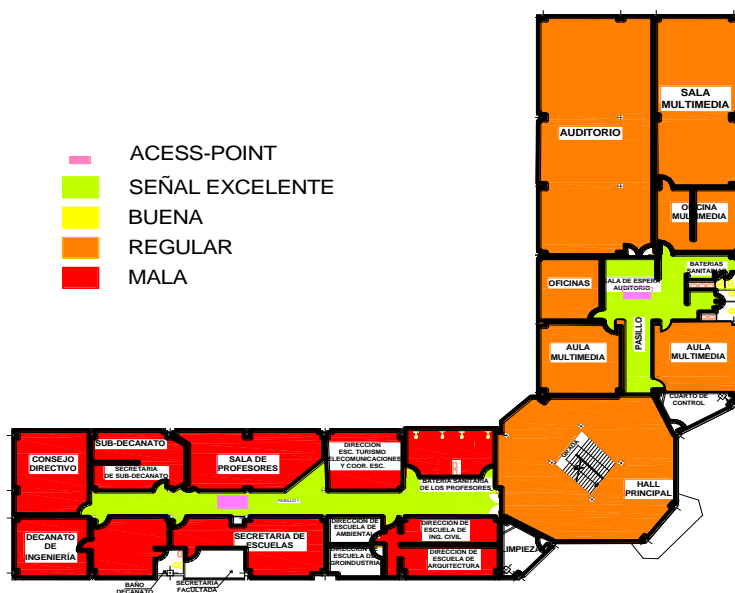


Figura 31. Medición de señal de la segunda planta.

Fuente: Los Autores

Comentado [D4]: Insisto arreglen las letras en el gráfico no se ve....
Van a tener problemas con los demás correctores...

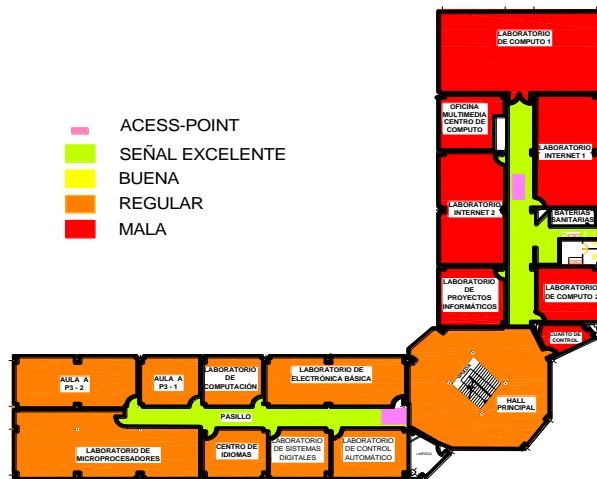


Figura 32. Medición De Señal De La Tercera Planta.
Fuente: Los Autores

Uno de los principales inconvenientes es que la velocidad de transmisión en los diferentes ambientes es variada ya que en algunos sitios tenemos una velocidad de 54Mbps y en otros hasta 144.4 Mbps, esto provoca que mientras más usuarios estén conectados a la red de 54Mbps la señal decaerá. Adicionalmente el nivel de la señal es alto en los lugares en donde se encuentran conectados los Access Point (AP); teniendo en cuenta que cada planta se sub divide en tres partes, el lugar más cercano al punto de acceso tendrá mejor recepción de la señal.

4.1.1. RECOMENDACIONES

Según lo anteriormente expuesto se determina que el nivel de señal en varios espacios no es eficiente razón por la cual se recomienda:

- Reubicación de los puntos de acceso de la primera planta a un lugar central en donde la señal pueda expandirse de lado y lado, el lugar que se cree acertado es en el descanso de la primera planta junto a las gradas; de no ser

posible esta solución se podría aumentar un punto de acceso para el lado B de la planta baja.

- En la segunda planta es conveniente instalar un nuevo punto de acceso en el descanso junto a las gradas.
- En la tercera planta habiendo aquí varias aulas y muchos estudiantes es recomendable instalar un punto de acceso en el lado B ya que solo se encuentra uno en el lado A o de otra manera se podría mover hacia el descanso de la tercera planta.
- Como solución a los problemas de calidad de recepción y mejorar la señal en la Facultad de Ingeniería se propone reemplazar los Access point marca LINKSYS por otros de más potencia (cisco systems), adicionalmente a esto trabajar ver la posibilidad de aumentar el ancho de banda y la velocidad de transmisión para que más usuarios puedan conectarse a las redes docentes y estudiantes sin inconvenientes.
- Con la adecuación de antenas directivas adaptando a los Access Point se puede dar servicio de datos en las áreas que la señal no es óptima.

CAPÍTULO V

5. METODOLOGÍA

5.1. MÉTODOS Y TÉCNICAS

Para realizar el proyecto la base se sustenta en los siguientes métodos y técnicas:

5.1.1. Método Analítico.

Esta metodología permite entender el funcionamiento de los APS de la red inalámbrica de la facultad, además de los componentes necesarios para el desarrollo del prototipo, conjuntamente con el software que manejará dicho sistema, y la forma de interactuar de cada uno de los componentes entre sí para permitir publicar la información en la pantalla LED.

5.1.2. Método Comparativo.

Este método permite conseguir información que ayudará a realizar las diferentes comparaciones para el mejoramiento de la red inalámbrica además del correcto funcionamiento del sistema.

5.1.3. Técnica de la Entrevistas.

Se utiliza esta técnica para conocer los criterios de los diferentes actores que serán los usuarios del sistema sobre la aplicación de esta tecnología.

5.2. OPERACIONALIZACIÓN DE LAS VARIABLES

En la Tabla 10, se asocia los variables, conceptos y técnicas que se utilizaron para la realización del proyecto.

Los principales elementos a ser tomados en cuenta son las variables, las cuales a través de los indicadores permitirá verificar el funcionamiento real de la aplicación, teniendo como resultado la verificación de la hipótesis propuesta. Para llevarla a cabo será necesaria la aplicación de algunas técnicas e instrumentos.

Tabla 10. Operación de Variables.

VARIABLES	CONCEPTO	INDICADORES	TÉCNICAS E INSTRUMENTOS
<p>Independiente</p> <p>El Diseño e Implementación de un Sistema de Comunicación Integrando Tecnologías para el Mejoramiento de Voz IP</p>	<p>Diseño e implementación sistema que administra la información publicada en una pantalla LED.</p>	<p>Informática Conexión Sistemas.</p>	<p>Programación en Arduino Programación en Processing</p>
<p>Dependiente</p> <p>Permitirá acceder a la comunicación de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo entre el personal.</p>	<p>Las personas pueden acceder de una mejor manera a la información</p>	<p>Veracidad Rapidez</p>	<p>Pantalla informativa</p>

Fuente: Los Autores

5.1. TIPO DE ESTUDIO

El tipo de estudio utilizado es teórico-práctico, vinculando la electrónica y el software necesarios para el correcto funcionamiento del sistema, utilizando la investigación bibliográfica referente al tema de estudio.

5.2. POBLACIÓN BENEFICIARIA

La población potencialmente usuaria de este sistema de información, serán los docentes, estudiantes, autoridades, personal administrativo y de servicio, así como el público en general que tenga acceso al campus universitario de la Unach.

5.3. PROCEDIMIENTOS

Para el desarrollo del sistema, se requirió seguir una serie de pasos y etapas, que en conjunto, cumplirían con el objetivo planteado. Cada etapa desempeña una función específica dentro del esquema global, y también sirve de apoyo para la siguiente etapa, haciendo del dispositivo implementado un sistema funcional, se puede referenciar estos procedimientos en la Figura 33.

La primera: se desarrolló en el edificio de la facultad un estudio de la cobertura actual de la red inalámbrica, en sitios abiertos y diferentes pasillos. Con ello se determinó el nivel de cobertura en los diferentes ambientes y analizó los problemas existentes de cobertura.

Segundo: para complementar el acceso a la información de la red inalámbrica se realiza el sistema que permita proyectar en una pantalla LED datos informativos. Además para posibilitar el envío de noticias o informaciones urgentes, es necesaria la interconexión de la red celular al servidor que almacena la información a ser proyectada.

Ante un evento o una noticia importante y urgente, como queda dicho, se enviaría vía SMS la información a ser proyectada en la pantalla LED. Para ello, se escogió dos módulos Arduino para tal efecto.

El módulo Arduino GSM / GPRS necesita una fuente de alimentación externa de 5 voltios. El módulo Arduino Leonardo se conecta a través de un cable USB al servidor, permitiendo el envío y recepción de la información.

Una vez desarrollado el código se compila y se carga en el módulo Arduino para su correcto funcionamiento.

La información está almacenada en un servidor que consta de una base de datos en mysql y apache que permite el acceso vía web.

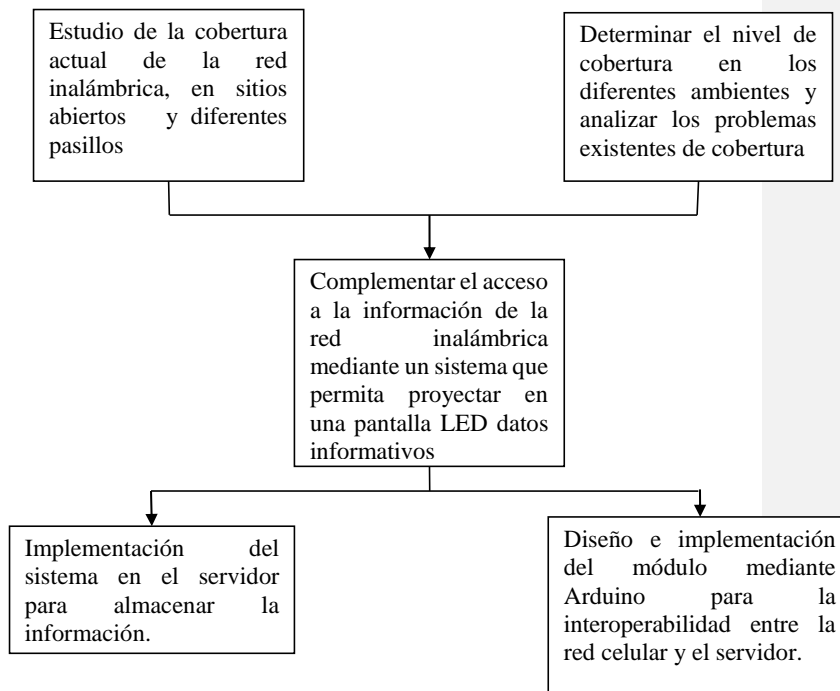


Figura 33. Diagrama de procedimientos.

Fuente: Los Autores

5.3.1. DISEÑO E IMPLEMENTACIÓN DE SOFTWARE

5.3.1.1. Programación en Arduino

La programación del módulo se la realiza en lenguaje de alto nivel, siendo este el más común, en este caso se necesita también un compilador para traducir el programa a lenguaje ensamblador. Para la programación del módulo Arduino se

empleó el software y los driver en la página oficial de Arduino. La pantalla principal se puede apreciar en la Figura 34.



Comentado [D5]: Traten que la imagen se ajuste al margen, se encuentra muy salida a la derecha...

Figura 34. Pantalla inicio Arduino Software.

Fuente: Los Autores

Para trabajar con el modelo Arduino Leonardo se necesita, seleccionar el modelo, además del puerto virtual serial para el envío y recepción de datos. En la Figura 35 y la Figura 36, se puede apreciar cómo realizarlo.

