



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN ELECTRÓNICA

Y REDES DE COMUNICACIÓN

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

TEMA:

**“SISTEMA DE MONITOREO DE RITMO CARDÍACO (S.M.R.C.) PARA
PERSONAS QUE REALIZAN EJERCICIO FÍSICO AERÓBICO”**

AUTOR: EDISON MARCOS FLORES BOSMEDIANO

DIRECTOR: ING. DANIEL JARAMILLO

IBARRA – ECUADOR

2015



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación,

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR			
CEDULA DE IDENTIDAD	1003431986		
APELLIDOS Y NOMBRES	FLORES BOSMEDIANO EDISON MARCOS		
DIRECCIÓN	LOS CEIBOS, RÍO JUBONES 3-42 Y PASTAZA		
E-MAIL	em_floresb@hotmail.com		
TELÉFONO FIJO	062-954-319	TELÉFONO MÓVIL	0993430040
DATOS DE LA OBRA			
TÍTULO	"SISTEMA DE MONITOREO DE RITMO CARDÍACO (S.M.R.C.) PARA PERSONAS QUE REALIZAN EJERCICIO FÍSICO AERÓBICO"		
AUTOR	FLORES BOSMEDIANO EDISON MARCOS		
FECHA	DICIEMBRE DEL 2015		
PROGRAMA	PREGRADO		
TÍTULO POR EL QUE OPTA	INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN		
ASESOR	ING. DANIEL JARAMILLO		

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Edison Marcos Flores Bosmediano, con cédula de identidad Nro. 1003431986, en calidad de autor (es) y titular (es) de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, al 01 día del mes de Diciembre de 2015.

EL AUTOR:

Firma: .....

Nombre: Edison Marcos Flores B.

Cedula: 1003431986

Firma: .....

Ing. Betty Chávez

Encargada Biblioteca



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIA

Yo, Edison Marcos Flores Bosmediano declaré bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondiente a este trabajo, a la Universidad Técnica del Norte, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

.....
Firma

Nombre: Edison Marcos Flores Bosmediano

Cédula: 1003431986

Ibarra, Diciembre del 2015



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA
UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, Edison Marcos Flores Bosmediano, con cédula de identidad Nro. 1003431986, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado: **“SISTEMA DE MONITOREO DE RITMO CARDÍACO (S.M.R.C.) PARA PERSONAS QUE REALIZAN EJERCICIO FÍSICO AERÓBICO”**, que ha sido desarrollado para optar por el título de: Ingeniero en Electrónica y Redes de Comunicación en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

.....
Firma

Nombre: Edison Marcos Flores Bosmediano

Cédula: 1003431986

Ibarra, Diciembre del 2015



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL DIRECTOR

Certifico que bajo mi dirección el trabajo de grado: **“SISTEMA DE MONITOREO DE RITMO CARDÍACO (S.M.R.C.) PARA PERSONAS QUE REALIZAN EJERCICIO FÍSICO AERÓBICO”**, fue desarrollado en su totalidad por el señor egresado Edison Marcos Flores Bosmediano, previo a la obtención del título de Ingeniero en Electrónica y Redes de Comunicación.

Certifico que ha sido dirigido en todas sus partes, cumpliendo con todas las disposiciones emitidas por la Universidad Técnica del Norte en lo referente a la elaboración del Trabajo de Grado.

Por lo expuesto:

Autorizo su presentación ante los organismos competentes para sustentación del mismo.

A handwritten signature in blue ink, which appears to read "D. Jaramillo", is written over a blue oval stamp. Below the signature, there are several horizontal blue lines.

Ing. Daniel Jaramillo

DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

Aunque no es suficiente, agradecer es muy importante; debido a todas las facilidades y herramientas prestadas, a la ayuda brindada, y el integrado conjunto de conocimientos entregados por parte de la Universidad Técnica del Norte para el desarrollo a favor del progreso del país, al ser educado con la visión, misión y valores éticos y morales para ser un futuro ingeniero en electrónica y redes de comunicación.

De igual forma, como parte fundamental del trabajo presente, es el agradecimiento a la labor de la Facultad de Ingeniería en Ciencias Aplicadas, sus autoridades, miembros de las diferentes áreas de la carrera de Electrónica y Redes de Comunicación.

A mi director de tesis, Ingeniero Daniel Jaramillo por su colaboración, ayuda y por guiarme durante el desarrollo de esta investigación, factor importante para culminar con éxito mi propósito

Un agradecimiento especial al Ing. Paúl Rosero por su ayuda incondicional y constante en el desarrollo de esta investigación; persona muy importante que ayudo con sus conocimientos y más elementos sustanciales en el proceso de esta investigación.

De igual manera agradezco a los ingenieros Calos Vásquez, Jaime Michilena, Edgar Maya y Roberto Marcillo por sus enseñanzas, por ejemplo de grandes profesionales a base de mucho esfuerzo y de ser personas de bien.

Edison Marcos Flores Bosmediano



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

El presente trabajo de grado, es fruto del constante compromiso hacia el estudio y el aprendizaje, está dedicado a “DIOS” que es mi luz en todos los ámbitos, a una persona especial que siempre ha estado a mi lado apoyándome y finalmente, a mi pilar fundamental en mi vida “Mis Padres”, Marco Antonio Flores Miño y Guadalupe de Lourdes Bosmediano Cabrera; los cuales me han guiado y apoyado siempre en todas las actividades realizadas para mejorar mi desempeño profesional.

Edison Marcos Flores Bosmediano

ÍNDICE DE CONTENIDO

AUTORIZACIÓN DE USO Y PUBLICACIÓN	II
CONSTANCIA.....	IV
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	V
CERTIFICACIÓN DEL DIRECTOR.....	VI
AGRADECIMIENTO.....	VII
DEDICATORIA.....	VIII
ÍNDICE DE CONTENIDO.....	IX
ÍNDICE DE FIGURAS	XXI
ÍNDICE DE TABLAS	XXVIII
RESUMEN.....	XXX
ABSTRACT	XXXI
PRESENTACIÓN	XXXII
CAPÍTULO I	1
1 ANTECEDENTES	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	1
1.2 JUSTIFICACIÓN	2
1.3 ALCANCE.....	4
1.4 OBJETIVOS	5
1.4.1 OBJETIVO GENERAL	5
1.4.2 OBJETIVOS ESPECÍFICOS	5
CAPÍTULO II	6
2 FUNDAMENTOS TEÓRICOS.....	6
2.1 FUNDAMENTOS DE TECNOLOGÍA ARDUINO	6
2.1.1 TECNOLOGÍA ARDUINO	6
2.1.2 CARACTERÍSTICAS DE ARDUINO.....	6
2.1.3 ALIMENTACIÓN	7
2.1.3.1 GND.....	9

2.1.3.2 VIN.....	9
2.1.3.3 PIN 5 V	10
2.1.3.4 PIN 3,3 V	10
2.1.4 ENTRADAS Y SALIDAS DIGITALES	10
2.1.5 ENTRADAS ANALÓGICAS	11
2.1.6 SALIDAS ANALÓGICAS (PWM)	11
2.1.7 PINES TRANSMISIÓN Y RECEPCIÓN.....	12
2.1.8 HARDWARE LIBRE: ARDUINO	12
2.1.9 SOFTWARE LIBRE DE ARDUINO.....	13
2.1.9.1 IDE DE ARDUINO.....	14
2.1.9.2 ESTRUCTURA GENERAL DE UN SKETCH DE ARDUINO.....	15
2.1.10 PLACAS DE ARDUINO.....	15
2.1.10.1 ARDUINO UNO	16
2.1.10.2 ARDUINO MEGA 2560	16
2.1.10.3 ARDUINO ETHERNET	16
2.1.10.4 ARDUINO NANO	17
2.1.10.5 ARDUINO MINI	17
2.1.10.6 ARDUINO LEONARDO.....	17
2.1.10.7 ARDUINO DUE	18
2.1.11 ARDUINO LILYPAD.....	18
2.1.11.1 LILYPAD USB	18
2.1.11.1.1 CARACTERÍSTICAS FÍSICAS DE ARDUINO LILYPAD USB	19
2.1.11.1.2 POTENCIA DE ARDUINO LILYPAD USB.....	19
2.1.11.1.3 MEMORIA DE ARDUINO LILYPAD USB	20
2.1.11.1.4 ENTRADAS Y SALIDAS DE ARDUINO LILYPAD USB	20
2.1.11.1.5 COMUNICACIÓN DE ARDUINO LILYPAD USB.....	21
2.1.11.1.6 PROGRAMACIÓN DE ARDUINO LILYPAD USB	21
2.1.11.2 LILYPAD ARDUINO ATMEGA168.....	22
2.1.11.2.1 PROGRAMACIÓN DE ARDUINO LILYPAD.....	22

2.1.11.2.2 POTENCIA DE ARDUINO LILYPAD	23
2.1.11.2.3 CARACTERÍSTICAS FÍSICAS DE ARDUINO LILYPAD.....	23
2.1.11.2.4 LAVABILIDAD DE ARDUINO LILYPAD.....	23
2.1.11.3 ARDUINO LILYPAD SIMPLE.....	23
2.1.11.3.1 PROGRAMACIÓN DE ARDUINO LILYPAD SIMPLE	24
2.1.11.3.2 POTENCIA DE ARDUINO LILYPAD SIMPLE	24
2.1.11.3.3 ENTRADAS Y SALIDAS DE ARDUINO LILYPAD SIMPLE	25
2.1.11.3.4 CARACTERÍSTICAS FÍSICAS DE ARDUINO LILYPAD SIMPLE	25
2.1.11.3.5 LAVABILIDAD DE ARDUINO LILYPAD SIMPLE	25
2.1.12 TABLA COMPARATIVA DE LAS PLACAS ELECTRÓNICAS ARDUINO.....	26
2.1.13 ARDUINO LILYPAD Y TEXTILES INTELIGENTES	28
2.1.13.1 TEXTILES INTELIGENTES	29
2.1.13.2 ROPA INTELIGENTE.....	30
2.2 FUNDAMENTOS DE COMUNICACIÓN INALÁMBRICA	30
2.2.1 INTRODUCCIÓN A LA COMUNICACIÓN INALÁMBRICA	30
2.2.2 REDES WPAN	31
2.2.2.1 FUNCIONAMIENTO DE REDES WPAN	31
2.2.3 INTRODUCCIÓN DE TECNOLOGÍA BLUETOOTH	31
2.2.3.1 COMUNICACIÓN BLUETOOTH.....	32
2.2.3.2 VERSIONES DE BLUETOOTH	32
2.2.3.2.1 BLUETOOTH VERSIÓN 1.1	32
2.2.3.2.2 BLUETOOTH VERSIÓN 1.1	33
2.2.3.2.3 BLUETOOTH VERSIÓN 2.0.....	33
2.2.3.2.4 BLUETOOTH VERSIÓN 2.1	34
2.2.3.2.5 BLUETOOTH VERSIÓN 3.0	34
2.2.3.2.6 BLUETOOTH VERSIÓN 4.0	34
2.2.3.3 CARACTERÍSTICAS DE FUNCIONAMIENTO DE BLUETOOTH	35
2.2.3.4 TOPOLOGÍAS DE RED	36
2.2.3.4.1 TOPOLOGÍA PUNTO A PUNTO	36

2.2.3.4.2 TOPOLOGÍA PICONET	37
2.2.3.4.3 TOPOLOGÍA SCATTER-NET	38
2.2.3.4.3.1 TOPOLOGÍA MAESTRO-ESCLAVO	38
2.2.3.4.3.2 TOPOLOGÍA ESCLAVO-ESCLAVO	38
2.2.3.5 ESTRUCTURA DE LA TRAMA DE BLUETOOTH	39
2.2.3.6 PRINCIPIO DE COMUNICACIÓN BLUETOOTH	39
2.2.3.7 ESTABLECIMIENTO DE CONEXIÓN BLUETOOTH	40
2.2.3.8 INTERFERENCIAS	41
2.2.4 MÓDULOS BLUETOOTH	42
2.2.4.1 MÓDULO BLUETOOTH HC-05	43
2.2.4.2 MÓDULO BLUETOOTH HC-06	43
2.2.5 NORMAS DE TECNOLOGÍA BLUETOOTH	44
2.2.6 BENEFICIOS DE COMUNICACIÓN BLUETOOTH	44
2.2.7 DESVENTAJAS DE COMUNICACIÓN BLUETOOTH VS ZIGBEE	45
2.2.8 COMPARACIÓN DE TECNOLOGÍAS: WI-FI, BLUETOOTH Y ZIGBEE	45
2.3 SENSORES ELECTRÓNICOS	46
2.3.1 SENSORES ANALÓGICOS	46
2.3.2 SENSORES ÓPTICOS	47
2.3.3 PULSE SENSOR	47
2.3.3.1 CARACTERÍSTICAS DE PULSE SENSOR	48
2.3.3.2 UBICACIÓN DE PULSE SENSOR	49
2.4 MATERIALES ELECTRÓNICOS ADICIONALES	50
2.4.1 BATERÍAS LIPO	50
2.4.1.1 TIEMPO DE DURACIÓN DE UNA BATERÍA	51
2.4.2 HILO CONDUCTOR	52
2.4.3 MINI INTERRUPTOR	53
2.4.4 CABLE CONDUCTOR	53
2.5 DISPOSITIVOS MÓVILES	54
2.5.1 SMARTPHONES (TELÉFONOS INTELIGENTES)	54

2.5.2 APLICACIONES EN TELÉFONOS INTELIGENTES.....	54
2.6 FUNDAMENTOS DE SISTEMA OPERATIVO ANDROID.....	54
2.6.1 LAS APLICACIONES EN ANDROID	55
2.6.1.2 APP STORE PARA ANDROID	55
2.6.2 CÓDIGO EN ANDROID	56
2.6.3 APP INVENTOR (PROGRAMACIÓN PARA ANDROID)	57
2.6.3.1 PROGRAMACIÓN EN BLOQUES.....	58
2.6.3.2 REQUISITOS DEL SISTEMA	58
2.6.3.3 FUNCIONAMIENTO.....	60
2.6.3.4 DISEÑADOR	60
2.6.3.5 APLICACIÓN PARA SMARTPHONE ANDROID: MIT AI2 COMPANION	61
2.7 FUNDAMENTOS DE ENTRENAMIENTO DEPORTIVO.....	63
2.7.1 PRINCIPIOS DEL ENTRENAMIENTO DEPORTIVO.....	64
2.7.2 PRINCIPIO DE LA CARGA INDIVIDUALIZADA.....	64
2.7.3 PREPARACIÓN FÍSICA.....	64
2.7.3.1 BENEFICIOS CARDIOVASCULARES DEL ENTRENAMIENTO FÍSICO.....	64
2.7.3.2 BENEFICIOS RESPIRATORIOS DEL ENTRENAMIENTO FÍSICO	65
2.7.3.3 BENEFICIOS ÓSEONEUROMUSCULARES DEL ENTRENAMIENTO FÍSICO	65
2.7.4 CAPACIDADES FÍSICAS.....	65
2.7.5 CAPACIDAD FÍSICA: RESISTENCIA AERÓBICA.....	66
2.7.5.1 BENEFICIOS DE LA RESISTENCIA AERÓBICA	67
2.7.5.2 OBJETIVOS DEL ENTRENAMIENTO AERÓBICO.....	67
2.7.6 MÉTODOS DE ENTRENAMIENTO DE LA RESISTENCIA AERÓBICA	68
2.7.6.1 MÉTODOS CONTINUOS.....	68
2.7.6.1.1 MÉTODO CONTINUO UNIFORME	68
2.7.6.1.2 MÉTODO CONTINUO VARIABLE.....	69
2.7.6.2 MÉTODOS FRACCIONADOS	69
2.7.7 CAPACIDAD FÍSICA: RESISTENCIA ANAERÓBICA.....	69
2.7.7.1 OBJETIVOS DEL ENTRENAMIENTO ANAERÓBICO.....	70

