

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGIA
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES



**DESARROLLO DE SISTEMAS ELECTRÓNICOS
ENLAZADOS A UNA BASE DE DATOS PARA MEJORAR
LA ATENCIÓN DE LOS PACIENTES INTERNADOS EN
HOSPITALES Y CLÍNICAS DE LA CIUDAD DE LA PAZ**

PROYECTO DE GRADO PRESENTADO PARA LA OBTENCION DEL GRADO DE LICENCIATURA

POR: Univ. ALVARO SAUL PATANA CHAMBI

TUTOR: LIC. JULIO DANIEL LIMA MARQUEZ

LA PAZ-BOLIVIA

2022

DEDICATORIA

El presente Proyecto de Grado va dedicado a Dios quien como guía estuvo presente en el caminar de mi vida, bendiciéndome y dándome la oportunidad de terminar mi carrera universitaria.

A mis Padres Silvestre Patana y Fabiana Chambi, que con su apoyo, amor y confianza permitieron que logre culminar mi carrera profesional

A mis hermanos, por estar siempre ahí apoyándome de manera incondicional.

A mi hijita Ashelem Patana, por su amor incondicional y ser el mejor regalo que me dio la vida.

AGRADECIMIENTOS

Quiero expresar mi gratitud a Dios, quien con su bendición llena siempre mi vida.

A mis padres por ser mi pilar fundamental y haberme apoyado incondicionalmente, pese a las adversidades e inconvenientes que se presentaron durante estos años de formación universitaria.

De manera especial a mi tutor de Proyecto de Grado Lic. Julio Daniel Lima Marquez que gracias a sus consejos y correcciones hoy puedo culminar este proyecto de grado.

Agradezco a todos los docentes y compañeros de la Facultad de Tecnología que con su sabiduría, conocimiento y apoyo me motivaron a seguir siempre adelante.

RESUMEN DEL PROYECTO

El presente proyecto describe el desarrollo de sistemas electrónicos de audio enlazados a una base de datos para mejorar la atención de hospitales y clínicas en la ciudad de La Paz.

Esta idea surgió de la necesidad existente en hospitales y clínicas de la ciudad de La Paz, donde por falta de personal, muchos pacientes quedan desatendidos, algunos de ellos pasan por dolores o incluso alguna emergencia que no logra ser atendida en su momento ya que el personal de cada centro hospitalario no es lo suficientemente eficiente para atender a todos.

Viendo este problema se plantea desarrollar un sistema electrónico de audio que sea capaz de contactar a cada paciente de forma remota, a través de dispositivos que reciben el nombre de intercomunicadores esclavos controlados por un intercomunicador maestro. Cuando un paciente requiera atención médica por alguna dolencia o necesidad, podrá presionar un botón que le permite realizar una llamada al personal de enfermería que se encuentra controlando todos los dispositivos desde una computadora. El personal de enfermería podrá responder al llamado del paciente o acudir a su asistencia sabiendo mediante una base de datos, su nombre, su diagnóstico, y si se trata de un paciente de emergencia

Esta comunicación se realizó a través de circuitos preamplificadores de micrófono y amplificadores de audio que son activados mediante un microcontrolador. El paciente solo podrá comunicarse con el personal si el personal médico así lo decide. Sin embargo, también existe la opción de configurarlo como paciente de emergencia, lo cual hará que, al llamado del paciente, la llamada se active automáticamente y se active una alarma desde la base de datos indicando la prioridad del paciente.

INDICE

CAPÍTULO I.....	I
ANTECEDENTES DEL PROYECTO	I
1.1. INTRODUCCIÓN.....	II
1.2. PLANTEAMIENTO DEL PROBLEMA.....	III
1.2.1. IDENTIFICACIÓN DEL PROBLEMA	III
1.2.2. FORMULACIÓN DEL PROBLEMA	III
1.3. OBJETIVOS.....	IV
1.3.1. OBJETIVO GENERAL	IV
1.3.2. OBJETIVOS ESPECÍFICOS	IV
1.4. JUSTIFICACIÓN.....	V
1.4.1. JUSTIFICACIÓN ECONÓMICA	V
1.4.2. JUSTIFICACIÓN TECNOLÓGICA.....	V
1.4.3. JUSTIFICACIÓN SOCIAL.....	V
1.5. DELIMITACIÓN	VI
1.5.1. DELIMITACIÓN TEMPORAL	VI
1.5.2. DELIMITACIÓN ESPACIAL	VI
1.6. METODOLOGIA.....	VII
1.6.1. ACTIVIDADES Y TAREAS	VII
1.6.2. MÉTODOS Y TÉCNICAS	VII
CAPÍTULO II.....	1
MARCO TEÓRICO.....	1
2.1. SONIDO.....	2
2.1.1. DEFINICIÓN	2
2.1.2. PROPIEDADES Y CARACTERÍSTICAS	2

2.1.3.	FRECUENCIA DEL SONIDO.....	3
2.2.	AMPLIFICADORES DE SONIDO	4
2.2.1.	DEFINICIÓN	4
2.2.2.	CIRCUITOS PARA AMPLIFICAR EL AUDIO	6
2.2.2.1.	PREAMPLIFICADOR.....	6
2.2.2.1.1.	DEFINICIÓN.....	6
2.2.2.1.2.	CARACTERÍSTICAS DE UN PREAMPLIFICADOR IDEAL.....	7
2.2.2.2.	AMPLIFICADOR DE POTENCIA.....	8
2.2.2.2.1.	AMPLIFICADORES DE POTENCIA INDEPENDIENTES.....	8
2.2.2.2.2.	MEZCLADORAS AUTO-AMPLIFICADAS (MEZCLADORAS CON AMPLIFICADORES DE POTENCIA INCORPORADOS)	8
2.2.2.2.3.	ALTAVOCES ACTIVOS (ALTAVOCES CON AMPLIFICADORES DE POTENCIA INCORPORADOS)	9
2.2.3.	MICRÓFONO	10
2.2.3.1.	DEFINICIÓN.....	10
2.2.3.2.	TIPOS DE MICRÓFONOS.....	10
2.2.3.2.1.	MICRÓFONOS CONDENSADORES	10
2.2.3.2.2.	MICRÓFONOS DINÁMICOS	11
2.2.3.2.3.	MICRÓFONOS DE LISTÓN.....	12
2.2.3.3.	PARTES	13
2.2.3.3.1.	DIAFRAGMA.....	13
2.2.3.3.2.	DISPOSITIVO TRANSDUCTOR (ELEMENTO O CÁPSULA).....	14
2.2.3.3.3.	REJILLA.....	14
2.2.3.3.4.	CARCASA.....	14
2.2.3.4.	PATRÓN POLAR	15
2.2.3.4.1.	OMNIDIRECCIONAL.....	15

2.2.3.4.2.	UNIDIRECCIONAL	15
2.2.3.4.3.	CARDIOIDE.....	16
2.2.4.	PARLANTES	16
2.3.	MICROCONTROLADORES	17
2.3.3.	DEFINICIÓN	17
2.3.4.	ELEMENTOS INTERNOS EN UN MICROCONTROLADOR	18
2.3.4.4.	PROCESADOR O MICROPROCESADOR.....	18
2.3.4.5.	PERIFÉRICOS.....	19
2.3.4.6.	MEMORIAS	19
2.3.4.6.1.	MEMORIA DE PROGRAMA – FLASH.....	19
2.3.4.6.2.	MEMORIA DE DATOS – RAM	20
2.3.4.6.3.	REGISTROS.....	20
2.3.4.6.4.	MEMORIA EXTERNA.....	20
2.4.	OPTOACOPLADOR.....	21
2.4.1.	DEFINICIÓN	21
2.4.2.	COMPONENTES.....	21
2.4.3.	APLICACIONES.....	22
2.5.	LENGUAJE PROGRAMACIÓN PYTHON	23
2.6.	BASE DE DATOS	24
2.6.1.	DEFINICIÓN	24
2.6.2.	DIFERENCIA ENTRE UNA BASE DE DATOS Y UNA HOJA DE CÁLCULO	
	25	
CAPÍTULO III.....		26
INGENIERÍA DEL PROYECTO.....		26
3.1.	DESCRIPCIÓN DEL PROYECTO	27
3.1.1.	DIAGRAMA EN BLOQUES.....	28

3.1.1.1.	ETAPA DE INICIACIÓN DE LLAMADA	28
3.1.1.2.	CAPTURA DE AUDIO	29
3.1.1.3.	CONTROLADOR.....	29
3.1.1.4.	COMPUTADORA CENTRAL.....	29
3.1.1.5.	BASE DE DATOS	29
3.1.1.6.	SALIDA DE AUDIO	29
3.1.2.	ALGORITMO	30
3.2.	DISEÑO DEL PROYECTO	31
3.2.1.	DISPOSITIVO CONTROLADOR.....	31
3.2.2.	DISPOSITIVOS ESCLAVOS	39
3.2.3.	SOFTWARE EN COMPUTADORA Y ARDUINO.....	41
3.2.4.	BASE DE DATOS.....	45
3.3.	FUNCIONAMIENTO DEL PROYECTO.....	47
CAPITULO IV		58
COSTOS DEL PROYECTO		58
4.1.	EVALUACIÓN DE RESULTADOS	59
4.1.	INVERSIÓN DEL PROYECTO.....	61
4.2.	CÁLCULO DEL COSTO BENEFICIO	65
4.3.	CÁLCULO DE LA VIABILIDAD DEL PROYECTO.....	65
4.4.	FACTIBILIDAD ECONÓMICA DEL PROYECTO.....	66
CAPITULO V.....		70
CONCLUSIONES Y RECOMENDACIONES		70
5.1.	CONCLUSIONES	71
5.2.	RECOMENDACIONES	71
REFERENCIAS BIBLIOGRAFICAS		73

ANEXOS.....	75
ANEXO 1. CÓDIGO DEL PROYECTO (PYTHON).....	75
ANEXO 2. CÓDIGO DEL PROYECTO (ARDUINO).....	87

INDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 FORMA DEL SONIDO.....	2
ILUSTRACIÓN 2 FRECUENCIA DEL SONIDO	4
ILUSTRACIÓN 3 AMPLIFICADOR DE AUDIO CON CONTROLES.....	5
ILUSTRACIÓN 4 EJEMPLO DE UNA MALA GANANCIA	6
ILUSTRACIÓN 5 AMPLIFICADOR DE POTENCIA INDEPENDIENTE	8
ILUSTRACIÓN 6 MEZCLADORAS CON AMPLIFICADORES DE POTENCIA	9
ILUSTRACIÓN 7 ALTAVOCES ACTIVOS.....	9
ILUSTRACIÓN 8 MICRÓFONOS DE CONDENSADOR	11
ILUSTRACIÓN 9 MICRÓFONOS DINÁMICOS	12
ILUSTRACIÓN 10 MICRÓFONOS DE LISTÓN	13
ILUSTRACIÓN 11 PARTES DE UN MICRÓFONO	14
ILUSTRACIÓN 12 PARTES DE UN MICROCONTROLADOR	17
ILUSTRACIÓN 13 DIAGRAMA INTERNO DE UN OPTOACOPLADOR.....	21
ILUSTRACIÓN 14 DIFERENTES TIPOS DE OPTOACOPLADOR	22
ILUSTRACIÓN 15 DIAGRAMA GENERAL DEL PROYECTO.....	27
ILUSTRACIÓN 16 DIAGRAMA EN BLOQUES	28

ILUSTRACIÓN 17 DIAGRAMA DE FLUJO DEL PROYECTO	30
ILUSTRACIÓN 18 DIAGRAMA CIRCUITAL DE ETAPA DE PREAMPLIFICADOR.....	31
ILUSTRACIÓN 19 DIAGRAMA SUGERENCIA DE HOJA DE DATOS DE TDA2003	34
ILUSTRACIÓN 20 DIAGRAMA CIRCUITAL DE REGISTROS PARALELO - SERIE	36
ILUSTRACIÓN 21 DIAGRAMA CIRCUITAL DE REGISTROS SERIE - PARALELO	37
ILUSTRACIÓN 22 DIAGRAMA GENERAL DEL DISPOSITIVO MAESTRO	38
ILUSTRACIÓN 23 DIAGRAMA CIRCUITAL DEL DISPOSITIVO ESCLAVO	39
ILUSTRACIÓN 24 INSTALACIÓN DE PYTHON.....	41
ILUSTRACIÓN 25 INSTALACIÓN DE LIBRERÍAS PYTHON	42
ILUSTRACIÓN 26 FORMA DE COMUNICACIÓN ENTRE ARDUINO Y PC CUANDO LOS DISPOSITIVOS ESCLAVOS SON ACTIVADOS	43
ILUSTRACIÓN 27 FORMA DE COMUNICACIÓN ENTRE ARDUINO Y PC PARA REALIZAR UNA LLAMADA	44
ILUSTRACIÓN 28 INICIO DE PHPMYADMIN LUEGO DE INSTALACIÓN	45
ILUSTRACIÓN 29 CREACIÓN DE BASE DE DATOS Y TABLA PARA CONTROL	46
ILUSTRACIÓN 30 CREACIÓN DE COLUMNAS DE TABLA “DISPOSITIVOS”	46
ILUSTRACIÓN 31 CREACIÓN DE COLUMNAS DE TABLA “COM”	47
ILUSTRACIÓN 32 PCB DE PREAMPLIFICADOR Y AMPLIFICADOR.....	47
ILUSTRACIÓN 33 PCB DE LOS REGISTROS DE DESPLAZAMIENTOS CONECTADOS A ARDUINO.....	48
ILUSTRACIÓN 34 PCB DEL RESTO DE CONECTORES A LOS REGISTROS DE	

DESPLAZAMIENTO.....	48
ILUSTRACIÓN 35 PCB DEL DISPOSITIVO ESCLAVO	49
ILUSTRACIÓN 36 TERMINADO EL PCB DEL AMPLIFICADOR	49
ILUSTRACIÓN 37 TERMINADO EL PCB DE LOS REGISTROS Y ARDUINO	50
ILUSTRACIÓN 38 VISTA LATERAL DE LOS CONECTORES A LOS DISPOSITIVOS ESCLAVOS	51
ILUSTRACIÓN 39 FORMA DE CONEXIÓN DEL DISPOSITIVO MAESTRO	52
ILUSTRACIÓN 40 FORMA DE CONEXIÓN ENTRE EL DISPOSITIVO MAESTRO Y UN DISPOSITIVO ESCLAVO	53
ILUSTRACIÓN 41 ACABADO DE LOS DISPOSITIVOS ESCLAVOS.....	53
ILUSTRACIÓN 42 ENTORNO GRÁFICO CREADO CON PYTHON PARA LA BASE DE DATOS	54
ILUSTRACIÓN 43 REGISTRANDO NUEVO PACIENTE EN BASE DE DATOS.....	54
ILUSTRACIÓN 44 RESPUESTA DEL DISPOSITIVO ESCLAVO PARA EL REGISTRO... ..	55
ILUSTRACIÓN 45 HABILITACIÓN DE NUEVO PACIENTE EN LA BASE DE DATOS ..	55
ILUSTRACIÓN 46 HABILITACIÓN DE MÁS PACIENTES EN LA BASE DE DATOS	56
ILUSTRACIÓN 47 LLAMADA DE UNA PACIENTE REGISTRADA EN EL PROGRAMA	57

CAPÍTULO I

ANTECEDENTES DEL PROYECTO

1.1.INTRODUCCIÓN

En salud, una atención de calidad es tan importante, que su ausencia tiene repercusiones no solamente sobre la eficiencia del sistema de salud o la percepción que los ciudadanos tienen de él, sino que, literalmente, puede costarles la vida. Un reporte del Lancet Global Health Commission sobre los sistemas de salud de alta calidad revela que en los países de pequeños y medianos ingresos que son la mayoría en América Latina y el Caribe mueren más de 8 millones de personas por enfermedades que pudieran ser perfectamente tratables por los sistemas de salud. La baja calidad asistencial es ahora un obstáculo para reducir la mortalidad junto al acceso insuficiente a servicios.

La calidad asistencial se convertirá en un impulsor cada vez más importante de la salud poblacional, a medida que aumente la utilización de los sistemas de salud y que la carga de morbilidad recaiga en enfermedades más complejas. (Fuente: https://www.thelancet.com/pb-assets/Lancet/stories/commissions/health-quality-health-systems/TLGH_HealthSystem_ExecSum_Spanish.PDF)

Este problema de baja calidad de la atención en salud se vive actualmente en Bolivia. Los hospitales se han visto saturados de clientes que sobrepasan la cantidad de camas disponibles en salas de especialidades y quedan esperando en los pasillos.

En la ciudad de La Paz de los hospitales públicos el más conocido es el Hospital de Clínicas, el problema en este hospital es que el personal médico y de enfermería es insuficiente frente a la enorme cantidad de pacientes que tienen en cada sala. No llegan a cumplir ni siquiera los requisitos de cantidad del personal mínimo necesario para las Unidades de Terapia Intensiva. En esto son los estudiantes de 6to año de medicina y los residentes, quienes colaboran con la tarea. (Fuente: LA IZQUIERDA DIARIO (2019). Claves para entender la crisis del sistema de salud en Bolivia. La Paz.)

1.2.PLANTEAMIENTO DEL PROBLEMA

1.2.1. IDENTIFICACIÓN DEL PROBLEMA

En clínicas y hospitales de La Paz, los pacientes internados en las diferentes especialidades sufren de una mala atención por parte del personal de enfermería.

Este problema se debe a que el personal tiene que pasar ronda por las salas de internación tratando de dar abasto a la sobrepoblación de pacientes internados.

El personal de enfermería es insuficiente para estar pendiente de todos los pacientes de forma personal.

Las clínicas y hospitales no cuentan con equipos de comunicación entre los pacientes y el personal de enfermería.

Debido a este problema, los pacientes tienen un trato de pésima calidad. Muchos de ellos sufren alguna dolencia, o necesitan asistencia y son descuidados por el personal de enfermería. Pacientes delicados con problemas de infarto podrían llegar a fallecer por solo un descuido del personal.

1.2.2. FORMULACIÓN DEL PROBLEMA

Teniendo en cuenta estos factores, se realiza la siguiente pregunta:

¿Cómo puede mejorarse la atención a pacientes internados en clínicas y hospitales de La Paz?

1.3.OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar sistemas electrónicos enlazados a una base de datos para mejorar la atención de los pacientes internados en hospitales y clínicas de la ciudad de La Paz.

1.3.2. OBJETIVOS ESPECÍFICOS

- Diseñar un circuito receptor de ondas de sonido a través de un micrófono.
- Diseñar un circuito reproductor de audio a través de un parlante.
- Elaborar un programa para microcontrolador que transmita datos de audio a través de cable a varios dispositivos
- Crear un software de computadora que permita la comunicación de audio para recibir y transmitir la voz humana.
- Elaborar una base de datos enlazada al software de comunicación que permita priorizar pacientes delicados.

1.4.JUSTIFICACIÓN

1.4.1. JUSTIFICACIÓN ECONÓMICA

En los hospitales de la ciudad de La Paz no cuentan con este tipo de tecnología, ya que un sistema como este es demasiado costoso y el hospital prefiere no realizar ese gasto. El sistema que se implementará utilizará componentes de bajo coste, pero de alto rendimiento.

1.4.2. JUSTIFICACIÓN TECNOLÓGICA

Actualmente la tecnología tiene un gran auge en cuanto intercomunicadores. Se aprovechará este tipo de tecnología para transmitir audio y poder comunicar a los pacientes internados con el personal de enfermería.

1.4.3. JUSTIFICACIÓN SOCIAL

Este proyecto tendrá mucha aceptación en la sociedad, porque actualmente la atención a los pacientes en las clínicas y hospitales de La Paz es deficiente y este tipo de sistema será muy aceptado por la sociedad.

1.5.DELIMITACIÓN

1.5.1. DELIMITACIÓN TEMPORAL

El presente proyecto se desarrolló en dos meses, considerando que solo se tiene un dispositivo terminado y realizadas todas las pruebas de calidad. Pero para implementar veinte dispositivos en algún centro hospitalario, se demoraría cuatro días en construirlos.

La implementación del proyecto se realizaría en dos días, tomando en cuenta que solo se tenga veinte dispositivos a comunicar y una capacitación al personal de enfermería para controlar el software.

En total el proyecto está planificado a ser implementado en el lapso de una semana.

1.5.2. DELIMITACIÓN ESPACIAL

El proyecto está planificado para ser implementado de manera inicial solo en la ciudad de La Paz, en clínicas y hospitales que deseen adquirirlo, pero el proyecto podría beneficiar a clínicas y hospitales en toda Bolivia.

1.6.METODOLOGIA

El tipo de metodología utilizada en este proyecto es experimental. Ya que se experimentará la distancia a la que se puede conectar un micrófono sin perder la señal o generar demasiado ruido, se experimentará la calidad de audio en parlantes y la activación de un número de dispositivos sin tener fallas en la activación por falta de corriente o voltaje.