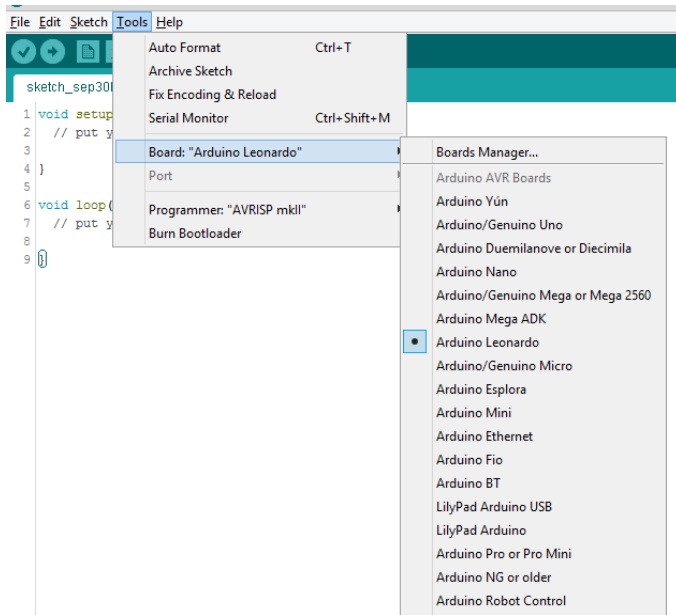


Figura 35. Selección del módulo Arduino Leonardo.

Fuente: Los autores

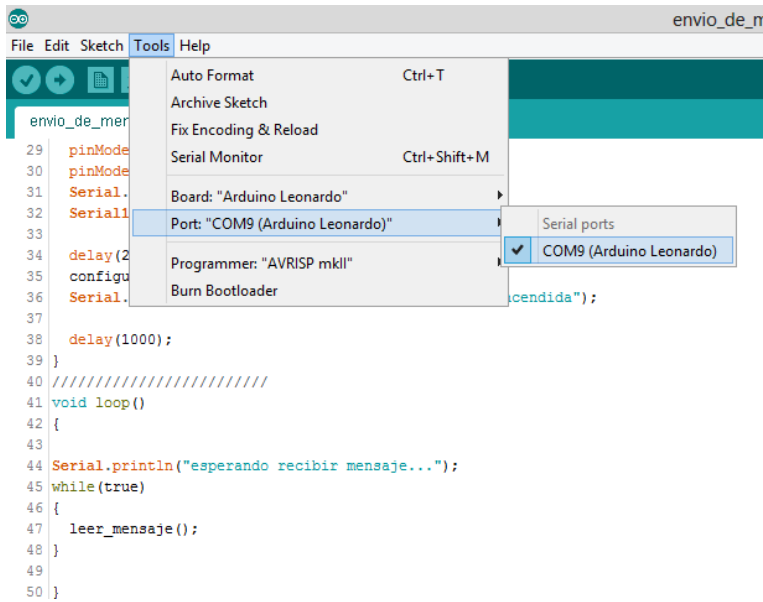


Figura 36. Selección del puerto virtual para conexión a la placa Arduino.

Fuente: Los autores

5.3.1.2. Código Comunicación con los módulos de Arduino

Para la programación en Arduino (Cisco Systems, 2006), se realizó una búsqueda exhaustiva a través de internet de programas similares desarrollados. El programa desarrollado por el ingeniero Edison Viveros, nos sirvió como referencia (Editronixx, 2015).

```
String numero_cell = "09792xxxxx";//meter numero de telefono
int pulsador =9;
int led =8;
boolean valor=true;
int i=0;
int j=0;
char DAT;
char DAT_dos;
boolean condicion=true;
char datosSERIAL[144];
char clave_uno[]={'U','N','A','C','H'};//AQUI CAMBIAMOS TODO EL MENSAJE DE CONTROL
```

Figura 37. Extracto código Arduino. Configuración inicial variables.

Fuente: Los autores

En la Figura 37, se puede observar la configuración inicial, siendo relevante el número de teléfono con el cuál se va a interactuar. El usuario de este móvil envía los mensajes al sistema para ser procesados. Se añade una clave para los envíos de SMS que tendría que enviar el remitente, siendo un mecanismo de seguridad para que los datos no lleguen alterados. El número de caracteres para la clave es de 5.

```
void configuracion_inicial();//configura los codigos de lectura de mensajes
{
  Serial1.println("configuracion inicial");
  Serial1.println("AT+IPR=9600");//modo texto
  Serial.println("AT+IPR=9600");//modo texto
  delay(300);
  Serial1.println("AT+CMGF=1");//modo texto
  Serial.println("AT+CMGF=1");//modo texto
  delay(300);
  Serial1.println("AT+CMGR=?");//ACTIVAMOS CODIGO PARA RECIBIR MENSAJES
  Serial.println("AT+CMGR=?");//ACTIVAMOS CODIGO PARA RECIBIR MENSAJES
  delay(300);
  Serial1.println("AT+CNMI=2,2,0,0");//ACTIVAMOS PARA VER MENSAJES
  Serial.println("AT+CNMI=2,2,0,0");//ACTIVAMOS PARA VER MENSAJES
  //Serial1.println("AT+CNMI=3,3,0,0");//ACTIVAMOS PARA VER MENSAJES
  //Serial.println("AT+CNMI=3,3,0,0");//ACTIVAMOS PARA VER MENSAJES
  delay(300);
  Serial.println("configuracion terminada");
  delay(300);
}
```

Figura 38. Extracto código Arduino. Configuración módulo Arduino.

Fuente: Los autores

```

void leer_mensaje()
{
  salir:
  if (Serial1.available()>0)
  {
    char DAT = Serial1.read();
    if(DAT=='@')//el arroba hace detectar el inicio del codigo
    //If arroba
    while(true)//ingresa en un while para leer solo los codigos futuros que estan por llegar despues de la arroba
    {
      if (Serial1.available() > 0)
      //cierre del segundo if
      char DAT_dos = Serial1.read();//@LEDON enter
      datosSERIAL[j]= DAT_dos;//almacena en cadena de caracteres, suma de caracteres
      j++;
      if (DAT_dos=='\n')//cuando termine de entregar todos los datos dara un enter
      {
        dato_recibido();//llama a la la funcion y verifica codigo
        delay(500);
        for(int i=0;i<=j;i++)
        {
          datosSERIAL[i] = 0;//borro array
          DAT_dos=0; DAT=0;
        }
        j=0;//borra el puntero o acumulador si no se hace esto no detecta los siguientes codigos
        goto salir;//sale de todos los ciclos y va al inicio para volver a leer codigo
      }//CIERRA AL /N
    } //cierre del segundo if
  } //while
} //arroba
} //serial available
}

```

Figura 39. Extracto código Arduino. Lectura de mensajes de la placa GSM/GPRS.

Fuente: Los autores

En la Figura 38, se puede observar la configuración del módulo Arduino Leonardo, para que permita enviar y recibir los datos con la tarjeta GSM/GPRS.

En la Figura 39, se puede observar el proceso para la lectura de los mensajes que provienen de la red GSM/GPRS, se activa un bucle para que monitoree constantemente el puerto Serial1, el cual permite una comunicación bidireccional entre las dos placas. Detecta el inicio del texto del SMS cuando lee una arroba (@), guardando los caracteres posteriores en una variable, para posteriormente verificar la clave, llamando a la función dato_recibido().

Para que el bucle no sea infinito, se reinicializa la variable datosSERIAL, para esperar la próxima información.


```

void dato_recibido()
{
  //////////////////////////////////////
  if(datosSERIAL[0]==clave_uno[0] && datosSERIAL[1]==clave_uno[1] &&
      datosSERIAL[2]==clave_uno[2] && datosSERIAL[3]==clave_uno[3]
      && datosSERIAL[4]==clave_uno[4])
  {
    int l = 144;
    //digitalWrite(led,HIGH);
    int k=0;
    char val[144];
    for(int i=4;i<=1;i++)
    {
      val[k]=(datosSERIAL[i]);
      k++;
    }
    Serial.print(val);
    mensaje = "Mensaje Recibido correctamente.";
  }
  else
  {
    mensaje = "Clave incorrecta.";
  }
  mensaje_recibido();
  //////////////////////////////////////
}

```

Figura 40. Extracto código Arduino. Comparación de la clave.

Fuente: Los autores

En la Figura 40, se puede observar el método utilizado para comprobar que los caracteres iniciales coincidan con la clave configurada en el sistema. Si el password es correcto se envía los datos a través del Serial, para ser administrados por el código de Processing, Si todo es correcto se envía un mensaje de respuesta al remitente por medio de la función mensaje recibido, Figura 41.

```

////////////////////////////////////ENVIO DE MENSAJES
void mensaje_recibido()
{
  Serial1.println("AT+CMGF=1");//modo texto
  //Imprime los datos al puerto serie como texto ASCII seguido de un retorno de carro
  delay(2000);
  Serial1.print("AT+CMGS=");// comando de envio de mensaje
  //Imprime los datos al puerto serie como texto ASCII
  Serial1.print((char)34);//ponemos las comillas ", para que lo tome debe ser char de lo contrario el serial envia caracter por caracter
  Serial1.print(numero_cell);//colocamos numero de telefono
  Serial1.println((char)34);//volvemos a poner el caracter "
  delay(200);//tiempo para que de respuesta el modulo >
  Serial1.print(mensaje);//mensaje que enviare
  Serial1.print((char)26);//ponemos el simbolo ascii 26, que corresponde a CTRL+Z, con lo que el modulo sabe que el sms termino

  //Serial.print("led uno activo ONN");//mensaje que enviare
}

```

Figura 41. Extracto código Arduino. Envío mensaje de respuesta al remitente.

Fuente: Los autores

Comentado [D6]: Ajusten al margen

5.3.1.3. Función principal de la PC que funciona como Servidor

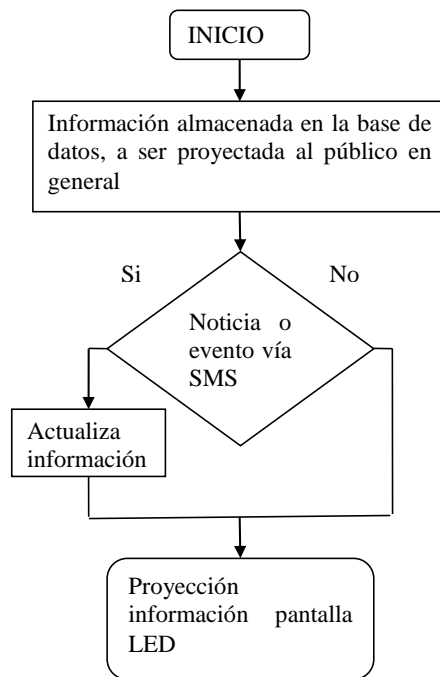


Figura 42. Funcionamiento servidor.

Fuente: Los Autores

El funcionamiento del servidor se basa en la Figura 42, en donde se pueden reconocer los siguientes procedimientos:

- La información se encuentra almacenada en un servidor, al cual se puede acceder mediante el programa desarrollado en base a Processing.
- El servidor estaría constantemente monitoreando las tarjetas conectadas a través del puerto USB.

- Al enviarse una noticia o evento importante vía SMS, se actualizaría la información a ser proyectada, la cual se mostraría posteriormente en la pantalla LED.

Para la comunicación entre el servidor y la pantalla LED, se realiza a través de un cable de video Vga A Vga, Figura 43, la distancia dependería de la ubicación de la pantalla. La información se muestra de forma dinámica a través de una página WEB.



Figura 43. Conector video Vga a Vga.

Fuente: <http://articulo.mercadolibre.com.mx/MLM-519238489-cable-vga-a-vga-10-metros-conectores-macho-macho-pc-y-laptop- JM>

5.3.1.4. Diseño de la Base de datos

A través de la herramienta wamp server se creó un servidor WEB provisional para brindar el acceso remoto. Por medio de la página WEB <http://localhost/phpmyadmin>, se puede acceder a su administración.