2.7.7.2	BENEFICIOS DE LA RESISTENCIA ANAERÓBICA	70
2.7.7.3	MÉTODOS DE ENTRENAMIENTO DE RESISTENCIA ANAERÓBICA.....	70
2.7.7.3.1	LA CARRERA CONTINUA	70
2.7.7.3.2	LA CARRERA CONTINUA	71
2.7.7.3.3	CARRERA CON VARIACIÓN DE VELOCIDAD.....	71
2.7.7.3.4	INTERVAL TRAINING.....	72
2.8	FRECUENCIA CARDÍACA	73
2.8.1	FACTORES QUE AFECTAN A LA FRECUENCIA CARDÍACA.....	73
2.8.2	FACTORES INTERNOS EN LA F.C.....	73
2.8.2.1	LA EDAD	73
2.8.2.1.1	LA GENÉTICA.....	74
2.8.2.1.2	EL GÉNERO	74
2.8.2.1.3	SOMATOTIPO O COMPOSICIÓN CORPORAL	74
2.8.2.1.4	LAS PSICOLÓGICAS	74
2.8.2.1.5	LA POSTURA.....	74
2.8.2.1.6	EL METABOLISMO.....	74
2.8.2.1.7	EL CONTROL MENTAL.....	75
2.8.2.1.8	MEDICAMENTOS	75
2.8.2.2	FACTORES EXTERNOS EN LA F.C.....	75
2.8.2.2.1	HORARIO DEL DÍA.....	75
2.8.2.2.2	LA TEMPERATURA.....	75
2.8.2.2.3	LA ALTURA	75
2.8.2.2.4	LA CONTAMINACIÓN	75
2.8.3	DEPORTES PARA EL CORAZÓN	76
2.8.3.1	LA CARRERA CONTINUA.....	76
2.8.3.2	LA NATACIÓN	77
2.8.3.3	EL CICLISMO.....	77
2.8.4	EL CORAZÓN Y EL EJERCICIO	77
2.8.5	CORAZÓN, LATIDOS Y FUNCIONAMIENTO.....	78

2.8.5.1 SÍSTOLE	78
2.8.5.2 DIÁSTOLE.....	78
2.8.6 PULSACIONES.....	78
2.8.6.1 TOMA DE PULSACIONES	79
2.8.6.2 TIEMPOS PARA CONTAR LAS PULSACIONES	80
2.8.6.3 ERRORES EN LA TOMA DE PULSACIONES.....	80
2.8.7 EJERCICIO FÍSICO Y FRECUENCIA CARDÍACA.....	80
2.8.8 FRECUENCIA CARDÍACA MÁXIMA	81
2.8.8.1 CÁLCULO DE LA FRECUENCIA CARDÍACA MÁXIMA	81
2.8.8.2 OTRAS FORMAS DE CÁLCULO DE LA FRECUENCIA CARDÍACA MÁXIMA	82
2.8.9 FRECUENCIA CARDIACA DE REPOSO.....	82
2.8.10 FRECUENCIA CARDIACA MÁXIMA DE RESERVA.....	83
2.9 ENTRENAMIENTO SEGÚN LA FRECUENCIA CARDÍACA.....	83
2.9.1 ZONAS DE ENTRENAMIENTO SEGÚN LA FRECUENCIA CARDÍACA	84
2.9.2 ZONA 1 - MEJORAMIENTO DE LA SALUD.....	84
2.9.2.1 CARACTERÍSTICAS DE LA ZONA 1	84
2.9.2.1.1 ZONA 2 – QUEMA DE GRASAS	85
2.9.2.2 CARACTERÍSTICAS DE LA ZONA 2	85
2.9.2.2.1 ZONA 3 - AERÓBICA.....	85
2.9.2.3 CARACTERÍSTICAS DE LA ZONA 3	86
2.9.2.3.1 ZONA 4 - ANAERÓBICA	86
2.9.2.4 CARACTERÍSTICAS DE LA ZONA 4	87
2.9.2.4.1 ZONA 5 – LÍNEA ROJA	87
2.9.2.5 CARACTERÍSTICAS DE LA ZONA 5.....	87
2.9.3 ENTRENAMIENTO: MÉTODO SIMPLIFICADO.....	88
2.9.3.1 CÁLCULO DE LAS ZONAS DE ENTRENAMIENTO POR EL MÉTODO SIMPLIFICADO	88
2.9.4 ENTRENAMIENTO: MÉTODO MEDIANTE FÓRMULA DE KARVONEN	89
2.9.4.1 CÁLCULO DE LAS ZONAS DE ENTRENAMIENTO SEGÚN KARVONEN	90

CAPÍTULO III	92
3 DESARROLLO DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO.....	92
3.1 ANÁLISIS DE REQUERIMIENTOS DE ELEMENTOS ELECTRÓNICOS MEDIANTE LA NORMA IEEE 29148.....	92
3.1.1 INTRODUCCIÓN	92
3.1.1.1 PROPÓSITO DEL SISTEMA	92
3.1.1.2 ALCANCE DEL SISTEMA.....	92
3.1.1.3 DESCRIPCIÓN GENERAL DEL SISTEMA	92
3.1.1.3.1 PERSPECTIVA DEL SISTEMA	92
3.1.1.3.2 FUNCIONES DEL PRODUCTO	93
3.1.1.3.3 CARACTERÍSTICAS DEL USUARIO	93
3.1.1.3.4 LIMITACIONES	94
3.1.1.4 DEFINICIONES.....	94
3.1.2 REFERENCIAS.....	95
3.1.3 REQUERIMIENTOS DEL SISTEMA.....	95
3.1.3.1 INTERFACES EXTERNAS	95
3.1.3.2 REQUERIMIENTOS FUNCIONALES.....	96
3.1.3.3 REQUERIMIENTOS DE USABILIDAD	98
3.1.3.4 REQUERIMIENTOS DE RENDIMIENTO	99
3.1.3.5 REQUERIMIENTOS DE BASES DE DATOS	99
3.1.3.6 RESTRICCIONES DE DISEÑO.....	99
3.1.3.7 ATRIBUTOS DEL SISTEMA DE SOFTWARE	99
3.1.4 APÉNDICES.....	100
3.2 HARDWARE DEL SISTEMA DE MONITOREO CARDÍACO	101
3.2.1 ELEMENTOS ELECTRÓNICOS UTILIZADOS	101
3.2.1.1 SENSOR CARDÍACO	101
3.2.1.2 ARDUINO LILYPAD ATMEGA 168.....	102
3.2.1.3 MÓDULO BLUETOOTH HC- 05	104
3.2.1.4 MÓDULO BLUETOOTH DEL SMARTPHONE	106
3.2.1.5 SMARTPHONE (TELÉFONO INTELIGENTE)	106

3.3 DISEÑO DE BLOQUES DEL SISTEMA	107
3.4 DIAGRAMA DE PINES DEL SISTEMA	108
3.4.1 DIAGRAMA DE PINES - ARDUINO LILYPAD.....	108
3.4.2 DIAGRAMA DE PINES - MÓDULO BLUETOOTH HC-05.....	109
3.5 DIAGRAMAS DE FLUJO DEL SISTEMA	109
3.5.1 DIAGRAMA DE FLUJO DE LA APLICACIÓN EN ANDROID.....	117
3.6 CARACTERIZACIÓN DEL SISTEMA	118
3.6.1 DATOS DE COMUNICACIÓN ARDUINO LILYPAD.....	118
3.6.2 DATOS COMUNICACIÓN APLICACIÓN ANDROID.....	119
3.7 FIRMWARE DEL SISTEMA	120
3.7.1 FIRMWARE DE ARDUINO LILYPAD	121
3.7.1.1 VARIABLES	121
3.7.1.1.1 DESCRIPCIÓN DE VARIABLES	122
3.7.1.1.2 DESCRIPCIÓN DEL PROGRAMA PRINCIPAL	124
3.7.1.1.2.1 EDAD.....	124
3.7.1.1.2.2 FRECUENCIA CARDÍACA	125
3.7.1.1.2.3 PULSO NORMAL.....	125
3.7.1.1.2.4 PULSO DE RESERVA.....	126
3.7.1.1.2.5 GÉNERO Y LÍMITES	126
3.7.1.1.2.6 ZONAS DE ENTRENAMIENTO.....	128
3.7.1.3 INTERRUPCIÓN EN EL SISTEMA.....	129
3.7.1.3.1 VARIABLES DE LA INTERRUPCIÓN	130
3.7.1.3.2 DESCRIPCIÓN DE VARIABLES EN LA INTERRUPCIÓN	130
3.7.1.3.3 CONFIGURACIÓN DE VOID INTERRUPTSETUP ().....	131
3.7.1.3.4 CONFIGURACIÓN DEL ISR (TIMER2_COMPA_VECT)	132
3.7.2 FIRMWARE DE ANDROID	136
3.7.2.1 CONFIGURACIÓN EN EL SCREEN 1 “MENÚ PRINCIPAL”	136
3.7.2.1.1 CONFIGURACIÓN DE CONEXIÓN AL MÓDULO BLUETOOTH – HC05.....	137
3.7.2.1.2 CONFIGURACIÓN DE LOS TIMER	138

3.7.2.1.3	INGRESO DE LA EDAD	139
3.7.2.1.4	VISUALIZACIÓN DE LA FRECUENCIA CARDÍACA	140
3.7.2.1.5	INGRESO DEL PULSO NORMAL	141
3.7.2.1.6	ELECCIÓN DEL GÉNERO	142
3.7.2.1.7	CÁLCULO DE LA F.C.M	143
3.7.2.1.8	ELECCIÓN DE UNA ZONA DE ENTRENAMIENTO	143
3.7.2.1.9	BOTONES ZONAS DE ENTRENAMIENTO	145
3.7.2.1.10	OPCIÓN RECONECTAR	146
3.7.2.2	CONFIGURACIÓN DEL “SCREEN 2” AL “SCREEN 6”	146
3.7.2.2.1	BOTÓN CONTINUAR	147
3.7.2.2.2	BOTÓN VOLVER	148
3.7.2.2.3	BOTÓN INICIAR ENTRENAMIENTO	148
3.7.2.2.4	OPCIÓN PARAR	150
3.7.2.2.5	SALIR DE LA APLICACIÓN	151
3.8	PROCESO DE IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO – PROTOTIPO #1	151
3.8.1	MATERIALES UTILIZADOS	152
3.8.2	FIRMWARE DEL PROTOTITPO	152
3.8.2.1	DISEÑO DE LA APLICACIÓN EN APP INVENTOR	152
3.8.2.1.1	INSTALACIÓN DE LA APLICACIÓN	156
3.8.2.1.2	APLICACIÓN EN EL SMARTPHONE	159
3.8.3	CONEXIONES DEL PROTOTIPO #1	160
3.8.4	PRUEBAS Y RESULTADOS DEL PROTOTIPO #1	161
3.9	PROCESO DE IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO – PROTOTIPO #2	163
3.9.1	MATERIALES UTILIZADOS	163
3.9.2	DISEÑO PARA LA IMPLEMENTACIÓN DEL PROTOTIPO #2	164
3.9.3	CARACTERÍSTICAS DE LA MUÑEQUERA	165
3.9.4	COSIDO DE ARDUINO LILYPAD A LA MUÑEQUERA	165
3.9.5	CONEXIÓN DEL MÓDULO BLUETOOTH	167

3.9.6 CONEXIÓN DEL SENSOR	167
3.9.7 ALIMENTACIÓN ELÉCTRICA DEL SISTEMA	168
3.9.8 PRUEBAS Y RESULTADOS DEL PROTOTIPO #2.....	168
3.9.8.1 PRUEBAS DE COMODIDAD DE LA MUÑEQUERA.....	169
3.9.8.2 PRUEBAS DEL SENSOR CARDIACO.....	170
3.9.8.3 PRUEBA DE ENVÍO DE CARACTERES.....	171
3.9.8.4 PRUEBAS DE ALIMENTACIÓN ELÉCTRICA.....	173
3.9.9 ANÁLISIS DE RESULTADOS DEL PROTOTIPO #2	175
3.10 PROCESO DE IMPLEMENTACIÓN DEL SISTEMA CARDIACO – PROTOTIPO #3	176
3.10.1 APLICACIÓN EN EL SMARTPHONE.....	176
3.10.2 CONFIGURACIONES DE ARDUINO LILYPAD	176
3.10.3 MATERIALES UTILIZADOS	176
3.10.4 DISEÑO DE LA MUÑEQUERA EN EL PROTOTIPO #3.....	177
3.10.5 CONEXIONES DE LOS ELEMENTOS.....	180
3.10.5.1 COSIDO Y CONEXIÓN DE LOS ELEMENTOS ELECTRÓNICOS	180
3.10.6 PRUEBAS Y RESULTADOS DEL PROTOTIPO #3.....	181
3.10.6.1 PRUEBAS DE COMODIDAD DE LA MUÑEQUERA.....	181
3.10.7 ANÁLISIS DE RESULTADOS DEL PROTOTIPO #3	182
3.11 PROCESO DE IMPLEMENTACIÓN DEL SISTEMA CARDIACO – DISEÑO FINAL.....	183
3.11.1 MATERIALES UTILIZADOS	183
3.11.2 DISEÑO PARA LA IMPLEMENTACIÓN.....	184
3.11.3 FIRMWARE PARA ARDUINO LILYPAD	185
3.11.4 APLICACIÓN.....	185
3.11.5 DISEÑO DE LA MUÑEQUERA.....	185
3.11.5.1 DIMENSIONES DE LA MUÑEQUERA	187
3.11.6 COSIDO DE ARDUINO LILYPAD A LA MUÑEQUERA	188
3.11.7 CONEXIÓN DEL MÓDULO BLUETOOTH	190
3.11.7.1 CONFIGURACIÓN DEL MÓDULO BLUETOOTH.....	192
3.11.8 CONEXIÓN DEL SENSOR CARDÍACO.....	197
3.11.9 ALIMENTACIÓN ELÉCTRICA DEL SISTEMA	197