1.6.1. ACTIVIDADES Y TAREAS

Se realizarán estudios acerca de diferentes micrófonos existentes en el mercado y cuál sería el más óptimo para hablar desde cierta distancia. Se estudiarán métodos de controlar un número considerable de dispositivos de manera simultánea y que no conlleve un gran gasto económico

Se investigará un caso particular de algún hospital para saber cuan útil resultará implementar el presente proyecto y que impacto causará en la sociedad

Teniendo ya un prototipo del sistema se realizarán pruebas de calidad en distancias grandes, así como el manejo de grandes cantidades de forma simultánea para poder determinar los límites que tiene el proyecto y la manera de poder subsanar o aumentar el rango de dichos límites.

1.6.2. MÉTODOS Y TÉCNICAS

El método de determinar un micrófono óptimo para este propósito será tomando en cuenta las características y rangos de cada uno. En caso de determinar los dispositivos será leyendo su hoja de datos y realizando simulaciones para verificar si es posible o no un control de muchos dispositivos con un solo microcontrolador.

Para poder investigar un caso particular, se realizará encuestas al personal de salud y a personas que estuvieron internadas en algún centro hospitalario. Estas encuestas no serán documentadas hoja por hoja, sino que solo se obtendrá los datos de un centro en especial,

en este caso el hospital corazón de Jesús, ubicado en la zona Gran Poder de El Alto. Quedando el servicio en el primer piso del centro médico.

Las pruebas no se realizarán en el centro médico, pues no se cuenta con la autorización para hacerlo, pero basta con contar con algún ambiente con distancias de cable y alejar los dispositivos unos de otros.

CAPÍTULO II
MARCO TEÓRICO

2.1.SONIDO

2.1.1. DEFINICIÓN

El sonido es una onda mecánica longitudinal que se propaga a través de un medio elástico. Nuestro oído es capaz de percibir algunos de estos sonidos, y nuestro cerebro los procesa de manera que podemos escucharlos.

El sonido humanamente audible consiste en ondas sonoras y ondas acústicas que se producen cuando las oscilaciones de la presión del aire, son convertidas en ondas mecánicas en el oído humano y percibidas por el cerebro. La propagación del sonido es similar en los fluidos, donde el sonido toma la forma de fluctuaciones de presión. En los cuerpos sólidos la propagación del sonido involucra variaciones del estado tensional del medio.



Ilustración 1 Forma del sonido

Fuente: <https://www.caracteristicas.co/sonido/>

2.1.2. PROPIEDADES Y CARACTERÍSTICAS

El sonido tiene tres cualidades fundamentales, así como todas las ondas en física. Una onda tiene amplitud, frecuencia y contenido armónico, y para el sonido. estas magnitudes

se llaman intensidad, tono y timbre, respectivamente. Pasamos a explicarlas una a una.

La intensidad hace referencia a la amplitud de la onda de sonido. Es la característica más sencilla de comprender: se relaciona con el volumen al que oímos el sonido. Cuánto más fuerte vibren las moléculas del medio, más intensidad de sonido. Normalmente nos referimos a ella empleando una unidad relacionada con la intensidad de sonido: el Decibelio [dB].

El tono es la frecuencia de vibración de la onda de sonido. Se mide en Hertz [Hz], y equivale al número de oscilaciones por segundo. Ya hemos visto que nuestro oído sólo es capaz de escuchar los sonidos con tonos comprendidos entre los 20 Hz y los 20 kHz, aunque por ejemplo muchos animales pueden escuchar por encima de nuestros límites, los conocidos como ultrasonidos. En música, cada nota equivale a una frecuencia concreta, por ejemplo, un La son 440 Hz. cuanto más alta es la frecuencia o tono del sonido, más agudo lo escuchamos.

El timbre es la cualidad del sonido más interesante, pero también más compleja de entender. El sonido, normalmente, no es una sola onda a una frecuencia concreta, sino que está compuesto por varias ondas de varias frecuencias distintas, que, al sumarlas, hace que oigamos el sonido de una forma muy concreta. La onda principal se le conoce como fundamental, y el resto de ondas acopladas tienen el nombre de armónicos. El timbre es el responsable de que distingamos a una persona solo por su voz. Cada uno tiene un timbre único, independientemente de la nota que esté cantando. (Fuente: <https://soportemultimedia.com/que-es-el-sonido/>)

2.1.3. FRECUENCIA DEL SONIDO

La frecuencia de las ondas sonoras determina su tono y se suele medir en hercios (Hz), una unidad que equivale a una onda por segundo. El oído humano es capaz de percibir un amplio rango de frecuencias: aproximadamente de 20 a 20.000 Hz. La amplitud es una descripción de la potencia de la onda de un sonido; el volumen de un sonido aumenta en

función de la amplitud de su onda sonora. La música consiste en mezclar diferentes frecuencias y amplitudes.

(Fuente:

<https://www.apple.com/es/sound/#:~:text=La%20frecuencia%20de%20las%20ondas,de%2020%20a%2020.000%20Hz.>)

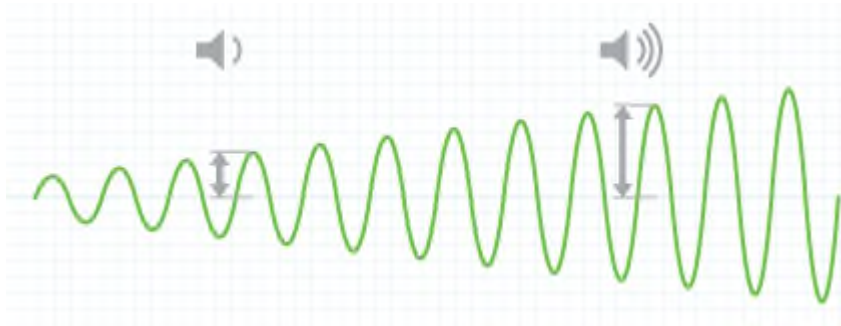


Ilustración 2 Frecuencia del sonido

Fuente:

<https://www.apple.com/es/sound/#:~:text=La%20frecuencia%20de%20las%20ondas,de%2020%20a%2020.000%20Hz.>

2.2.AMPLIFICADORES DE SONIDO

2.2.1. DEFINICIÓN

Un amplificador de audio es un dispositivo electrónico utilizado en la reproducción de sonido, cuya función es aumentar la amplitud de la señal de salida, actuando sobre la tensión de la señal de entrada. La relación entre el nivel de salida y de entrada es la ganancia, de tal manera que la ganancia expresa el grado de amplificación de la señal.

El amplificador, junto con los altavoces, es el componente más importante en un sistema de reproducción de sonido, ya que sin amplificador no podríamos oír nada en los dispositivos multimedia.

Un amplificador es un transformador de tensión que permite transformar una tensión débil

de una señal de audio procedente, por ejemplo, de un reproductor de CD, en una tensión mayor, para transmitirla posteriormente a otro aparato.



Ilustración 3 Amplificador de audio con controles

Fuente: https://www.freepik.es/fotos-premium/mezclador-control-remoto-grabacion-sonido-ingeniero-sonido-trabajo-estudio-amplificador-sonido-mezclador-ecualizador-consola-dj_7686892.htm

Generalmente dispone de conexiones para varias fuentes de sonido, como un reproductor de CD, un sintonizador de radio o instrumentos musicales, que son seleccionadas mediante un conmutador de entrada y reproducidas mediante un regulador progresivo de volumen o potenciómetro. (Fuente: <https://bticino.com.pe/como-funciona-un-amplificador-de-audio-usos-y-tipos/>)

2.2.2. CIRCUITOS PARA AMPLIFICAR EL AUDIO

2.2.2.1.PREAMPLIFICADOR

2.2.2.1.1. DEFINICIÓN

Un preamplificador es un dispositivo electrónico que se encarga de amplificar una señal eléctrica débil hasta convertirla en una señal lo suficientemente fuerte y limpia, de manera que pueda ser procesada por el resto del equipo.

Esta definición es muy general. En el mundo del audio esa señal eléctrica vendría a ser la señal de audio, el equipo que produce esa señal sería un micrófono o un instrumento grabado en línea y el resto del equipo serían los EQ's, compresores, altavoces...

La señal que produce por ejemplo un micrófono tiene una amplitud muy inferior a la que necesitan el resto de tu equipo. Por lo tanto, se necesita mucha ganancia (30-60 dB) para alcanzar el nivel de línea, es decir, el nivel estándar del resto del equipo.

Además, todas las señales de audio son propensas a interferencias, que provocan ruidos indeseados. En las señales débiles el ruido tendrá mucho mayor volumen en relación con la señal.

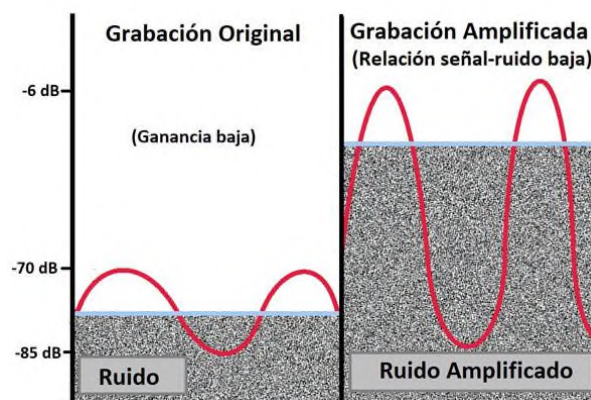


Ilustración 4 Ejemplo de una mala ganancia

<https://www.homestudioproductions.com/que-es-un-preamplificador-por-que-lo-necesitas/>

Es por esto que es tan importante el papel del preamplificador, pues amplifica la señal de manera óptima antes de que sea grabada o procesada. De esta manera el ruido es casi imperceptible. (Fuente: <https://www.homestudioproductions.com/que-es-un-preamplificador-por-que-lo-necesitas/>)

2.2.2.1.2. CARACTERÍSTICAS DE UN PREAMPLIFICADOR IDEAL

Para que podamos decir que un preamplificador es ideal, este debe ser esencialmente lineal; cuando decimos que un preamplificador debe ser lineal, significa que tendrá una ganancia de valor constante en su rango de trabajo. Otras características importantes, es que debe tener una alta impedancia de entrada, con el objetivo de que detecte con una mínima corriente la señal de entrada que se desea amplificar, y una baja impedancia de salida, para que exista un cambio mínimo en el voltaje de salida. El rendimiento de ruido es otro aspecto que se debe destacar en los preamplificadores.

En resumen, estas son las características de trabajo más importantes en un amplificador ideal:

- Linealidad.
- Alta impedancia de entrada.
- Baja impedancia de salida.
- Rendimiento ante el ruido y la interferencia.

Si la ganancia en el preamplificador es alta, la relación señal-ruido (SNR) de la señal final es producto de la SNR de la señal de entrada.

Ahora que conocemos las características que debe tener un preamplificador ideal, es tiempo de conocer los tipos de preamplificadores que existen. (Fuente: <https://amplificadores.info/amplificadores-de-potencia/preamplificador>)

2.2.2.2.AMPLIFICADOR DE POTENCIA

Los amplificadores de potencia son dispositivos que amplifican las señales eléctricas que se han ajustado a un volumen y tono adecuados en la consola, a un nivel que se puede convertir en sonidos acústicos por los altavoces. Hay varios tipos y modelos de amplificadores de potencia, varían en tamaño, forma y potencia de salida, y se adaptan a diferentes situaciones.

Hay tres tipos principales de amplificadores de potencia, cuyas características se describen a continuación.

2.2.2.2.1. AMPLIFICADORES DE POTENCIA INDEPENDIENTES.

Este es un tipo de amplificador de potencia para uso general. Los amplificadores de potencia como lo dice su nombre amplifican las señales enviadas desde las mezcladoras para hacer que los altavoces produzcan sonido. Se pueden instalar varios amplificadores de potencia en un rack, lo que permite construir un sistema de PA a gran escala con facilidad.



Ilustración 5 Amplificador de potencia independiente

Fuente:

<https://mx.yamaha.com/es/products/contents/proaudio/musicianspa/equipments/amplifier.html>

2.2.2.2.2. MEZCLADORAS AUTO-AMPLIFICADAS (MEZCLADORAS CON AMPLIFICADORES DE POTENCIA INCORPORADOS)

Este tipo de mezcladora incluye un amplificador de potencia incorporado. Como la fuente

de alimentación que requieren la mezcladora y el amplificador de potencia están integradas al sistema de amplificación, hace falta solo una conexión de alimentación, y se requieren menos conexiones que al utilizar amplificadores independientes.



Ilustración 6 Mezcladoras con amplificadores de potencia

Fuente:

<https://mx.yamaha.com/es/products/contents/proaudio/musicianspa/equipments/amplifier.html>

2.2.2.2.3. ALTAVOCES ACTIVOS (ALTAVOCES CON AMPLIFICADORES DE POTENCIA INCORPORADOS)

Este tipo de altavoz incluye un amplificador de potencia incorporado. Puesto que el amplificador de potencia incorporado está diseñado específicamente para el altavoz, la combinación del altavoz y el amplificador de potencia ofrecen una compatibilidad superior, permitiendo que el altavoz rinda al máximo. La salida de la mezcladora se puede conectar directamente al altavoz activo, por lo que se necesitan menos cables.



Ilustración 7 Altavoces activos

Fuente:

<https://mx.yamaha.com/es/products/contents/proaudio/musicianspa/equipments/amplifier.html>

2.2.3. MICRÓFONO

2.2.3.1.DEFINICIÓN

Es un instrumento que recoge el sonido y los transmite cambiando la naturaleza de la señal captada. Puede afirmarse que se trata de un transductor, es decir, un aparato encargado de transformar la energía acústica (por ejemplo, la voz) en energía eléctrica.

Es el funcionamiento inverso a lo que hacen los altavoces o los amplificadores, los cuales transforman la energía eléctrica en sonido. Existen varios tipos de micrófonos, sin embargo, su funcionamiento es similar en todos. (Fuente: <https://mx.yamaha.com/es/products/contents/proaudio/musicianspa/equipments/amplifier.html>)

2.2.3.2.TIPOS DE MICRÓFONOS

Existen tres tipos de micrófonos: Condensadores, dinámicos y de listón.

2.2.3.2.1. MICRÓFONOS CONDENSADORES

Son los micrófonos que ves tradicionalmente en fotografías de estudios profesionales. Son muy llamativos, finos, con clase y robustos. Este se subdivide en condensador con diafragma grande y con diafragma chico, y se usan dependiendo la necesidad:

Ambos modelos son parecidos con la diferencia que el diafragma grande puede capturar la señal de forma más amplia mientras que el de diafragma chico lo hace de forma más localizada. Estos micrófonos requieren de un poder externo para que puedan funcionar, se llama Phantom Power y se ve como un switch. Este viene generalmente integrado a la interfaz de audio y es fácil de encender.

Las características que destacan de los micrófonos condensadores son:

Sonido claro, detallado y fino.

Sonoramente son los más coloridos.

Altamente sensibles al sonido.



Ilustración 8 Micrófonos de condensador

Fuente: <https://como-funciona.co/un-microfono/>

2.2.3.2.2. MICRÓFONOS DINÁMICOS

Estos son los que sueles ver en los sistemas karaoke o en las situaciones de audios en vivo. También son utilizados en estudios de grabación en distintos casos. Entre sus características se pueden mencionar:

Efectivos si la fuente está relativamente cerca y si predominan sonidos bajos y medios (voces mayormente).

No representa bien las frecuencias agudas.

Son resistentes, baratos, duraderos y no requieren de poder externo.

Soportan presiones altas de sonido (SPL)



Ilustración 9 Micrófonos dinámicos

Fuente: <https://como-funciona.co/un-microfono/>

2.2.3.2.3. MICRÓFONOS DE LISTÓN

El micrófono de listón tal vez sea el menos popular por que suelen ser más caros por la manera en que están confeccionados. Son frágiles y requieren cuidado al manipularlos. Son llamados de listón porque tienen una hoja de metal frágil que capta el sonido y que se puede romper facialmente si la presión del sonido alcanza niveles altos.

En las versiones actuales esto casi no sucede, pero es mejor ser precavido. Sus características son:

Caros y frágiles.

El sonido es robusto, meloso y opaco.

Sensibles a la presión ejercida por el sonido

Populares usando técnicas de posicionamiento lejano del micrófono. (Fuente: <https://como-funciona.co/un-microfono/>)



Ilustración 10 Micrófonos de listón

3. Fuente: <https://como-funciona.co/un-microfono/>

2.2.3.3.PARTES

2.2.3.3.1. DIAFRAGMA

Es la parte más delicada de un micrófono. En algunos lugares también recibe el nombre de pastilla, aunque generalmente este término se refiere al dispositivo que capta las vibraciones en los instrumentos como, por ejemplo, en una guitarra eléctrica. El diafragma es una membrana que recibe las vibraciones sonoras y está unido al sistema que transforma estas ondas en electricidad.

2.2.3.3.2. DISPOSITIVO TRANSDUCTOR (ELEMENTO O CÁPSULA)

El dispositivo transductor sensible de un micrófono se llama «elemento» o «cápsula». Esta cápsula microfónica puede estar construida de diferentes maneras y, dependiendo del tipo de transductor, se pueden clasificar los micrófonos como dinámicos, de condensador, de carbón o piezoeléctricos.

2.2.3.3.3. REJILLA

Protege el diafragma. Evita tanto los golpes de sonido (las “p” y las “b”) así como los daños físicos que sufra por alguna caída.

2.2.3.3.4. CARCASA

Es el recipiente donde se colocan los componentes del micrófono. En los de mano, que son los más comunes, esta carcasa es de metales poco pesados, ligeros de portar pero resistentes a la hora de proteger el dispositivo transductor.



Ilustración 11 Partes de un micrófono

3. Fuente: <https://como-funciona.co/un-microfono/>

2.2.3.4.PATRÓN POLAR

2.2.3.4.1. OMNIDIRECCIONAL

La respuesta de un micrófono omnidireccional (o no direccional) se considera generalmente que es una esfera perfecta en tres dimensiones. En el mundo real, este no es el caso. Como con los micrófonos direccionales, el patrón polar de un micrófono “omnidireccional” es una función de la frecuencia. El cuerpo del micrófono no es infinitamente pequeño y, como consecuencia, tiende a interferir en su propio campo con respecto a los sonidos que llegan desde la parte trasera, provocando un ligero aplanamiento de la respuesta polar. Este aplanamiento aumenta a medida que el diámetro del micrófono (asumiendo que es cilíndrico) llega a la longitud de onda de la frecuencia en cuestión. Por lo tanto, el micrófono de diámetro más pequeño da las mejores características omnidireccionales a altas frecuencias.

La longitud de onda del sonido a 10 kHz es poco más de una pulgada (3,4 cm). Los micrófonos de medición más pequeños suelen ser de 1/4" (6 mm) de diámetro, lo que prácticamente elimina la direccionalidad incluso hasta de las frecuencias más altas. Los micrófonos omnidireccionales, a diferencia de los cardioides, no emplean cavidades resonantes, por lo que pueden ser considerados los micrófonos “más puros” en términos de baja coloración; agregan muy poca distorsión al sonido original. Ser sensible a la presión puede requerir una respuesta de baja frecuencia muy plana hasta los 20 Hz o por debajo, por lo que los micrófonos sensibles a la presión también responden mucho menos al ruido del viento y a las oclusivas (velocidad sensible) que los micrófonos direccionales.

2.2.3.4.2. UNIDIRECCIONAL

Un micrófono unidireccional es sensible a los sonidos de una sola dirección, el micrófono está orientado hacia arriba. La intensidad del sonido de una frecuencia particular se mide perimetralmente de 0 a 360°. Los diagramas profesionales muestran estas escalas e incluyen varias gráficas con diferentes frecuencias. Los diagramas anteriores solo

proporcionan una visión general de las formas típicas de los patrones habituales, y facilitan sus nombres.

2.2.3.4.3. CARDIOIDE

El micrófono unidireccional más común es el micrófono cardioide, llamado así debido a que el patrón de sensibilidad tiene “forma de corazón”, es decir, una curva cardioide. La familia de micrófonos cardioides se utilizan comúnmente como micrófonos vocales o del habla, ya que son buenos en el rechazo de los sonidos de otras direcciones. En tres dimensiones, el cardioide tiene la forma de una manzana, centrada alrededor del micrófono que sería el «tallo» de la manzana. La respuesta cardioide reduce la captación trasera y desde los lados, ayudando a evitar la retroalimentación de los monitores. Estos micrófonos son direccionales respecto al gradiente de presión del transductor, por lo que ponerlos muy cerca de la fuente de sonido (a distancias de unos pocos centímetros) se traduce en un refuerzo de los graves. Esto se conoce como el “efecto de proximidad”. (Fuente: <https://soportemultimedia.com/que-es-el-microfono/>)

2.2.4. PARLANTES

Un altavoz es un transductor electroacústico, es decir, convierte energía eléctrica en energía acústica. Esta conversión tiene lugar en dos etapas: la señal eléctrica produce el movimiento del diafragma del altavoz y este movimiento produce a su vez ondas de presión (sonido) en el aire que rodea al altavoz.

La cantidad de aire que debe moverse depende de la potencia sonora deseada y de la frecuencia. Es muy difícil construir un altavoz que funcione en todo el espectro de frecuencias audible. Para producir un nivel acústico determinado a bajas frecuencias, es necesario mover una gran cantidad de aire, mientras que en los agudos se obtiene el mismo nivel acústico con una menor cantidad de aire.