Para la configuración del servidor se crea una base de datos denominada: pantalla, en la cual se añade varias tablas para el almacenamiento de la información relevante del sistema, detalladas a continuación:

- Base de datos: pantalla
- Tablas:
 - noticia (id, titulo, texto1, texto2, texto3, fecha)

- email_fuente (id, nombre, email, clave)
- email_destino (id, nombre, email)
- intervalo (tiempo)
- Información a almacenar en las tablas
 - La tabla noticias, almacena la información a ser proyectada en la pantalla LED.
 - La tabla email_fuente, almacena el correo fuente del cual vamos a enviar la respuesta de confirmación de la recepción del mensaje SMS.
 - La tabla email_destino, contiene el correo del receptor del mensaje de confirmación.
 - La tabla intervalo almacena el tiempo en segundos, por medio de los cuales se puede rotar la información a ser proyectada en la pantalla LED.

5.3.1.5. Programación en Processing

Para permitir la administración de la información receptada por el módulo de Arduino se generó el código en processing, por medio del cual se realizó la aplicación.

Entre las principales funciones utilizadas para la programación en processing tenemos las siguientes:

- Librerías necesarias de JAVA para realizar la parte grafica de la aplicación PANTALLA, contiene el manejo de los menus, Textbox, Fechas. Descrito en la Figura 44.

```
import java.awt.Menu;
import java.awt.MenuBar;
import java.awt.MenuItem;
import java.awt.*;
import java.awt.event.*;
```

Figura 44. Librerías Processing.

Fuente: Los Autores

- Esta librería permite la comunicación con el puerto serial, en este caso permite recibir los datos enviados de la aplicación de Arduino.⁹⁴ Descrito en la Figura 45.

```
import processing.serial.*; //Importamos la librería Serial
```

Figura 45. Librerías Processing

Fuente: Los Autores

- Librería que permite la comunicación con cualquier base de datos, en nuestro caso es mysql. Descrito en la Figura 46.

```
import de.bezier.data.sql.*;
```

Figura 46. Librerías Processing

Fuente: Los Autores

- Estas librerías as permiten el enviar mensajes vía mail, en nuestro caso desde un correo de gmail preconfigurado hacia cualquier destinatario, por el momento solo uno. Descrito en la Figura 47.

```
import javax.mail.*;
import javax.mail.internet.*;
```

Figura 47. Librerías Processing

Fuente: Los Autores

- Declaración de las variables para los menús principales del programa, basados en el lenguaje JAVA. Descrito en la Figura 48 y Figura 49.

```
MenuBar myMenu;
Menu file, notice, configuration, screenControl, help;
MenuItem close,notice_list,notice_new,screenView,;
MenuItem mailIN, mailOUT, timeView, screenStart, screenStop, help1;
```

Figura 48. Variables para los menús y submenús

Fuente: Los Autores

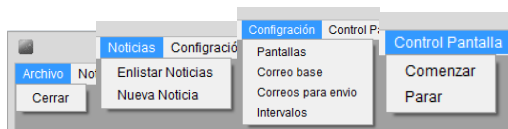


Figura 49. Menús y submenús principales de la aplicación

Fuente: Los Autores

- Declaración de las variables para el manejo de las fechas. Descrito en la Figura 50.

```
Date date = new Date();
DateFormat fecha = new SimpleDateFormat("yyyy/MM/dd");
String convertido = "";
convertido = fecha.format(date);
```

Figura 50. Variables para el manejo de fechas

Fuente: Los Autores

- Tipo de letra y color de la letra de la aplicación. Descrito en la Figura 51.

```
color bg = color(255);
PFont font = createFont("arial",20);
```

Figura 51. Variables color y tipo de letra

Fuente: Los Autores

- Declaración de la variable que permite la colocación del sello de la institución en la aplicación. Descrito en la Figura 52.

```
PImage img;
// Images must be in the "data" directory to load correctly
img = loadImage("images/unach.png");
image(img, 1050, 0);
```

Figura 52. Colocación imagen de la UNACH

Fuente: Los Autores

- Declaración de la variable para almacenar los datos provenientes del puerto serial. Descrito en la Figura 53.

```
String [] datosSERIAL;
```

Figura 53. Variable para almacenar los datos provenientes del puerto serial

Fuente: Los Autores

- Declaración de las variables para el manejo del tiempo de rotación entre cada una de las noticias. Descrito en la Figura 54.

```
int intervaloNoticia = 2000;//timer's interval
int lastRecordedTime = 0;
int counterNoticia = 0;
```

Figura 54. Variables para el manejo de la rotación de las noticias

Fuente: Los Autores

- Declaración de la variable que permite la comunicación con el Puerto serial. Descrito en la Figura 55.

```
Serial port; //Nombre del puerto serie
```

Figura 55. Variables para el manejo del puerto serial

Fuente: Los Autores

- Declaración de la variable que permite realizar consultas query hacia la base de datos MYSQL. Descrito en la Figura 56.

```
MySQL msqL;
```

Figura 56. Variables para el manejo de la conexión y consultas a la base de datos

Fuente: Los Autores

- Conjunto de variables para la parte grafica de la aplicación cuadros de texto y botones, del menú Listar Noticias. Descrito en la Figura 57, Figura 58 y Figura 59.

```

TextField LTitulo = new TextField();
TextField LFecha = new TextField();
TextField LId = new TextField();
TextField LTitulo1 = new TextField();
TextField LFecha1 = new TextField();
TextField LId1 = new TextField();
TextField LTitulo2 = new TextField();
TextField LFecha2 = new TextField();
TextField LId2 = new TextField();
TextField LTitulo3 = new TextField();
TextField LFecha3 = new TextField();
TextField LId3 = new TextField();

Button LmodButton = new Button("Modificar");
Button LdelButton = new Button("Eliminar");
Button LmodButton1 = new Button("Modificar");
Button LdelButton1 = new Button("Eliminar");
Button LmodButton2 = new Button("Modificar");
Button LdelButton2 = new Button("Eliminar");
Button LmodButton3 = new Button("Modificar");
Button LdelButton3 = new Button("Eliminar");

```

Figura 57. Declaración de Variables para los cuadros de texto y botones

Fuente: Los Autores

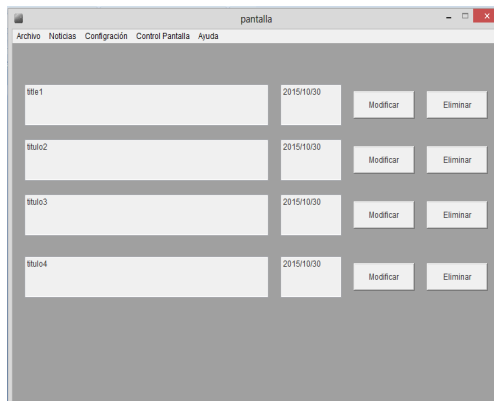


Figura 58. Pantalla principal para La Lista de Noticias

Fuente: Los Autores

```

LTitulo.setBounds(20, 60, 400, 60); //posicion izq-derecha, tamaño largo-ancho
add(LTitulo);
LTitulo.setVisible(false); LTitulo.setEditable(false);
LFecha.setText(convertido);
LFecha.setBounds(440, 60, 100, 60); //posicion izq-derecha, tamaño largo-ancho
add(LFecha);
LFecha.setVisible(false); LFecha.setEditable(false);
LmodButton.setBounds(560, 70, 100, 40); //posicion izq-derecha, tamaño largo-ancho
LdelButton.setBounds(680, 70, 100, 40); //posicion izq-derecha, tamaño largo-ancho
add(LdelButton); add(LmodButton);
LdelButton.setVisible(false); LmodButton.setVisible(false);

```

Figura 59. Código para definir el lugar, en el cual se ubicaran los cuadros de texto y botones

Fuente: Los Autores

- Conjunto de variables para la parte grafica de la aplicación cuadros de texto y botones, del menú Nueva Noticia. Descrito en la Figura 60, Figura 61 y Figura 62.

```

TextField NTitulo = new TextField();
TextField NTexto1 = new TextField();
TextField NTexto2 = new TextField();
TextField NTexto3 = new TextField();

Button NsaveButton = new Button("Guardar");
Button NcancelButton = new Button("Cancelar");

```

Figura 60. Declaración de Variables para los cuadros de texto y botones

Fuente: Los Autores

```

NTitulo.setText("Titulo Nueva Noticia");
NTitulo.setBounds(60, 60, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(NTitulo);
NTitulo.setVisible(false);
NTexto1.setText("Texto 1");
NTexto1.setBounds(60, 140, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(NTexto1);
NTexto1.setVisible(false);
NTexto2.setText("Texto 2");
NTexto2.setBounds(60, 220, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(NTexto2);
NTexto2.setVisible(false);
NTexto3.setText("Texto 3");
NTexto3.setBounds(60, 300, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(NTexto3);
NTexto3.setVisible(false);

NsaveButton.setBounds(160, 380, 100, 40); //posicion izq-derecha, tamaño largo-ancho
NcancelButton.setBounds(320, 380, 100, 40); //posicion izq-derecha, tamaño largo-ancho
add(NsaveButton); add(NcancelButton);
NsaveButton.setVisible(false);
NcancelButton.setVisible(false);

```

Figura 61. Código para definir el lugar, en el cual se ubicaran los cuadros de texto y botones

Fuente: Los Autores

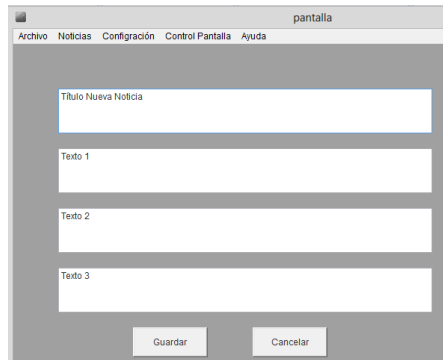


Figura 62. Pantalla Principal para el ingreso de una nueva noticia

Fuente: Los Autores

- Conjunto de variables para la parte grafica de la aplicación cuadros de texto y botones, del menú Ingresar email. Descrito en la Figura 63, Figura 64 y Figura 65.

```
//--- para los emails
TextField mNamebre = new TextField();
TextField mMail = new TextField();
TextField mMailClave = new TextField();

Button msaveMail = new Button("Guardar");
Button mcancelMail = new Button("Cancelar");

TextField nNombre = new TextField();
TextField nMail = new TextField();

Button nsaveMail = new Button("Guardar");
Button ncancelMail = new Button("Cancelar");
```

Figura 63. Declaración de Variables para los cuadros de texto y botones

Fuente: Los Autores

```
//-----
//--- para los emails
mNombre.setText("Nombre");
mNombre.setBounds(60, 60, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(mNombre);
mNombre.setVisible(false);
mMail.setText("Email");
mMail.setBounds(60, 140, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(mMail);
mMail.setVisible(false);
mMailClave.setText("Clave Correo");
mMailClave.setBounds(60, 220, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(mMailClave);
mMailClave.setVisible(false);

msaveMail.setBounds(250, 320, 100, 40); //posicion izq-derecha, tamaño largo-ancho
mcancelMail.setBounds(250, 320, 100, 40); //posicion izq-derecha, tamaño largo-ancho
add(msaveMail); add(mcancelMail);
msaveMail.setVisible(false); mcancelMail.setVisible(false);
```

Figura 64. Código para definir el lugar, en el cual se ubicaran los cuadros de texto y botones

Fuente: Los Autores



Figura 65. Pantalla principal para el ingreso de los emails

Fuente: Los Autores

- Conjunto de variables para la parte grafica de la aplicación cuadros de texto y botones, del menú Ingresar email. Descrito en la Figura 66 y Figura 67.

```

TextField interval = new TextField();

Button saveInterval = new Button("Guardar");

//----interval-----
interval.setText("2000");
interval.setBounds(60, 140, 500, 60); //posicion izq-derecha, tamaño largo-ancho
add(interval);
interval.setVisible(false);

saveInterval.setBounds(250, 240, 100, 40); //posicion izq-derecha, tamaño largo-ancho
saveInterval.setBounds(250, 240, 100, 40); //posicion izq-derecha, tamaño largo-ancho
add(saveInterval);
saveInterval.setVisible(false);