3.11.9.1 CONSUMO DE ENERGÍA DEL MÓDULO BLUETOOTH HC-05.....	199
3.11.9.2 CONSUMO DE ENERGÍA DE LA PLACA ELECTRÓNICA LILYPAD ARDUINO.....	200
3.11.9.3 CONSUMO DE ENERGÍA DEL SENSOR DE PULSOS CARDÍACOS.....	200
3.11.9.4 CONSUMO DE ENERGÍA DE LA BATERÍA LIPO	200
3.11.9.5 TIEMPO DE DURACIÓN DE LA BATERÍA LIPO	201
3.11.10 PRUEBAS Y RESULTADOS DEL DISEÑO FINAL	202
3.11.10.1 PRUEBA DE ENVIÓ DE CARACTERES CORRECTOS	202
3.11.10.2 PRUEBAS DE ALIMENTACIÓN ELÉCTRICA.....	204
3.11.10.3 PRUEBAS DE COMODIDAD DE LA MUÑQUERA.....	205
3.11.11 ANÁLISIS DE RESULTADOS DEL DISEÑO FINAL	207
CAPÍTULO IV	208
4 ANÁLISIS ECONÓMICO.....	208
4.1 PRESUPUESTO PARA EL DISEÑO DE LA MUÑQUERA	208
4.2 PRESUPUESTO DEL HARDWARE UTILIZADO	209
4.3 PRESUPUESTO DEL SOFTWARE UTILIZADO	211
4.4 ANÁLISIS COSTO BENEFICIO	211
CAPÍTULO V	213
5 CONCLUSIONES Y RECOMENDACIONES.....	213
5.1 CONCLUSIONES.....	213
5.2 RECOMENDACIONES	214
5.3 BIBLIOGRAFÍA	215
5.4 LINKOGRAFÍA	216
5.5 ANEXOS.....	219
ANEXO 1: CÓDIGO PRINCIPAL DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO ...	219
ANEXO 2: CÓDIGO DE LA INTERRUPCIÓN DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO.....	227
ANEXO 3: HOJA DE DATOS DE ARDUINO LILYPAD	230
ANEXO 4: HOJA DE DATOS DE PULSE SENSOR.....	232
ANEXO 5: HOJA DE DATOS DE MODULO BLUETOOTH HC-05	243
ANEXO 6: MANUAL DE USUARIO PARA EL USO CORRECTO DEL SISTEMA DE MONITOREO CARDÍACO	249

ÍNDICE DE FIGURAS

FIGURA 1: IDE de Arduino	14
FIGURA 2: Lilypad USB.....	19
FIGURA 3: Lilypad Arduino.....	22
FIGURA 4: Lilypad Arduino Simple.....	23
FIGURA 5: Hilo conductor para Arduino Lilypad	28
FIGURA 6: Hilo conductor conectado a Lilypad	28
FIGURA 7: Textiles Inteligentes.....	29
FIGURA 8: Uso de Textil Inteligente.....	29
FIGURA 9: Ropa Inteligente	30
FIGURA 10: Logo de Bluetooth	31
FIGURA 11: Topología Punto a punto	37
FIGURA 12: Topología Piconet	37
FIGURA 13: Topología Scatter-net.....	38
FIGURA 14: Trama de Datos Bluetooth	39
FIGURA 15: Módulo HC-05 conectado a cable serial	43
FIGURA 16: Módulo HC-06 conectado a Tarjeta Arduino.....	44
FIGURA 17: Factores a considerar en la elección de un sensor	46
FIGURA 18: Esquema básico de un Sensor Analógico	47
FIGURA 19: Diagrama básico de un sensor óptico.....	47
FIGURA 20: Pulse Sensor	48
FIGURA 21: Señal de onda del corazón	49
FIGURA 22: Terminales de Pulse Sensor	49
FIGURA 23: Pulse Sensor en el dedo	50
FIGURA 24: Pulse Sensor en la oreja	50
FIGURA 25: Hilo conductor	52
FIGURA 26: Mini interruptor	53
FIGURA 27: Porcentajes de uso de Sistema Operativos para aplicaciones móviles	56
FIGURA 28: Pantalla inicial del software App Inventor	57

FIGURA 29: Diagrama de Funcionamiento de App Inventor	60
FIGURA 30: Pantalla de Diseño de la aplicación en App Inventor.....	61
FIGURA 31: Aplicación MIT y dispositivo móvil.....	62
FIGURA 32: Aplicación MIT y Ordenador.....	62
FIGURA 33: Aplicación MIT y dispositivo móvil mediante USB	63
FIGURA 34: Clasificación de Capacidades Físicas.....	66
FIGURA 35: Toma de pulso en el cuello	79
FIGURA 36: Toma de pulso en la muñeca.....	79
FIGURA 37: Zonas de Entrenamiento	84
FIGURA 38: Diseño General del Sistema Cardíaco.....	101
FIGURA 39: Pulse Sensor	102
FIGURA 40: Placa Arduino Lilypad ATmega 168.....	103
FIGURA 41: Diagrama de pines de ATmega168	103
FIGURA 42: Modulo Bluetooth HC-05.....	104
FIGURA 43: Modulo Samsung S4.....	106
FIGURA 44: Smartphone Samsung S3	106
FIGURA 45: Diseño de bloques del Sistema de monitoreo de ritmo cardíaco	107
FIGURA 46: Diagrama de pines de la placa Arduino Lilypad.....	108
FIGURA 47: Módulo Bluetooth HC-05 (vista frontal).....	109
FIGURA 48: Diagrama de pines del módulo Bluetooth HC-05.....	109
FIGURA 49: Declaración de variables.....	121
FIGURA 50: Declaración de variables tipo int y float.....	122
FIGURA 51: Variables tipo "volatile".....	122
FIGURA 52: Configuración de la edad	124
FIGURA 53: Llamado a la interrupción desde programa principal	125
FIGURA 54: Configuración del pulso Normal	125
FIGURA 55: Configuración Pulso de Reserva.....	126
FIGURA 56: Configuración de género masculino + fórmulas.....	127
FIGURA 57: Configuración de género femenino + fórmulas	127

FIGURA 58: Configuración zonas de entrenamiento	128
FIGURA 59: Configuraciones de límites	129
FIGURA 60: Inicialización de variables en la interrupción.....	130
FIGURA 61: Configuración de Registros	132
FIGURA 62: Configuración del vector ISR	133
FIGURA 63: Configuración de punto bajo y alto de la onda.....	133
FIGURA 64: Configuración para evitar ruido a altas frecuencias	133
FIGURA 65: Configuración de valor real de BPM	134
FIGURA 66: Configuración de RunningTotal.....	135
FIGURA 67: Configuración de variable pulse	135
FIGURA 68: Configuración de "sin latidos"	136
FIGURA 69: Diseño del menú principal.....	137
FIGURA 70: Llamada a Bluetooth client.....	137
FIGURA 71: Configuración para la conexión a la MAC del módulo HC-05	137
FIGURA 72: Configuración Timer1 en aplicación.....	138
FIGURA 73: Configuración de variable global en menú principal (aplicación).....	138
FIGURA 74: Configuración Timer2 en aplicación.....	139
FIGURA 75: Teclado numérico para ingreso de edad.....	139
FIGURA 76: Configuración para habilitar teclado numérico.....	140
FIGURA 77: Configuración de edad en la aplicación	140
FIGURA 78: Visualización de la Frecuencia Cardiaca	141
FIGURA 79: Configuración del pulso normal en la aplicación.....	141
FIGURA 80: Visualización de ingreso del pulso normal.....	142
FIGURA 81: Visualización de elección del género en la aplicación	142
FIGURA 82: Configuración para elección del género.....	143
FIGURA 83: Configuración para cálculo de la F.C.M.	143
FIGURA 84: Visualización para la F.C.M.	143
FIGURA 85: Visualización de Llamada de notificación	144
FIGURA 86: Configuración de "Elegir Entrenamiento"	144

FIGURA 87: Visualización de la notificación	144
FIGURA 88: Visualización de las zonas de entrenamiento	145
FIGURA 89: Configuración de ingreso a zonas de entrenamiento en la aplicación	145
FIGURA 90: Configuración para salir de la aplicación	146
FIGURA 91: Configuración de reconectar	146
FIGURA 92: Visualización de la zona 1 en la aplicación.....	147
FIGURA 93: Configuración botón continuar en la aplicación	147
FIGURA 94: Configuración de variables globales (aplicación).....	148
FIGURA 95: Configuración de botón volver (aplicación).....	148
FIGURA 96: Configuración de botón iniciar entrenamiento en la aplicación	149
FIGURA 97: Configuración de timer en la zona de entrenamiento específica dentro de la aplicación.....	149
FIGURA 98: Visualización de iniciar entrenamiento en la aplicación	149
FIGURA 99: Visualización de los rangos de pulsaciones	150
FIGURA 100: Configuración botón de parar (aplicación)	150
FIGURA 101: Visualización de botón parar en la aplicación	151
FIGURA 102: Configuración para cerrar screen de zona de entrenamiento (aplicación)	151
FIGURA 103: Aplicación Demo para prototipo#1	153
FIGURA 104: Menú principal de la aplicación (Versión Demo).....	154
FIGURA 105: Elección de zonas de entrenamiento en la aplicación (versión Demo)	154
FIGURA 106: Visualización de Zona 1 en la aplicación (Versión Demo).....	155
FIGURA 107: Visualización de Zona 2 en la aplicación (Versión Demo).....	155
FIGURA 108: Instalación de la aplicación mediante código QR	156
FIGURA 109: Compilación de la aplicación.....	156
FIGURA 110: Generación de código QR para instalar la aplicación.....	156
FIGURA 111: Icono de MIT AI2 Companion	157
FIGURA 112: Aplicación MIT AI2 Companion.....	157
FIGURA 113: Enlace para descargar aplicación al Smartphone	158
FIGURA 114: Instalador de paquetes de la aplicación.....	158
FIGURA 115: Instalación de la aplicación en el Smartphone.....	158

FIGURA 116: Pantalla de aplicación instalada en el Smartphone	159
FIGURA 117: Icono de la aplicación Demo en el Smartphone	159
FIGURA 118: Conexiones del prototipo#1	160
FIGURA 119: Verificación de funcionamiento de pulse sensor.....	161
FIGURA 120: Verificación de pulsos en monitor serial de Arduino	161
FIGURA 121: Icono de Blue Term+	162
FIGURA 122: Visualización de pulsos a través de Blue Term+	162
FIGURA 123: Visualización de pulsos a través de la aplicación	162
FIGURA 124: Diseño para implementación del prototipo #2.....	164
FIGURA 125: Banda de brazo para celular	164
FIGURA 126: Muñequera "Nike Premier".....	165
FIGURA 127: Primeros pasos cosido de Arduino Lilypad a la muñequera (Prototipo#2)	166
FIGURA 128: Arduino Lilypad cosida totalmente a la muñequera (Prototipo#2)	166
FIGURA 129: Diagrama de conexión (prototipo#2).....	166
FIGURA 130: Conexión del módulo Bluetooth en la muñequera (prototipo#2)	167
FIGURA 131: Conexión del sensor cardíaco (prototipo#2)	167
FIGURA 132: Conexión a batería lipo (prototipo#2).....	168
FIGURA 133: Prototipo#2 (Muñequera)	168
FIGURA 134: Smartphone con la aplicación (prototipo#2)	169
FIGURA 135: Vista frontal del sistema (prototipo#2).....	169
FIGURA 136: Ubicación de la muñequera y el sensor (prototipo#2)	170
FIGURA 137: Puesta de cinta aislante en partes del hilo conductor.....	170
FIGURA 138: Verificación de pulsos cardíacos en prototipo#2	171
FIGURA 139: Verificación de ingreso de edad.....	171
FIGURA 140: Verificación de elección de zona de entrenamiento 1	172
FIGURA 141: Verificación de elección de zona de entrenamiento 2	172
FIGURA 142: Envío de carácter de límite inferior en la zona 1.....	173
FIGURA 143: Envío de carácter de límite superior en la zona 1.....	173
FIGURA 144: Pruebas de Alimentación eléctrica mediante batería lipo	174

FIGURA 145: Fuente utilizada para alimentar eléctricamente al prototipo#2	174
FIGURA 146: Pruebas de alimentación eléctrica mediante fuente	175
FIGURA 147: Diseño de muñequera para prototipo#3.....	177
FIGURA 148: Cintas de velcro en la muñequera del prototipo #3	178
FIGURA 149: Primer doblaje de la muñequera en el prototipo #3.....	178
FIGURA 150: Segundo Doblaje de la muñequera en el prototipo #3.....	179
FIGURA 151: Dimensiones de la muñequera completa (prototipo#3)	179
FIGURA 152: Dimensiones de la muñequera con 1 doblaje (prototipo#3)	179
FIGURA 153: Dimensiones de la muñequera y el antebrazo (prototipo#3)	180
FIGURA 154: Cosido y conexión de elementos en la muñequera (prototipo#3)	180
FIGURA 155: Conexión de los elementos (prototipo#3)	181
FIGURA 156: Pruebas de comodidad de muñequera (prototipo#3)	181
FIGURA 157: Vista lateral de muñequera (prototipo#3).....	182
FIGURA 158: Esquema previo a la implementación del sistema.....	184
FIGURA 159: Icono de la aplicación (diseño final).....	185
FIGURA 160: Muñequera (Diseño final).....	186
FIGURA 161: Zonas de velcro en la muñequera (Diseño final)	186
FIGURA 162: Ubicación de la muñequera (diseño final).....	187
FIGURA 163: Muñequera junto con sensor cardíaco (Diseño final)	187
FIGURA 164: Tiras para sostener elementos en la muñequera	187
FIGURA 165: Dimensiones de la muñequera (diseño final).....	188
FIGURA 166: Cosido de la placa Arduino LilyPad (diseño final)	188
FIGURA 167: Intersección de cables con hilo conductor (diseño final)	189
FIGURA 168: Caminos de cosido del hilo conductor	189
FIGURA 169: Protección al hilo conductor	190
FIGURA 170: Conexión del Módulo Bluetooth (Diseño Final).....	190
FIGURA 171: Modulo Bluetooth más extensiones de cable para conexión.....	191
FIGURA 172: Conexión de pines Tx y Rx del módulo Bluetooth	191
FIGURA 173: Conexión del USB-TTL al módulo Bluetooth	192

FIGURA 174: Puerto de conexión para el módulo Bluetooth	193
FIGURA 175: Descripción de la conexión del módulo Bluetooth	193
FIGURA 176: Elección del puerto de comunicación para configurar el módulo Bluetooth	194
FIGURA 177: Propiedades de comunicación del módulo Bluetooth	194
FIGURA 178: Propiedades de configuración ASCII	195
FIGURA 179: Configuración del nombre para el módulo Bluetooth HC-05	195
FIGURA 180: Configuración para verificación de velocidad y Contraseña del módulo Bluetooth HC-05	196
FIGURA 181: Conexión del Sensor Cardíaco	197
FIGURA 182: Recubrimiento para el cable del sensor	197
FIGURA 183: Batería Lipo para sistema de monitoreo	198
FIGURA 184: Medición de voltaje de batería lipo por medio de multímetro	198
FIGURA 185: Alimentación de entrada para la placa Arduino Lilypad	199
FIGURA 186: Voltaje out de placa Lilypad	199
FIGURA 187: Conexión serial para verificación de datos con el IDE de Arduino	203
FIGURA 188: Verificación de edad a través de monitor serial de Arduino (diseño final)	203
FIGURA 189: Verificación de zonas a través de monitor serial de Arduino (diseño final)	203
FIGURA 190: Conexión eléctrica con una batería en el diseño final	204
FIGURA 191: Pruebas de switch en protoboard (diseño final)	205
FIGURA 192: Sistema con dos baterías y switch incorporado	205
FIGURA 193: Ubicación de los switch en la muñequera	205
FIGURA 194: Vista aérea de la muñequera (diseño final)	206
FIGURA 195: Vista lateral de la muñequera (diseño final)	206
FIGURA 196: Muñequera más banda de brazo	206