(Fuente: <https://www.ehu.eus/acustica/espanol/electricidad/altaves/altaves.html>)

2.3. MICROCONTROLADORES

2.3.3. DEFINICIÓN

El Microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Es como una pequeña computadora que incluye sistemas para controlar elementos de entrada/salida. También incluye a un procesador y por supuesto memoria que puede guardar el programa y sus variables (flash y RAM). Funciona como una mini PC. Su función es la de automatizar procesos y procesar información.

El microcontrolador se aplica en toda clase de inventos y productos donde se requiere seguir un proceso automático dependiendo de las condiciones de distintas entradas.

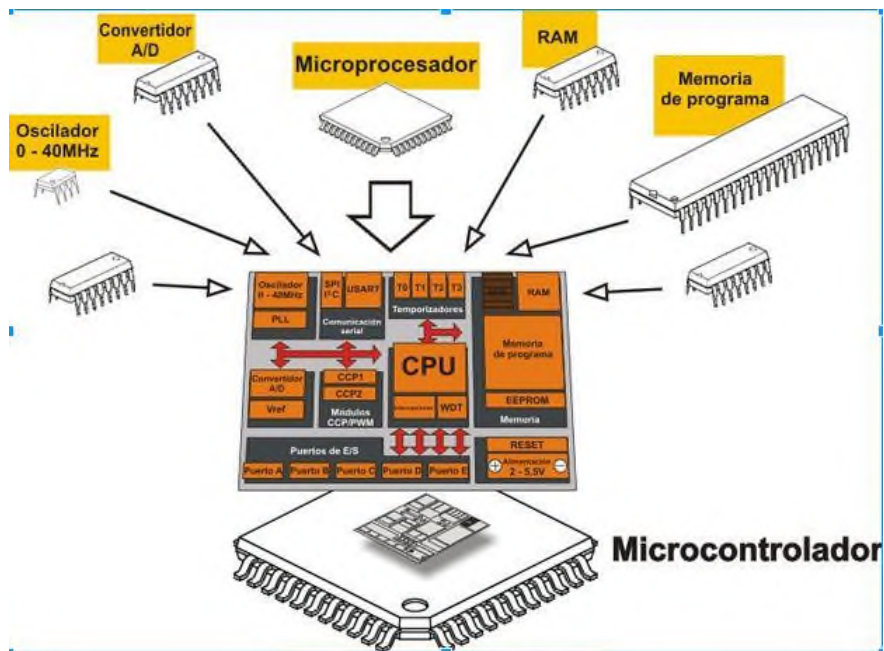


Ilustración 12 Partes de un microcontrolador

Fuente: <https://aprendiendoarduino.wordpress.com/2015/03/23/microcontroladores/>

2.3.4. ELEMENTOS INTERNOS EN UN MICROCONTROLADOR

2.3.4.4. PROCESADOR O MICROPROCESADOR.

Un microprocesador es el cerebro encargado de realizar toda las instrucciones y operaciones aritméticas y lógicas. Es quien controla al resto de elementos según las instrucciones que lea en una memoria de programa. Un procesador incluye al menos tres elementos, ALU, unidad de control y registros.

- ALU. También conocida como Unidad Aritmética y Lógica. Esta unidad está compuesta por los circuitos electrónicos digitales del tipo combinatorios (compuertas, sumadores, multiplicadores), cuya principal función es el realizar operaciones. Estas operaciones están divididas en tres tipos: Lógicas. Como las operaciones básicas de las compuertas lógicas, como la suma lógica (OR), multiplicación lógica (AND), diferencia lógica (XOR) y negación (NOT). Una operación lógica sólo puede tener como entradas y como salidas una respuesta lógica (0 o 1). Esto dependiendo de los niveles de voltajes de una señal digital. Aritméticas. Las operaciones aritméticas son la suma, resta, multiplicación y división. Dependiendo del procesador (8, 16, 32 o 64 bits) será la rapidez con la que se pueden hacer dichas operaciones. Misceláneas. En estas operaciones caen todas las demás operaciones como la transferencia de bits.
- Unidad de control. La unidad de control es el conjunto de sistemas digitales secuenciales (aquellos que tienen memoria) que permiten distribuir la lógica de las señales.
- Registros. Los registros son las memorias principales de los procesadores, ya que funcionan a la misma velocidad que el procesador a diferencia de otras memorias un tanto más lentas (como la RAM, FLASH o la CACHE). Los registros están contruidos por Flip-Flops. Los Flip-Flops son circuitos digitales secuenciales.

2.3.4.5.PERIFÉRICOS

Los periféricos son los circuitos digitales que permiten una interacción con el mundo “exterior” al microcontrolador. Su función es la de poder habilitar o deshabilitar las salidas digitales, leer sensores analógicos, comunicación con terminales digitales o sacar señales analógicas de una conversión digital.

- Puertos de entrada/salida paralelos. Los puertos están relacionados al tamaño del procesador, es decir que un puerto de 8 bits es porque el procesador es de 8 bits. Un procesador de 64 bits, tiene la capacidad de tener un puerto de 64 bits.
- Puertos seriales. Permiten transformar la información digital paralela (bytes de información) en tramas que se pueden transferir por una o varias líneas de comunicación. Existen, por ejemplo: puerto serial, i2c, SPI, USB, CAN, etc.
- Periféricos analógicos. Convierten señales analógicas a digitales (ADC) o señales digitales a analógicas (DAC).

2.3.4.6.MEMORIAS

La memoria de los microcontroladores puede dividirse en cuatro tipos distintos: para el programa, generalmente es una memoria interna del tipo FLASH. También para las variables de los programas se llama RAM. Los registros de propósito general que utiliza el procesador para guardar los resultados de las operaciones, así como los datos que se traen y llevan a la RAM y finalmente la memoria externa que se utiliza para guardar mediciones y/o datos de calibración.

2.3.4.6.1. MEMORIA DE PROGRAMA – FLASH

La memoria de programa es la que se utiliza cuando se programa una aplicación. Cada instrucción del lenguaje ensamblador o de lenguaje C o de un lenguaje de alto nivel, es convertida a instrucciones máquina que requieren de un tamaño particular de bits para ser guardadas. Estos bits o instrucciones se guardan en la FLASH.

Si se tiene poca FLASH, entonces sólo se puede hacer programas pequeños.

2.3.4.6.2. MEMORIA DE DATOS – RAM

La memoria de datos se utiliza cada vez que se agrega una variable al programa. Como la variable seguramente se utiliza como entrada para un proceso o cálculo, está requiere de estar disponible a una velocidad relativamente rápida. Este tipo de información se guarda en la RAM – Random Access Memory ó Memoria de Acceso Aleatorio por sus siglas en inglés.

Si se tiene poca RAM, la aplicación no podrá tener muchas variables.

2.3.4.6.3. REGISTROS

Los registros son las memorias digitales más rápidas. Se construyen con Flip-Flops y generalmente funcionan a una velocidad cercana a la del procesador. En algunos procesadores, también incluyen un tipo de memoria llamada CACHE. Está no puede guardar operaciones y sólo es un puente entre el procesador y la memoria principal.

2.3.4.6.4. MEMORIA EXTERNA

Se utiliza cuando se quiere guardar parte de la información recolectada o procesada. Se suelen usar también para procesos de calibración. Existen dos distintos tipos de memorias no volátiles que pueden ser usadas externamente para los microcontroladores.

- EEPROM – Memoria Electricamente Borrable de Solo Lectura. Ojo, también se pueden programar y escribir.
- FLASH – Como la que tienen internamente los microcontroladores. Pero estas suelen funcionar a una velocidad mucho más lenta a través de protocolos seriales como I2C, Serial o SPI.

(Fuente: <https://hetpro-store.com/TUTORIALES/microcontrolador/>)

2.4. OPTOACOPADOR

2.4.1. DEFINICIÓN

Un optoacoplador también llamado optoaislador, es un circuito electrónico que funciona como un interruptor aislado ópticamente. Es decir, que permite una conexión eléctricamente aislada entre dos circuitos que operan a distintos voltajes. Esta construido por un led y un circuito de control activado por luz infrarroja. Entre otras cosas, una de las ventajas principales de los optoacopladores es su aislación eléctrica entre la carga y la electrónica de control. La única conexión entre ambos elementos es la luz del led que activa al foto-transistor. En la siguiente figura se muestra un diagrama general para un optoacoplador con salida a foto-transistor.

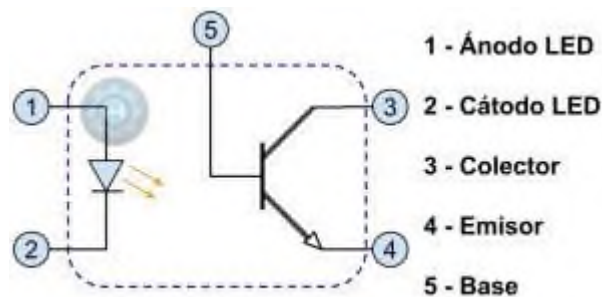


Ilustración 13 Diagrama interno de un optoacoplador

Fuente: <https://hetpro-store.com/TUTORIALES/optoacoplador/>

2.4.2. COMPONENTES

Un optoacoplador está diseñado con dos elementos principalmente. El primero es un LED infrarrojo, este dispositivo activa “remotamente” al opto-transistor. El segundo elemento es el dispositivo electrónico de control. Dependiendo del tipo, este puede ser un opto-transistor, un TRIAC, un transistor Darlington, SCR o una compuerta digital. Por ejemplo, un optoacoplador común es el 4N25, este incluye como dispositivo para controlar, a un opto-transistor. Por el contrario, el MOC3011 incluye a un TRIAC activado ópticamente.

Finalmente, el propósito de led es el activar al elemento de control.

Los optoacopladores se pueden clasificar de acuerdo al tipo de elemento de controlador que tengan.

- TRIAC
- Transistor
- TRIAC con detector de cruce por cero
- Transistor Darlington
- Lógica
- Mosfet

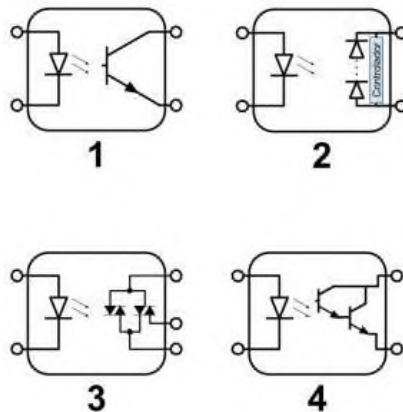


Ilustración 14 Diferentes tipos de optoacoplador

Fuente: <https://hetpro-store.com/TUTORIALES/optoacoplador/>

2.4.3. APLICACIONES

Las aplicaciones de optoacopladores incluyen el de activar cargas que puedan inducir ruido eléctrico al sistema de control. Cuando una carga inductiva como un motor se activa

y desactiva produce perturbaciones como por ejemplo eléctricas en la alimentación del sistema. Incluso cargas que consumen mucha potencia de la fuente pueden drenar momentáneamente el voltaje o la corriente que dicha fuente suministra. Los optoacopladores se usan para aislar a estas perturbaciones electrónicas.

Comúnmente se usan a los optoacopladores con otros elementos de control como MOSFET's, TRIACS, transistores de potencia, relevadores mecánicos o relevadores de estado sólido. En este caso, cuando se usan en conjunto con otros circuitos electrónicos, el objetivo es aislar a la fuente del sistema de control de las perturbaciones que puedan ocasionar el encendido o apagado de los actuadores como motores, luces, etc.

Generalmente no se usan solos debido a que no tienen demasiada capacidad para disipar mucha potencia. En otras palabras, está limitada en cuanto a la corriente y el voltaje que pueden pasar por sus terminales de control, es por eso que se recomienda usarlos en conjunto con otros elementos de mayor potencia de operación como lo sería un relevador o un TRIAC. (Fuente: <https://hetpro-store.com/TUTORIALES/optoacoplador/>)

2.5. LENGUAJE PROGRAMACIÓN PYTHON

Python es un lenguaje de programación de propósito general, que es otra forma de decir que puede ser usado para casi todo. Lo más importante es que se trata de un lenguaje interpretado, lo que significa que el código escrito no se traduce realmente a un formato legible por el ordenador en tiempo de ejecución.

Este tipo de lenguaje también se conoce como “lenguaje de scripting” porque inicialmente fue pensado para ser usado en proyectos sencillos.

El concepto de “lenguaje de scripting” ha cambiado considerablemente desde su creación, porque ahora se utiliza Python para programar grandes aplicaciones de estilo comercial, en lugar de sólo las simples aplicaciones comunes.

Una encuesta realizada en 2019 entre los usuarios de Python indicó que los usos más

populares eran para el desarrollo web y el análisis de datos. Sólo alrededor del 6 % de los encuestados lo utilizaron para el desarrollo de juegos o el desarrollo de aplicaciones.

Esta dependencia de Python ha crecido aún más a medida que Internet se ha hecho más popular. Una gran mayoría de las aplicaciones y plataformas web dependen de su lenguaje, incluido el motor de búsqueda de Google, YouTube, y el sistema de transacciones orientado a la web de la Bolsa de Nueva York (NYSE).

En definitiva, es un lenguaje realmente importante cuando se encarga de impulsar un sistema bursátil. De hecho, la NASA lo utiliza cuando programan sus equipos y maquinaria espacial.

Existen muchas aplicaciones comerciales para la programación en Python, pero el lenguaje también se ha afianzado en los círculos académicos, especialmente entre los que trabajan con grandes cantidades de datos.

También, puede ser usado para procesar texto, mostrar números o imágenes, resolver ecuaciones científicas y guardar datos. (Fuente: <https://www.cursosaula21.com/que-es-python/>)

2.6. BASE DE DATOS

2.6.1. DEFINICIÓN

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar

la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

2.6.2. DIFERENCIA ENTRE UNA BASE DE DATOS Y UNA HOJA DE CÁLCULO

Tanto las bases de datos como las hojas de cálculo (como Microsoft Excel) son modos cómodos de almacenar información. Las principales diferencias entre los dos son:

- Cómo se almacenan y se manipulan los datos
- Quién puede acceder a los datos
- Cuántos datos pueden almacenarse

Las hojas de cálculo se diseñaron originalmente para un usuario y sus características así lo reflejan. Son perfectas para un único usuario o para un pequeño número de usuarios que no necesiten hacer una gran manipulación de datos increíblemente complicada. Las bases de datos, por otro lado, están diseñadas para contener recopilaciones mucho más grandes de información organizada, a veces en cantidades masivas. Las bases de datos permiten que muchos usuarios accedan y consulten los datos de forma rápida y segura al mismo tiempo mediante una lógica y un lenguaje muy complejos. (Fuente: <https://www.oracle.com/mx/database/what-is-database/>)

CAPÍTULO III
INGENIERÍA DEL PROYECTO

3.1.DESCRIPCIÓN DEL PROYECTO

Para lograr el objetivo planteado se realizó el siguiente proyecto mostrado en diagrama:

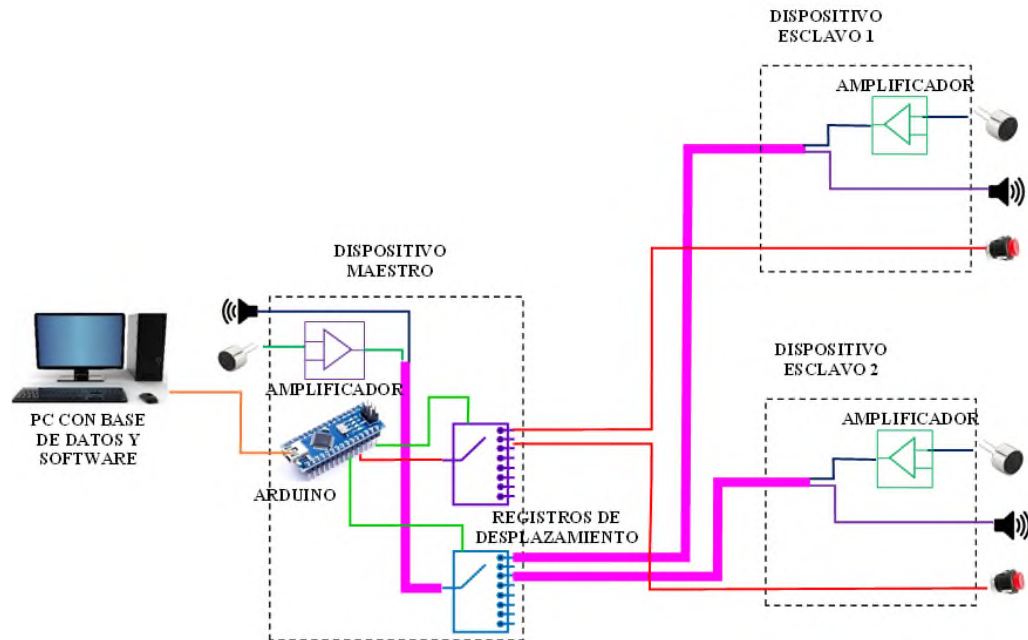


Ilustración 15 Diagrama general del proyecto

Fuente propia

Se tiene un intercomunicador especial; donde un dispositivo hará de maestro logrando activar y desactivar a los demás dispositivos esclavos. También se conectó a una computadora a través de un puerto USB. Para poder manejar este dispositivo maestro se necesita una computadora, y en esta se tendrá instalada una base de datos donde se guardará los pacientes a los que pertenecen los dispositivos esclavos y donde se hablará mediante el micrófono y se escuchará mediante audífonos a cada uno de los pacientes.

Los dispositivos esclavos serán instalados en cada cama de un paciente. Este dispositivo esclavo incluye un pulsador, un micrófono y un altavoz o parlante que sirve para la comunicación con el maestro. El pulsador se utiliza para alertar al maestro que el dispositivo pide comunicarse con él.

Se podrá implementar cuantos dispositivos esclavos se necesiten, gracias a registros de desplazamiento que estarán en el dispositivo maestro.

3.1.1. DIAGRAMA EN BLOQUES

A continuación, se muestra el diagrama en bloques del proyecto:

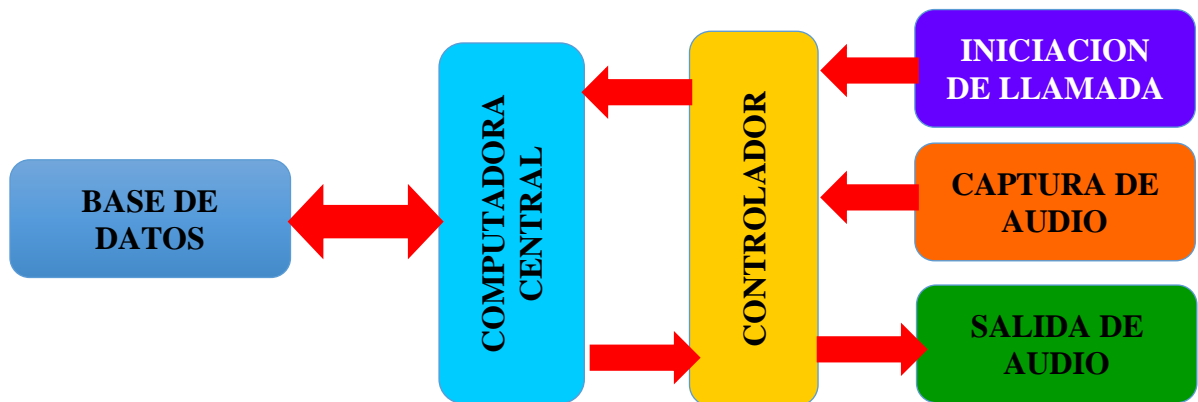


Ilustración 16 Diagrama en bloques

Fuente propia

Este diagrama en bloques está dimensionado para un solo dispositivo esclavo, en caso de contar con más esclavos, se multiplican los bloques de captura de audio y salida de audio. Este diagrama consta de las siguientes etapas:

3.1.1.1. ETAPA DE INICIACIÓN DE LLAMADA

Esta etapa consiste primeramente en un pulsador que inicia sesión de comunicación de audio para que los pacientes se comuniquen con el personal de enfermería.

En la base de datos se puede determinar si es un paciente de emergencia o no, y en el caso de ser un paciente de emergencia la comunicación se iniciará inmediatamente. Esto solo se permitirá cuando por base de datos se estipule que se trata de un paciente delicado o con algún problema de comunicación oral.

3.1.1.2.CAPTURA DE AUDIO

Esta etapa consta de un micrófono y un amplificador de la señal transmitida por el micrófono. La señal resultante debe variar entre 0 y 12 voltios que serán transmitidos a la etapa de registros de desplazamiento para que la señal sea direccionada entre un dispositivo esclavo y el maestro de manera audible por altavoz

3.1.1.3.CONTROLADOR

En esta etapa se tiene una tarjeta Arduino nano, una etapa de preamplificador de audio y etapa de amplificador de potencia, además de un circuito selector en base a un registro de desplazamiento. Esta etapa recibe órdenes de la computadora central para habilitar la salida de audio de los dispositivos esclavos. La señal de salida del amplificador va a un parlante o a auriculares.