```

Figura 66. Código declarar variables y definir el lugar, en el cual se ubicaran los cuadros de texto y botones

Fuente: Los Autores

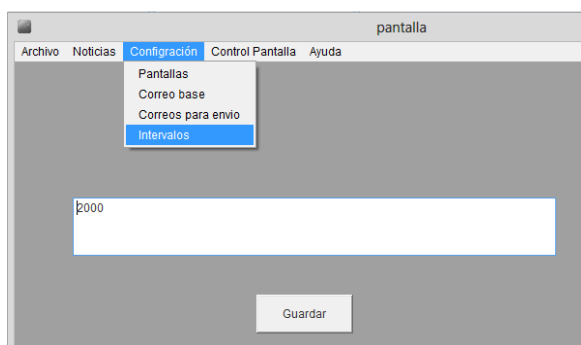


Figura 67. Pantalla principal para modificar el intervalo

Fuente: Los Autores

- Inicialización del programa a través del void setup, este código se ejecutará una sola vez al inicio del programa

```
void setup(){
```

- Declaración del usuario, password y base de datos a la cual se accederá. Descrito en la Figura 68.

```
String user    = "root";
String pass    = "";
String database = "pantalla";
```

Figura 68. Variables conexión a la base de datos

Fuente: Los Autores

- Realizará un intento de la conexión a la base de datos, de no ser posible aparecerá un mensaje de error en la conexión a la base de datos. Descrito en la Figura 69.

```
try{
    msq1 = new MySQL( this, "127.0.0.1", database, user, pass );
    //--table: noticia (id, titulo, texto1, texto2, texto3, fecha)
}
catch(Exception e){
}
}
```

Figura 69. Conexión a la base de datos

Fuente: Los Autores

- Tamaño estándar de la pantalla. Descrito en la Figura 70.

```
size(1200,600);
setLayout(new BorderLayout());
```

Figura 70. Definir el tamaño de la pantalla

Fuente: Los Autores

- Color de fondo estándar de la pantalla

```
background(160);
```

- Código que permite detectar si un mensaje se ha recibido por el puerto serial, este mensaje procede del serial del código de Arduino que comunica la placa de Arduino Leonardo con la PC. Descrito en la Figura 71.

```

try
{
    port = new Serial(this, Serial.list()[0], 9600); //Abre el puerto serie COM3

    if(port.available(>0) //Comprobamos si en el buffer hay datos
    {
        //int dato=Serial.read(); //Lee cada carácter uno por uno y se almacena en una variable
        String num = "0987444571";
        port.write(num); //Imprimimos en la consola el carácter recibido

        String myString = port.readString();
        if (myString != null) {
            println(myString);
            actionTextSMS(myString);
        }
    }
}
catch (Exception e) {
    println("Error en inicializar el puerto. SETUP!!!!.");
    //e.printStackTrace();
    //line = null;
}
}

```

Figura 71. Código para recepción del mensaje recibido a través del puerto serial

Fuente: Los Autores

- Código que permite adicionar los menús en la pantalla. Descrito en la Figura 72.

```

menuListen = new myMenuListener();
myMenu = new MenuBar();

//create the file-archivo button
file = new Menu("Archivo");
//create all the Menu Items and add the menuListener to check their state.
close = new MenuItem("Cerrar");
close.addActionListener(menuListen);
//add the items to the top level Button
file.add(close);
//add the button to the menu
myMenu.add(file);

//create the notice-list,new button
notice = new Menu("Noticias");
//create all the Menu Items and add the menuListener to check their state.
notice_list = new MenuItem("Enlistar Noticias");
notice_list.addActionListener(menuListen);
//add the items to the top level Button
notice.add(notice_list);

notice_new = new MenuItem("Nueva Noticia");
notice_new.addActionListener(menuListen);
//add the items to the top level Button
notice.add(notice_new);
//add the button to the menu
myMenu.add(notice);

```

Figura 72. Código manejo de menús

Fuente: Los Autores

- Inicio del void draw, código que se ejecutara constantemente, mientras este active la aplicación

```
void draw(){
```

- Código que permite detectar si un mensaje se ha recibido por el puerto serial, este mensaje procede del serial del código de Arduino que comunica la placa de Arduino Leonardo con la PC. Descrito en la Figura 73.

```
try
{
//Recibir datos SMS del Arduino
if(port.available() > 0) // si hay algún dato disponible en el puerto
{
valor=port.read();//Lee el dato y lo almacena en la variable "valor"

String myString = port.readString();
if (myString != null) {
println(myString);

actionTextSMS(myString);
actionCrearNoticias();
}
//println(valor);
} //if variable
} catch (Exception e) {
//e.printStackTrace();
//println("Error en inicializar el puerto. DRAW!!!!.");
//line = null;
}
```

Figura 73. Código para recepción del mensaje recibido a través del puerto serial

Fuente: Los Autores

- Código para cerrar la aplicación si se da click en el menú cerrar. Descrito en la Figura 74.

```
if(menu=="cerrar"){
exit();
menu="";
}
```

Figura 74. Código para cerrar la aplicación

Fuente: Los Autores

- Al añadir, modificar o consultar desde los diferentes menús se realice el llamado a la función que permite la comunicación con la base de datos, que ejecuta unas sentencias SQL para su comunicación. Descrito en la Figura 75 y Figura 76.

```

private void actionButton(String butt, String value){
    if ( msql.connect() )
    {
        try{
            if(butt == "consultaTodos"){
                //--table: noticia (id, titulo, texto1, texto2, texto3, fecha)
                int i=0;
                msql.query( "SELECT id,titulo,texto1,texto2,texto3,fecha FROM noticia" );
                while(msql.next()){
                    if(i==0)
                    {
                        LId.setText(msql.getString(1)); LTitulo.setText(msql.getString(2));
                        LFecha.setText(msql.getString(6));
                        i++;
                    }
                    else if(i==1)
                    {
                        LId1.setText(msql.getString(1)); LTitulo1.setText(msql.getString(2));
                        LFecha1.setText(msql.getString(6));
                        i++;
                    }
                    else if(i==2)
                    {
                        LId2.setText(msql.getString(1)); LTitulo2.setText(msql.getString(2));
                        LFecha2.setText(msql.getString(6));
                        i++;
                    }
                    else if(i==3)
                    {
                        LId3.setText(msql.getString(1)); LTitulo3.setText(msql.getString(2));
                    }
                }
            }
        }
    }
}

```

Figura 75. Código para consultas query a la base de datos

Fuente: Los Autores

```

if(butt=="consultaId"){
    //--table: noticia (id, titulo, texto1, texto2, texto3, fecha)
    msql.query( "SELECT titulo,texto1,texto2,texto3,fecha FROM noticia " + "WHERE id = " + value + " " );
    if(msql.next()){
        {
            NTitulo.setText(msql.getString(1));
            NTexto1.setText(msql.getString(2));
            NTexto2.setText(msql.getString(3));
            NTexto3.setText(msql.getString(4));
        }
    }
}
else if(butt=="borrar"){
    //--table: noticia (id, titulo, texto1, texto2, texto3, fecha)
    //msql.query( );
    msql.execute("DELETE FROM noticia " + "WHERE id = " + value + " ");
}
}

```

Figura 76. Código para consultas query a la base de datos

Fuente: Los Autores

- Al llegar un mensaje SMS se realice el llamado a la función que permite la comunicación con la base de datos, que ejecuta unas sentencias SQL para su comunicación. Descrito en la Figura 77.

```

private void actionTextSMS(String myString){
    if ( mysql.connect() )
    {
        int id=1;
        try{
            //--table: noticia (id, titulo, texto1, texto2, texto3, fecha)
            mysql.query( "SELECT MAX(id) FROM noticia" );
            mysql.next();
            id = mysql.getInt(1) + 1;
        }
        catch(Exception ex){
            id=1;
        }
        //println( "number of rows: " + mysql.getInt(1) );
        //System.out.println( fecha.getDay());
        //System.out.println( fecha.getMonth());

        mysql.execute("INSERT INTO 'pantalla'.noticia ('id','titulo','texto1','texto2','texto3','fecha') "
            + "VALUES ('" + id + "','" + "Noticia Urgente" + "','" + myString + "','" + fecha + "','" + fecha + "'");
        println( "Ingresado correctamente.");
    }
}

```

Figura 77. Código para consultas query a la base de datos

Fuente: Los Autores

- Valores por defecto del correo del cual se envía el mensaje vía email

```

String Fnombre = "Tesis Unach 2015", Femail = "pruebatesisunach2015@gmail.com", Fclave = "unach2015";
String Dnombre = "Sebastian", Demail = "escobar_sebastian@hotmail.com";
String Subject = "Reporte_pantalla_informativa";

```

- Código que permite el envío de mensajes vía email a través de una cuenta de google, esta se ejecuta al detectar que existen datos enviados en el puerto serial. Descrito en la Figura 78 y Figura 79.

```

import javax.mail.Authenticator;
import javax.mail.PasswordAuthentication;

public class Auth extends Authenticator {

    public Auth() {
        super();
    }

    public PasswordAuthentication getPasswordAuthentication() {
        String username, password;
        username = "pruebatesisunach2015@gmail.com";
        password = "unach2015";
        System.out.println("authenticating. . ");
        return new PasswordAuthentication(username, password);
    }
}

```

Figura 78. Código para consultas envió de mensajes vía email

Fuente: Los Autores

```

void sendMail(String Fnombre,String Femail,String Fclave,String Dnombre,String Demail, String Subject, String myString) {
// Create a session
String host="smtp.gmail.com";
Properties props=new Properties();
// SMTP Session
props.put("mail.transport.protocol", "smtp");
props.put("mail.smtp.host", host);
props.put("mail.smtp.port", "587");
props.put("mail.smtp.auth", "true");
// We need TLS, which gmail requires
props.put("mail.smtp.starttls.enable","true");
// Create a session
Session session = Session.getDefaultInstance(props, new Auth());
try
{
// Make a new message
MimeMessage message = new MimeMessage(session);
// Who is this message from
message.setFrom(new InternetAddress(Femail, Fnombre));
// Who is this message to (we could do fancier things like make a list or add CC's)
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(Demail, false));
// Subject and body
message.setSubject(Subject);
message.setText(myString);
// We can do more here, set the date, the headers, etc.
Transport.send(message);
println("Mail sent!");
}
catch(Exception e)

```

Figura 79. Código para consultas envío de mensajes vía email

Fuente: Los Autores

5.3.1.6. Pantallas principales de la aplicación

La pantalla principal se puede apreciar en la Figura 80.

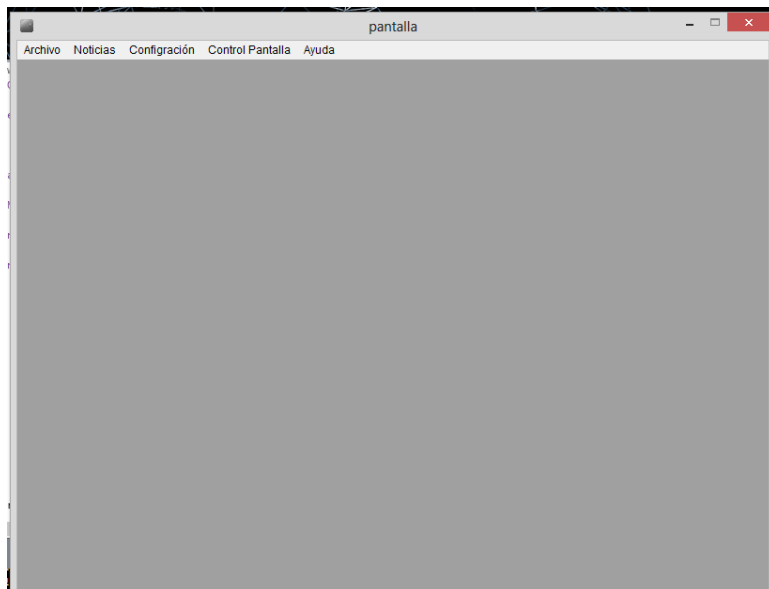


Figura 80. Pantalla inicial Aplicación.