ÍNDICE DE TABLAS

TABLA 1: Comparación entre placas Arduino (Parte 1)	26
TABLA 2: Comparación entre placas Arduino (Parte 2)	27
TABLA 3: Representación del ancho de banda y sus versiones	35
TABLA 4: Clases y Potencias de Bluetooth	36
TABLA 5: Obstrucciones absorbentes y reflectantes de radiofrecuencia.....	42
TABLA 6: Comparación entre tecnologías.....	45
TABLA 7: Requisitos App Inventor para ordenadores	58
TABLA 8: Compatibilidad App Inventor para Navegadores.....	58
TABLA 9: Compatibilidad App Inventor para Smartphones y tablets	59
TABLA 10: Métodos de entrenamiento de carrera continua.....	71
TABLA 11: Interfaces Externas.....	96
TABLA 12: Requerimientos Funcionales (Visualización de Pulsaciones)	96
TABLA 13: Requerimientos Funcionales (Visualización de FCM).....	97
TABLA 14: Requerimientos Funcionales (Elección de la Z.E.).....	97
TABLA 15: Requerimientos Funcionales (Monitoreo Z.E.)	98
TABLA 16: Características Pulse Sensor	102
TABLA 17: Características ATmega168	103
TABLA 18: Distribución de pines de Módulo Bluetooth HC-05.....	105
TABLA 19: Características adicionales de Módulo Bluetooth HC-05	105
TABLA 20: Caracteres Arduino Lilypad	119
TABLA 21: Caracteres de App Android	120
TABLA 22: Descripción de Variables	123
TABLA 23: Variables de la interrupción	130
TABLA 24: Lista de Timers	131
TABLA 25: Materiales utilizados en prototipo#1.	152
TABLA 26: Materiales Utilizados.....	163
TABLA 27: Materiales utilizados en el Prototipo#3.....	177
TABLA 28: Materiales utilizados en el diseño final del sistema.....	183

TABLA 29: Funciones de cableado del diseño final.....	184
TABLA 30: Distribución de pines de Modulo Bluetooth HC-05 y TTL.....	192
TABLA 31: Comandos AT para configurar módulo Bluetooth HC-05.....	196
TABLA 32: Consumo de energía del módulo Bluetooth	199
TABLA 33: Consumo de corriente límite en pines	200
TABLA 34: Consumo de energía de batería Lipo	201
TABLA 35: Presupuesto del sistema de monitoreo cardíaco	208
TABLA 36: Presupuesto de componentes electrónicos implementados	209
TABLA 37: Costos del diseño final.....	210
TABLA 38: Presupuesto del dispositivo móvil inteligente	210
TABLA 39: Presupuesto del software	211
TABLA 40: Costos del proyecto	212

RESUMEN

El presente trabajo consiste en el desarrollo de un sistema de monitoreo de ritmo cardíaco para las personas que realizan ejercicio físico aeróbico. Este proyecto beneficiará a la población que efectúa ejercicio físico, ya que mediante este sistema, ahora se podrá realizar dicha actividad de una manera controlada y lo más importante es que será muy fácil de interpretar para cualquier persona, tenga o no experiencia con el manejo de instrumentos para el control de su frecuencia cardiaca.

El primer capítulo contiene los antecedentes del proyecto, aporte esencial para el desarrollo inicial del mismo, aquí se detallan el problema, su justificación, objetivos entre otros.

El segundo capítulo contiene los fundamentos teóricos útiles para el desarrollo del proyecto, en este se detallan los aspectos más importantes del sistema electrónico como la placa Arduino, el modulo Bluetooth y los demás componentes necesarios para la implementación. Además se explica todo lo referente al entrenamiento deportivo, específicamente la capacidad física: resistencia aeróbica.

Posteriormente en el tercer capítulo se describen los requerimientos de los sistemas mediante la norma IEEE 29148, el diseño del sistema y adicionalmente se describe el desarrollo del sistema junto con su configuración, tanto en la parte del sistema electrónico, como en la aplicación móvil. Al final de este capítulo se detalla también el proceso de implantación que sufrió el sistema de monitoreo de ritmo cardiaco y las respectivas pruebas necesarias para lograr un óptimo funcionamiento del sistema.

En el cuarto capítulo se realiza un análisis económico referencial que abarca todos los componentes de software y hardware empleados. Finalmente en el quinto capítulo se encuentran las conclusiones y recomendaciones obtenidas durante el desarrollo del presente trabajo.

ABSTRACT

This work involves the development of a system for monitoring heart rate for people who do aerobic exercise. This project will benefit the population that performed physical exercise, because by this system, now can perform this activity in a controlled manner and most importantly it will be very easy to play for anyone with or without experience handling instruments for monitoring your heart rate.

The first chapter contains the background of the project, essential for the initial development of the contribution, the problem is listed here, its justification, goals and more. The second chapter contains the useful theoretical foundations for the project, in this the most important aspects of the electronic system as Arduino, the Bluetooth module and other components necessary for implementation are detailed. Besides everything related to sports training is explained, specifically physical ability endurance.

Later in the third chapter the system requirements are described by the IEEE 29148, the standard system design and further development of the system is described together with its configuration, both part of the electronic system, as in the mobile application. At the end of this chapter the process of implementing the system suffered heart rate monitoring and respective tests necessary for optimal system performance is also detailed.

In the fourth chapter a referential economic analysis covering all components of software and hardware used is made. Finally in the fifth chapter are the conclusions and recommendations obtained during the development of this work.

PRESENTACIÓN

En la actualidad el ejercicio físico adecuado es un factor fundamental para mantener, mejorar y conservar la salud, ya que ayuda a prevenir múltiples enfermedades. El ejercicio es una necesidad básica. El cuerpo humano está hecho para ser usado, y con la falta de ejercicio se renuncia a la oportunidad de mejorar o conservar la salud. Cuando el cuerpo no es ejercitado se vuelve flácido, lento y, en general, ineficaz. Sin embargo, al realizar ejercicio físico, la mayoría de personas piensa que mientras mayor sea el volumen e intensidad de trabajo, mejor serán sus resultados; siendo este concepto incorrecto.

Por tal motivo, se considera que para la realización de ejercicio físico aeróbico en cualquier deporte, se lo debe realizar de manera controlada y planificada. Las personas que caminan, corren, trotan en los parques, por la ciudad o en sus caminadoras, lo hacen siempre queriendo obtener un beneficio para su salud; pero nunca pensando si lo están haciendo bien o mal.

Para controlar dicho problema, se utiliza instrumentos de verificación de nuestra frecuencia cardiaca como es el pulsómetro, no obstante casi nadie que hace ejercicio, a excepto de deportistas de alto rendimiento, lo utilizan. Las pocas personas que compran un pulsómetro, lo hacen con el afán de ver solo sus pulsaciones, pero eso no es suficiente a la hora de controlar una correcta práctica de ejercicio aeróbico.

El sistema de monitoreo de control cardiaco para personas que realizan ejercicio físico aeróbico, pretende dar al usuario deportista o no, la facilidad de obtener un correcto entrenamiento aeróbico y anaeróbico; con datos fiables, procedentes de estudios previos realizados por investigadores en la preparación física. Para esto el sistema presentará rangos de pulsaciones fiables, en cada zona de intensidad de entrenamiento, y haciendo fácil de usar para cualquier personas con la experiencia o sin ella de haber usado un pulsómetro convencional

CAPÍTULO I

1 ANTECEDENTES

1.1 PLANTEAMIENTO DEL PROBLEMA

La actividad física es un conjunto de movimientos corporales en un determinado período, donde interviene el trabajo de varios músculos del ser humano. La mayoría de personas que entrenan o realizan actividad física como trotar, caminar, ir al gimnasio o practicar algún deporte de manera recreativa, cree que cuanto mayor es el volumen de trabajo físico, mayor será sus beneficios; siendo este pensamiento incorrecto. El cuerpo humano tiene un límite que no se debe sobrepasar (sobreentrenamiento), si esto sucede el cuerpo responderá de forma contraria a la que se busca con el entrenamiento o práctica física. Por tal motivo, lo más recomendado por entrenadores y profesionales de la salud, es realizar esta actividad de manera programada y personalizada, acorde a los requerimientos de cada persona.

Cuando se realiza ejercicios cardiovasculares, se pueden ejecutar en 5 zonas de intensidad establecidas, que para su aplicación, dependerá de las necesidades acordes al entrenamiento o actividad física de cada individuo. Si la persona que realiza actividad física, es sometida a un excesivo esfuerzo físico (80-90-100% intensidad), es decir, ejecuta un esfuerzo fuera de la zona apropiada para él; se presentarán varios síntomas que pueden ser de carácter psicológicos como la fatiga, pérdida de apetito, cansancio excesivo, entre otros. También se encontrarán los de mayor peligro como los fisiológicos, especialmente en el corazón; el cual a la hora de realizar estos esfuerzos excesivos tiende a aumentar su ritmo cardíaco de una manera muy acelerada, y la mayoría de veces; la persona ni siquiera sabe lo que está pasando internamente.

Realizar ejercicios físicos aeróbicos, es sano; pero en la moderación también está la clave para sacar el máximo provecho de sus beneficios. En otras palabras, abusar del ejercicio físico puede ser contraproducente y perjudicial para la salud, y especialmente en las personas que pasan del sedentarismo al ejercicio físico rápidamente; causando desde lesiones musculares hasta problemas cardíacos; como por ejemplo, la muerte súbita cardíaca en los ejercicios de alta intensidad.

La principal causa es por el desconocimiento del entrenamiento de ejercicios físicos basados en la frecuencia cardíaca, establecidas en las zonas de intensidad del entrenamiento. Por lo tanto se debe de tener un control a la hora de querer adquirir una condición física adecuada, esto es para mejorar el entrenamiento, prevenir lesiones y problemas cardíacos en las personas.

Por otra parte, existen varios elementos y dispositivos electrónicos que pueden cumplir diversas funciones y utilidades en el mundo tecnológico. En la actualidad, estos son mal utilizados en el mundo técnico-científico, ya que la mayoría de dispositivos, por mencionar, el Smartphone; son utilizados como principal función para el entretenimiento y pasatiempo; y no explotando todas sus características técnicas, para soluciones específicas de problemas en la vida diaria.

1.2 JUSTIFICACIÓN

El realizar ejercicio físico de manera planificada y controlada, es la clave para obtener beneficios como fortalecimiento del corazón, activar el cerebro, mantener fuertes los huesos y músculos. Además, también es uno de los mejores aliados junto a una dieta adecuada para perder peso, reducir la grasa abdominal y prevenir patologías como la diabetes o la hipertensión, entre otras.

Las personas que realizan ejercicio físico deben conocer las características de su entrenamiento, como el rango de pulsaciones del corazón y la intensidad del ejercicio, al que cada persona deberá someterse; es decir, que antes de realizar ejercicio físico, se debe saber qué zona de entrenamiento se desea entrenar.

Un individuo, por ejemplo, que este desentrenado o no practique ejercicio físico, no deberá realizar las mismas actividades que un deportista de alto rendimiento o que una persona con huella deportiva; ya que al realizar estas acciones de entrenamiento indebidas, rompen principios del entrenamiento, que podría causar problemas cardíacos, como la muerte súbita, infarto, sobreentrenamiento, entre otros. En otras palabras cuando se somete al cuerpo a un esfuerzo excesivo varios beneficios se pierden.

Este proyecto contribuirá con el Plan Nacional del Buen Vivir 2013 - 2017, ya que el mismo contempla en el objetivo 3 la mejora de calidad de vida de la población. La finalidad de este objetivo es mejorar la calidad y calidez de los servicios sociales de atención, garantizar la salud de la población desde la generación de un ambiente y prácticas saludables.

Además en la política 3.7 en el literal g), manifiesta que impulsar la organización, el asociativismo o la agrupación en materia deportiva o cualquier actividad física permanente o eventual, de acuerdo a las necesidades, aptitudes y destrezas de sus integrantes.

Así también, la actividad física y el deporte son elementos dinamizadores de las capacidades y potencialidades de las personas. En el campo de la salud pública, la práctica deportiva es funcional a la expectativa de un estilo de vida activo y relativamente prolongado, en tanto produce beneficios fisiológicos y psicológicos.

Hoy se cuentan con diferentes herramientas para poder solucionar problemas como este. El uso de dispositivos tecnológicos en la vida cotidiana es una realidad, permitiéndonos ahora realizar la prevención y control de frecuencia cardiaca en las personas de una manera sencilla y fácil a la hora de interpretar los datos. Esta información solo era posible monitorearla en laboratorios de centros de entrenamiento de alto rendimiento mediante un ordenador convencional con varios electrodos conectados a la persona ubicada en una banda.

Para solucionar problemas como el mencionado anteriormente, se puede aprovechar el gran avance que ha tenido el software y el hardware libre en los últimos años, como ventajas de tratamiento de datos y procesamiento de información, dirigido a las personas que deseen realizar ejercicio físico de una forma planificada y controlada

. El uso de algunas placas con microcontroladores, permiten tener un sistema de hardware y software completo, con varias posibilidades de aplicaciones que pueden adaptarse a diferentes necesidades en entornos diversos. Además la elección de herramientas libres permite obtener soluciones funcionales, escalables, adaptables, flexibles y sobre todo económicas; ya que no es necesario adquirir licencias para su uso o modificación, como es el caso de las herramientas propietarias.

El desarrollo del software y hardware libre en la actualidad, hace que se tengan herramientas de desarrollo avanzadas, con las cuales se pueden solucionar problemas de este tipo. Placas con circuitos en forma de parche, con características de un ordenador y con sensores acoplados, los cuales tienen la ventaja de ser accesibles por sus precios económicos. Además en combinación con tecnologías inalámbricas hacen que se puedan desarrollar soluciones específicas a problemas puntuales. Si a todo esto

se suman los beneficios de la Internet y los dispositivos móviles inteligentes; se puede concebir un sistema de monitoreo y prevención cardíaco dirigido a personas que realizan actividad física aeróbica.

Por último, uno de los aspectos más importantes de la misión de la “Universidad Técnica Del Norte” es la vinculación con la sociedad en distintos aspectos, por lo cual esta institución tiene la obligación de dar soluciones a problemas como el mencionado.