3.1.1.4.COMPUTADORA CENTRAL

Para el funcionamiento del presente proyecto se requirió de una computadora en la que se instaló un software que enlaza la base de datos con cada dispositivo esclavo y a su vez manda instrucciones al controlador y lee las instrucciones de pulsador enviadas desde la etapa de inicialización de llamada. El software está hecho en base a Python.

3.1.1.5.BASE DE DATOS

La computadora central cuenta con una base de datos en la que se anota los datos de cada paciente al que se asignó cada dispositivo. En esta base de datos existe un campo donde se ponga si el paciente requiere o no de la función de activación de emergencia. Todos los dispositivos de comunicación son iguales, pero mediante base de datos se podrá activar la función de comunicación de emergencia o quitar la misma.

3.1.1.6.SALIDA DE AUDIO

En cada dispositivo esclavo se cuenta con una salida de audio con altavoz, para escuchar

la respuesta de la enfermera que acudirá al llamado de cada paciente. Esta etapa consiste en un preamplificador para un micrófono electred y un amplificador de potencia para 10 watts.

3.1.2. ALGORITMO

Una vez estén todos los bloques conectados, este es el algoritmo que explica su funcionamiento:

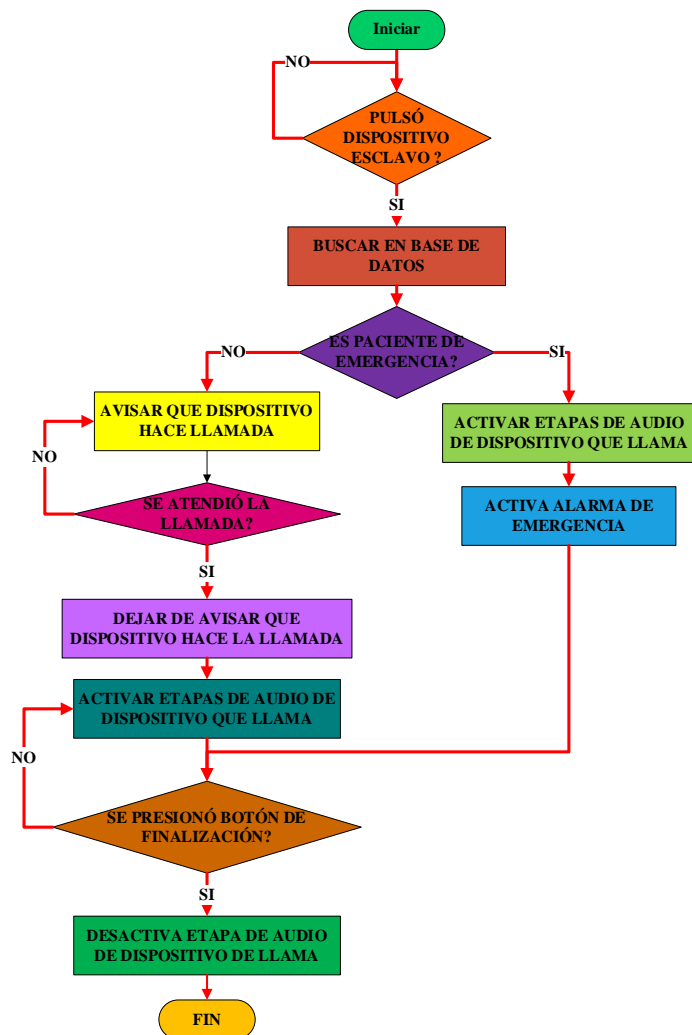


Ilustración 17 Diagrama de flujo del proyecto

Fuente propia

Se puede apreciar que dependiendo la configuración que se dará a cada dispositivo esclavo, el algoritmo que se ejecuta es diferente, siendo prioridad el dispositivo de emergencia, pues activa la llamada automáticamente y una alarma que en el software se expresa como un mensaje emergente. Y en caso de tratarse de un paciente normal, el personal médico decide en qué momento responde a la llamada activando sus etapas de audio. En caso de existir una segunda llamada desde otro dispositivo mientras se encuentra en una llamada, solo quedará como indicador de que hay otro dispositivo esperando ser respondido.

3.2.DISEÑO DEL PROYECTO

A continuación, se describe en detalle cómo se diseñó el presente proyecto:

3.2.1. DISPOSITIVO CONTROLADOR

Este dispositivo compone un preamplificador, un amplificador de potencia, una etapa de activación y desactivación de dispositivos y su interconexión con Arduino.

Para el preamplificador se tiene el siguiente circuito:

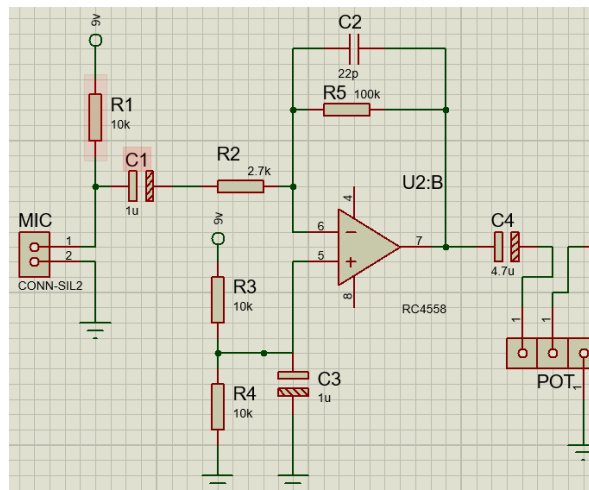


Ilustración 18 Diagrama circuital de etapa de preamplificador

Fuente propia

Se tiene un amplificador inversor en base a un amplificador operacional.

La resistencia R1 brinda un voltaje de funcionamiento al micrófono, pues internamente tiene un transistor FET con drenador abierto. El valor de esta resistencia es de $10\text{K}\Omega$, suficiente para excitar el transistor del micrófono.

El capacitor C1 es un capacitor de acople, este anulará la corriente continua del micrófono, y solo dejará pasar la corriente alterna que resulta ser la voz audible.

A su vez, C1 y R2 actúan como filtro pasa altos. Su fórmula en cuanto a frecuencia de corte es:

$$F_c = \frac{1}{2\pi C_1 R_2} \quad (1)$$

Para este caso es:

$$F_c = \frac{1}{2\pi * 1\mu\text{F} * 2,7\text{K}\Omega} = 58,97 \text{ Hz}$$

Se debe tomar en cuenta que la voz masculina se encuentra entre 50 y 200 hercios y la femenina entre 150 y 350 hercios. Esta frecuencia de corte hará que, a frecuencias más bajas, la señal se atenúe un poco.

La configuración de este amplificador es un amplificador inversor. Pero la señal de entrada debe tener voltajes positivos y negativos, por lo que se crea una “tierra analógica”. Para ello se usan las resistencias R3 y R4, como divisor de tensión. El voltaje equivalente se calcula de la siguiente manera:

$$V_a = V_{CC} * \frac{R_4}{R_3 + R_4} \quad (2)$$

Para este caso es:

$$V_a = 9\text{v} * \frac{10\text{K}\Omega}{10\text{k}\Omega + 10\text{k}\Omega} = 4,5 \text{ v}$$

El capacitor C3 es un capacitor de desacople, que se encarga de mandar a tierra todas las corrientes analógicas, esto se utiliza para estabilizar el voltaje resultante. Este a su vez funciona como filtro pasa bajos con la resistencia R4. La fórmula de su frecuencia de corte es:

$$F_c = \frac{1}{2\pi C_3 R_4} \quad (3)$$

Para este caso es:

$$F_c = \frac{1}{2\pi * 1\mu\text{F} * 10\text{K}\Omega} = 15,9 \text{ Hz}$$

Este filtro quita la mayor parte del ruido que produzca la fuente.

Para la ganancia del amplificador se utiliza la siguiente fórmula:

$$\Delta G = -\frac{R_5}{R_2} \quad (4)$$

Para este caso:

$$\Delta G = -\frac{100\text{K}}{2,7\text{K}} = 37$$

Este preamplificador es de ganancia 37, lo cual es suficiente para ser enviado a una etapa de potencia.

El capacitor C2 en paralelo a R5 forman un filtro pasa bajos. Su frecuencia de corte viene dada por la siguiente fórmula

$$F_c = \frac{1}{2\pi C_2 R_5} \quad (5)$$

Para este caso:

$$F_c = \frac{1}{2\pi * 10pF * 100K\Omega} = 159,2 \text{ Hz}$$

Se puede ver que las frecuencias de corte superior e inferior están en los rangos de frecuencia de la voz audible, por tanto, este preamplificador está hecho para un micrófono exclusivamente.

Antes de pasar a la etapa de amplificador, se coloca un potenciómetro para control de volumen del dispositivo.

Para el amplificador de potencia se utilizó uno en base al circuito integrado TDA2003

Este circuito es un amplificador de potencia integrado que es exclusivo para amplificadores de potencia de hasta 10w, suficientes para ser audibles.

En la hoja de datos del TDA2003 se tiene un circuito de prueba, el cual fue el implementado para este proyecto. Su diagrama es el siguiente:

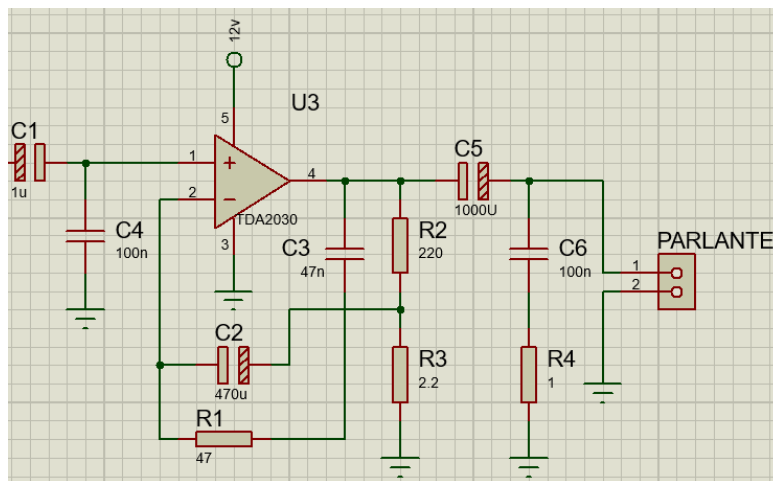


Ilustración 19 Diagrama sugerencia de hoja de datos de TDA2003

Fuente propia

Donde:

R1 con C3 determinan la frecuencia de corte superior, en un filtro pasa bajos.

R2 determina la ganancia de retroalimentación

R3 configura la ganancia y SVR o eliminación del golpe que emite el altavoz.

R4 y C6 tienen el propósito de generar una Red de Zobel o bloqueo de oscilación

C1 es capacitor de acople entre la etapa de preamplificador y la etapa de potencia.

C2 realiza un rechazo de rizado

C4 es de desacople entre la entrada y tierra para evitar ruido

Estas dos etapas deben reflejarse también en los dispositivos esclavo, ya que se necesita hablar y escuchar al mismo tiempo y no es muy conveniente que la comunicación se lleve a cabo en un solo sentido, así que tanto el punto A y el punto B de una comunicación tendrán un micrófono y un altavoz para la comunicación. Tampoco es conveniente que los dispositivos esclavos compartan un mismo amplificador, pues la extensión del cable traería consigo mucho ruido adherido a la señal de micrófono.

El circuito de activación de dispositivos tiene dos etapas: una de registros PARALELO – SERIE; y otra de registros SERIE – PARALELO.

El registro serie – paralelo sirve para conectar los pulsadores de cada dispositivo esclavo, mediante estos se solicita una llamada entre el paciente y el personal de enfermería.

El circuito integrado es el 74HC165, es un registro Paralelo – Serie que tiene las siguientes características:

- 8 pines de entrada en paralelo.
- 1 pin de salida serie.
- 1 pin de clock.
- 1 pin de carga de datos y desplazamiento.

- Se pueden interconectar hasta 8 chips en serie.

Su diagrama con este circuito es el siguiente:

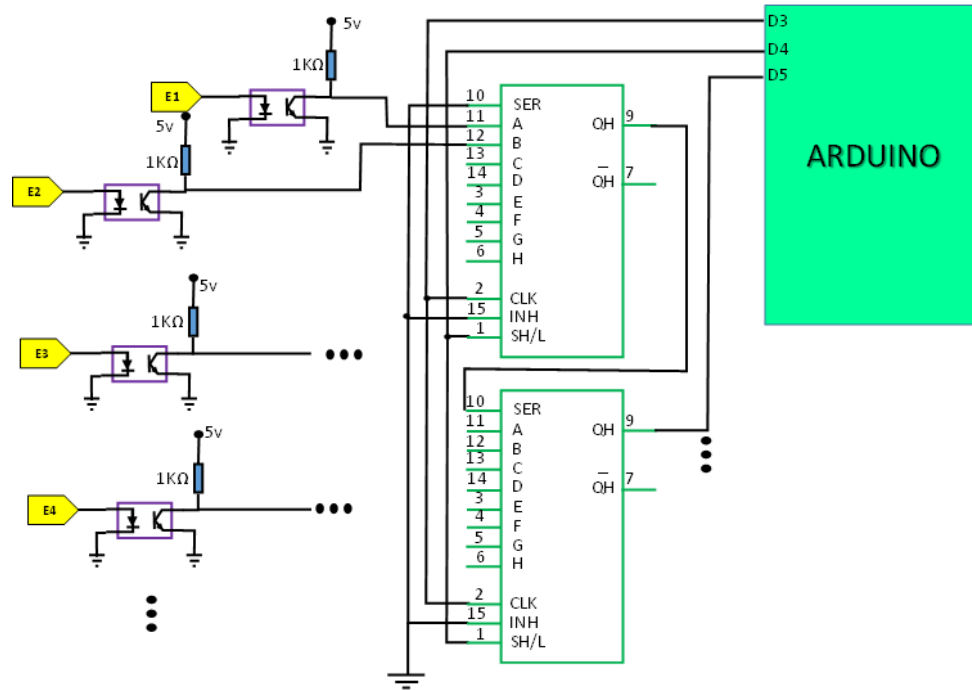


Ilustración 20 Diagrama circuital de registros PARALELO - SERIE

Fuente propia

Todas las resistencias usadas para esta etapa son de valor $1K\Omega$, se usa optoacopladores PC817 puesto que los pulsadores de los dispositivos esclavo activarán un pulso de 12 voltios, que van por encima a los que puede soportar un microcontrolador, así que un optoacoplador logra separar etapas de diferentes voltajes.

Gracias a los registros Paralelo – Serie, solo es necesario utilizar 3 pines del Arduino para controlar todas las entradas. Además, estas entradas se podrán conectar de forma modular; cuando se requiera más entradas, es decir más dispositivos esclavos, se aumentará otro módulo de registro.

La etapa de registro Serie – Paralelo se usó para habilitar la llamada con diferentes

dispositivos esclavos, uno a la vez. Para esto se utilizó el chip 74HC595, y a su vez se utilizará amplificadores de corriente, el circuito integrado ULN2003, pues es necesario activar los dispositivos esclavos con un voltaje de 12 voltios para el micrófono, y la habilitación del altavoz a través de un relé.

A continuación, se muestra el diagrama de esta etapa

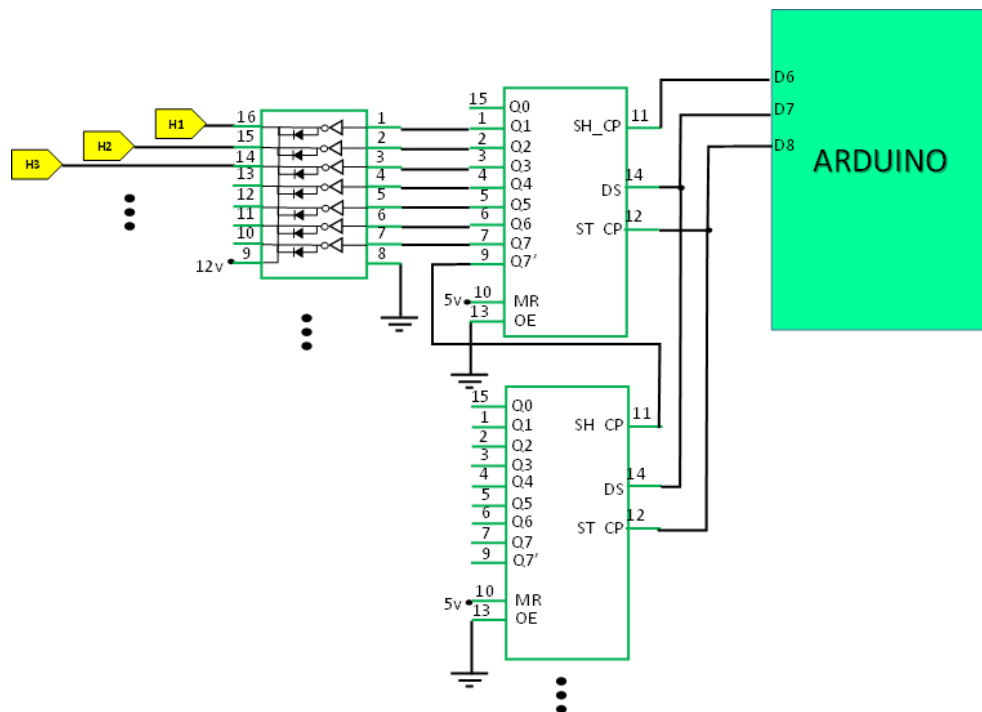


Ilustración 21 Diagrama circuital de registros SERIE - PARALELO

Fuente propia

Del diagrama se puede observar lo siguiente:

Un ULN2003 solo tiene siete amplificadores de corriente, por lo que por cada módulo solo se utilizó siete salidas a controlar, para que se tenga la misma cantidad en entradas, también se utilizó solo 7 entradas por módulo, pero eso se soluciona en el programa de Arduino.

Las salidas del amplificador de corriente ULN2003 van directamente a los dispositivos esclavos, y se encargan de habilitar la comunicación con cada uno de ellos. El pin en común de este circuito integrado (pin 9) se alimenta con 12 voltios, suficientes para alcanzar distancias largas.

Para este caso, gracias a los registros Serie – Paralelo, también fue necesario únicamente 3 pines del Arduino, lo cual es muy ventajoso para no tener que cablear demasiados pines o tener un límite de salidas.

Mostrando el diagrama completo del dispositivo maestro, para solo un módulo de registros de desplazamiento y para un solo dispositivo esclavo, quedó de la siguiente forma:

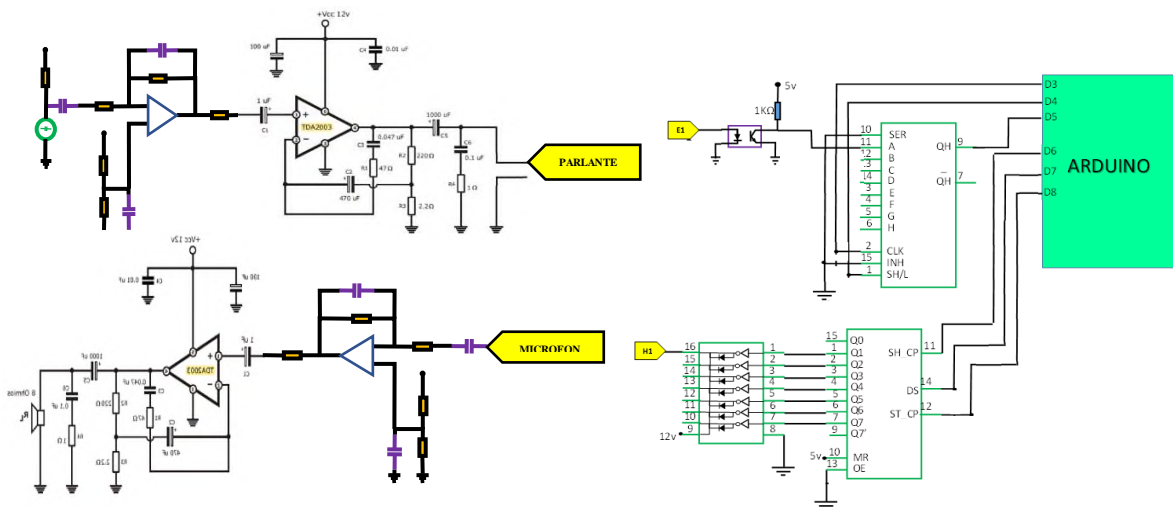


Ilustración 22 Diagrama general del dispositivo maestro

Fuente propia

Se puede apreciar que el dispositivo maestro tendrá 4 salidas, sin tomar en cuenta a (12v) y a (GND), entonces cada dispositivo esclavo tendrá que conectarse al maestro a través de un cable de 6 vías. El cable más recomendado para esto es UTP.

Todos los dispositivos esclavos comparten los pines de PARLANTE y MICROFONO, pero se diferencian en los pines E1 y H1.

Se necesitará dos fuentes: de 5 y 12 voltios, iniciando en 2 amperios para la fuente de 12 voltios, y a medida que se aumente los dispositivos esclavos se debe aumentar la potencia de la fuente de 12 voltios. La de 5 voltios también debe aumentar su amperaje a medida que aumenta la cantidad de dispositivos esclavos, pues debe mantener a varios registros de desplazamiento. De principio puede ser alimentado por el cable USB conectado a una computadora, pero en un futuro debe tener una fuente independiente. La fuente más recomendada para este proyecto es una de computadora o similar.