Fuente: Los Autores

A continuación se muestran las funcionalidades de la aplicación:

En la Figura 81, se puede apreciar los submenús presentes para la interacción con el usuario.

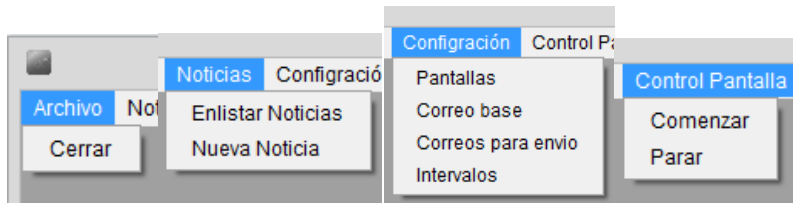


Figura 81. Menús principales.

Fuente: Los Autores

La Figura 82, se puede observar la pantalla, para administrar las noticias recibidas, donde se puede modificar o eliminar.

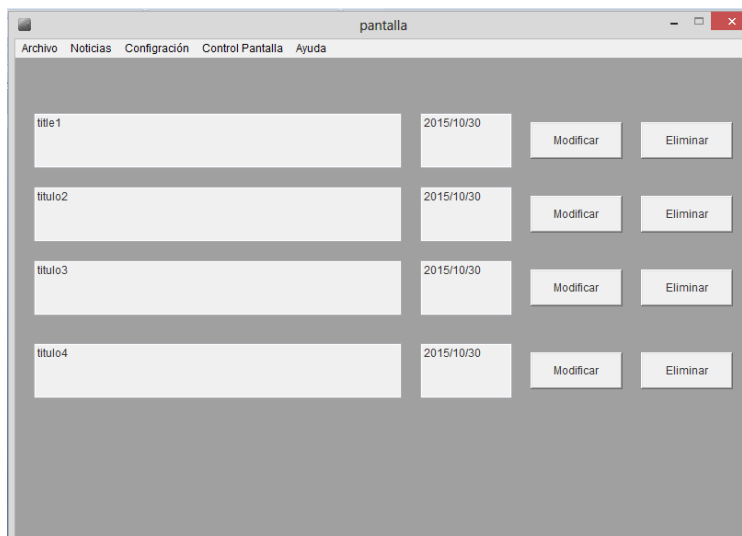


Figura 82. Menú enlistar noticias.

Fuente: Los Autores

En la Figura 83, se puede observar la pantalla para modificar una noticia.

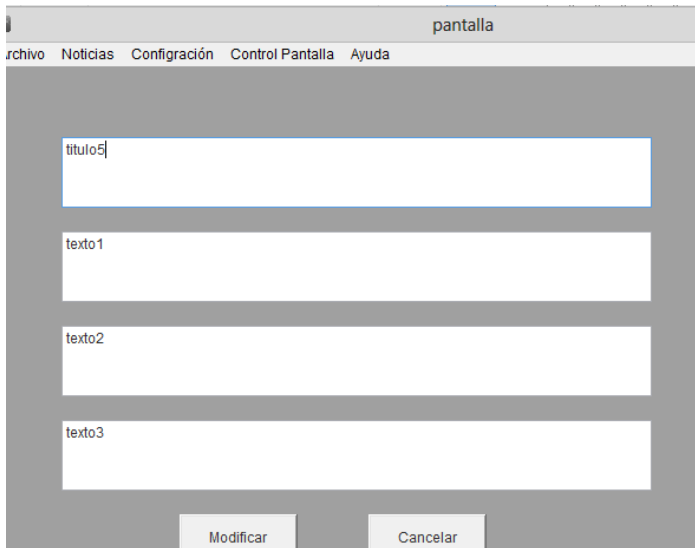


Figura 83. Menú modificar noticia.

Fuente: Los Autores

En la Figura 84, se puede observar la pantalla para ingresar una nueva noticia.

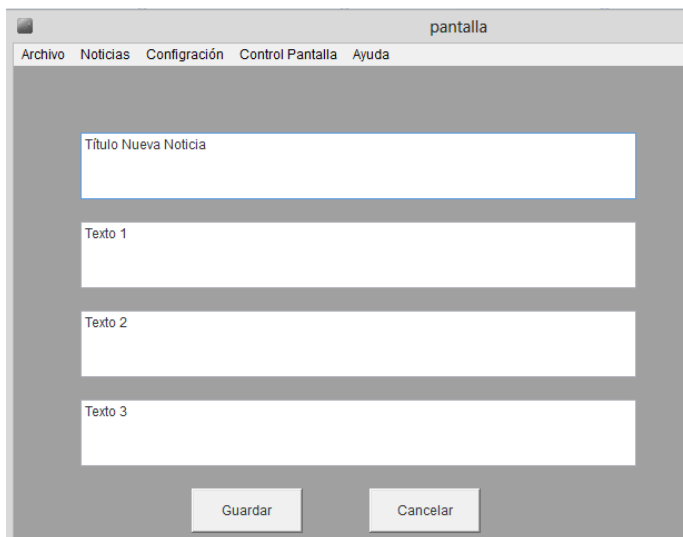
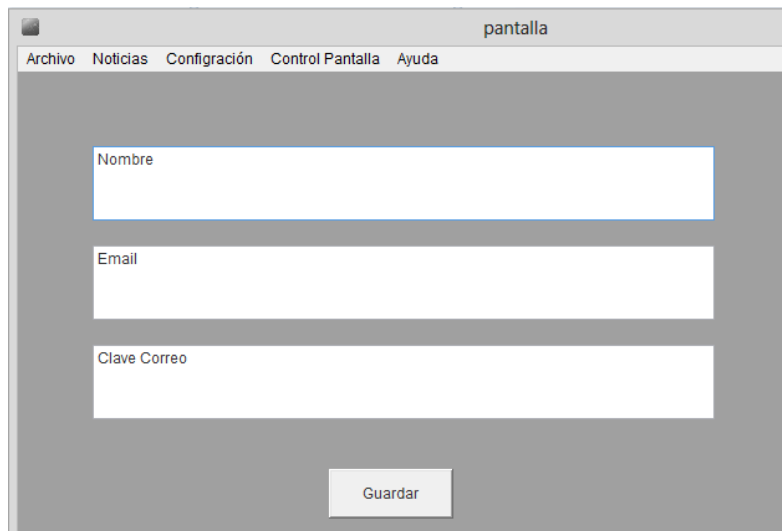


Figura 84. Menú ingresar nueva noticia.

Fuente: Los Autores

En la Figura 85, se puede observar la pantalla para modificar los datos del usuario fuente.




The screenshot shows a window titled "pantalla" with a menu bar containing "Archivo", "Noticias", "Configuración", "Control Pantalla", and "Ayuda". The main area contains three text input fields: "Nombre", "Email", and "Clave Correo". A "Guardar" button is located at the bottom center of the window.

Figura 85. Menú Datos usuario fuente.

Fuente: Los Autores

En la Figura 86, se puede observar la pantalla para modificar los datos del usuario destino.



The screenshot shows a window titled "pantalla" with a menu bar containing "Archivo", "Noticias", "Configuración", "Control Pantalla", and "Ayuda". The main area contains two text input fields: "Nombre" and "Email". A "Guardar" button is located at the bottom center of the window.

Figura 86. Menú datos usuario destino.

Fuente: Los Autores

En la Figura 87, se puede observar la pantalla para modificar el intervalo de rotación de las noticias, para mostrar en la pantalla.

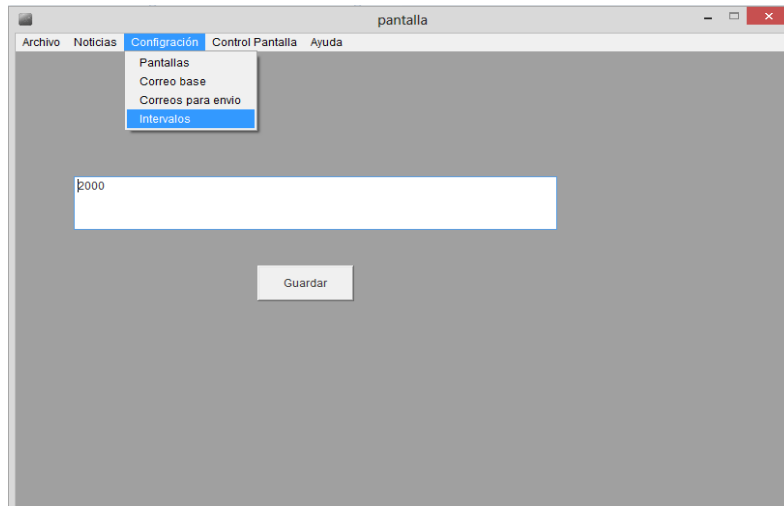


Figura 87. Intervalos.

Fuente: Los Autores

En la Figura 88, se puede observar la captura de las pantallas, en las cuales se observa la rotación de las noticias, en este sentido, si se desea administrar la información a ser proyectada, se puede modificar con las opciones dentro del menú Control Pantalla. Al seleccionar Control Pantalla -> Parar, se muestra la pantalla vacía, Figura 89.



Figura 88. Rotación de noticias en la pantalla.

Fuente: Los Autores



Figura 89. Pantalla vacía, luego de accionar Parar.

Fuente: Los Autores

Los módulos Arduino utilizados se pueden observar en la Figura 90.



Figura 90. Módulos en funcionamiento.

Fuente: Los Autores

Al momento de enviar un mensaje SMS, y cumplir con todas las condiciones, como la clave y el formato del texto, la aplicación almacena en la base de datos, esa información, además enviándole una respuesta de confirmación, tanto al móvil, como al correo configurados. En la Figura 91, Figura 92, Figura 93, se puede observar los resultados del funcionamiento.

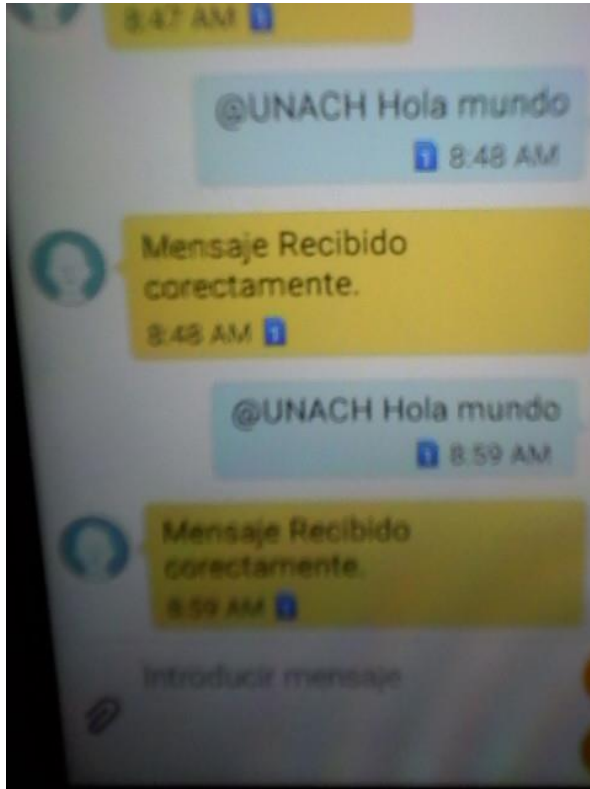


Figura 91. Envío y recepción del mensaje SMS.