1.3 ALCANCE

Se realizará un estudio teórico de las características del ejercicio físico aeróbico, para obtener información detallada sobre el entrenamiento según la frecuencia cardíaca con sus respectivas zonas e intensidades de trabajo.

Se realizará un análisis de las características técnicas y físicas de los elementos que se utilizarán para el sistema electrónico, el mismo que constará principalmente de una placa con microcontrolador diseñado para textiles inteligentes, que tendrá como características especiales, la impermeabilidad, facilidad y flexibilidad, para las personas que realicen ejercicio físico de desplazarse sin ninguna restricción, será ubicado en la camiseta o licra del individuo.

Dentro del sistema, también estará conectado junto a la placa especificada, un sensor de signos vitales, que servirá para captar los pulsos de frecuencia cardíaca de la persona. Además el sistema constará de una alimentación (Lipo) de corriente directa, con características específicas para conectar de manera externa a la placa.

Para este proyecto se realizará una aplicación para Sistema Operativo Android compatible con dispositivos móviles. Posteriormente se implementará en el dispositivo móvil del usuario. Mediante la aplicación, se podrá observar la información de frecuencia cardíaca máxima (F.C.M.), a partir del ingreso de la edad y otros factores.

Esta información de la FCM, servirá como punto de partida para elegir una de las cinco zonas de entrenamiento según la frecuencia cardíaca. Además, inicialmente, el usuario indicará su estado físico actual y si ha tenido huella deportiva; esto será para que la aplicación verifique los niveles de entrenamiento hasta donde el individuo pueda realizar. La aplicación constará de una alarma que indique si la persona se encuentra en el rango de pulsaciones adecuado y si se pasan de los mismos.

Este proyecto realizará la función específica de monitorear las pulsaciones de la frecuencia cardíaca de la persona, solo cuando esta esté efectuando actividad y ejercicio físico. También, se menciona, que se utilizará tecnología inalámbrica Bluetooth, la cual servirá para la comunicación entre el Sistema Electrónico y la Aplicación.

Como parte final del proyecto, se realizará las pruebas del sistema de monitoreo cardíaco en las personas. Esta actividad se realizará como evaluación del funcionamiento del sistema, y para una corrección en posibles fallos.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

- Diseñar un sistema electrónico de monitoreo de ritmo cardíaco para personas que realizan ejercicio físico aeróbico utilizando software y hardware libre.

1.4.2 OBJETIVOS ESPECÍFICOS

- Elaborar un estudio teórico del ejercicio físico aeróbico, para obtener información sobre el entrenamiento según la frecuencia cardíaca.
- Analizar los elementos electrónicos necesarios mediante la norma IEEE 29148, para el diseño del sistema de monitoreo cardíaco y su configuración tanto en hardware y software para uso conjunto con textiles inteligentes.
- Diseñar e implementar la aplicación en Sistema Operativo Android, que permita el monitoreo del ritmo cardíaco en el sistema electrónico.
- Realizar pruebas del sistema que garantice un correcto funcionamiento y corregir posibles fallos.
- Realizar un análisis de costo beneficio que puede ofrecer el implementar un sistema de monitoreo de ritmo cardíaco en las personas que realicen actividad física aeróbica.

CAPÍTULO II

2 FUNDAMENTOS TEÓRICOS

2.1 FUNDAMENTOS DE TECNOLOGÍA ARDUINO

2.1.1 TECNOLOGÍA ARDUINO

Con la existencia de la tecnología Arduino, se ha facilitado el realizar proyectos en donde se unan los microcontroladores y sensores electrónicos de manera más fácil.

Arduino (2010)

Arduino es una plataforma electrónica de código abierto basado en el hardware y el software fácil de usar. Está dirigido a cualquier persona que hace proyectos interactivos. Arduino detecta el medio ambiente mediante la recepción de las aportaciones de muchos sensores, y afecta a su entorno por las luces de control, motores y otros actuadores.

2.1.2 CARACTERÍSTICAS DE ARDUINO

Las placas de Arduino cuentan con varias características, cada una de estas puede ser utilizada en escenarios distintos lo que facilita a la resolución de problemas específicos.

Torrente (2013)

Existen muchas otras placas de diferentes fabricantes que, aunque incorporan diferentes modelos de microcontroladores, son comparables y ofrecen una funcionalidad más o menos similar a las de las placas Arduino. Todas ellas también vienen acompañados de un entorno de desarrollo agradable y cómodo y de un lenguaje de programación sencillo y completo. (p. 69)

Arduino cumple con varias características, que lo hacen importante en varias aplicaciones de la electrónica, entre ellas están:

Torrente (2013) manifiesta:

Arduino es libre y extensible: Esto quiere decir que cualquiera que desee ampliar y mejorar tanto el diseño hardware como el entorno de desarrollo software y el propio lenguaje de programación, puede hacerlo sin problemas.

Esto permite que exista un rico ecosistema de extensiones, tanto de variantes de placas no oficiales como de librerías software de terceros, que puedan adaptarse a nuestras necesidades concretas.

Arduino tiene una gran comunidad: Muchas personas lo utilizan, enriquecen la documentación y comparten continuamente sus ideas. Su entorno de programación en multiplataforma: Se puede instalar y ejecutar en sistemas Windows, Mac OS X y Linux. Esto no ocurre con el software de muchas otras placas.

Su entorno de lenguaje de programación son simples y claros: Son muy fáciles de aprender y utilizar, a la vez que flexibles y completos para que los usuarios avanzados puedan aprovechar y exprimir todas las posibilidades del hardware. Además, están bien documentados, con ejemplos detallados y gran cantidad de proyectos publicados en diferentes formas.

Las placas Arduino son baratas: la placa Arduino estándar (llamada Arduino Uno) ya preensablada y lista para funcionar cuesta alrededor de 20 euros.

Las placas Arduino son reutilizables y versátiles: Reutilizables porque se puede aprovechar la misma placa para varios proyectos (ya que es muy fácil de desconectarla, reconectarla y programarla), y versátiles porque las placas Arduino proveen varios tipos diferentes de entradas y salidas de datos, los cuales permiten capturar información de sensores y enviar señales a actuadores de múltiples formas. (p. 71)

2.1.3 ALIMENTACIÓN

Torrente (2013) manifiesta:

El voltaje de funcionamiento de la placa Arduino (incluyendo el microcontrolador y el resto de componentes) es de 5 V. Podemos obtener esta alimentación eléctrica de varias formas:

Conectando la placa Arduino a una fuente externa, tal como un adaptador AC/DC o una pila. Para el primer caso, la placa dispone de un zócalo donde se puede enchufar una clavija de 2,1 milímetros de tipo "jack". Para el segundo, los cables salientes de los bornes de la pila se pueden conectar a los pines-hembra marcados como "Vin" y "Gnd" (positivo y negativo respectivamente) en la zona de la placa marcada con la etiqueta "POWER".

En ambos casos, la placa está preparada en teoría para recibir una alimentación de 6 a 20 voltios, aunque realmente, el rango de entrada recomendado de voltaje (teniendo en cuenta el deseo de obtener una estabilidad y seguridad eléctrica de nuestros circuitos) es menor de 7 a 12 voltios.

En cualquier caso, este voltaje ofrecido por la fuente externa siempre es rebajado a los 5 V. de trabajo mediante un circuito regulador de tensión que ya viene incorporado dentro de la placa.

Conectando la placa Arduino a nuestro computador mediante una cable USB. Para ello la placa dispone de un conector USB hembra de tipo B. La alimentación recibida de esta manera está regulada permanentemente a los 5V de trabajo y ofrece un máximo de hasta 500mA de corriente (por lo tanto, la potencia consumida por la placa es en este caso de unos 2,5W).

Si en algún momento por el conector USB pasa más intensidad de la deseable, la placa Arduino está protegida mediante un polifusible reseteable que automáticamente rompe la conexión hasta que las condiciones eléctricas vuelvan a la normalidad.

Una consecuencia de esta protección contra posibles picos de corriente es que la intensidad de corriente recibida a través de USB puede no ser suficiente para proyectos que contengan componentes tales como motores, solenoides o motrices de LED, los cuales consumen mucha potencia.

Sea cual sea la manera elegida para alimentar la placa, es lo suficientemente inteligente para seleccionar automáticamente en cada momento la fuente eléctrica disponible y utilizar una u otra sin que tengamos que hacer nada especial al respecto.

Si utilizamos una pila como alimentación externa, una ideal sería la de 9V (está dentro del rango recomendado de 7 a 12 voltios), y si se utilizan un adaptador AC/DC, se recomienda el uso de las siguientes características

El voltaje de salida ofrecido ha de ser de 9 a 12 V DC. En realidad, el circuito regulador que lleva incorporado la placa Arduino es capaz de manejar voltajes de salida de 20V DC. No obstante, esta no es una buena idea porque se pierde la mayoría del voltaje en forma de calor (lo cual es terriblemente ineficiente) y además puede provocar el sobrecalentamiento del regulador y, como consecuencia dañar la placa.

La intensidad de corriente ofrecida ha de ser de 250 mA (o más). Si conectamos a nuestra placa Arduino muchos componentes o unos pocos pero consumidores de mucha energía (como por ejemplo una matriz de LEDs, una tarjeta SD o un motor) el adaptador debería suministrar al menos 500 mA o incluso 1 A. De esta manera nos aseguraremos de que tenemos suficiente corriente para que cada componente pueda funcionar de forma fiable. (p. 85) Por otro lado, dentro de la zona etiquetada como “POWER” en la placa Arduino existe una serie de pines-hembra relacionados con la alimentación eléctrica, como son:

2.1.3.1 GND

“Pines-hembra conectados a tierra. Es muy importante que todos los componentes de nuestros circuitos compartan una tierra común como referencia. Estos pines-hembra se ofrecen para realizar esta función”. (Torrente, 2013, p.87)

2.1.3.2 VIN

Torrente (2013)

Este pin-hembra se puede utilizar para dos cosas diferentes: si la placa está conectada mediante la clavija de 2,1mm a alguna fuente externa que aporte un voltaje dentro de los márgenes de seguridad, podemos conectar a este pin-hembra cualquier componente electrónico para alimentarlo directamente con el nivel de voltaje que esté aportando la fuente en ese momento.

Si la placa está alimentada mediante USB, entonces ese pin-hembra aportará 5 V regulados. En cualquier caso, la intensidad de corriente máxima aportada es de 40 mA (esto hay que tenerlo en cuenta cuando conectemos dispositivos que consuman mucha corriente, como por ejemplo motores).

También podemos usar el pin-hembra “Vin” para otra cosa: para alimentar la propia placa directamente desde alguna fuente de alimentación externa sin utilizar ni la clavija ni el cable USB. Esto se hace conectándole el borne positivo de la fuente (por ejemplo, una pila de 9 V) y conectando el borne negativo al pin de tierra. Si se usa este montaje, el regulador de tensión que incorpora la placa reducirá el voltaje recibido de la pila al voltaje de trabajo de la placa (los 5 V). (p.87)

2.1.3.3 PIN 5 V

Torrente (2013)

Este pin-hembra se puede utilizar para dos cosas diferentes: tanto si la placa está alimentada mediante el cable USB como si está alimentada por una fuente externa que aporte un voltaje dentro de los márgenes de seguridad, podemos conectar a este pin-hembra cualquier componente para que pueda recibir 5 V regulados. En cualquier caso, la intensidad de corriente máxima generada será de 40 mA. Pero también podemos usar este pin hembra para otra cosa: para alimentar la propia placa desde una fuente de alimentación externa previamente regulada a 5 V sin utilizar el cable USB ni la clavija de 2,1mm. (p.87)

2.1.3.4 PIN 3,3 V

Torrente (2013)

Este pin-hembra ofrece un voltaje de 3,3 voltios. Este voltaje se obtiene a partir del recibido indistintamente a través del cable USB o de la clavija de 2,1 mm, y está regulado (con un margen de error del 1%) por un circuito específico incorporado en la placa: el LP2985.

En este caso particular, la corriente máxima generada es de 50 mA. Al igual que con los pines anteriores, podremos usar este pin para alimentar componentes de nuestros. (p.87)

2.1.4 ENTRADAS Y SALIDAS DIGITALES

Torrente (2013) manifiesta:

La placa Arduino UNO dispone de 14 pines-hembra de entradas o salidas (según lo que convenga) digitales, numeradas desde la 0 hasta la 13. Es aquí donde conectaremos nuestros sensores para que la placa pueda recibir datos del entorno, y también donde conectaremos los actuadores para que la placa pueda enviarles las órdenes pertinentes, además de poder conectar cualquier otro componente que necesite comunicarse con la placa de alguna manera. A veces a estos pines-hembra digitales de “propósito general” se les llama pines GPIO (de “General Purpose Input/Output”).

Todos estos pines-hembra digitales funcionan a 5 V, pueden proveer o recibir un máximo de 40 mA y disponen de una resistencia “pull-up” interna de entre 20 KΩ y 50 KΩ que inicialmente está desconectada (salvo que nosotros indiquemos lo contrario mediante programación software).

Hay que tener en cuenta, no obstante, que aunque cada pin individual pueda proporcionar hasta 40 mA como máximo, en realidad, internamente la placa agrupa los pines digitales de tal forma que tan solo pueden aportar 100 mA a la vez el conjunto de los pines nº 0, 1, 2, 3 y 4, y 100 mA más el resto de pines (del 5 al 13). Esto quiere decir que como mucho podríamos tener 10 pines ofreciendo 20 mA a la vez. (p. 90)

2.1.5 ENTRADAS ANALÓGICAS

Torrente (2013) manifiesta:

La placa Arduino dispone de 6 entradas analógicas (en forma de pines hembra etiquetados como “A0”, “A1”... hasta “A5”) que pueden recibir voltajes dentro de un rango de valores continuos de entre 0 y 5 V. No obstante, la electrónica de la placa tan solo puede trabajar con valores digitales, por lo que es necesaria una conversión previa del valor analógico recibido a un valor digital lo más aproximado posible. Esta se realiza mediante un circuito conversor analógico/digital incorporado en la propia placa.

El circuito conversor es de 6 canales (uno por cada entrada) y cada canal dispone de 10 bits (los llamados “bits de resolución”) para guardar el valor del voltaje convertido digitalmente. Por último, decir que estos pines-hembra de entrada analógica tienen también toda la funcionalidad de los pines de entrada-salida digitales.