3.2.2. DISPOSITIVOS ESCLAVOS

Cada dispositivo esclavo se conecta al dispositivo maestro a través de seis cables, dos de ellos son su alimentación y los otros de control y de audio.

El diagrama del circuito es el siguiente:

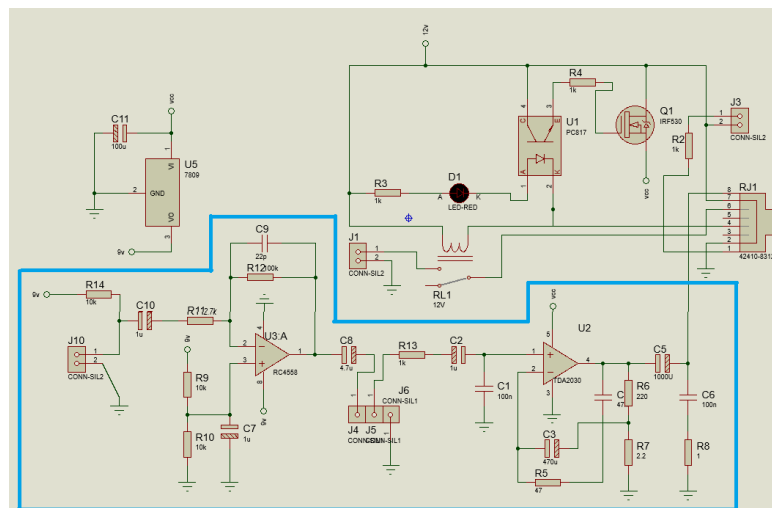


Ilustración 23 Diagrama circuital del dispositivo esclavo

Fuente propia

Se puede ver que los dispositivos esclavos llevan la misma etapa de preamplificador y amplificador que el dispositivo maestro, estas etapas ya fueron anteriormente explicadas.

Los pines de 12v y GND están presentes y son compartidos desde el dispositivo maestro. El pin de 12v se utiliza para habilitar el pulsador que llegará al optoacoplador del dispositivo maestro, mientras no se presione el pulsador, no hay consumo de energía. Esto lo hace mediante la resistencia R2 de la ilustración. Aparte se usa este pin para alimentar la etapa de amplificador.

Se puso un conector RJ45 para comunicar este dispositivo esclavo con el maestro. El pin 4 de este conector representa la entrada H1. Mientras la entrada H1 no se habilite, no funciona nada de este circuito. Este pin H1 es cada uno de los pines de salida del circuito integrado ULN2003 del dispositivo maestro.

Este pin pasa por la bobina del relé de 12 voltios, y en paralelo a una resistencia R3, un led y el optoacoplador PC817, cerrando circuito con el pin 12 voltios. Cuando se activa el pin H1, se activa el relé, se enciende el led, y se activa el optoacoplador.

Del otro lado del optoacoplador se tiene una resistencia de 1k ohmios, para alimentar el pin Gate de un transistor MOSFET IRF520. Este transistor tiene la misma configuración que un transistor BJT en corte y saturación, pero con la ventaja de mayor potencia y sin pérdida de voltaje o corriente. En este caso el IRF520 activa a un regulador de voltaje 7809 el cual alimenta toda la etapa de preamplificador y amplificador. Del mismo modo, el pin Común del relé está conectado a la salida del amplificador que viene del dispositivo maestro, el cual representa la salida de audio en parlante, pero la salida del parlante es en este caso el pin NO (Normalmente abierto).

Con este dispositivo, al momento de activarse H1, se activa el amplificador del esclavo, y se habilita el parlante del maestro. Y a la vez, se tiene el pulsador para solicitar la habilitación.

Para que no haya interferencia entre un dispositivo y otro es que el maestro solo activa un esclavo a la vez, y no así a varios.

3.2.3. SOFTWARE EN COMPUTADORA Y ARDUINO

En el presente proyecto existe una comunicación entre la PC y el Arduino a través de su puerto USB, que en realidad es una comunicación serial RS232 con un conversor a TTL, así que el software en la computadora se comunicará a través de un puerto serie.

Para ello se utilizó el software Python, el cual es muy versátil para comunicarse con una base de datos y con el puerto serie.

Primeramente, se instaló el software, descargando el archivo e instalando como se muestra en la figura:

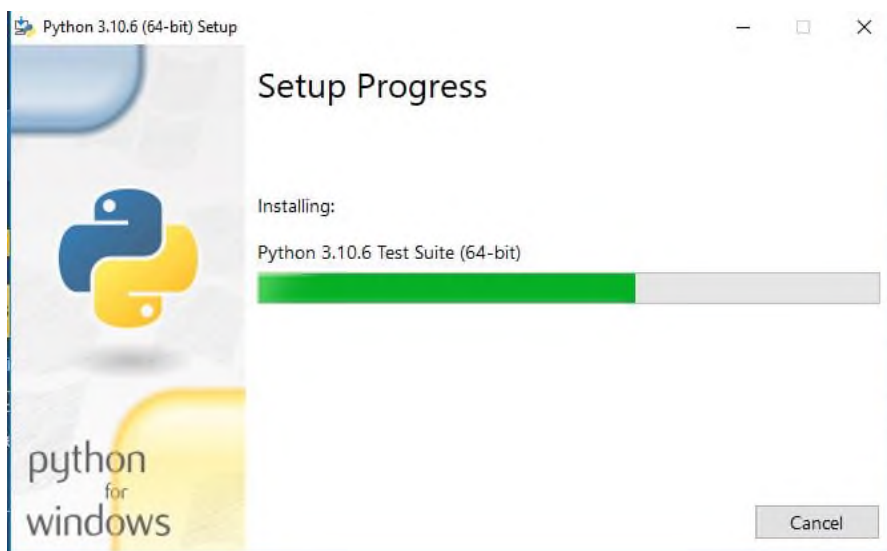


Ilustración 24 Instalación de Python

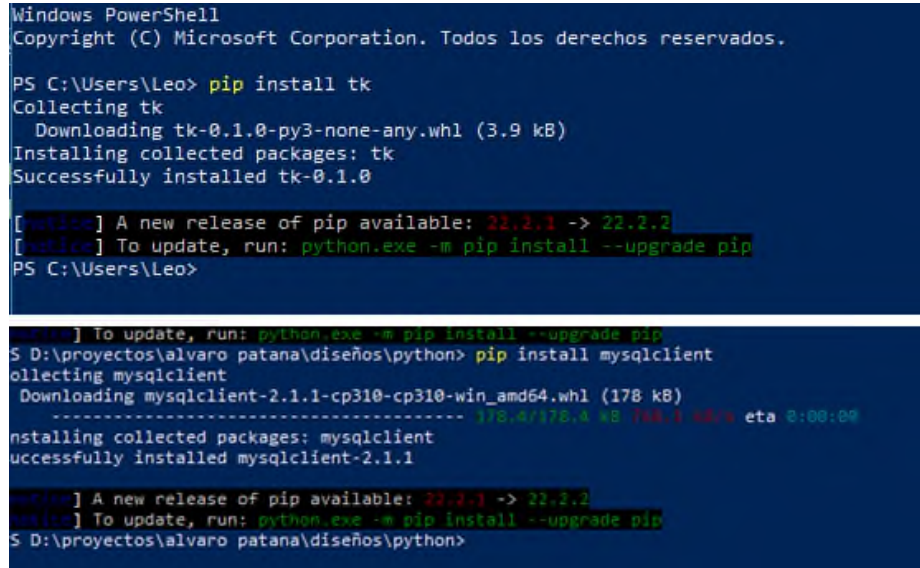
Fuente propia

Se utilizaron diversas librerías para la comunicación:

La librería “MySQL.db” es la que permite la comunicación con una base de datos MySQL instalada en la misma computadora. La librería “serial” permite la comunicación a un puerto serial a través de un COM, el número que se le vaya a asignar a un Arduino en una PC, por defecto se puso COM4, y ese puerto debe ser asignado en el Arduino. La librería

“tkinter” es la encargada de generar una ventana con botones de todas las camas disponibles en el centro de salud.

A continuación, se muestra los comandos para instalar las librerías:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Leo> pip install tk
Collecting tk
  Downloading tk-0.1.0-py3-none-any.whl (3.9 kB)
Installing collected packages: tk
Successfully installed tk-0.1.0

[notice] A new release of pip available: 22.2.1 -> 22.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\Leo>

PS D:\proyectos\alvaro patana\diseños\python> pip install mysqlclient
Collecting mysqlclient
  Downloading mysqlclient-2.1.1-cp310-cp310-win_amd64.whl (178 kB)
----- 178.4/178.4 kB 740.1 kB/s eta 0:00:00
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.1.1