Fuente: Los Autores

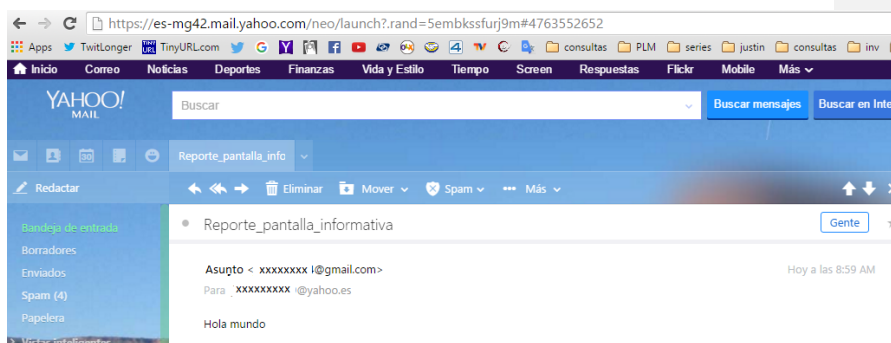


Figura 92. Recepción de la respuesta en el correo electrónico

Fuente: Los Autores

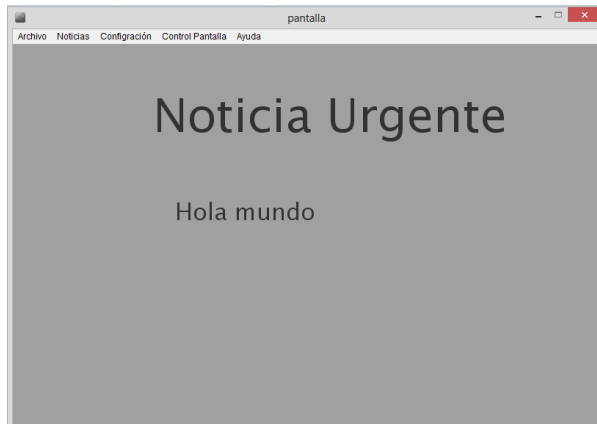


Figura 93. Adición de la información en la pantalla, al momento de la recepción.

Fuente: Los Autores

Para permitir la comunicación a través del correo, es necesaria una cuenta en gmail, hay que desactivar las seguridades, para poder acceder desde aplicaciones menos seguras: <https://www.google.com/settings/security/lesssecureapps>

En el código en Processing para el envío de mensajes hay que configurar el host a smtp.gmail.com, además de configurar en el puerto: mail.smtp.port", "587"

CAPÍTULO VI

6. RESULTADOS

6.1. RESULTADOS OBTENIDOS

Los módulos Arduino utilizados se pueden observar en la Figura 94.



Figura 94. Módulos en funcionamiento.

Fuente: Los Autores

Al momento de enviar un mensaje SMS, y cumplir con todas las condiciones, como la clave y el formato del texto, la aplicación almacena en la base de datos, esa información, además enviándole una respuesta de confirmación, tanto al móvil, como al correo configurados.

En la Figura 95, Figura 96, Figura 97, se puede observar los resultados del funcionamiento.

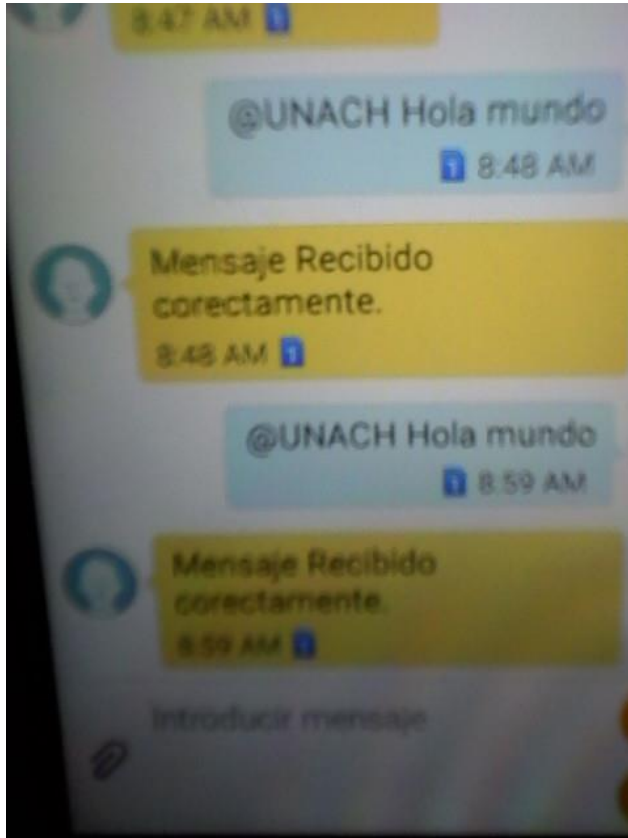


Figura 95. Envío y recepción del mensaje SMS.

Fuente: Los Autores

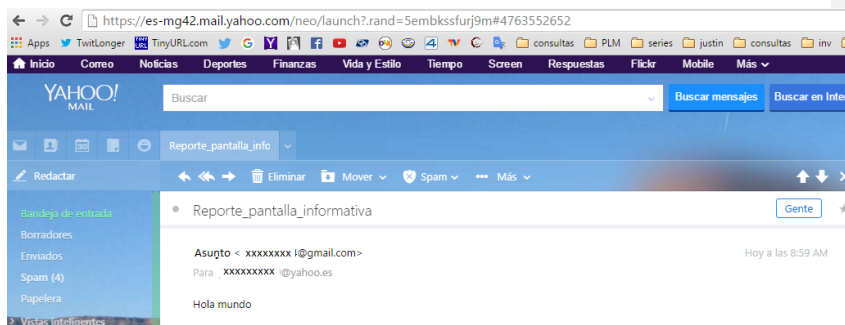


Figura 96. Recepción de la respuesta en el correo electrónico.

Fuente: Los Autores

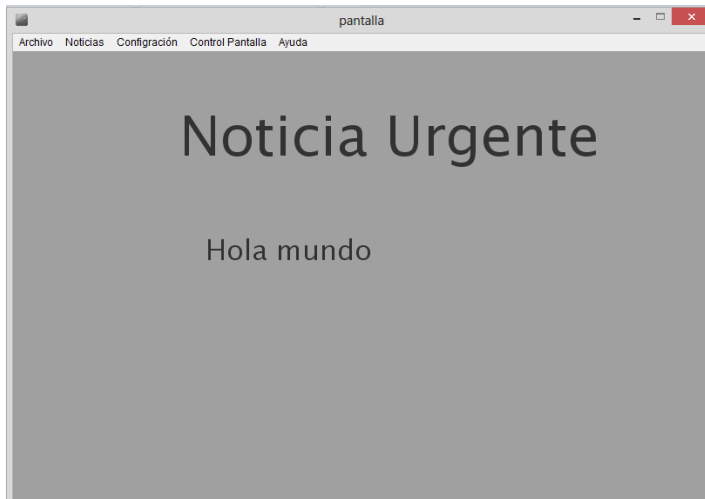


Figura 97. Adición de la información en la pantalla, al momento de la recepción.

Fuente: Los Autores

Para permitir la comunicación a través del correo, es necesaria una cuenta en gmail, hay que desactivar las seguridades, para poder acceder desde aplicaciones menos seguras: <https://www.google.com/settings/security/lesssecureapps>

En el código en Processing para el envío de mensajes hay que configurar el host a smtp.gmail.com, además de configurar en el puerto: mail.smtp.port", "587"

Debido a la distancia entre el servidor y la localización de la pantalla es necesaria la utilización de un dispositivo que permita crear una extensión para la comunicación. Este dispositivo permite la combinación del cable VGA con el cable UTP, de este modo cubre una distancia mayor llegando a 80 metros como máximo. Además por este medio abarata los costos ya que solamente es necesaria la adquisición de cable UTP y no un cable VGA.

A continuación en la Figura 98 y Figura 99, se muestra la colocación de la pantalla y los conectores utilizados.



Figura 98. Colocación de la pantalla.

Fuente: Los Autores



Figura 99. Conectores VGA / UTP utilizados.

Fuente: Los Autores

6.2. COMPROBACIÓN DE LA HIPÓTESIS

- **Escenario A:** El tiempo que se demora en enviar un oficio desde el decano a los directores de escuela comunicando una disposición tomada por las autoridades.
- **Escenario B:** El tiempo que demora en publicar la información en la pantalla LED que se encuentra en el pasillo del primer piso del bloque B de la facultad.

En la Tabla 11 se ingresa los tiempos transcurridos al momento de enviar un oficio o publicar una noticia a cada una de los directores de escuela.

Tabla 11. Datos de la variable

	Escenario A (Segundos)	Escenario B (Segundos)
Prueba 1	1800	160s
Prueba 2	1830	150
Prueba 3	1860	200
Prueba 4	1890	190
Prueba 5	2100	180

Fuente: Los Autores

6.3.DEMOSTRACIÓN DE LA HIPÓTESIS

En este proyecto de investigación se utiliza para la comprobación de la hipótesis con el método de la prueba del Chi- Cuadrado. Para realizar la comprobación de la hipótesis planteada se utiliza los tiempos que se demora el decano al enviar un oficio el a cada uno de los directores de carrera y se tomó también los tiempos que se demora al llegar un oficio al momento de enviar a la pantalla y a los correos de los directores.

Ho: El Diseño e Implementación de un Sistema de Comunicación Integrando Tecnologías para el Mejoramiento de Voz IP permitirá acceder a la comunicación de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo entre el personal.

Se calcula los valores esperados, que constan en la tabla 12 y son los posibles valores que debería tomar para que se acepte la hipótesis.

Tabla 12. Valores calculados

	Escenario A (Segundos)	Escenario B (Segundos)
Prueba 1	1647	166
Prueba 2	1812	168
Prueba 3	1885	175
Prueba 4	1903	177
Prueba 5	2086	194

Fuente: Los Autores

Para dar por aceptada la hipótesis el valor de la Prueba del Chi cuadrado X^2_{Prueba} debe ser menor a la X^2_{Tabla} . Por tanto la propuesta presentada permite mejorar la eficiencia al momento de enviar un mensaje de texto que se publicara en la pantalla y el correo de un director de escuela.

Comentado [D7]: De esta manera se encuentra la comprobación de hipótesis, cambien la redacción por favor.... Y de acuerdo a lo de ustedes.... Revisen el Chi cuadrado con una variable....

Tabla 13. Comprobación de Hipótesis

	R	5
	K	2
Grados de Libertad	$(r-1)(k-1)$	4
Nivel de significación	α	0,005
X^2_{Tabla}	X2	23,51
Probabilidad	P	0,00
X^2_{Prueba}		22,66

Fuente: Los Autores

De esta manera se comprueba la hipótesis propuesta en el proyecto de investigación que se presento donde permite mejorar la eficiencia al momento de enviar un mensaje de texto vía celular que se publicara en la pantalla y el correo de un director de escuela.

Comentado [D8]: De esta manera se encuentra la comprobación de hipótesis, cambien la redacción por favor.... Y de acuerdo a lo de ustedes.... Revisen el Chi cuadrado con una variable....

CAPÍTULO VII

7. DISCUSIÓN

Como primer paso se desarrolló un análisis de la cobertura presente en el edificio de la Facultad, en este sentido se presentó las recomendaciones necesarias para mejorar el acceso para la red wifi.

Para el análisis se utilizó el software Acrylic WI-FI HeadMaps. Es un software que facilita la detección de redes inalámbricas WLAN que operen con el estándar 802.11 (WI-FI) y trabaja con los protocolos 802.11^a, 802.11b y 802.11g.

Las utilidades importantes que permitió el software son:

- Comprobación las redes WLAN presentes
- Análisis la cobertura de cada red y la intensidad de señal.
- Se determinó la ubicación de puntos de acceso así como también su marca y configuración.
- Permitió conocer que estaciones base se encuentran abiertas o con seguridad WEP, WAP, WAP2.

Para complementar el acceso a la información de la red inalámbrica se realiza el sistema que permita proyectar en una pantalla LED datos informativos. Además para posibilitar el envío de noticias o informaciones urgentes, siendo necesario la interconexión de la red celular al servidor que almacena la información a ser proyectada.