Es decir, que si en algún momento necesitamos más pines-hembra digitales más allá de los 14 que la placa Arduino ofrece (del 0 al 13), los 6 pines-hembra analógicos pueden ser usados como unos pines-hembra digitales más (numerándose entonces del 14 al 19) sin ninguna distinción. (p.91)

2.1.6 SALIDAS ANALÓGICAS (PWM)

Torrente (2013)

En nuestros proyectos a menudo necesitaremos enviar al entorno señales analógicas, por ejemplo, para variar progresivamente la velocidad de un motor, la frecuencia de un sonido emitido por un zumbador o la intensidad con la que luce

un LED. No basta con simples señales digitales: tenemos que generar señales que varíen continuamente. La placa Arduino no dispone de pines-hembra de salida analógica propiamente dichos (porque su sistema electrónico interno no es capaz de manejar este tipo de señales), sino que utiliza algunos pines-hembra de salida digitales concretos para “simular” un comportamiento analógico. Los pines-hembra digitales que son capaces trabajar en este modo no son todos: solo son los marcados con la etiqueta “PWM¹”. En concreto para el modelo Arduino UNO son los pines número: 3, 5, 6, 9, 10 y 11. (p. 91)

2.1.7 PINES TRANSMISIÓN Y RECEPCIÓN

Torrente (2013)

Permiten que el microcontrolador ATmega328P pueda recibir directamente datos en serie (por el pin RX) o transmitirlos (por el pin TX) sin pasar por la conversión USB-Serie que realiza el chip ATmega16U2. Es decir, estos pines posibilitan la comunicación sin intermediarios de dispositivos externos con el receptor/transmisor serie (de tipo TTL-UART) que incorpora el propio ATmega328P. De todas maneras, estos pines están internamente conectados (mediante resistencias de 1 K Ω) al chip ATmega16U2, por lo que los datos disponibles en el USB también lo estarán en estos pines. (p.94)

Hay que aclarar que en la placa están incrustados un par de LEDs etiquetados como “RX” y “TX”, pero que, a pesar de su nombre, no se encienden cuando se reciben o transmiten datos de los pines 0 y 1, sino solamente cuando se reciben o transmiten datos provenientes de la conexión USB a través del chip ATmega16U2.

2.1.8 HARDWARE LIBRE: ARDUINO

Torrente (2013)

La placa Arduino es hardware libre porque sus ficheros esquemáticos están disponibles para descargar de la página web del proyecto con la licencia Commons Attribution Share-Alike, la cual es una licencia libre que permite realizar trabajos derivados tanto personales como comerciales (siempre que estos den crédito a Arduino y publiquen sus diseños bajo la misma licencia). Así pues, uno mismo se

¹ Pulse Width Modulation

puede construir su propia placa Arduino “a mano”. No obstante, lo más normal comprarlas de un distribuidor ya preensambladas y listas para usar; en ese caso, lógicamente, la placa Arduino, aunque sea libre, no puede ser gratuita, ya que es un objeto físico y su fabricación cuesta dinero. (...) El Objetivo del hardware libre es, por lo tanto, facilitar y acercar la electrónica, la robótica y en definitiva la tecnología actual a la gente, no de una manera masiva, meramente consumista, sino de manera activa, involucrando al usuario final para que entienda y obtenga más valor de la tecnología actual e incluso ofreciéndole la oportunidad de participar en la creación de futuras tecnologías. Básicamente el hardware abierto significa tener la posibilidad de mirar que hay dentro de las cosas, y que eso sea éticamente correcto. Permite en definitiva, mejorar la educación de las personas. Por eso el concepto de software y hardware libre es tan importante, no solo para el mundo de la informática y de la electrónica, sino para la vida en general. (p. 69)

2.1.9 SOFTWARE LIBRE DE ARDUINO

Torrente (2013)

El software de Arduino es software libre porque se publica con una combinación de la licencia GPL (para el entorno visual de programación propiamente dicho) y la licencia LGPL (códigos de fuente de gestión y control del microcontrolador a nivel más interno).

La consecuencia de esto es, en pocas palabras, que cualquier persona que quiera (y sepa), puede formar parte del desarrollo del software Arduino y contribuir a mejorar dicho software, aportando nuevas características, sugiriendo ideas de nuevas funcionalidades, compartiendo soluciones a errores existentes, etc. Esta manera de funcionar provoca la creación espontánea de una comunidad de personas que colaboran mutuamente a través de Internet, y consigue que el software Arduino evolucione según lo que la propia comunidad decida.

Esto va mucho más allá de la simple cuestión de si el software Arduino es gratis o no, porque el usuario deja de ser un sujeto pasivo para pasar a ser (si quiere) un sujeto activo y partícipe del proyecto. (p. 68)

2.1.9.1 IDE DE ARDUINO

Torrente (2013)

Un programa es un conjunto concreto de instrucciones, ordenadas y agrupadas de forma adecuada y sin ambigüedades que pretende obtener un resultado determinado. Cuando decimos que un microcontrolador es “programable”, estamos diciendo que permite grabar en su memoria de forma permanente (hasta que regrabemos de nuevo si es necesario) el programa que deseemos que dicho microcontrolador ejecute. Si no introducimos ningún programa en la memoria del microcontrolador, este no sabrá qué hacer. (p. 130)

El IDE² es simplemente una forma de llamar a un conjunto de herramientas por software, que se encuentran en un programa; en el cual podemos escribir el también llamado sketch, lo podemos escribir, modificar varias veces y probar.

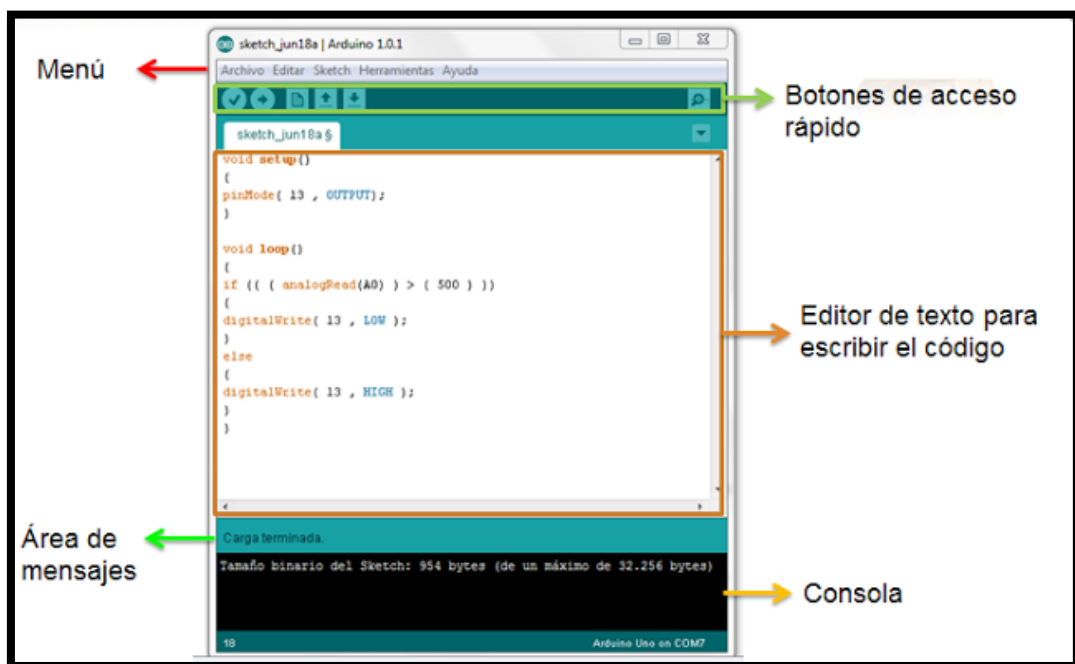


FIGURA 1: IDE de Arduino

Fuente: [http://4.bp.blogspot.com/-PAk-](http://4.bp.blogspot.com/-PAk-1ZLISfo/T_eVzsFq8tl/AAAAAAAAAJQ/I7q_xXrkIm8/s640/ide+arduino+2.png)

[1ZLISfo/T_eVzsFq8tl/AAAAAAAAAJQ/I7q_xXrkIm8/s640/ide+arduino+2.png](http://4.bp.blogspot.com/-PAk-1ZLISfo/T_eVzsFq8tl/AAAAAAAAAJQ/I7q_xXrkIm8/s640/ide+arduino+2.png)

² Integrated Development Environment=Entorno interactivo de desarrollo

2.1.9.2 ESTRUCTURA GENERAL DE UN SKETCH DE ARDUINO

Torrente (2013), manifiesta:

Un programa diseñado para ejecutarse sobre un Arduino (un “sketch”) siempre se compone de tres secciones:

- La sección de declaraciones de variables globales: ubicada directamente al principio del sketch.
- La sección llamada “void setup()”: delimitada por llaves de apertura y cierre.
- La sección llamada “void loop()”: delimitada por llaves de apertura y cierre.

La primera sección del sketch (que no tiene ningún tipo de símbolo delimitador de inicio o de final) está reservada para escribir, tal como su nombre indica, las diferentes declaraciones de variables que necesitemos. En un apartado posterior explicaremos ampliamente qué significa todo esto.

En el interior de las otras dos secciones (es decir, dentro de sus llaves) deberemos escribir las instrucciones que deseemos ejecutar en nuestra placa, teniendo en cuenta lo siguiente:

Las instrucciones escritas dentro de la sección “void setup()” se ejecutan una única vez, en el momento de encender (o resetear) la placa Arduino.

Las instrucciones escritas dentro de la sección “void loop()” se ejecutan justo después de las de la sección “void setup()” infinitas veces hasta que la placa se apague (o se resetee). Es decir, el contenido de “void loop()” se ejecuta desde la 1ª instrucción hasta la última, para seguidamente volver a ejecutarse desde la 1ª instrucción hasta la última, para seguidamente ejecutarse desde la 1ª instrucción hasta la última, y así una y otra vez.

Por tanto, las instrucciones escritas en la sección “void setup()” normalmente sirven para realizar ciertas preconfiguraciones iniciales y las instrucciones del interior de “void loop()” son, de hecho, el programa en sí que está funcionando continuamente. (p. 154)

2.1.10 PLACAS DE ARDUINO

Entre las placas de Arduino, existen varias con características diferentes, citaremos algunas de las placas más utilizadas y las más conocidas en el medio local.

2.1.10.1 ARDUINO UNO

Arduino (2014)

El Arduino Uno es una placa electrónica basada en el ATmega328. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 pueden utilizarse para salidas PWM), 6 entradas analógicas, un 16 MHz resonador cerámico, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador o la batería AC-to-DC para empezar. El Uno se diferencia en que no utiliza el chip controlador de USB a serial FTDI. En lugar de ello, cuenta con la Atmega16U2 (Atmega8U2 hasta la versión R2) programado como un convertidor de USB a serie.

2.1.10.2 ARDUINO MEGA 2560

Torrente (2013)

Placa basada en el microcontrolador ATmega2560. Como características más destacables diremos que tiene 54 pines de entrada/salida digitales (de los cuales 14 pueden ser usados como salidas analógicas PWM), 16 entradas analógicas y 4 receptores/transmisores serie TTL-UART. Consta de una memoria Flash de 256 Kilobytes (de los cuales 8 están reservados para el bootloader), una memoria SRAM de 8 KB y una EEPROM de 4 KB. Su voltaje de trabajo es igual al del modelo UNO: 5 V. (p. 102)

2.1.10.3 ARDUINO ETHERNET

Torrente (2013)

Al igual que el modelo UNO, la placa Ethernet está basada en el microcontrolador ATmega328P (y por lo tanto, tiene la misma cantidad de memoria Flash, SRAM y EEPROM), y también tiene el mismo número de pines de entrada/salida digitales y de entradas analógicas. El resto de características también es muy similar al modelo UNO. La mayor diferencia que existe con la placa UNO es que la placa Ethernet incorpora un zócalo de tipo RJ-45 para poder conectarse mediante el cable adecuado (cable de par trenzado de categoría 5 o 6) a una red de tipo Ethernet. (p. 104)

2.1.10.4 ARDUINO NANO

Torrente (2013)

La característica más destacable de esta placa es que a pesar de su tamaño (0,73 pulgadas de anchura por 1,70 de longitud), sigue ofreciendo el mismo número de salidas y entradas digitales y analógicas que la Arduino UNO y la misma funcionalidad que esta. La consecuencia más evidente de su reducido tamaño es que carece del conector de alimentación de 2,1 mm (aunque puede seguir siendo alimentada por una fuente externa mediante el pin “Vin” o “5 V”) e incorpora un conector USB mini-B en vez del conector USB tipo B. Esta placa está especialmente pensada para conectarla a una breadboard mediante las patillas que sobresalen de su parte posterior, pudiendo formar parte así de un circuito complejo de una manera relativamente fija. (p. 111)

2.1.10.5 ARDUINO MINI

Torrente (2013)

Esta placa es muy parecida a la placa Arduino Nano: está basada igualmente en el microcontrolador ATmega328P SMD funcionando a 16MHz, tiene 14 pines de entrada/salida digitales (6 de los cuales pueden funcionar como salidas PWM) y 8 entradas analógicas. Y al igual que la placa Arduino Nano, la Arduino Mini está especialmente pensada para conectarla a una breadboard mediante las patillas que sobresalen de su parte posterior, pudiendo formar parte así de un circuito complejo de una manera relativamente fija. (p. 112)

2.1.10.6 ARDUINO LEONARDO

Torrente (2013)

La gran novedad de esta placa es que el microcontrolador que incorpora es el ATmega32U4 (en formato SMD), el cual tiene todas las funcionalidades que ofrece el Atmega328P pero incorpora además 0,5 kilobytes más de memoria SRAM y sobretodo, soporta comunicaciones USB directamente (y por tanto, no necesita de ningún chip suplementario como el ATmega16U2 o el FTDI). (p. 113)

2.1.10.7 ARDUINO DUE

Torrente (2013)

Esta placa pertenece a una familia totalmente distinta de la del resto de placas Arduino. Incluye el microcontrolador SAM3X8E, el cual, aunque fabricado también por Atmel, es de una arquitectura interna muy diferente a la AVR (concretamente es de tipo ARM Cortex-M3) y además, sus registros son cuatro veces más grandes de lo habitual en las otras placas (concretamente, son de 32 bits). Su velocidad de reloj está también muy por encima del resto de placas Arduino (concretamente, es de 84 MHz). Además, el microcontrolador SAM3X8E dispone de muchas más memoria (concretamente, 96 KB de SRAM y 512 KB de memoria Flash) y también de un circuito especializado (llamado controlador “DMA”) que permite a la CPU acceder a la memoria de una manera mucho más rápida. (p. 115)

2.1.11 ARDUINO LILYPAD

Torrente (2013)

La placa de Arduino Lilypad es una placa con tecnología más flexible en cuanto al uso en textiles. Se pueden coser a la ropa muy fácilmente. La placa de Arduino está diseñada para ser cocida a material textil. Permite además conectarle (mediante hilos conductores) fuentes de alimentación, sensores y actuadores de forma que se puedan “llevar encima”, haciendo posible la creación de vestidos y ropa “inteligente”. Además, se puede lavar. Esta placa incorpora el microcontrolador ATMEGA328V (una versión de bajo consumo del Atmega328P), el cual se programa acoplado a la placa un adaptador o cable USB. (p. 111)

2.1.11.1 LILYPAD USB

Arduino (2014)

El LilyPad Arduino USB es una placa electrónica basada en el ATMEGA32U4 (ficha técnica). Cuenta con 9 pines digitales de entrada / salida (de los cuales 4 pueden utilizarse para salidas PWM y 4 entradas analógicas), como un núcleo de 8 MHz resonador, una conexión micro USB, un conector JST de 3.7V LiPo batería y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o el poder con una batería para empezar.