[notice] A new release of pip available: 22.2.1 -> 22.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\proyectos\alvaro patana\diseños\python>
```

Ilustración 25 Instalación de librerías Python

Fuente propia

Primero se establece una comunicación con la base de datos, para luego iniciar la comunicación con el Arduino a través del COM4 o el COM que reconozca la computadora.

En pantalla se generará automáticamente la cantidad de botones equivalentes al número de dispositivos generados en la base de datos, y cada botón tendrá el nombre del paciente registrado en cada espacio.

Una vez Python lee la cantidad de dispositivos registrados, este mandará un mensaje por el puerto serial al Arduino con la cantidad de módulos de expansión para dispositivos (se incluirá los códigos de Python y Arduino en la sección “Anexos” del presente informe).

Para que ambos puedan diferenciar entre una instrucción y otra, se utilizaron letras

mayúsculas como encabezado y así saber de qué tipo de instrucción se trata:

- E: instrucciones para solicitud de llamada por algún esclavo
- N: instrucciones para ingreso de nuevo dispositivo
- H: instrucciones para habilitar una llamada
- F: instrucciones para finalizar una llamada
- D: instrucciones para señalar la cantidad de módulos registro SERIE-PARALELO

utilizando estas letras como encabezado, la comunicación entre la computadora y Arduino es más ordenada y sincronizada como se muestra a continuación:

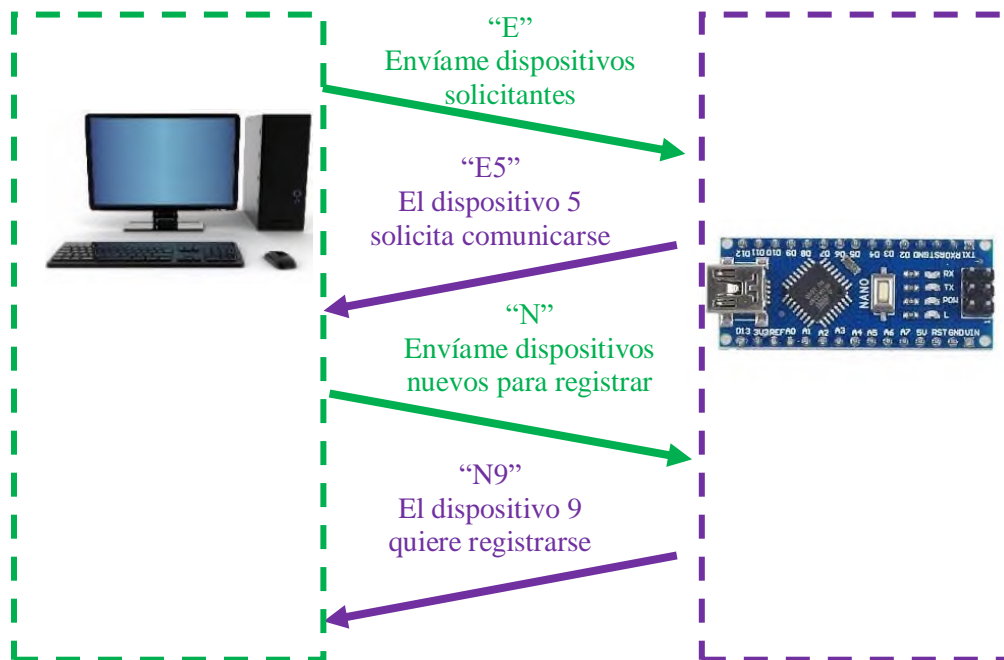


Ilustración 26 Forma de comunicación entre Arduino y PC cuando los dispositivos esclavos son activados

Fuente Propia

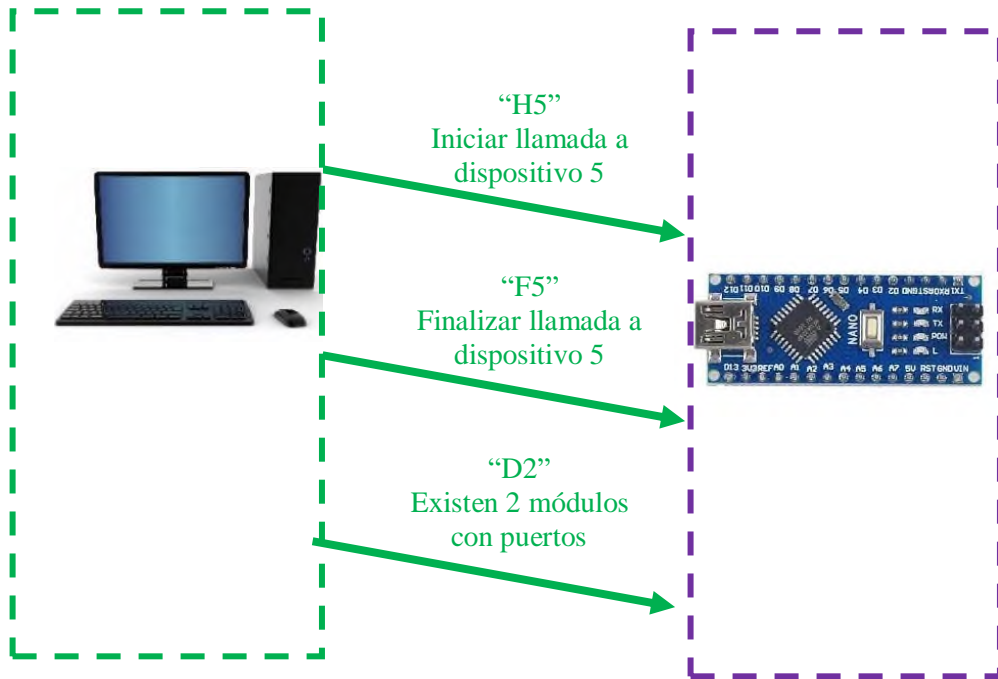


Ilustración 27 Forma de comunicación entre Arduino y PC para realizar una llamada

Fuente propia

El Arduino al momento de iniciar estará en modo "E", o modo envío de dispositivos de solicitudes de llamada hasta que Python le indique vía serial que cambie a modo "N" o modo de nuevo dispositivo a registrar.

Cuando un dispositivo esclavo activa su pulsador, el Arduino detecta el número del dispositivo, y envía esa información a Python, por ejemplo, un "E8". Con esta información, Python buscará en la base de datos, a que paciente corresponde el dispositivo 8, y avisará en el entorno gráfico que dicho paciente quiere comunicarse, y si es un paciente de emergencia, la llamada iniciará automáticamente.

Si los dispositivos esclavos no son de un paciente de emergencia, solo Python puede decidir iniciar o no una llamada con los dispositivos solicitantes.

3.2.4. BASE DE DATOS

Para el presente proyecto, fue necesario crear una base de datos en la computadora, y esta misma base de datos se creó en la computadora.

Para ello primero se instaló phpmyadmin en la computadora.

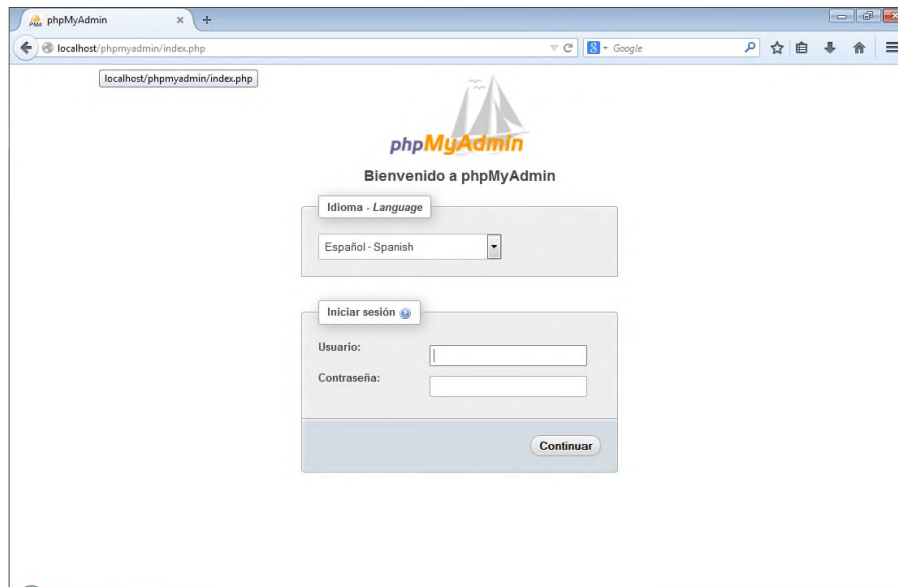


Ilustración 28 Inicio de phpMyAdmin luego de instalación

Fuente propia

Al instalarlo en la computadora, se incluye MySQL en la instalación, así que luego se procedió a crear una base de datos con el nombre “base” y dos tablas tabla llamadas “dispositivos” y “COM”.

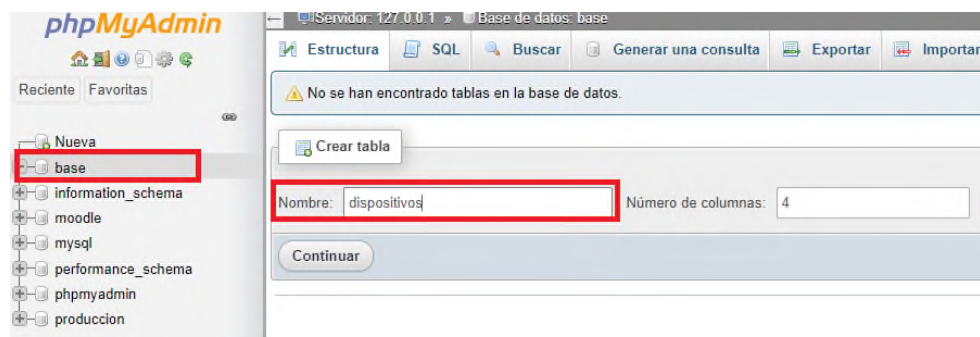


Ilustración 29 Creación de base de datos y tabla para control

Fuente propia

A continuación, se crea las columnas respectivas para tener todos los datos del paciente, como se muestra a continuación:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 ID	int(4)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	2 PACIENTE	varchar(30)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 DIAGNOSTICO	varchar(60)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4 EMERGENCIA	int(2)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	5 CAMA	varchar(10)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más

Ilustración 30 Creación de columnas de tabla “dispositivos”

Fuente propia

En la columna ID estará el número que el Arduino le dé según el dispositivo esclavo instalado en una cama de paciente. En la columna PACIENTE va el nombre del paciente. En la columna DIAGNOSTICO se incluye información adicional del cuadro clínico, las más relevantes como si tiene algún problema cardiaco, alguna dolencia que debe ser atendida regularmente, etc. La columna EMERGENCIA será llenada con los números “0” y “1”; donde 0 simbolizará que no es un paciente de emergencia, y 1 es paciente de emergencia. Y la columna CAMA indicará la ubicación del dispositivo esclavo que hace la llamada para poder acudir a donde se encuentra de forma certera.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 ID_COM	int(4)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/>	2 COM	int(4)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	3 NOMBRE_HOSP	varchar(30)	latin1_swedish_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/>	4 N_MODULOS	int(3)			No	Ninguna			Cambiar Eliminar Más

Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice

Ilustración 31 Creación de columnas de tabla “com”

Fuente propia

Y en la tabla “com” solo se contará con una fila de datos, donde ID_COM está por defecto como “1”, “COM” señala el último puerto COM conectado con éxito, para agilizar la comunicación en próximas ocasiones. La columna NOMBRE_HOSPITAL almacena el nombre del hospital al cual se registra toda la base de datos. Y la columna N_MODULOS señala la cantidad de módulos que se configuró estaban conectados para esa base de datos. Recordar que cada módulo contiene a 7 dispositivos esclavo, con lo que si la base de datos señala 4 en N_MODULOS, el Arduino solo realizará la lectura a 28 dispositivos esclavo

3.3.FUNCIONAMIENTO DEL PROYECTO

Una vez diseñado el proyecto se procedió a construir el mismo. para ello se construyó los PCBs tanto del dispositivo maestro, como de los esclavos.

Para ello se utilizó el software “Proteus”, obteniendo los siguientes resultados:

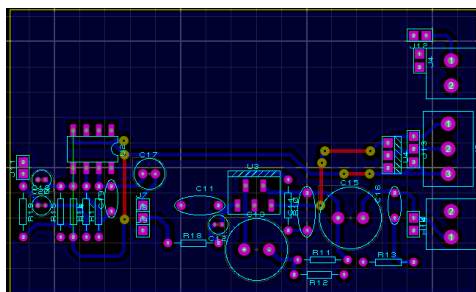


Ilustración 32 PCB de preamplificador y amplificador

Fuente propia

Para el PCB de los registros con el Arduino “Nano” se vio necesario hacer dos placas para montar una sobre otra debido a que ocuparía mucho espacio, y este diseño está pensado para ser modular; es decir que en caso de necesitar mas dispositivos esclavos, se debe agregar otra placa de registros. Pensando en esto se tiene el siguiente diseño de PCB:

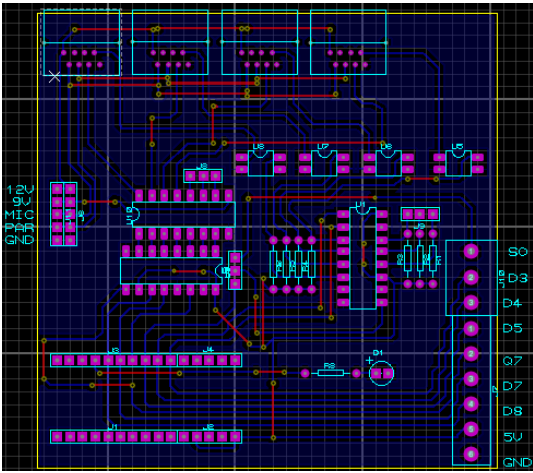


Ilustración 33 PCB de los registros de desplazamientos conectados a Arduino

Fuente propia

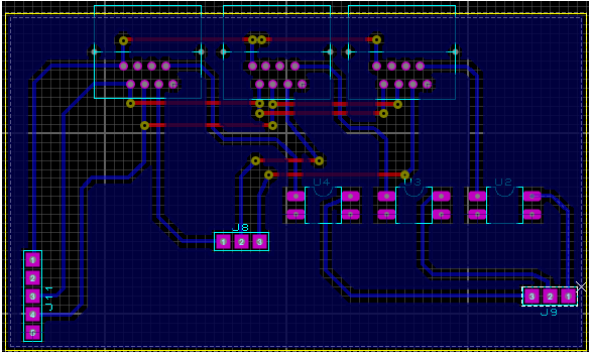


Ilustración 34 PCB del resto de conectores a los registros de desplazamiento

Fuente propia

Finalmente, por cada cama, se debe implementar un dispositivo esclavo, este se muestra a continuación:

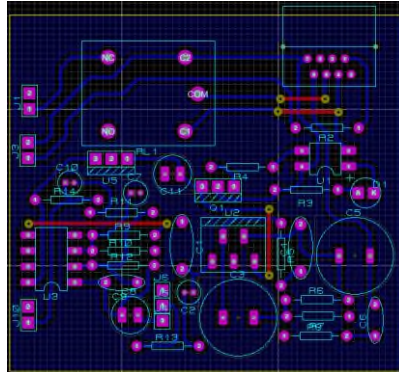


Ilustración 35 PCB del dispositivo esclavo

Fuente propia

En todos estos diseños se puede notar que se les implementó conectores RJ45, esto hará que el cableado a cada dispositivo sea mucho más sencillo, y a su vez se puede aprovechar el revestimiento con el que cuenta el cable UTP categoría 6 en las transmisiones de datos.

Una vez diseñados los PCBs, se procedió a implementar cada uno de ellos. A continuación, se muestra los resultados:

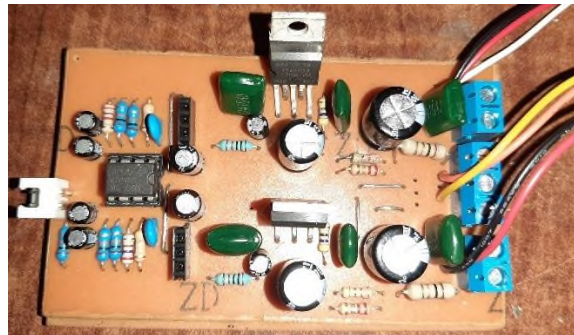


Ilustración 36 Terminado el PCB del amplificador

Fuente propia

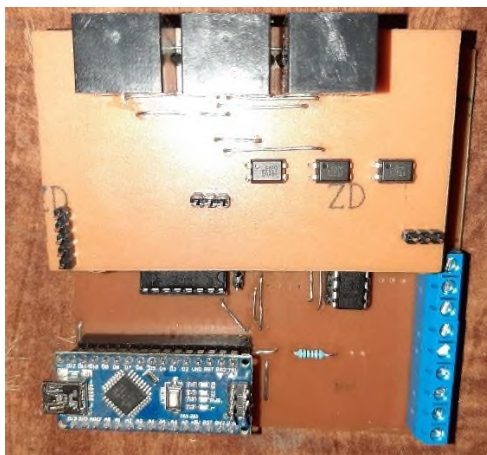


Ilustración 37 Terminado el PCB de los registros y Arduino

Fuente propia

Estas placas están conectadas a la computadora donde se monitorea a los pacientes. La placa de amplificación de audio cuenta con un conector JST de dos pines al inicio; este se conecta al micrófono que debe utilizar el personal de salud que está a cargo de los pacientes. A la mitad se cuenta con un conector de 3 pines, este se conecta a un potenciómetro que cumple la función de volumen para el micrófono del dispositivo maestro. En la parte final hay un conector JST de 2 pines, este se conecta a un parlante de 8 ohmios para el personal de salud. También se puede apreciar tres borneras, una de 3 pines y dos de 2 pines. Estos se conectan al micrófono de los pacientes, al parlante de los pacientes y el de 3 pines es para la alimentación del circuito, con 12 voltios y 9 voltios DC.

En el PCB de los registros SERIE – PARALELO y PARALELO – SERIE, se puede ver que se tiene una placa montada sobre otra, esto para tener una mayor eficiencia al momento de situar los conectores RJ45. Se obtuvo un buen resultado como se muestra en la siguiente figura:

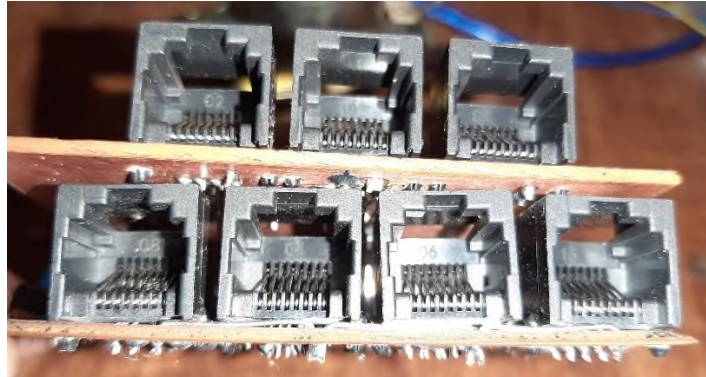


Ilustración 38 Vista lateral de los conectores a los dispositivos esclavos

Fuente propia

Como se puede apreciar, en una placa de 9 centímetros se pudo situar 7 conectores RJ45 lo que hará que, en caso de tener una cantidad mayor de dispositivos, podrán ser situados de forma más eficiente.

De la ilustración 37 se puede apreciar además que hay tres borneras de 3 pines, haciendo un total de 9 pines. Estos pines son los pines necesarios para poder expandir los registros, pues por cada uno es necesario conectar 3 pines, y compartir 5 voltios y GND a cada uno.

Las borneras que tienen las placas son para que a estas se le conecten cables hacia el siguiente módulo de registros SERIE – PARALELO y PARALELO – SERIE.

Una vez implementados todos los PCB se realizó cajas para producto terminado y se los interconectó para que queden listos a comunicarse. La forma de conectarlos es como se muestra en la siguiente figura:

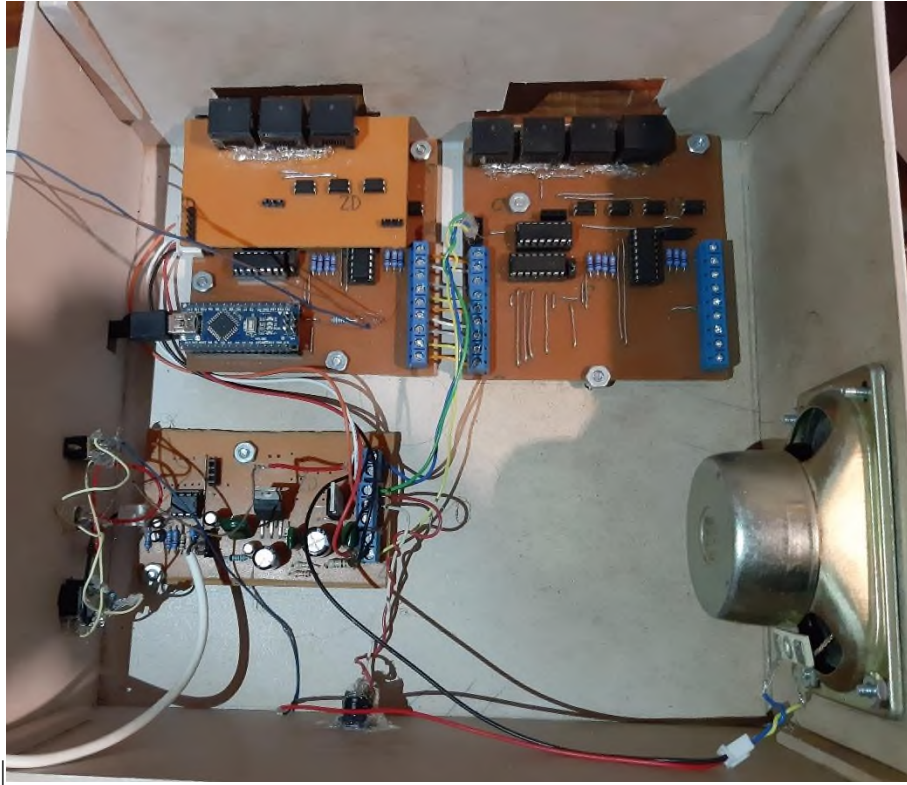


Ilustración 39 Forma de conexión del dispositivo maestro

Fuente propia

El PCB de registros tiene un conector JST de 5 pines, a los cuales se les conectan las señales de micrófono, parlante y voltajes de alimentación de los amplificadores. A su vez se conecta un propio parlante a la salida de un amplificador y un micrófono electret para el personal de salud. Sin embargo, para una mayor comodidad se adicionó un conector de audio jack de 3.5 mm para cualquier micrófono – audífono que se tenga. La forma de comunicación del Arduino “Nano” a la computadora se realiza mediante un cable USB, pero el protocolo que utiliza es RS232 mediante puerto Serial.

Para distribuir desde el PCB maestro a los PCBs esclavos, se conecta un cable UTP categoría 6, a través de los conectores RJ45, estos deben ser cableados de forma directa, y no cruzada. Quedando como se muestra en la figura:

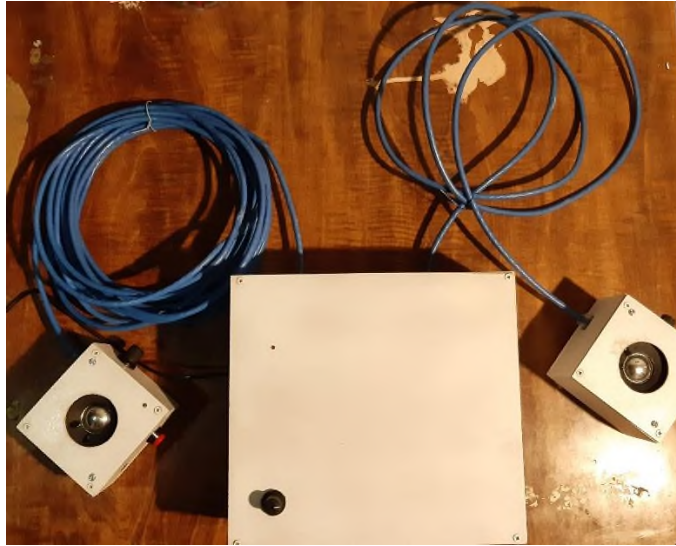


Ilustración 40 Forma de conexión entre el dispositivo maestro y un dispositivo esclavo

Fuente propia

Esta forma de conectividad permite una mayor facilidad de manejar el cableado para cada cama.

Y el acabado de los dispositivos esclavo es como se muestra en la siguiente figura:

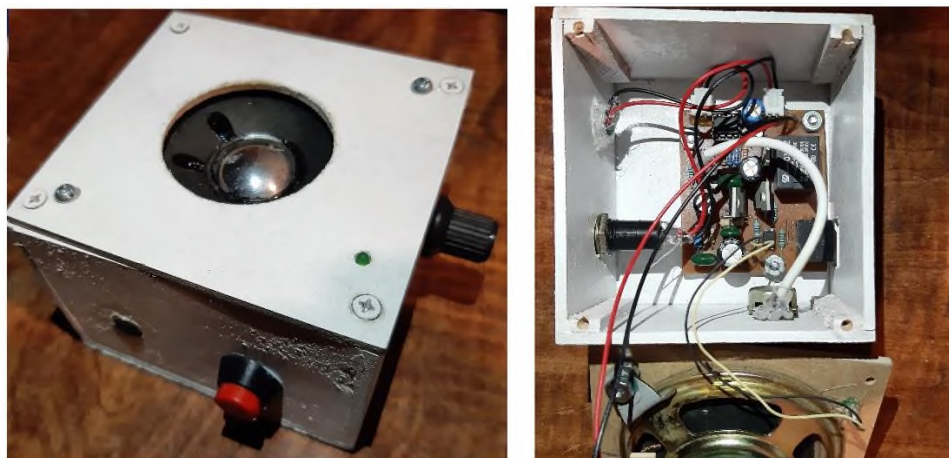


Ilustración 41 Acabado de los dispositivos esclavos

Fuente propia

Una vez conectado por lo menos un dispositivo esclavo, se procedió a configurarlo en la base de datos. Para ello se conectó el Arduino a la computadora del personal de salud, y se abrió el programa elaborado en base a Python y Tkinter.



Ilustración 42 Entorno gráfico creado con Python para la base de datos

Fuente propia

Al abrir el programa, no se puede ver ningún paciente registrado, pues la base de datos está vacía, primero se debe conectar al puerto del Arduino buscando su número de puerto en el control “Combox” que aparece en la parte izquierda. Una vez conectado con el puerto, se agrega un nuevo paciente, haciendo clic en el botón de la derecha “NUEVO”.

Ilustración 43 Registrando nuevo paciente en base de datos

Fuente propia

Aparece una ventana donde se llenan los datos del paciente al cual se le habilitará el dispositivo, pero no se puede llenar el espacio donde aparece el dispositivo. Para llenar este espacio, el programa pide que desde el dispositivo esclavo se presione el pulsador. Apenas se presiona el pulsador de algún dispositivo esclavo, el Arduino reconoce que número de dispositivo presionó el pulsador, y envía esta información a Python, y se aprecia que se llena automáticamente el espacio de “DISPOSITIVO”.

The screenshot shows a web application window titled "HOSPITAL DE LA MUJER" with a sub-header "NUEVO PACIENTE". The form contains the following fields and buttons:

- N° DE MÓDULOS: 2 (dropdown)
- DISPOSITIVO: 7 (text input, highlighted with a red box)
- AGREGAR (button)
- NOMBRE DEL PACIENTE: Jorge Pérez (text input)
- MODIFICAR (button)
- DIAGNOSTICO: Preembolia (text input)
- BORRAR (button)
- EMERGENCIA: NO (dropdown)
- CANCELAR (button)
- LOCALIZACIÓN CAMA: Sala 2, cama 1 (text input)

At the bottom of the form, there is a instruction: "PRESIONE BOTÓN DEL NUEVO DISPOSITIVO".

Ilustración 44 Respuesta del dispositivo esclavo para el registro

Fuente propia

Una vez introducidos todos los datos, ya se puede presionar el botón “AGREGAR” y el paciente queda registrado en la base de datos.



Ilustración 45 Habilitación de nuevo paciente en la base de datos

Fuente propia

Adicionalmente se puede agregar a más pacientes como dispositivos se tengan disponibles



Ilustración 46 Habilitación de más pacientes en la base de datos

Fuente propia

Una vez se registran los pacientes en la base de datos, en la pantalla del programa se muestran botones con los nombres de cada paciente, así como dos botones con las letras “D” y “M”. el botón “D” muestra una ventana con el diagnóstico del paciente, donde se detalla si requiere algún medicamento o tratamiento en especial. El botón “M” sirve para modificar los datos del paciente o para eliminarlo de la base de datos en caso de que ya sea dado de alta. Y el botón con el nombre del paciente sirve para iniciar o finalizar la comunicación con cada uno de ellos. Cuando se inicia todo el sistema, los amplificadores están inhabilitados, y no consumen energía, ningún micrófono puede emitir alguna señal de audio.

Cuando sucede que algún paciente tiene alguna necesidad, dolencia o emergencia por lo que requiere al personal de salud, puede presionar el pulsador habilitado a la cabecera de su cama. Apenas sucede esto, el Arduino detecta que dispositivo presionó el pulsador y manda esa información a Python, en la pantalla cambia de color el botón correspondiente a ese paciente.



Ilustración 47 Llamada de una paciente registrada en el programa

Fuente propia

La comunicación no inicia hasta que el personal de salud haga clic en el botón que cambió de color. En ese instante Arduino manda una señal para habilitar el dispositivo esclavo que presionó el pulsador, habilitando la alimentación del amplificador, así como el micrófono y parlante del dispositivo esclavo del paciente que solicitó la llamada. Una vez termina la llamada, el personal de salud solo debe volver a presionar el botón del paciente que habilitó la llamada, el botón vuelve a su estado inicial y los dispositivos quedan sin energía nuevamente.

CAPITULO IV
COSTOS DEL PROYECTO

4.1.EVALUACIÓN DE RESULTADOS

Culminadas las pruebas con todos los dispositivos se pudo determinar que su funcionamiento es óptimo, ya que de forma inmediata se puede realizar una llamada con el dispositivo esclavo según el paciente así lo requiera.

La calidad de audio es muy buena, y la respuesta es inmediata. Realizando una comparación con el anterior sistema de control de pacientes a como se realiza con este proyecto se puede ver que este proyecto hace más efectivo el trabajo del personal de salud, ya que en lugar de tener que realizar rondas para ver a los pacientes, y tener que dar dos vueltas en caso de que algún paciente en plena ronda haya necesitado algo, con este sistema, el personal de salud puede hablar directamente con los pacientes y puede en una sola ronda podrá acudir a los pacientes que lo necesiten con los medicamentos o cosas que los pacientes necesiten.

Se realizaron pruebas para determinar las limitaciones de este proyecto y se tiene las siguientes anotaciones (tomando en cuenta una fuente de alimentación de 12 voltios a 2 amperios, y utilizando únicamente la alimentación a través del puerto UBS de la computadora):

- Debido a que cada registro de desplazamiento consume 1 mA, y que a medida que se conectan más registros, el pin CLK del Arduino debe soportar simultáneamente a más chips, el máximo de registros SERIE – PARALELO o PARALELO – SERIE que se puede conectar simultáneamente es de 30. Cada registro abarca a 7 dispositivos, con lo que se puede controlar 210 dispositivos esclavos en total antes de exceder los 40mA del pin de Arduino.
- Mientras se tenga una mayor distancia en el cable, se tiene más probabilidades de error de envío de datos a través del cable UTP categoría 6. La distancia máxima recomendada es de 100 metros, pero se recomienda que sea menor a esta, como unos 90 metros.
- No se permite realizar llamadas simultáneas, ya que esto implicaría un mayor

consumo de energía debido a que se debe alimentar varios relés al mismo tiempo, y la fuente de 2 amperios no lo soportaría. Mientras solo sean llamadas de un paciente a la vez, la fuente de 2 amperios soporta 210 dispositivos esclavos

- Si varios pacientes pulsan solicitando una llamada de forma simultánea, el Arduino es capaz de enviar a la computadora todos los datos de las solicitudes, aun estando en plena llamada en curso, debido a que el programa Python o Arduino ejecutan la comunicación serial en segundo plano de forma simultánea a su programa principal.

Sin tomar en cuenta las limitaciones técnicas del presente proyecto, se puede ver que cubre con los requerimientos de un centro de salud, ya que el requerimiento no pasa de las 100 camas, haciendo que este proyecto sea más que suficiente.

De estas anotaciones, es posible realizar mejoras a futuro modificando los dispositivos, pero esto elevaría el costo de los dispositivos. En caso de necesitar este proyecto para lugares con una cantidad mayor de usuarios o a mayor distancia, estas son las propuestas a futuro:

- Se debe alimentar el proyecto con una fuente que disponga de dos salidas (12 y 5 voltios) con un mayor amperaje, por lo menos de 5 amperios. a su vez se podría modificar el dispositivo esclavo, cambiando el relé por un transistor MOSFET IRF540 o similar, que permite un funcionamiento similar.
- En caso de requerir realizar una llamada a varios dispositivos también es necesario aumentar la potencia del amplificador, para que soporte a varios parlantes en paralelo.
- Para aumentar el número de registros sin dañar los pines del Arduino, se debe agregar optoacopladores a cada pin del Arduino alimentados con la fuente de 5 voltios, pero la fuente de estos optoacopladores no debe ser la de la computadora mediante USB, sino de la fuente de 5 voltios que ya se dispondría.
- Hay una manera de convertir los dispositivos en dispositivos que se conecten a un

router o switch, inclusive que sean inalámbricos comunicados vía Wifi, pero esto significaría que cada dispositivo esclavo necesite su propia fuente de alimentación, que tenga su propio microcontrolador con conexión Ethernet o wifi, un DAC (convertor digital - analógico) y deben estar conectados a una misma red LAN.

Como se puede ver, este proyecto tiene una alta escalabilidad, pero cada modificación involucra una mayor inversión, lo que significaría volver a replantear si el proyecto es rentable o no.

4.1.INVERSIÓN DEL PROYECTO

Para establecer el costo total del proyecto se debe tomar en cuenta todos los gastos realizados, y que estos sean benéficos, es decir que el proyecto sea rentable.

A continuación, se muestran los costos que incluye la implementación del proyecto, tomando como ejemplo al hospital “Corazón de Jesús” que cuenta con 20 camas.

Esta tabla es de la inversión en materiales para el desarrollo del proyecto:

CANTIDAD	UNIDAD	DESCRIPCIÓN	PRECIO UNITARIO (Bs)	SUBTOTAL (Bs)
6	Unidades	Resistencias 1K Ω , ¼ watt	0,30	1,8
2	Unidades	Resistencia 2,7K Ω , ¼ watt	0,30	0,6
2	Unidades	Resistencia 100K Ω , ¼ watt	0,30	0,6
2	Unidades	Resistencia 270K Ω , ¼ watt	0,30	0,6
6	unidades	Capacitor 1uF	0,50	3

2	Unidades	Capacitor 10nF	0,50	1
43	Unidades	Micrófono electred	3	129
2	Unidades	Circuito integrado LM324	4	8
2	Unidades	Capacitor 100uF	1	2
2	Unidades	Capacitor 0,01uF	0,50	1
2	Unidades	Capacitor 1uF	1	2
2	Unidades	Capacitor 470uF	2	4
2	Unidades	Capacitor 1000uF	2,5	5
2	Unidades	Capacitor 0,047uF	0,50	1
2	Unidades	Capacitor 0,1uF	0,50	1
2	Unidades	Resistencia 47Ω, 1 watt	0,50	1
2	Unidades	Resistencia 220Ω, 1 watt	0,50	1
2	Unidades	Resistencia 2,2KΩ, 1 watt	0,50	1
2	Unidades	Resistencia 1KΩ, 10 watt	3	6
2	Unidades	Circuito integrado TDA2003	15	30
3	Unidades	Circuito integrado 74HC165	5	15

42	Unidades	Resistencia 1K Ω , ¼ watt	0,30	12,6
21	Unidades	Optoacoplador PC817	1,5	31,5
1	Unidad	Arduino NANO	50	50
3	Unidades	Circuito integrado ULN2003	3,5	10,5
3	Unidades	Circuito integrado 74HC595	5	15
21	Unidades	Resistencia 10K Ω , ¼ watt	0,30	6,3
21	Unidades	Relé de 12v	5	105
1	Unidad	Fuente conmutada 12 voltios DC, 2 amperios	50	50
21	Unidades	Carcasa de comunicadores	10	210
1	Unidad	Carcasa de controlador	10	10
5	Hojas	Placa de baquelita	10	50
42	unidades	Conector RJ45 hembra	5	210
42	unidades	Conector RJ45 macho	3	126
40	unidades	Bornera de 3 pines	3	120
2	unidades	Bornera 2 pines	2	4

43	unidades	Conector JST de 2 pines	2	86
1	unidades	Conector JST de 5 pines	3,5	3,5
10	metros	Cable para audio	2	20
100	Metros	Cable UTP CAT6	3	300
TOTAL INVERSIÓN EN MATERIALES				1635

Se tiene un total de 1635 Bs de inversión en materiales

También se realiza el cálculo de los costos en recursos de desarrollo

CANTIDAD	UNIDAD	DESCRIPCIÓN	PRECIO UNITARIO (Bs)	SUBTOTAL (Bs)
10	unidades	Tarjetas de teléfono	10	100
60	unidades	Pasajes de minibús	2	120
TOTAL RECURSOS DE DESARROLLO				240

Se tiene un total de 240 Bs por inversión en recursos de desarrollo.

En cuanto a recursos humanos, se puede señalar la siguiente tabla:

TIEMPO (MESES)	DESCRIPCIÓN	COSTO (Bs/MES)	SUBTOTAL (Bs)
2	Diseño, implementación y pruebas del proyecto.	2500	5000
TOTAL RECURSOS HUMANOS			5000

4.2.CÁLCULO DEL COSTO BENEFICIO

Para el cálculo del costo beneficio se utilizará la siguiente fórmula:

$$B = \text{total recursos humanos} + \text{costo de mantenimiento} \\ * \text{tiempo de vida} \quad (6)$$

El tiempo de vida del proyecto es de 5 años y se prevé realizar mantenimientos preventivos anuales del sistema. Tomando en cuenta la mano de obra del mantenimiento es a 500Bs por año.

$$B = 5000 + 500 \text{ Bs} \times 5(\text{años})$$

$$B = 7500 \text{ Bs}$$

4.3.CÁLCULO DE LA VIABILIDAD DEL PROYECTO

Para calcular el índice de viabilidad del proyecto se utilizará la siguiente fórmula:

$$\text{Si: } \frac{B}{\text{total inversion en materiales} + \text{total recursos de desarrollo}} > 1 \quad (7)$$

Reemplazando:

$$\frac{7500}{1635 + 240} > 1$$

$$4 > 1$$

Si $B/C > 1$, esto indica que los beneficios son mayores a los costos. En consecuencia, el proyecto debe ser considerado.

4.4.FACTIBILIDAD ECONÓMICA DEL PROYECTO

Este estudio permitirá realizar una evaluación sobre la conveniencia de invertir o no en este proyecto. Para ello, se realizará una comparación de los costos del sistema actual de control a los pacientes, con los costos proyectados con este proyecto en base a dispositivos electrónicos.

Para ello, se presenta la siguiente tabla con la descripción del sistema actual del personal de salud del centro de salud “Corazón de Jesús” donde se dispone de 20 camas.

Cargo	Salario/mes	Salario/día	Salario/hora	Salario/minuto
Personal de enfermería	3000 Bs	100 Bs	8,33 Bs	0,139 Bs

La fórmula a utilizar es:

$$Ca = \frac{\text{salario minuto} * \text{minutos utilizados al día}}{\text{días trabajados}} \quad (8)$$

En 12 horas de trabajo por turno, el personal realiza 10 rondas, y cada ronda le toma 10 minutos, haciendo un total de 100 minutos por día que se utilizan para realizar la ronda en promedio.

Como el salario por minuto es de 0,139 Bs y se trabaja 30 días, se debe multiplicar estas 3 cifras para tener un costo total para realizar esta tarea.

$$C_{\text{mes}} = 100 * 0,139 * 30 = 416,67 \text{ Bs/mes}$$

Para un año el resultado se multiplica por 12

$$C_a=416,67*12 = 5000 \text{ Bs/año}$$

Anualmente se tendría un gasto de 5000 Bs para realizar esta tarea actualmente.

Comparando con todos los costos que se requirieron para realizar este proyecto, los cuales se muestran a continuación:

Recursos	Monto (Bs)
Recursos humanos	5000
Recursos para el desarrollo	240
Recursos materiales	1635
TOTAL, SIN IMPREVISTOS	6875
Imprevistos (10%)	687,5
COSTO TOTAL EN RECURSOS	7562,5

Teniendo un total de 7562,5 Bs en recursos.

Con este proyecto, ya no será necesario realizar las 10 rondas por día, sino que todo el tiempo se estará en realizando otras tareas, mientras se tiene la pantalla abierta del control por base de datos, y en caso de necesitar comunicarse con un paciente, el tiempo se reduce a un promedio de 1 minuto, realizar la misma tarea de cada ronda y, como se

realizaban 10 rondas, el tiempo equivalente con el proyecto implementado será de 10 minutos

Como el salario por minuto es de 0,139 Bs y se trabaja 30 días, se debe multiplicar estas 3 cifras para tener un costo total para realizar esta tarea. Se utiliza nuevamente la ecuación (8).

$$C_{\text{mes}} = 10 * 0,139 * 30 = 41,667 \text{ Bs/mes}$$

Para un cálculo anual se multiplica el resultado por 12.

$$C_p = 41,667 * 12 = 500 \text{ Bs}$$

Anualmente se tendría un costo de 500 Bs

Tomando en cuenta el respectivo mantenimiento anual de 500 el gasto con el proyecto asciende a de 1000 Bs.

Como el presente proyecto tiene un tiempo de vida de 5 años, se debe considerar los montos a este tiempo con la siguiente fórmula:

$$\text{Gasto en 5 años} = \text{gasto anual} * 5 + \text{recurso total del proyecto} \quad (9)$$

Reemplazando en ambos sistemas:

$$\text{Gasto (sistema actual) en 5 años} = 5000 * 5 = 25000 \text{ Bs}$$

$$\text{Gasto (con proyecto) en 5 años} = 1000 * 5 + 7562,5 = 12562,5 \text{ Bs}$$

- Con el sistema actual se tendrá un gasto de 25000 Bs en 5 años
- Con el presente proyecto, se tendrá un gasto de 12562,5 Bs en 5 años, reduciendo a la mitad los gastos para realizar este trabajo.

Por tanto, el proyecto es factible y ayuda a la economía de los centros de salud que lo vayan a implementar.

A continuación, se muestra la tabla resumen de la inversión del proyecto:

Descripción	Costos totales de recursos (Bs)
Recursos humanos	5000
Recursos para el desarrollo	240
Recursos materiales	1635
TOTAL, SIN IMPREVISTOS	6875
Imprevistos (10%)	687,5
COSTO TOTAL DEL PROYECTO	7562,5
Mantenimiento por 5 años	2500
Salario a personal	2500
COSTO TOTAL DEL PROYECTO + MANTENIMIENTO Y SALARIOS POR 5 AÑOS	12562,5

CAPITULO V

CONCLUSIONES Y

RECOMENDACIONES

5.1.CONCLUSIONES

Luego de haber culminado el proyecto, se llega a las siguientes conclusiones:

- Se desarrolló sistemas electrónicos enlazados a una base de datos que mejoran la atención de los pacientes en hospitales y clínicas en la ciudad de La Paz, poniendo como ejemplo el hospital Corazón de Jesús en el que solo se contaban con dos enfermeras para veinte pacientes a los cuales ahora pueden atender de forma eficiente.
- Se diseñó un circuito receptor de ondas de audio a través de un micrófono, y ahora los pacientes pueden comunicarse con el personal médico de forma instantánea, sin tener que esperar a que los enfermeros pasen por ahí.
- Se diseñó un circuito reproductor de audio a través de un parlante, por el cual el personal médico puede comunicarse con sus pacientes sabiendo que necesitan.
- Se elaboró un programa para Arduino que transmite datos de audio a través de cable a varios dispositivos. La comunicación es en ambos sentidos y con un paciente a la vez para evitar confusiones.
- Se creó un software en base a Python para computadora que permite la comunicación de audio para recibir y transmitir la voz humana seleccionando de forma gráfica con que dispositivo se desea realizar la comunicación.
- Se elaboró una base de datos enlazada al software de comunicación que permita priorizar pacientes delicados. Ahora se puede ver el historial de cada paciente que solicita una llamada con el personal de enfermería, y en caso de una emergencia, la llamada es iniciada inmediatamente.

5.2.RECOMENDACIONES

- Para tener una buena calidad de audio, se recomienda no modificar la distancia de los micrófonos a los PCBs de los dispositivos esclavos o del amplificador, mucho menos poniendo un cable sin recubrimiento, ya que el ruido generado por los

cables estará presente.

- Si bien es posible implementar parlantes de 4 ohmios, se recomienda utilizar de 8 ohmios, para evitar que se quemen.
- Para el caso de pacientes de emergencia, se recomienda acercar más el dispositivo a su acceso, y configurar de la manera correcta en el programa de Python, para que la atención sea eficaz.
- Se recomienda no exceder las limitantes establecidas en el capítulo 3 para evitar dañar el sistema.

REFERENCIAS BIBLIOGRAFICAS

- Harvard T H Chan School of Public Health, Boston, MA, USA (2018). Sistemas de salud de alta calidad en la era de los Objetivos de Desarrollo Sostenible: es hora de una revolución. Recuperado de: <https://blogs.iadb.org/salud/es/la-mala-calidad-de-la-atencion-de-salud-mata/>
- LA IZQUIERDA DIARIO (2019). Claves para entender la crisis del sistema de salud en Bolivia. La Paz.
- Compupalace. (2020) ¿Qué es el sonido? Recuperado de: <https://soportemultimedia.com/que-es-el-sonido/>
- Apple Inc. (2022) Sonido y audición. Recuperado de: <https://www.apple.com/es/sound/#:~:text=La%20frecuencia%20de%20las%20ondas,de%2020%20a%2020.000%20Hz.>
- HOME STUDIO PRODUCTIONS. (2020) ¿Qué Es Un Preamplificador? ¿Por Qué Lo Necesitas? Recuperado de: <https://www.homestudioproductions.com/que-es-un-preamplificador-por-que-lo-necesitas/>
- Juan Carlos G. Amplificadores.info. (2022) PREAMPLIFICADOR: ¿QUÉ ES Y CÓMO FUNCIONA? TIPOS, ANÁLISIS Y APLICACIONES. Recuperado de: <https://amplificadores.info/amplificadores-de-potencia/preamplificador>
- YAMAHA Make Waves. (2022) ¿Qué son los amplificadores de potencia? Recuperado de: <https://mx.yamaha.com/es/products/contents/proaudio/musicianspa/equipments/amplifier.html>
- Compupalace. (2020) ¿Qué es el micrófono? Recuperado de: <https://soportemultimedia.com/que-es-el-microfono/>
- José Luis R. ComoFunciona. (2022) Como funciona un micrófono. Recuperado de: <https://como-funciona.co/un-microfono/>
- Curso de Acústica creado por GA. (2003) Altavoces. Recuperado de: <https://www.ehu.eus/acustica/espanol/electricidad/altaves/altaves.html>
- HETPRO/TUTORIALES. (2021) Microcontrolador – qué es y para qué sirve. Recuperado

de: <https://hetpro-store.com/TUTORIALES/microcontrolador/>

- HETPRO/TUTORIALES. (2021) Optoacoplador, que es y cómo funciona. Recuperado de: <https://hetpro-store.com/TUTORIALES/optoacoplador/>
- AULA21. (2020) Python: qué es, para qué sirve y cómo se programa. Recuperado de: <https://www.cursosaula21.com/que-es-python/>
- Oracle Mexico (2022) Base de datos definida. Recuperado de: <https://www.oracle.com/mx/database/what-is-database/>
- BTICINO DEL PERU S A. (2010) ¿Cómo funciona un amplificador de audio? Usos y tipos. Recuperado de: <https://bticino.com.pe/como-funciona-un-amplificador-de-audio-usos-y-tipos/>
- CARACTERÍSTICAS. (2022) SONIDO. Recuperado de: <https://www.caracteristicas.co/sonido/>
- Freepikcompany. (2022) El mezclador control remoto para grabación de sonido. Recuperado de: https://www.freepik.es/fotos-premium/mezclador-control-remoto-grabacion-sonido-ingeniero-sonido-trabajo-estudio-amplificador-sonido-mezclador-ecualizador-consola-dj_7686892.htm
- Aprendiendo Arduino. (2022) Microcontroladores. Recuperado de: <https://aprendiendoarduino.wordpress.com/2015/03/23/microcontroladores/>

ANEXOS

ANEXO 1. CÓDIGO DEL PROYECTO (PYTHON)