Ante un evento o una noticia importante y urgente, como queda dicho, se enviaría vía SMS la información a ser proyectada en la pantalla LED. Para ello, se escogió dos módulos Arduino para tal efecto.

CAPÍTULO VIII

8. CONCLUSIONES Y RECOMENDACIONES

8.1. CONCLUSIONES

El mejoramiento de la red inalámbrica interna permite un mejor acceso a la información para distintos usuarios.

La adopción de pantallas informativas permite ofrecer de manera más abierta y segura datos de relevancia tanto para docentes, estudiantes y público en general.

El sistema permite la comunicación entre distintas tecnologías, proporcionando distintas opciones para el acceso a la información.

Los módulos de Arduino son una buena alternativa para ocupar en sistemas donde los altos costos de equipos son una limitante.

Es necesario tener de medios a altos conocimiento en programación para poder configurar la aplicación, enfocando sobre todo en el lenguaje JAVA.

Se puede crear un mecanismo para que las personas interactúen con el sistema adaptando otros módulos.

Se puede mejorar realizando la aplicación para conectarse a través de la red de datos para conectarse a varias pantallas. Además se puede añadir otra información para presentar en la pantalla como: imágenes y videos.

Se puede además migrar el código desarrollado en processing para aplicarlo en un teléfono móvil, permitiendo en este sentido acceder a la información a través de una red wifi.

8.2. RECOMENDACIONES

Según lo anteriormente expuesto se determina que el nivel de señal en varios espacios no es eficiente razón por la cual se recomienda:

- Reubicación de los puntos de acceso de la primera planta a un lugar central en donde la señal pueda expandirse de lado y lado, el lugar que se cree acertado es en el descanso de la primera planta junto a las gradas; de no ser posible esta solución se podría aumentar un punto de acceso para el lado B de la planta baja.
- En la segunda planta es conveniente instalar un nuevo punto de acceso en el descanso junto a las gradas.
- En la tercera planta habiendo aquí varias aulas y muchos estudiantes es recomendable instalar un punto de acceso en el lado B ya que solo se encuentra uno en el lado A o de otra manera se podría mover hacia el descanso de la tercera planta.
- Como solución a los problemas de calidad de recepción y mejorar la señal en la Facultad de Ingeniería se propone reemplazar los Access point marca LINKSYS por otros de más potencia (cisco systems), adicionalmente a esto trabajar ver la posibilidad de aumentar el ancho de banda y la velocidad de transmisión para que más usuarios puedan conectarse a las redes docentes y estudiantes sin inconvenientes.
- Con la adecuación de antenas directivas adaptando a los Access Point se puede dar servicio de datos en las áreas que la señal no es óptima.

CAPÍTULO IX

9. PROPUESTA

9.1. TÍTULO DE LA PROPUESTA

Diseño e Implementación de un Sistema de Comunicación Integrando Tecnologías para el Mejoramiento de Voz Ip Interna de la Facultad de Ingeniería de la Universidad Nacional de Chimborazo.

9.2. INTRODUCCIÓN

La Universidad Nacional de Chimborazo cubre grandes extensiones de terreno, en el cual se encuentran distribuidas varias edificaciones entre ellas el de la facultad de ingeniería.

Teniendo la necesidad de publicar notas informativas como el calendario académico, ubicación de los docentes, información de parte de las autoridades hacia los docentes, estudiantes y ciudadanía en general; la mayoría de veces se ha realizado mediante carteles, pancartas incluso volantes, surgiendo la necesidad de implementar un mecanismo que permita emitir esta información de mejor manera. Este mecanismo sería a través del mejoramiento de la red inalámbrica y por medio del uso de pantallas informativas, optimizando así el tiempo y recursos.

Realizando un estudio de cobertura se delimitará la eficiencia de la red inalámbrica. Resultado de este estudio se brindará recomendaciones y sugerencias para el mejoramiento de la red.

El sistema permite administrar de mejor manera la proyección de información en una pantalla LED.

9.3. OBJETIVOS

9.3.1. OBJETIVO GENERAL

- Mejorar el desempeño de la red inalámbrica y por medio del uso de pantallas informativas ofrecer de forma óptima el acceso a la información de los docentes, estudiantes y ciudadanía en general.

9.3.2. OBJETIVOS ESPECÍFICOS

- Mejorar el desempeño de la red inalámbrica ofreciendo recomendaciones y sugerencias, resultado de un estudio de cobertura de la red.
- Mediante el uso de pantallas informativas ofrecer de mejor forma el acceso a la información a los docentes, estudiantes y ciudadanía en general.
- Implementar un prototipo que permita emitir una noticia con urgencia en una pantalla LED, enviando información a través de mensajes de texto desde un teléfono celular.

Enviar un reporte vía email, al emitir una noticia a través de un SMS.

9.4. FUNDAMENTACIÓN CIENTÍFICO – TÉCNICA

Gracias al avance tecnológico de las pantallas a color, ahora se cuenta con pantallas electrónicas tipo LED. Las mismas que han evolucionado con el objetivo de ofrecer mayores prestaciones como son una gran brillantez, mejores niveles de contraste y resolución lo que ofrece una calidad única, además que consumen menos energía, es menos nociva con el medio ambiente y su vida útil es mayor. En el Ecuador todavía no existen empresas dedicadas a la fabricación de Pantallas LED gigantes para exteriores, obligando la importación de estas pantallas tipo LED provenientes de otros países.

La estructura de pantallas publicitarias es modular, lo que permite construir pantallas de diferentes dimensiones y agiliza el montaje de la misma. Para la implementación de una pantalla comercial se unen hasta dieciséis gabinetes que son estructuras donde se montan varios módulos pequeños cada uno con su propio controlador y en comunicación entre sí.

El ahorro energético es una ventaja en la implementación de estas pantallas por el mismo hecho de estar construidas con diodos LED que presentan dichas características de ahorro energético.

Las pantallas son habitualmente utilizadas en televisores, monitores para los ordenadores, cámaras fotográficas e incluso a gran escala como son las pantallas gigantes. El color en las pantallas ha ido evolucionando, lo que da como resultado que se puede crear imágenes más nítidas e inimaginables, como se muestra en la Figura 100.



Figura 100. Pantalla Tipo LED.

Fuente: (Jácome, Montenegro, 2012)

9.5. DESCRIPCIÓN DE LA PROPUESTA

Para el desarrollo del sistema, se requirió seguir una serie de pasos y etapas, que en conjunto, cumplirían con el objetivo planteado. Cada etapa desempeña una función específica dentro del esquema global, y también sirve de apoyo para la siguiente etapa, haciendo del dispositivo implementado un sistema funcional.

La primera se desarrolló en el edificio de la facultad un estudio de la cobertura actual de la red inalámbrica, en sitios abiertos y diferentes pasillos. Determinar el nivel de cobertura en los diferentes ambientes y analizar los problemas existentes de cobertura.

Lo segundo para complementar el acceso a la información de la red inalámbrica se realiza el sistema que permita proyectar en una pantalla LED datos informativos. Para ello es necesaria la interconexión de la red celular al servidor que almacena la información a ser proyectada.

La información está almacenada en un servidor que consta de una base de datos en mysql y apache que permite el acceso vía web.

Ante un evento o una noticia importante se enviaría vía SMS la información a ser proyectada en la pantalla LED. Se escogió dos módulos Arduino para tal efecto.

El módulo Arduino GSM / GPRS necesita una fuente de alimentación de 5 voltios externa. El módulo Arduino Leonardo se conecta a través de un cable USB al servidor, permitiendo el envío y recepción de la información.

Una vez desarrollado el código se compila y se carga en el módulo Arduino para su correcto funcionamiento.

9.5.1. ESPECIFICACIONES

Para la implementación del sistema se necesita un computador, el cual realiza las funciones de servidor; además de un celular y dos módulos de arduino:

Un Shield Modem Gsm Gprs Sms Sim900 y el módulo de Arduino Leonardo.

9.6. DISEÑO ORGANIZACIONAL

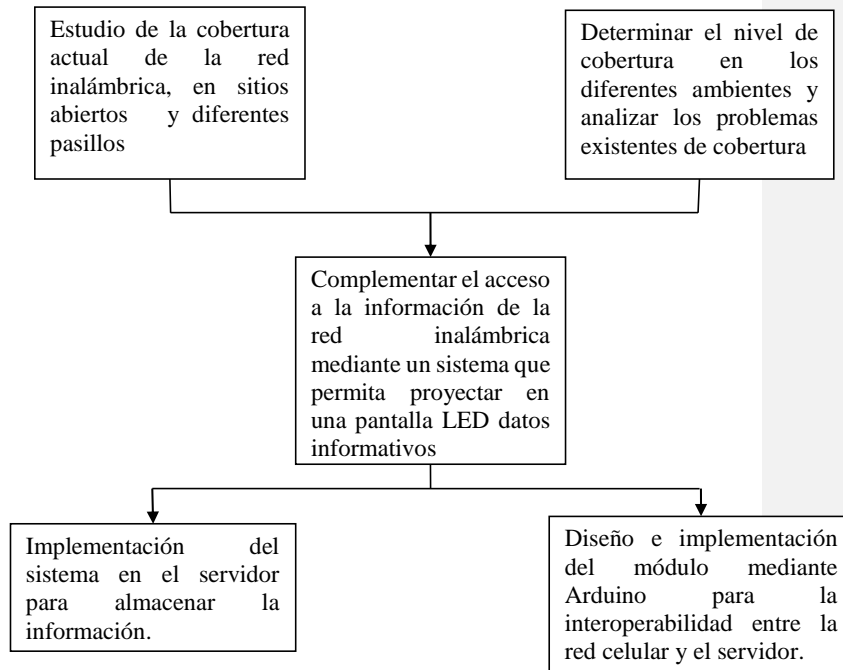


Figura 101. Diagrama de procedimientos.

Fuente: Los Autores

9.7. MONITOREO Y EVALUACIÓN DE LA PROPUESTA

Los módulos Arduino utilizados se pueden observar en la Figura 90



Figura 102. Módulos en funcionamiento.

Fuente: Los Autores

Al momento de enviar un mensaje SMS, y cumplir con todas las condiciones, como la clave y el formato del texto, la aplicación almacena en la base de datos, esa información, además enviándole una respuesta de confirmación, tanto al móvil, como al correo configurados. En la Figura 103, Figura 104, Figura 105, se puede observar los resultados del funcionamiento.

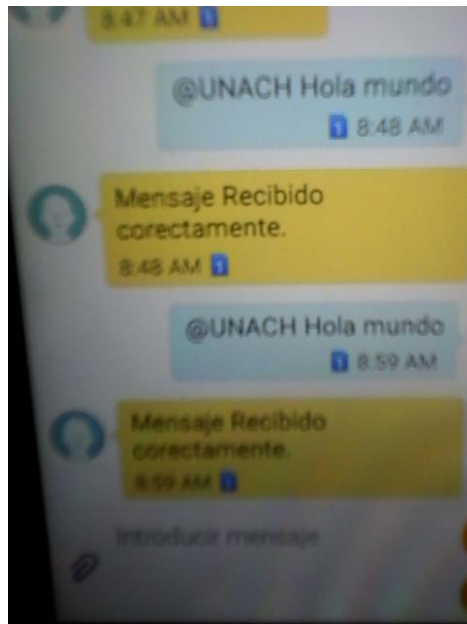


Figura 103. Envío y recepción del mensaje SMS.

Fuente: Los Autores

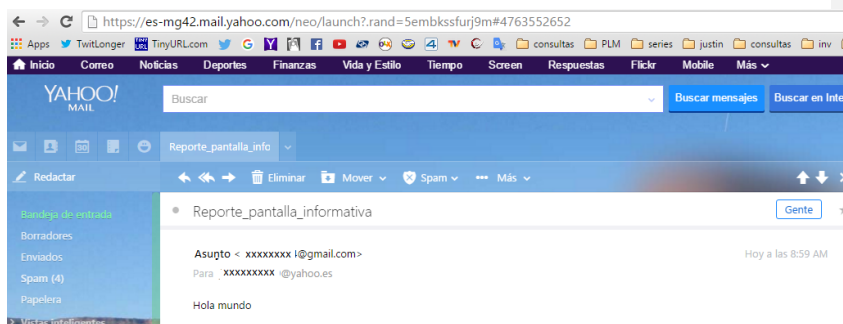


Figura 104. Recepción de la respuesta en el correo electrónico.