El LilyPad Arduino USB difiere de anteriores LilyPad tableros en que el ATMEGA32U4 ha incorporado en la comunicación USB, eliminando la necesidad de un adaptador independiente-USB a serie. Esto permite que la LilyPad Arduino USB aparezca a una computadora conectada como un ratón y el teclado, además de una virtual (CDC) de puerto serie / COM.

2.1.11.1.1 CARACTERÍSTICAS FÍSICAS DE ARDUINO LILYPAD USB

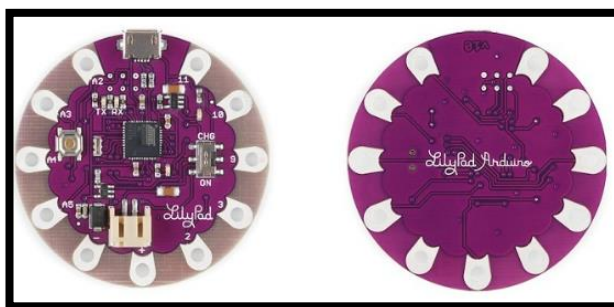


FIGURA 2: Lilypad USB

Recuperado de: <http://arduino.cc/en/Main/ArduinoBoardLilyPadUSB>

Arduino (2014)

El Lilypad Arduino USB es un círculo, de aproximadamente 50 mm (2 ") de diámetro. El conector USB Micro se extiende ligeramente más allá de este diámetro. La placa en sí es .8mm (1/32") de espesor (aproximadamente 6,5 mm (1/4") incluidos los componentes, el más alto de los cuales es el conector de la batería JST.

2.1.11.1.2 POTENCIA DE ARDUINO LILYPAD USB

El LilyPad Arduino USB puede ser alimentado a través de la conexión micro USB o con un 3.7V LiPo batería (conectado al conector JST en el tablero). De cualquier fuente de energía es regulada a la tensión de funcionamiento (3.3V) por el de a bordo MIC5219.

La placa se puede encender y apagar con el interruptor de a bordo. Con el interruptor en la posición ON, el microcontrolador recibe energía y del funcionamiento del Consejo. Con el interruptor en la posición "CHG", el microcontrolador no recibe energía. (Esto es cierto si el tablero es alimentado a través de USB o una batería.)

La tarjeta contiene un MCP73831 LiPo batería chip de carga. Si la tarjeta está conectado a USB y una batería, la potencia USB cargará la batería. Esto es cierto independientemente de la posición del interruptor. El LED encima de la palabra "CHG" se enciende mientras la batería se está cargando. La carga se detendrá automáticamente cuando la batería está completamente cargada. (Arduino, 2014)

Los pines de alimentación son como sigue:

- (+) El poder 3.3V de alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esto puede venir de la conexión USB o una batería, tanto a través del regulador de a bordo. Este pin sólo se enciende cuando el interruptor de a bordo está en la posición ON.
- (-) Pin de tierra.

2.1.11.1.3 MEMORIA DE ARDUINO LILYPAD USB

El ATMEGA32U4 tiene 32 KB (con 4 KB utilizado por el gestor de arranque). También tiene 2,5 KB de SRAM y 1 KB de EEPROM.

2.1.11.1.4 ENTRADAS Y SALIDAS DE ARDUINO LILYPAD USB

Cada uno de los pines en el 9 i digitales / LilyPad Arduino USB se puede utilizar como entrada o salida, utilizando pinMode (), digitalWrite (), y digitalRead () funciones. Operan a 3,3 V voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectada por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

- TWI: 2 (SDA) y 3 (SCL). Apoyar la comunicación TWI usando la librería Wire.
- Interrupciones externas: 2 y 3 Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor. Ver el attachInterrupt () función para más detalles.
- PWM: 3, 9, 10, 11, y 13 para salidas PWM de 8 bits con el analogWrite () función.
- LED: 13 Hay un built-in LED conectado al pin digital 13 Cuando el pin es de alto valor, el LED está encendido, cuando el pasador es bajo, es apagado.

- Entradas analógicas: A2-A5 El LilyPad Arduino USB cuenta con 4 entradas analógicas, etiquetados A0 a A11, todos los cuales también se pueden utilizar como E / S digitales. Cada entrada analógica proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto, la medida de las entradas analógicas de la tierra a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el `analogReference` función ().

2.1.11.1.5 COMUNICACIÓN DE ARDUINO LILYPAD USB

El LilyPad Arduino USB tiene una serie de instalaciones para la comunicación con un ordenador, otro Arduino, u otros microcontroladores. El 32U4 también permite la serie (CDC) de comunicación a través de USB y aparece como un puerto virtual para el software en el ordenador. El chip también actúa como un dispositivo de máxima velocidad USB 2.0, el uso de controladores USB COM estándar.

En Windows, se requiere un archivo `inf`. El software de Arduino incluye un monitor de serie que permite a los datos textuales sencillos para ser enviados hacia y desde la placa Arduino. Las RX y TX LED en el tablero parpadean cuando se están transmitiendo datos a través de la conexión USB a la computadora.

Una biblioteca `Software Serial` permite la comunicación en serie en cualquiera de la LilyPad pines digitales. El LilyPad Arduino USB también es compatible con I2C (TWI).

El software de Arduino incluye una librería `Wire` para simplificar el uso del I2C bus; consulte la documentación para obtener más información. El LilyPad Arduino USB aparece como un teclado y un ratón genérico, y puede ser programado para controlar estos dispositivos de entrada utilizando los teclados.

2.1.11.1.6 PROGRAMACIÓN DE ARDUINO LILYPAD USB

El LilyPad Arduino USB se puede programar con el software Arduino. Seleccione "LilyPad Arduino USB" de la Herramientas> Placa de menú (de acuerdo con el microcontrolador en la placa). Para obtener más información, consulte la referencia y tutoriales.

Los ATmega32U4 en el LilyPad Arduino USB viene precargado con un gestor de arranque que le permite cargar nuevo código a la misma sin el uso de un programador de hardware externo. Se comunica usando el AVR109 protocolo.

También puede pasar por alto el gestor de arranque y programar el microcontrolador a través de la (programación serial en circuito) cabeza ICSP. Mientras que los agujeros son demasiado pequeños para insertar clavijas en, puede insertar pines del conector macho en el conector ISP en el programador y presiona contra la cabecera ICSP en el tablero (desde arriba).

2.1.11.2 LILYPAD ARDUINO ATMEGA168

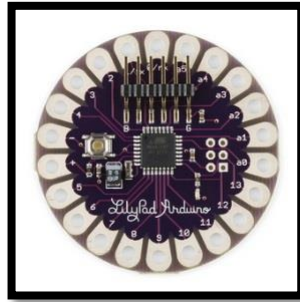


FIGURA 3: Lilypad Arduino

Recuperado de: <http://arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardLilyPad>

El LilyPad Arduino es una placa electrónica diseñada para prendas de vestir, textiles, entre otros. Puede ser cosido a la tela y montado de manera similar a fuentes de alimentación, sensores y actuadores con hilo conductor. El consejo se basa en la ATmega168V (la versión de bajo consumo del ATmega168) o el ATmega328V. El LilyPad Arduino fue diseñado y desarrollado por Leah Buechley y Sparkfun Electrónica.

2.1.11.2.1 PROGRAMACIÓN DE ARDUINO LILYPAD

El LilyPad Arduino se puede programar con el software Arduino. El LilyPad Arduino sólo debe programarse con versiones de software 0.010 o superior. Se puede programar con las versiones anteriores, pero todas las funciones relacionadas con el tiempo se apagará (el doble de lento como debe ser).

El ATmega168V o ATmega328V en el Arduino LilyPad viene precargado con un gestor de arranque que le permite cargar nuevo código para que con el software de Arduino. También puede pasar por alto el gestor de arranque y el programa del ATmega a través del ICSP (In-Circuit Serial Programming).

2.1.11.2.2 POTENCIA DE ARDUINO LILYPAD

El LilyPad Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. Si se utiliza una fuente de alimentación externa, se debe proporcionar entre 2,7 y 5,5 voltios. Esto puede venir con un adaptador de CA a CC (pared-verruga) o la batería. Una vez más, no alimentar el LilyPad Arduino con más de 5,5 voltios, o conecte la fuente al revés.

2.1.11.2.3 CARACTERÍSTICAS FÍSICAS DE ARDUINO LILYPAD

El LilyPad Arduino es un círculo, de aproximadamente 50 mm (2 ") de diámetro. La placa en sí es .8mm (1/32") de espesor (aproximadamente 3 mm (1/8")), donde se unen la electrónica).

2.1.11.2.4 LAVABILIDAD DE ARDUINO LILYPAD

Se puede lavar bajo propio riesgo. Se recomienda lavar a mano con un detergente suave. Escurrir. Asegúrese de retirar la fuente de alimentación primero.

2.1.11.3 ARDUINO LILYPAD SIMPLE

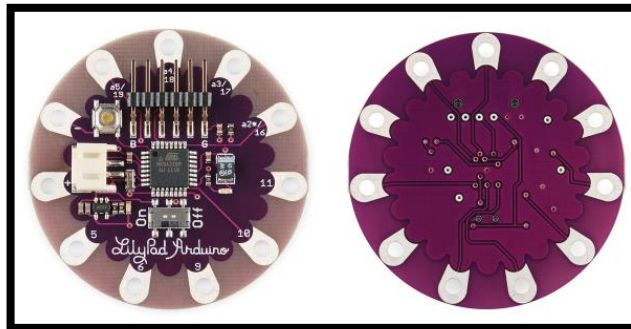


FIGURA 4: Lilypad Arduino Simple

Recuperado de: <http://arduino.cc/en/uploads/Guide/LilyPadBoards.jpg>

El LilyPad simple Arduino es una placa electrónica diseñado para e-textiles. Puede ser cosido a la tela y montado de manera similar fuentes de alimentación, sensores y actuadores con hilo conductor. A diferencia de la placa principal Arduino LilyPad, el LilyPad simple tiene sólo 9 pines para entrada / salida. Además, cuenta con un conector JST y construido en un circuito de carga para las baterías de polímero de litio.

2.1.11.3.1 PROGRAMACIÓN DE ARDUINO LILYPAD SIMPLE

El LilyPad simple Arduino se puede programar con el software Arduino. Seleccione "LilyPad Arduino" de la Herramientas, después ir a la placa de menú (de acuerdo con el microcontrolador en la placa). Los ATmega328 en el LilyPad Arduino simple viene precargado con un gestor de arranque que le permite cargar nuevo código para que con el software de Arduino. El LilyPad simple no tiene un puerto USB serial. Para programar la tarjeta, usted tendrá que utilizar un adaptador compatible FTDI como el adaptador de luz USB serial.

2.1.11.3.2 POTENCIA DE ARDUINO LILYPAD SIMPLE

El Arduino LilyPad simple funciona con una fuente de alimentación externa o con un adaptador compatible FTDI. La placa se puede encender y apagar con el interruptor de a bordo. Cuando la placa no está conectado a un adaptador FTDI y esta alimentado a través de una batería, se puede encender y apagar con el interruptor de la palca; con el interruptor en la posición ON, el microcontrolador recibe energía y con el interruptor en la posición OFF, el microcontrolador no recibe alimentación. Cuando la placa se alimenta a través de un adaptador FTDI, la placa se mantiene todo el tiempo encendida; con el interruptor en la posición ON, el microcontrolador recibe energía de la batería (o la placa FTDI a través del circuito de carga de la batería, si la batería no está conectado) y con el interruptor en la posición OFF, el microcontrolador recibe energía del adaptador FTDI.

Una fuente de alimentación externa debe suministrar entre 2,7 y 5,5 voltios. El Arduino Lilypad simple está diseñado con el uso de una batería de polímero de litio de 3,7 voltios que se pueden conectar directamente al conector JST de a bordo, no alimentar el Arduino LilyPad simple con más de 5,5 voltios, o conectar la fuente al revés ya que quemaría la placa.

La tarjeta contiene un MCP73831 chip de carga de batería lipo. Si la tarjeta está conectada tanto a una conexión FTDI y una batería, la potencia FTDI cargará la batería. Esto es cierto independientemente de la posición del interruptor. El LED junto al interruptor se enciende mientras la batería se está cargando. La carga se detendrá automáticamente cuando la batería está completamente cargada.

2.1.11.3.3 ENTRADAS Y SALIDAS DE ARDUINO LILYPAD SIMPLE

El Arduino LilyPad simple tiene un menor número de entradas y salidas que la placa principal Arduino LilyPad. Hay un total de 9 pins E / S en el tablero simple, una patilla al descubierto para + 3.3 VCC, y un pin de tierra.

Cada uno de los pasadores 9 E / S digitales en el LilyPad simple Arduino se puede utilizar como una entrada o salida, utilizando `pinMode ()`, `digitalWrite ()`, y `digitalRead ()` funciones. Ellos operan a 5V voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectada por defecto) de 20 kOhm. Además, algunos pines tienen funciones especializadas:

- PWM: 5, 6, 9, 10, 11 para salidas PWM de 8 bits con el `analogWrite ()` función.
- Entradas analógicas: A2-A5 El LilyPad Arduino simple cuenta con 4 entradas analógicas, etiquetada A2 a A5, todos los cuales también se pueden utilizar como E / S digitales. Cada entrada analógica proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto, la medida de las entradas analógicas de la tierra a 5 voltios, aunque es posible cambiar el extremo superior de su rango usando el `analogReference` función (`analogReference ()`).

2.1.11.3.4 CARACTERÍSTICAS FÍSICAS DE ARDUINO LILYPAD SIMPLE

El Arduino LilyPad Simple es un círculo, de aproximadamente 50 mm (2 ") de diámetro. La placa en sí es .8mm (1/32") de espesor (aproximadamente 3 mm (1/8"), donde se unen la electrónica).

2.1.11.3.5 LAVABILIDAD DE ARDUINO LILYPAD SIMPLE

Recomendamos lavar los proyectos realizados con esta placa, en agua fría a mano con un detergente suave. Extraiga la batería antes de lavar el tablero.