```
#importar librerias
from tkinter import ttk, PhotoImage, Label
from tkinter import messagebox
import os
import sys
import time
import tkinter
import MySQLdb
import numpy as np
import serial
import threading
```

```
tipo_letra = "KGRedHands"
tama_letra_tit = 20
tama_letra_bot = 11
estilo_letra = "bold"
#funciones
def cerrando():
    global b_tarea2
    global bandera
    b_tarea2=False
    bandera=False
    try:
        ser.close()
        tarea2.join(0.1)
    except:
        pass
    ventana.quit()
    ventana.destroy()
```

```
def lectura_ser():
    global b_tarea2
    global e
    global b_llamada
    ser.reset_input_buffer()
    while (b_tarea2):
        val = ser.readline()
        cade = val.decode('ascii')
        if cade:
```

```

encabezado = cade[:1]
dato = cade[1:]
if encabezado == "N":
    t_id.config(state="normal")
    t_id.delete(0, 5)
    t_id.insert(0,dato)
    t_id.config(state="readonly")
if encabezado == "E" and e[int(dato)] == 0:
    try:
        dat1=int(dato)
        codigo = 'bot_nuevo'+str(dat1)+''.config(bg="brown1")'
        exec(codigo)
        sql= """SELECT * FROM `dispositivos` WHERE
`ID`='""" +str(dat1)+"''";"""
        try:
            cursor.execute(sql)
            cursor.fetchall()
            for row in cursor:
                emer=row[3]
                paciente=row[1]
                if emer == 1:
                    b_llamada = False
                    llamada(dat1)
                    messagebox.showinfo(message="EL PACIENTE "+paciente+"
REQUIERE ATENCIÓN URGENTE", title="EMERGENCIA")
            except:
                mensaje_e.config(text="DISPOSITIVO NO REGISTRADO INTENTA
CONECTARSE")
                print ("error")
            except:
                mensaje_e.config(text="DISPOSITIVO NO REGISTRADO INTENTA
CONECTARSE")

def llamada(men):
    global e
    global b_llamada
    if e[men] == 0 and b_llamada == False:
        e[men]=1
        b_llamada = True
        codigo = 'bot_nuevo'+str(men)+''.config(bg="DarkOliveGreen1")'
        exec(codigo)
        cad = "H"+str(men)
        ser.write(cad.encode('ascii')+ b'\n')
    elif e[men] ==1:

```



```

e[men]=0
b_llamada = False
codigo = 'bot_nuevo'+str(men)+'.config(bg="SteelBlue1")'
exec(codigo)
cad = "F"+str(men)
ser.write(cad.encode('ascii')+ b'\n')

def diagnosticos_p(pac):
    global ventana_diag
    global ventana
    ventana_diag.state(newstate = "normal")
    ventana.state(newstate = "withdraw")

    sql= """SELECT * FROM `dispositivos` WHERE `ID`="""+str(pac)+""";"""
    try:
        cursor.execute(sql)
        cursor.fetchall()
        for row in cursor:
            diag=row[2]
            paciente=row[1]
            mensaje_d.config(text="El paciente "+paciente+" tiene un cuadro de: "+diag)
    except:
        mensaje_d.config(text="FALLO AL CONECTAR A BASE DE DATOS")

def modificar_p(pac):
    global ventana_nueva
    global ventana
    ventana_nueva.state(newstate = "normal")
    ventana.state(newstate = "withdraw")
    modi_b.config(state = "normal")
    borrar.config(state = "normal")
    agregar.config(state = "disabled")
    sql= """SELECT * FROM `dispositivos` WHERE `ID`="""+str(pac)+""";"""
    try:
        cursor.execute(sql)
        cursor.fetchall()
        for row in cursor:
            Id=row[0]
            paciente=row[1]
            diag=row[2]
            emer=row[3]
            cama=row[4]
    except:
        mensaje_d.config(text="FALLO AL CONECTAR A BASE DE DATOS")

```

```

t_id.config(state="normal")
t_id.insert(0,Id)
t_id.config(state="readonly")
t_paciente.insert(0,paciente)
t_diagnostico.insert(0,diag)
t_emergencia.current(emer)
t_cama.insert(0,cama)

def serial_Con(mens):
    global ser
    try:
        ser = serial.Serial(mens, 9600)
        time.sleep(1)
        mensaje_e.config(text="PUERTO CONECTADO")
        time.sleep(0.5)
        cad = "E"
        ser.write(cad.encode('ascii')+ b'\n')

        sql2= """SELECT * FROM `com`"""
        try:
            cursor.execute(sql2)
            cursor.fetchall()
            for row in cursor:
                cad = "D"+str(row[3])
                ser.write(cad.encode('ascii')+ b'\n')
            db_local.commit()
        except:
            db_local.rollback()
            mensaje_e.config(text="FALLO AL ENVIAR DISPO")

        conectar.config(state="disabled")
        nuevo.config(state="normal")
        tarea2.start()
        sql3="""UPDATE `com` SET `COM` = """+mens[3:]+"""" WHERE
`com`.`ID_COM` = 1;"""
        try:
            # Ejecutar un comando SQL
            cursor.execute(sql3)
            db_local.commit()
        except:
            db_local.rollback()

```