Fuente: Los Autores

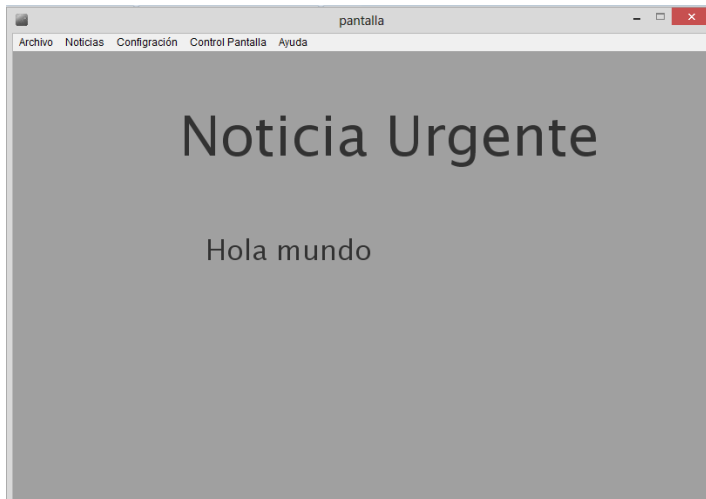


Figura 105. Adición de la información en la pantalla, al momento de la recepción.

Fuente: Los Autores

Para permitir la comunicación a través del correo, es necesaria una cuenta en gmail, hay que desactivar las seguridades, para poder acceder desde aplicaciones menos seguras: <https://www.google.com/settings/security/lesssecureapps>

En el código en Processing para el envío de mensajes hay que configurar el host a `smtp.gmail.com`, además de configurar en el puerto: `mail.smtp.port`, "587"

CAPITULO X

10. BIBLIOGRAFÍA

- H. Congreso de la Unión de los Estados Unidos. (2002). Ley de Ciencia y Tecnología. EEUU.
- Comisión Interamericana De Derechos Humanos. (2011). El derecho de acceso a la información en el marco jurídico interamericano, Segunda Edición.
- Jane Butler, Ermanno Pietrosemoli, Marco Zennaro. (2013). Redes inalámbricas en los países en desarrollo, Cuarta Edición
- Hueso Ibañez Galindo Luis. (2012). Gestión de bases de datos, Segunda Edición. Cfgs.
- Abel, P. (1996). Lenguaje ensamblador y programación para pc ibm y compatibles. México, D.F.: Prentice Hall Inc.
- Ambadar, A. (2002). Procesamiento de señales analógicas y digitales. México, D. F.: Thomson Editores, S. A.
- Stallings, W. (2004). Comunicaciones y redes de computadores. México, D.F.: Pearson Prentice Hall.
- Jorge Pomares Baeza. (2009). Control por computador – Arduino. España. GITE – IEA
- Gabriela Jácome, David Montenegro. (2012) Diseño e Implementación De Un Módulo Prototipo Para Pantallas Gigantes Tipo LED. Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército Quito, Ecuador
- Cisco Systems. Fundamentos de red inalámbricas. (2006). Cisco systems.
- Artículo Inseguridad en red inalámbricas WI-FI. (2008). Oliva Chavez, Karen Paola. Especialización en seguridad Informática UPB.
- Proyecto de grado: estudio de la red inalámbrica de la Universidad Pontificai Bolivariana Seccional Bucarmanga. (2006). Jaime Andrés Chaparro Sánchez.
- Redes inalámbricas IEEE 802.11 (2015). Wikipedia. Disponible en: https://es.wikipedia.org/wiki/IEEE_802.11
- Introducción a las redes inalámbricas IEEE 802.11 (2015). Wikipedia. Disponible en: <http://es.ccm.net/contents/789-introduccion-a-wi-fi-802-11-o-wifi>
- Redes inalámbricas IEEE 802.11 (2015). Wikipedia. Disponible en: https://es.wikipedia.org/wiki/IEEE_802.11n
- Telefonía Móvil (2015). Wikipedia. Disponible en: https://es.wikipedia.org/wiki/Telefon%C3%ADa_m%C3%B3vil
- Redes de celdas (2015). Wikipedia. Disponible en: https://es.wikipedia.org/wiki/Red_de_celdas
- Pixel. (2015) Wikipedia. Disponible en <http://es.wikipedia.org/wiki/P%C3%ADxel>.

- Arduino. (2015) Wikipedia. Disponible en: <https://es.wikipedia.org/wiki/Arduino>.
- "Interfacing Arduino to other languages". (2015) Proyecto Arduino. Disponible en: <http://www.arduino.cc/playground/Main/InterfacingWithSoftwar>
- "Language Reference (extended)". (2015) Proyecto Arduino. Disponible en: <http://arduino.cc/en/Reference/Extended>
- "Editronixx" (2015). Edison Viveros. Disponible en: <http://www.editronixx.com>
- Página oficial arduino. (2015). <https://www.arduino.cc/en/Main/Software>
- Redes Inalámbricas: IEEE 802.11. (2015). Enrique de Miguel Ponce, Enrique Molina Tortosa
- Redes wifi por mallas. (2006). Disponible en: <http://eveliux.com/mx/Redes-Wi-Fi-en-malla-wi-fi-mesh-networks.html>
- Email Processing (2015). Disponible en: <http://shiffman.net/2007/11/13/e-mail-processing/>
- Estructura Arduino. Disponible en: <http://playground.arduino.cc/ArduinoNotebookTraduccion/Structure>
- Estructura Processing. Disponible en: <http://dunadigital.com/processing/2013/07/14/estructura-basica-de-un-programa-setup-y-draw/>

ANEXOS

ANEXO A:

Módulo Arduino Leonardo



INTRODUCTION

The GPRS/GSM Shield provides you a way to use the GSM cell phone network to receive data from a remote location. The shield allows you to achieve this via any of the three methods:

- Short Message Service
- Audio
- GPRS Service

The GPRS Shield is compatible with all boards which have the same form factor (and pinout) as a standard Arduino Board. The GPRS Shield is configured and controlled via its UART using simple AT commands. Based on the SIM900 module from SIMCOM, the GPRS Shield is like a cell phone. Besides the communications features, the GPRS Shield has 12 GPIOs, 2 PWMs and an ADC.

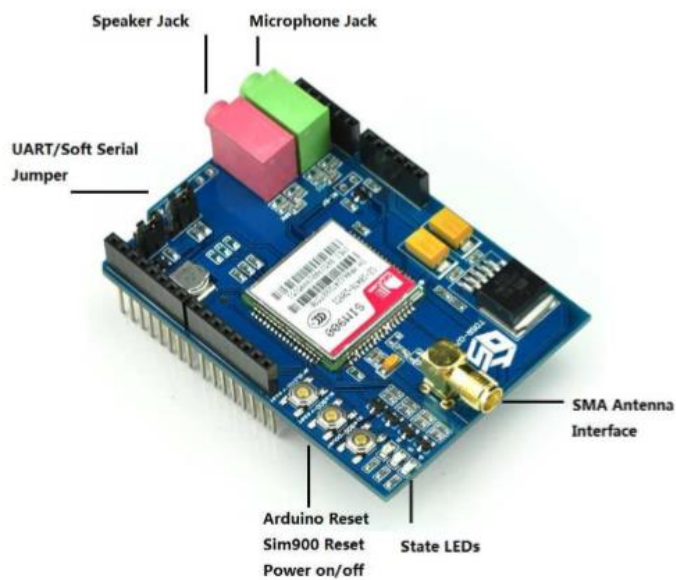
ANEXO B:

Módulo GSM/GPRS SIM 900

SPECIFICATIONS

- Quad-Band 850 / 900/ 1800 / 1900 MHz - would work on GSM networks in all countries across the world.
- GPRS multi-slot class 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Class 4 (2W@850/900MHz)
- Class 1 (1W@1800/1900MHz)
- Control via commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands)
- Short message service
- Free serial port selection
- RTC supported with Super Cap
- Power on/off and reset function supported by Arduino interface

Overview



The indicator LEDs

The GSM Shield has three indicator LEDs for the GSM Shield power, SIM900 power and net status.

GSM Shield power(P):

This LED is used to indicate the power status of the GSM Shield. If the external power supply is connected to the arduino board, then the GSM Shield will get power, this LED will light on.

SIM900 power(S):

This LED is used to indicate the power status of the SIM900. After the SIM900 is power on, the status LED will light on.

Net Status(N):

This LED is used to indicate the net status. The LED will blink slowly or quickly according to different states.



www.tinysine.com

Status	Description
Off	SIM900 is not running
64ms On/800ms Off	SIM900 not registered the network
64ms On/3000ms Off	SIM900 registered to the network
64ms On/300ms Off	GPRS communication is established

The buttons

There are 3 buttons on the GSM shield board.

SIM900 Power Button:

Stack the GSM shield on your arduino main board. The GSM shield will running automatically when you power up your arduino main board. If you want turn on/off the SIM900 module manually. Just press the the sim900 power button one second.

SIM900 Reset Button

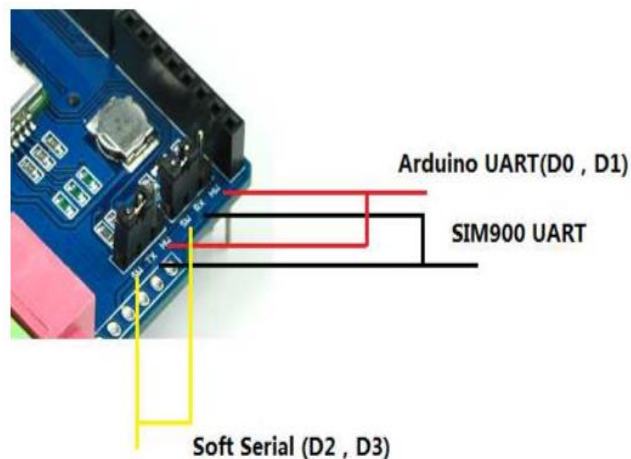
Reset the SIM900 module.

Arduino Reset Button:

Reset the Arduino main board. Same function as the Reset Button which on the arduino main board.

Note: Our test sketch will control D8 pin to turn on the SIM900 when the arduino board is powered up.

Setting Jumper



If you want use UART send AT commands. Set the jumper to HW position. If you use soft serial. Set to SW position.

Our test sketch use soft serial. So we keep it on SW position normally.

If you are using MEGA2560. Just need to connect the middle pins of RX and TX jumpers to RX1 TX1 on Arduino Mega 2560. And change the code setting.

The Arduino Leonardo is a microcontroller board based on the ATmega32u4 (datasheet). It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Leonardo differs from all preceding boards in that the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Leonardo to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port. It also has other implications for the behavior of the board; these are detailed on the getting started page.

This version of the Leonardo comes assembled.

Summary

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	20
PWM Channels	7
Analog Input Channels	12
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz

If you want to give a closer look to this board we advice you to visit the official Arduino Leonardo page in the Hardware Section.



INTRODUCTION

The GPRS/GSM Shield provides you a way to use the GSM cell phone network to receive data from a remote location. The shield allows you to achieve this via any of the three methods:

- Short Message Service
- Audio
- GPRS Service

The GPRS Shield is compatible with all boards which have the same form factor (and pinout) as a standard Arduino Board. The GPRS Shield is configured and controlled via its UART using simple AT commands. Based on the SIM900 module from SIMCOM, the GPRS Shield is like a cell phone. Besides the communications features, the GPRS Shield has 12 GPIOs, 2 PWMs and an ADC.