```

        print ("No se pudo guardar")
    except:
        mensaje_e.config(text="FALLO AL CONECTAR AL PUERTO")

def nuevo_Pac():
    global ventana_nueva
    global ventana
    cad = "N"
    ser.write(cad.encode('ascii')+ b'\n')
    ventana_nueva.state(newstate = "normal")
    ventana.state(newstate = "withdraw")
    modi_b.config(state = "disabled")
    borrar.config(state = "disabled")
    agregar.config(state = "normal")

def modifiko():
    global b_tarea2
    global ventana_nueva
    global ventana
    sql3="""UPDATE `dispositivos` SET `PACIENTE` = """+t_paciente.get()+""",
`DIAGNOSTICO` = """+t_diagnostico.get()+""", `EMERGENCIA` =
"""+str(t_emergencia.current()+""", `CAMA` = """+t_cama.get()+""", WHERE
`dispositivos`.`ID` = """+t_id.get()+""";"""
    try:
        # Ejecutar un comando SQL
        cursor.execute(sql3)
        db_local.commit()
        print("Guardando en base de datos...OK")
        ventana_nueva.state(newstate = "withdraw")
        ventana.state(newstate = "normal")
        b_tarea2=False
        ser.close()
        tarea2.join(0.1)
        ventana.quit()
        ventana.destroy()
    except:
        db_local.rollback()
        print ("No se pudo guardar")
def cambio(event):
    # Crear una ventana secundaria.
    ventana_nombre = tkinter.Toplevel()
    ventana_nombre.title("HOSPITAL")
    ventana_nombre.geometry("300x200")
    t_hospital=tkinter.Entry(ventana_nombre)

```

```

t_hospital.place(x=50, y=50)
boton_cerrar = tk.Button(ventana_nombre,text="CAMBIAR",
command=lambda:nuevo_hosp(t_hospital.get()))
boton_cerrar.place(x=35, y=75)
boton_cambiar = tk.Button(ventana_nombre,text="CANCELAR",
command=ventana_nombre.destroy)
boton_cambiar.place(x=135, y=75)

def nuevo_hosp(nombre):
    sql3=""""UPDATE `com` SET `NOMBRE_HOSP` = """+nombre+"""" WHERE
`com`.`ID_COM` = 1;""""
    try:
        # Ejecutar un comando SQL
        cursor.execute(sql3)
        db_local.commit()
    except:
        db_local.rollback()
        print ("No se pudo guardar")
    titulo.config(text=nombre)

def borra_p():
    global b_tarea2
    global ventana_nueva
    global ventana
    respuesta=messagebox.askyesno("Cuidado", "¿Quiere borrar el paciente de la base de
datos?")
    if respuesta==True:
        sql3=""""DELETE FROM `dispositivos` WHERE `dispositivos`.`ID` =
"""+t_id.get()
        try:
            # Ejecutar un comando SQL
            cursor.execute(sql3)
            db_local.commit()
            print("Borrado de la base de datos...OK")
            ventana_nueva.state(newstate = "withdraw")
            ventana.state(newstate = "normal")
            b_tarea2=False
            ser.close()
            tarea2.join(0.1)
            ventana.quit()
            ventana.destroy()
        except:
            db_local.rollback()
            print ("No se pudo borrar")

```

```

def regreso():
    ventana_nueva.state(newstate = "withdraw")
    ventana.state(newstate = "normal")
    cad = "E"
    ser.write(cad.encode('ascii')+ b'\n')

def registro():
    global b_tarea2
    global ventana_nueva
    global ventana
    sql = """INSERT INTO `dispositivos` (`ID`, `PACIENTE`, `DIAGNOSTICO`,
`EMERGENCIA`, `CAMA`) VALUES ("""+t_id.get()+""", """+t_paciente.get()+""",
"""+t_diagnostico.get()+""", """+str(t_emergencia.current()+""",
"""+t_cama.get()+""");"""
    try:
        # Ejecutar un comando SQL
        cursor.execute(sql)
        db_local.commit()
        print("Guardando en base de datos...OK")
        ventana_nueva.state(newstate = "withdraw")
        ventana.state(newstate = "normal")
        b_tarea2=False
        ser.close()
        tarea2.join(0.1)
        ventana.quit()
        ventana.destroy()
    except:
        db_local.rollback()
        print ("No se pudo guardar")

def acepta():
    ventana_diag.state(newstate = "withdraw")
    ventana.state(newstate = "normal")

def selection_changed(event):
    cad = "D"+str(t_modulo.current()+1)
    print (cad)
    ser.write(cad.encode('ascii')+ b'\n')

bandera = True
while bandera:
    #variables

opciones=["COM1","COM2","COM3","COM4","COM5","COM6","COM7","COM8","

```

```

COM9","COM10"]
nombre_hospital = "HOSPITAL DE PRUEBA"
opciones_emer=["NO","SI"]
opciones_modul = ["1","2","3","4","5","6","7","8","9","10"]
tarea2 = threading.Thread(target = lectura_ser)
b_tarea2=True
e = list(range(160))
b_llamada = False
# inicio tkinter
ventana = tkinter.Tk()
ventana.title("SISTEMA DE CONTROL DE PACIENTES")
ventana.geometry("1250x600")
bg = PhotoImage(file = "fondo.png")
bg_botonpac = PhotoImage(file = "botonin.png")
frame1 = tkinter.Frame(ventana, bg="LightBlue3")
frame1.place(relx=0.0,relx=0.0, relwidth=1.0, relheight=0.1)

frame2 = tkinter.Frame(ventana, bg="LightBlue3")
frame2.place(relx=0.0,relx=0.1, relwidth=0.5, relheight=0.05)
frame3 = tkinter.Frame(ventana, bg="LightBlue3")
frame3.place(relx=0.5,relx=0.1, relwidth=0.5, relheight=0.05)
frame4 = tkinter.Frame(ventana, bg="black")
frame4.place(relx=0.0,relx=0.15, relwidth=1.0, relheight=0.8)
frame5 = tkinter.Frame(ventana)
frame5.place(relx=0.0,relx=0.95, relwidth=1.0, relheight=0.05)

background = Label(frame4, image = bg)
background.place(x = 0, y = 0, relwidth = 1, relheight = 1)
ventana_nueva = tkinter.Toplevel()
ventana_nueva.title("NUEVO PACIENTE")
ventana_nueva.state(newstate = "withdraw")
ventana_nueva.geometry("500x400")

frame6 = tkinter.Frame(ventana_nueva, bg="LightBlue1")
frame6.place(relx=0.0,relx=0.0, relwidth=1.0, relheight=0.1)
frame7 = tkinter.Frame(ventana_nueva, bg="LightBlue2")
frame7.place(relx=0.0,relx=0.1, relwidth=1.0, relheight=0.8)
frame8 = tkinter.Frame(ventana_nueva, bg="LightBlue3")
frame8.place(relx=0.0,relx=0.9, relwidth=1.0, relheight=0.1)

ventana_diag = tkinter.Toplevel()
ventana_diag.title("DIAGNOSTICO")
ventana_diag.state(newstate = "withdraw")
ventana_diag.geometry("500x400")

```

```

frame9 = tkinter.Frame(ventana_diag, bg="LightBlue1")
frame9.place(relx=0.0,rely=0.0, relwidth=1.0, relheight=0.1)
frame10 = tkinter.Frame(ventana_diag, bg="LightBlue2")
frame10.place(relx=0.0,rely=0.1, relwidth=1.0, relheight=0.8)
frame11 = tkinter.Frame(ventana_diag, bg="LightBlue3")
frame11.place(relx=0.0,rely=0.9, relwidth=1.0, relheight=0.1)

# agregando titulos

conec_p = tkinter.Label(frame2,bg="LightBlue3", text="SELECCIONAR
PUERTO",font=(tipo_letra, tama_letra_bot, estilo_letra))
conec_p.place(relx=0.05,rely=0.0, relwidth=0.3, relheight=1)
mensaje_e = tkinter.Label(frame5, text=" ",font=(tipo_letra, tama_letra_bot,
estilo_letra))
mensaje_e.pack()
nuevo_p = tkinter.Label(frame3, bg="LightBlue3", text="MODIFICAR BASE DE
DATOS",font=(tipo_letra, tama_letra_bot, estilo_letra))
nuevo_p.place(relx=0.05,rely=0.0, relwidth=0.4, relheight=1)

l_modulo=tkinter.Label(frame7, text="N° DE MÓDULOS",
bg="LightBlue2",font=(tipo_letra, tama_letra_bot, estilo_letra))
l_id=tkinter.Label(frame7, text="DISPOSITIVO", bg="LightBlue2",font=(tipo_letra,
tama_letra_bot, estilo_letra))
l_paciente=tkinter.Label(frame7, text="NOMBRE DEL PACIENTE",
bg="LightBlue2",font=(tipo_letra, tama_letra_bot, estilo_letra))
l_diagnostico=tkinter.Label(frame7, text="DIAGNOSTICO",
bg="LightBlue2",font=(tipo_letra, tama_letra_bot, estilo_letra))
l_emergencia=tkinter.Label(frame7, text="EMERGENCIA",
bg="LightBlue2",font=(tipo_letra, tama_letra_bot, estilo_letra))
l_cama=tkinter.Label(frame7, text="LOCALIZACIÓN CAMA",
bg="LightBlue2",font=(tipo_letra, tama_letra_bot, estilo_letra))
l_modulo.place(relx=0.05,rely=0.1, relwidth=0.4, relheight=0.1)
l_id.place(relx=0.05,rely=0.25, relwidth=0.4, relheight=0.1)
l_paciente.place(relx=0.05,rely=0.4, relwidth=0.4, relheight=0.1)
l_diagnostico.place(relx=0.05,rely=0.55, relwidth=0.4, relheight=0.1)
l_emergencia.place(relx=0.05,rely=0.7, relwidth=0.4, relheight=0.1)
l_cama.place(relx=0.05,rely=0.85, relwidth=0.4, relheight=0.1)
t_id=tkinter.Entry(frame7, state="readonly",font=(tipo_letra, tama_letra_bot,
estilo_letra))

```

```

t_modulo=ttk.Combobox(frame7,width="2",values=opciones_modul,
state="readonly",font=(tipo_letra, tama_letra_bot, estilo_letra))
t_modulo.bind("<<ComboboxSelected>>", selection_changed)
t_paciente=tkinter.Entry(frame7,font=(tipo_letra, tama_letra_bot, estilo_letra))
t_diagnostico=tkinter.Entry(frame7,font=(tipo_letra, tama_letra_bot, estilo_letra))
t_emergencia=ttk.Combobox(frame7,width="2",values=opciones_emer,
state="readonly",font=(tipo_letra, tama_letra_bot, estilo_letra))
t_cama=tkinter.Entry(frame7,font=(tipo_letra, tama_letra_bot, estilo_letra))
t_modulo.place(relx=0.45,rely=0.1, relwidth=0.2, relheight=0.1)
t_id.place(relx=0.45,rely=0.25, relwidth=0.2, relheight=0.1)
t_paciente.place(relx=0.45,rely=0.4, relwidth=0.2, relheight=0.1)
t_diagnostico.place(relx=0.45,rely=0.55, relwidth=0.2, relheight=0.1)
t_emergencia.place(relx=0.45,rely=0.7, relwidth=0.2, relheight=0.1)
t_cama.place(relx=0.45,rely=0.85, relwidth=0.2, relheight=0.1)
t_emergencia.current(0)

```

```

mensaje_e2 = tkinter.Label(frame8, text="PRESIONE BOTÓN DEL NUEVO
DISPOSITIVO",font=(tipo_letra, tama_letra_bot, estilo_letra))
mensaje_e2.pack()

```

```

mensaje_d = tkinter.Label(frame10, text="",font=(tipo_letra, tama_letra_bot,
estilo_letra))
mensaje_d.pack()

```

```

#conectando a base de datos

```

```

try:

```

```

    db_local = MySQLdb.connect("127.0.0.1","root","", "base")
    mensaje_e.config(text="CONECTADA A BASE DE DATOS")

```

```

except:

```

```

    mensaje_e.config(text="FALLO AL CONECTAR A BASE DE DATOS")
    time.sleep(3)
    sys.exit()

```

```

# Preparando cursor

```

```

cursor = db_local.cursor()

```

```

sql= """"SELECT * FROM `com`""""

```

```

try:

```

```

    cursor.execute(sql)
    cursor.fetchall()
    for row in cursor:
        modu=row[3]-1
        nombre_hospital=row[2]

```

```

except:

```

```

    mensaje_d.config(text="FALLO AL CONECTAR A BASE DE DATOS")

```



```

t_modulo.current(modu)

titulo = tkinter.Label(frame1,bg= 'LightBlue3', text=nombre_hospital,font=(tipo_letra,
tama_letra_tit, estilo_letra))
titulo.pack()
titulo.bind("<Double-Button-1>",cambio)

titulo2 = tkinter.Label(frame6, text=nombre_hospital,bg=
'LightBlue1',font=(tipo_letra, tama_letra_tit, estilo_letra))
titulo2.pack()

titulo3 = tkinter.Label(frame9, text=nombre_hospital,bg=
'LightBlue1',font=(tipo_letra, tama_letra_tit, estilo_letra))
titulo3.pack()

# agregando botones

combo = ttk.Combobox(frame2,width="5",values=opciones,
state="readonly",font=(tipo_letra, tama_letra_bot, estilo_letra))
combo.place(relx=0.4,rely=0.0, relwidth=0.2, relheight=1)
combo.current(0)

conectar = tkinter.Button(frame2,bg="lime green",font=(tipo_letra, tama_letra_bot,
estilo_letra), text = "CONECTAR", command = lambda:serial_Con(combo.get()))
conectar.place(relx=0.7,rely=0.0, relwidth=0.25, relheight=1)

nuevo = tkinter.Button(frame3,bg="lime green",font=(tipo_letra, tama_letra_bot,
estilo_letra), state= "disabled" ,text = "NUEVO", command = nuevo_Pac)
nuevo.place(relx=0.7,rely=0.0, relwidth=0.25, relheight=1)

sql= """SELECT * FROM `dispositivos`"""
try:
    cursor.execute(sql)
    cursor.fetchall()
    xx=1
    yy=1
    for row in cursor:
        idcod=row[0]
        codigo= 'cod'+str(idcod)+'=idcod'
        exec(codigo)
        e[idcod]=0
        paciente = row[1]
        diagnostico = row[2]
    # crear botones

```

```

        codigo='bot_nuevo'+str(idcod)+'                                     =
tkinter.Button(frame4,bg="SteelBlue1",font=(tipo_letra, tama_letra_bot, estilo_letra),
text = paciente, command =lambda:llamada(cod'+str(idcod)+'))'
        exec(codigo)
        codigo='bot_nuevo'+str(idcod)+'place(relx=(0.1*(xx-
1)+0.03*xx),rely=(0.1*yy+0.03*yy), relwidth=0.1, relheight=0.1)'
        exec(codigo)

        codigo='diag_nuevo'+str(idcod)+'                                     =
tkinter.Button(frame4,bg="plum2",font=(tipo_letra, tama_letra_bot, estilo_letra), text =
"D", command =lambda:diagnosticos_p(cod'+str(idcod)+'))'
        exec(codigo)
        codigo='diag_nuevo'+str(idcod)+'place(relx=(0.1*(xx-
1)+(0.03*xx)+0.1),rely=(0.1*yy+0.03*yy), relwidth=0.015, relheight=0.05)'
        exec(codigo)

        codigo='modi_nuevo'+str(idcod)+'                                     =
tkinter.Button(frame4,bg="tan1",font=(tipo_letra, tama_letra_bot, estilo_letra), text =
"M", command =lambda:modificar_p(cod'+str(idcod)+'))'
        exec(codigo)
        codigo='modi_nuevo'+str(idcod)+'place(relx=(0.1*(xx-
1)+(0.03*xx)+0.1),rely=(0.1*yy+0.03*yy+0.05), relwidth=0.015, relheight=0.05)'
        exec(codigo)
        xx=xx+1
        if xx==8:
            xx=1
            yy=yy+1
        db_local.commit()

except:
    db_local.rollback()
    mensaje_e.config(text="FALLO AL BUSCAR PACIENTES EN BASE DE
DATOS")

agregar = tkinter.Button(frame7,bg="DarkOliveGreen1",text = "AGREGAR",
command = registro,font=(tipo_letra, tama_letra_bot, estilo_letra))
agregar.place(relx=0.7,rely=0.2, relwidth=0.2, relheight=0.1)
modi_b = tkinter.Button(frame7,bg="DarkOliveGreen1",text = "MODIFICAR",
command = modifiko,font=(tipo_letra, tama_letra_bot, estilo_letra))
modi_b.place(relx=0.7,rely=0.4, relwidth=0.2, relheight=0.1)
borrar = tkinter.Button(frame7,bg="DarkOliveGreen1",text = "BORRAR", command
= borra_p,font=(tipo_letra, tama_letra_bot, estilo_letra))
borrar.place(relx=0.7,rely=0.6, relwidth=0.2, relheight=0.1)

```

```

cancelar_n = tkinter.Button(frame7,bg="DarkOliveGreen1",text = "CANCELAR",
command = regreso,font=(tipo_letra, tama_letra_bot, estilo_letra))
cancelar_n.place(relx=0.7,rely=0.8, relwidth=0.2, relheight=0.1)

aceptar_diag = tkinter.Button(frame11,bg="DarkOliveGreen1",text = "ACEPTAR",
command = acepta,font=(tipo_letra, tama_letra_bot, estilo_letra))
aceptar_diag.pack()

# intentado conectar a puerto COM
sql= """SELECT * FROM `com`"""
try:
    cursor.execute(sql)
    cursor.fetchall()
    for row in cursor:
        puerto = "COM"+str(row[1])
        serial_Con(puerto)
        db_local.commit()
except:
    db_local.rollback()
    mensaje_e.config(text="FALLO AL CONECTARSE A PUERTO COM")

#cerrando ventanas
ventana.wm_protocol("WM_DELETE_WINDOW", cerrando)
ventana_nueva.wm_protocol("WM_DELETE_WINDOW", cerrando)
ventana_diag.wm_protocol("WM_DELETE_WINDOW", cerrando)

#para guardar registro tkinter
ventana_nueva.mainloop()
ventana_diag.mainloop()
ventana.mainloop()

```

ANEXO 2. CÓDIGO DEL PROYECTO (ARDUINO)

```

//variables
#define SL_165 4
#define CLK_165 3
#define SO_165 5
#define SER_595 6
#define RCLK_595 7
#define SRCLK_595 8
#define led 9
String cadena;
String cad1;

```

```

int dispositivos = 0;
int modo = 0;
int esclavo=0;
//funciones

void leer_dispositivo(){
  digitalWrite(SL_165, HIGH);
  delay(1);
  digitalWrite(SL_165, LOW);
  delay(1);
  digitalWrite(SL_165, HIGH);
  for (int i=dispositivos; i>0; i--){
    int n = digitalRead(SO_165);
    if (n==0){
      if (modo == 0){
        Serial.print("E");
      }
      if (modo == 1){
        Serial.print("N");
      }
      Serial.println(i);
      delay(100);
    }
    digitalWrite(CLK_165, LOW);
    delay(1);
    digitalWrite(CLK_165, HIGH);
    delay(1);
  }
  delay(1);
}

void llamar_dispositivo(int num){
  int habilitado = num % 8;
  int espacios = num / 8;
  int complemento = dispositivos / 8;
  int pot = potencia(habilitado);
  digitalWrite(RCLK_595, LOW);
  for (int i=0; i<(complemento - 1 -espacios);i++){
    shiftOut(SER_595, SRCLK_595, MSBFIRST, 0);
  }
  shiftOut(SER_595, SRCLK_595, MSBFIRST, pot);
  for (int i=0; i<espacios;i++){
    shiftOut(SER_595, SRCLK_595, MSBFIRST, 0);
  }
}

```

```

    Serial.println(pot);
    digitalWrite(RCLK_595, HIGH);
}

int potencia(int exponente){
    int resul=1;
    for (int i =0; i<exponente; i++){
        resul = resul*2;
    }
    return resul;
}

void led_mod0(){
    if (modo == 1){
        digitalWrite(led, HIGH);
        delay(100);
        digitalWrite(led, LOW);
        delay(100);
    }
}

void setup() {
    pinMode(SL_165, OUTPUT);
    pinMode(CLK_165, OUTPUT);
    pinMode(SO_165, INPUT);
    digitalWrite(SL_165, HIGH);
    pinMode(SER_595, OUTPUT);
    pinMode(RCLK_595, OUTPUT);
    pinMode(SRCLK_595, OUTPUT);
    pinMode (9, OUTPUT);
    digitalWrite(9, HIGH);
    Serial.begin(9600);
    llamar_dispositivo(0);
}

void loop() {
    if (Serial.available() > 0) {
        cadena = Serial.readStringUntil('\n');
        //cadena = cadena.substring(0, cadena.length()-1);
        cad1 = cadena.substring(0,1);
        if (cadena == "N"){
            modo = 1;
        }
        else if (cadena == "E"){

```

```

    modo = 0;
    digitalWrite(led, HIGH);
}
else if (cad1 == "H"){
    esclavo =cadena.substring(1).toInt()-1;
    Serial.print (esclavo);
    llamar_dispositivo(esclavo);
}
else if (cad1 == "F"){
    esclavo =0;
    llamar_dispositivo(esclavo);
}
else if (cad1 == "D"){
    dispositivos =(cadena.substring(1).toInt()*8;
    Serial.println (dispositivos);
    llamar_dispositivo(0);
}
//Serial.print(cadena);
}
leer_dispositivo();
led_mod0();
}

```