2.1.12 TABLA COMPARATIVA DE LAS PLACAS ELECTRÓNICAS ARDUINO

TABLA 1: Comparación entre placas Arduino (Parte 1)

CARACTERÍSTICAS	ARDUINO UNO	ARDUINO MEGA2560	ARDUINO NANO	ARDUINO YUN	ARDUINO LEONARDO
Microcontrolador	Atmega328	Atmega2560	Atmega2560	Atmega 32u4	Atmega 32u4
Tensión de funcionamiento	5 voltios	5 voltios	5 voltios	5 voltios	5 voltios
Voltaje de entrada recomendando	7-12 voltios	7-12 voltios	7-12 voltios	5 voltios	7-12 voltios
Voltaje de entrada límite	6-20 voltios	6-20 voltios	6-20 voltios	5 voltios	6-20 voltios
Pines de entrada/salida digitales	14 los cuales 6 proporcionan salida PWM	54 los cuales 15 proporcionan salida PWM	54 los cuales 15 proporcionan salida PWM	20 los cuales 7 proporcionan salida PWM	20 los cuales 7 proporcionan salida PWM
Pines de entrada analógicos	6	16	16	12	12
Corriente DC por pin de entrada/salida	40Ma	40mA	40mA	40mA	40mA
Corriente DC del pin de 3.3 Voltios	50mA	50mA	50mA	50mA	50mA
SRAM	2 KB de memoria flash	256 KB de memoria flash	256 KB de memoria flash	2,5 KB	2,5 KB
EEPROM	1 KB	4 KB	4 KB	1 KB	1 KB
Velocidad de reloj	16 MHz	16MHz	16MHz	16 MHz	16 MHz
Costo	28 dólares	35 dólares	20 dólares	110 dólares	40 dólares

Fuente: <http://arduino.cc/en/Products.Compare>

TABLA 2: Comparación entre placas Arduino (Parte 2)

CARACTERÍSTICAS	ARDUINO MICRO	ARDUINO ETHERNET	ARDUINO FIO	ARDUINO ROBOT Control - Motor		ARDUINO EXPLORA	ARDUINO LILYPAD
Microcontrolador	Atmega32u4	Atmega328	Atmega328P	Atmega 32u4	Atmega 32u4	Atmega 32u4	Atmega 128V-328V
Tensión de funcionamiento	5 voltios	5 voltios	3.3 voltios	5 voltios	5 voltios	5 voltios	2.7 a 5.5 voltios
Voltaje de entrada recomendando	7-12 voltios	7-12 voltios	3,7-7 voltios	5 voltios	9 voltios	5 voltios	2.7 a 5.5 voltios
Voltaje de entrada límite	6-20 voltios	6-20 voltios	3,35-12 voltios	5 voltios	9 voltios	5 voltios	2.7 a 5.5 voltios
Pines de entrada/salida digitales	20 y 7 proporcionan salida PWM	14 y 4 proporcionan salida PWM	14 y 6 proporciona salida PWM	5 los cuales 6 proporcionan salida PWM	4 y 1 proporciona salida PWM	--	14 los cuales 6 son salida PWM
Pines de entrada analógicos	12	6	16	4 de los pines de E/S	4	--	6
Corriente DC por pin de entrada/salida	40mA	40mA	40mA	40mA	40mA	--	40mA
Corriente DC del pin de 3.3 Voltios	50mA	50Ma	--	--	--	--	--
Memoria Flash	32 KB	32 KB	32 KB (2KB arranque)	32 KB (4Kb arranque)	32 KB (4Kb arranque)	32 KB (4 KB arranque)	16 KB (2 KB arranque)
SRAM	2,5 KB	2 KB	2 KB	2,5 KB	2,5 KB	2,5 KB	1 KB
EEPROM	1 KB	4 KB	1 KB	1 KB	1 KB	1 KB	512 bytes
Velocidad de reloj	16 MHz	16 MHz	8 MHz	16 MHz	16 MHz	16 MHz	8 MHz
Costo (Dólares)	28 dólares	20 dólares	25 dólares	110 dólares		60 dólares	16 dólares

Fuente: <http://arduino.cc/en/Products.Compare>

A pesar de la gran variedad de placas electrónicas Arduino existentes y de la gran cantidad de características que estas poseen, es importante tomar en cuenta primero la aplicación a desarrollar por el usuario para tener una buena elección de la placa a utilizar. El campo de los textiles inteligentes está tomando grandes desarrollos dentro del área de diseño e ingeniería, por lo que es importante recalcar que la elección del Arduino LilyPad se adapta perfectamente a las condiciones del proyecto de sistema de monitoreo cardiaco; la placa electrónica será cocida con hilo conductor en una muñequera, logrando adaptar los sensores, el módulo de comunicación Bluetooth y los elementos electrónicos pasivos.

2.1.13 ARDUINO LILYPAD Y TEXTILES INTELIGENTES

LilyPad es un conjunto de piezas electrónicas diseñadas para ayudarle a construir tejidos blandos interactivos. Un conjunto de módulos, incluyendo electrónicos, en un pequeño ordenador programable llamado LilyPad Arduino puede ser cosida con un hilo conductor para crear prendas y accesorios interactivos.

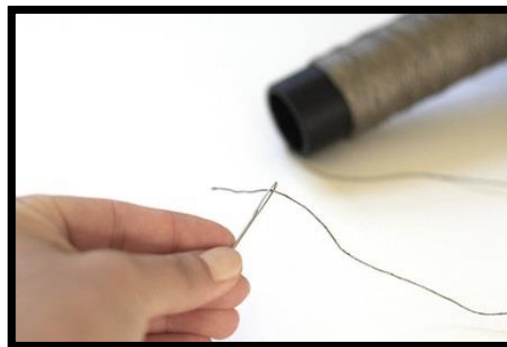


FIGURA 5: Hilo conductor para Arduino Lilypad

Recuperado de: http://lilypadarduino.org/?page_id=1260

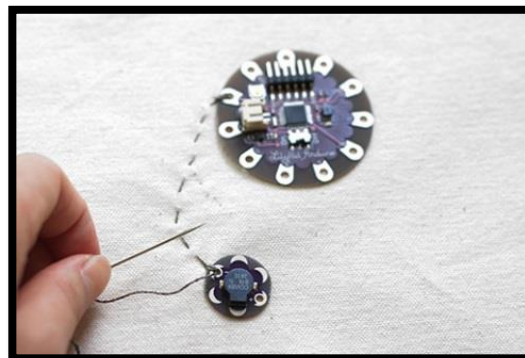


FIGURA 6: Hilo conductor conectado a Lilypad

Recuperado de: http://lilypadarduino.org/?page_id=1260

2.1.13.1 TEXTILES INTELIGENTES

Garcia (2005)

Se entiende por textiles inteligentes los que tienen incorporados determinados elementos o sistemas que les permiten responder de forma automática a las necesidades del cuerpo humano en función de las características del entorno. La determinación de textiles de uso técnico se aplica a los productos textiles que dan respuesta a determinadas solicitudes elevadas, requeridas por un problema industrial para una aplicación concreta. Las prendas inteligentes proporcionan al usuario determinados beneficios funcionales. (p. 68)



FIGURA 7: Textiles Inteligentes

Fuente: <http://image.slidesharecdn.com/lilypadarduinoytextilesintel1-130607032709-phpapp02/95/lilypad-arduino-y-textiles-intel1-4-638.jpg?cb=1370593733>

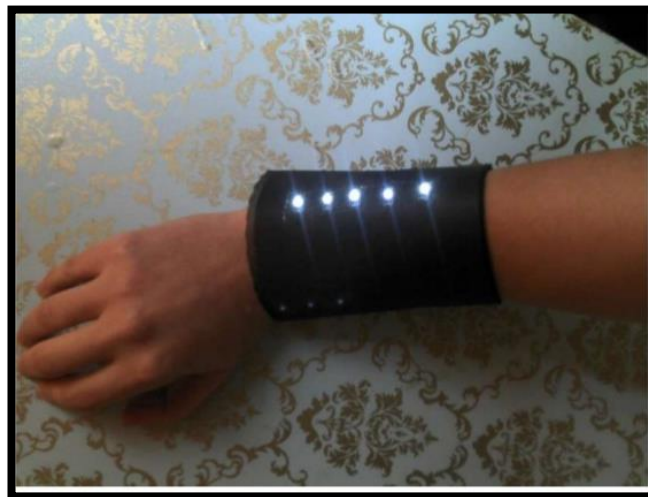


FIGURA 8: Uso de Textil Inteligente

Fuente: <http://image.slidesharecdn.com/lilypadarduinoytextilesintel1-130607032709-phpapp02/95/lilypad-arduino-y-textiles-intel1-5-638.jpg?cb=1370593733>

2.1.13.2 ROPA INTELIGENTE

Gómez (2011)

La ropa inteligente en algunos casos combina la ingeniería de tejidos con la microelectrónica, con la incorporación de sensores que cumplen distintas funciones (telefonía, control remoto, alarma de seguridad, diversión, etcétera) y tienen la posibilidad de que la ropa se pueda lavar, ya sea a mano o a máquina.



FIGURA 9: Ropa Inteligente

Recuperado de: <http://image.slidesharecdn.com/lilypadarduinoytextilesintel1-130607032709-phpapp02/95/lilypad-arduino-y-textiles-intel1-2-638.jpg?cb=1370593733>

2.2 FUNDAMENTOS DE COMUNICACIÓN INALÁMBRICA

2.2.1 INTRODUCCIÓN A LA COMUNICACIÓN INALÁMBRICA

Andrew (2003)

La comunicación inalámbrica digital no es una idea nueva. A principios de 1901, el físico italiano Guillermo Marconi demostró un telégrafo inalámbrico desde un barco a tierra utilizando el código Morse (después de todo, los puntos y rayas son binarios). Los sistemas inalámbricos digitales de la actualidad tienen un mejor desempeño, pero la idea básica es la misma. (p. 21)

2.2.2 REDES WPAN

“Una WPAN³ es una red inalámbrica de área personal. Las WPAN se definen mediante el estándar 802.15. Este estándar incluye la comunicación de dispositivos mediante Bluetooth.” (RODIL, 2010, p.127)

2.2.2.1 FUNCIONAMIENTO DE REDES WPAN

RODIL, 2010 manifiesta:

- La comunicación se realiza sin cables y consume poca energía.
- Su alcance es de pocos metros, unos 10 metros a la redonda
- En una red de ordenadores, se utilizan para conectar al ordenador periféricos y dispositivos personales como un PDA, un teléfono móvil, etc. (p.127)

2.2.3 INTRODUCCIÓN DE TECNOLOGÍA BLUETOOTH

Bluetooth empieza a concebirse en Ericsson Mobile Communications AB (Suecia) en 1994 como el efecto colateral de un proyecto sobre enlaces de comunicadores múltiples conectados a la red celular mediante teléfonos; cuatro años más tarde, aparece el grupo de interés en la tecnología Bluetooth SIG (Special Interest Group) en el que aparecían Ericsson, Toshiba, IBM e Intel como empresas promotoras de la tecnología; más tarde se sumaron otros promotores como 3Com, Lucent, Microsoft y Motorola generándose el consorcio Bluetooth; actualmente el Bluetooth SIG cuenta con más de 332000 empresas.

Plaza (2011) manifiesta “El logo es una combinación de dos letras del alfabeto rúnico, precisamente la H (Hagall) y la B (Berkanan)” (párr. 5).

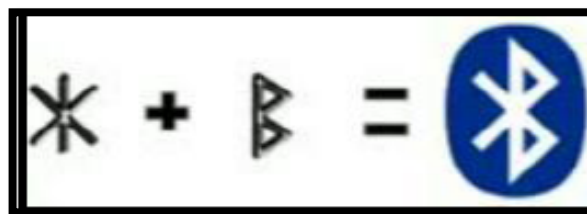


FIGURA 10: Logo de Bluetooth

Fuente: https://static.betazeta.com/www.wayerless.com/up/2011/09/imag_logo_bluetooth.jpg

³ Wireless Personal Area Network

2.2.3.1 COMUNICACIÓN BLUETOOTH

“El Bluetooth es un sistema de comunicación por radiofrecuencia desarrollado por Ericsson que permite la transmisión e intercambio de datos entre aparatos móviles tales como PDA, PC, teléfonos, etc.” (Domínguez, 2012, p. 107)

“Se diseñó específicamente para reemplazar a los puertos infrarrojos y cables de conexión de periféricos. La transmisión, al igual que WI-FI, se realiza por ondas de radio.” (Rodil, 2010, p.127)

2.2.3.2 VERSIONES DE BLUETOOTH

Los primeros emisores receptores de Bluetooth, fueron el v1.0 y v1.0B, los cuales ya están prácticamente obsoletos, y dieron muchísimos problemas a los fabricantes de teléfonos para la interacción entre dispositivos. Cada versión de Bluetooth incorpora notables mejoras en lo que respecta al consumo de energía y velocidades de transferencia, así como algunas mejoras de seguridad:

2.2.3.2.1 BLUETOOTH VERSIÓN 1.1

Chacón (2010) manifiesta:

Ericsson inició un estudio para investigar la viabilidad de una nueva interfaz de bajo costo y consumo para la interconexión vía radio (eliminando así cables) entre dispositivos como teléfonos móviles y otros accesorios. El estudio partía de un largo proyecto que investigaba unos multi comunicadores conectados a una red celular, hasta que se llegó a un enlace de radio de corto alcance, llamado MC link. Conforme este proyecto avanzaba se fue haciendo claro que éste tipo de enlace podía ser utilizado ampliamente en un gran número de aplicaciones, ya que tenía como principal virtud que se basaba en un chip de radio. (párr. 3).

Ratificado como estándar IEEE 802.15.1-2002, en 1999 Bluetooth lanzó su primera versión, permitía una velocidad de hasta 0.8 ~ 1Mbps⁵ a una distancia menor de 10 metros. Obviamente todo lo anterior era teórico y nunca se alcanzaban los 1Mbps y los supuestos 125 KB/s.

⁴ IEEE = Instituto de Ingeniería Eléctrica y Electrónica

⁵ Mbps = Mega Bits por Segundo, Unidad empleada para cuantificar un caudal de datos.

2.2.3.2.2 BLUETOOTH VERSIÓN 1.1

Ratificado como estándar IEEE 802.15.1-2005, esta versión es compatible con USB⁶ 1.1, a diferencia de la versión 1.1, provee una solución inalámbrica complementaria para coexistir Bluetooth y Wi-Fi en el espectro de los 2.4 GHz, sin interferencia entre ellos.

Chacón (2010) manifiesta:

La versión 1.2 usa la técnica “Adaptive Frequency Hopping (AFH), que ejecuta una transmisión más eficiente y un cifrado más seguro. Para mejorar las experiencias de los usuarios, la versión 1.2 ofrece una calidad de voz con menor ruido ambiental, y provee una más rápida configuración de la comunicación con los otros dispositivos bluetooth dentro del rango del alcance, como pueden ser: PDAs⁷, HIDs⁸, computadoras portátiles, computadoras de escritorio, Headsets⁹, impresoras y celulares. Mayor velocidad de transmisión en la práctica, de hasta 721 kbit/s, que en la versión 1.1 (párr. 4).

2.2.3.2.3 BLUETOOTH VERSIÓN 2.0

Fue lanzado en el año 2004 y es compatible con la versión anterior 1.2, “creada para ser una especificación separada, principalmente incorpora la técnica EDR¹⁰, que le permite mejorar las velocidades de transmisión en hasta 3Mbps” (Chacón, 2010, párr. 5), aunque en la realidad su tasa de transferencia es de 2.1 Mbps; a la vez que intenta solucionar algunos errores de la versión 1.2.

La especificación se publica como "Bluetooth v2.0 + EDR", lo que implica que EDR es una característica opcional. Aparte de EDR, hay otras pequeñas mejoras en la especificación 2.0, y los productos pueden reclamar el cumplimiento de "Bluetooth v2.0" sin el apoyo de la mayor tasa de datos. La inclusión en el mercado de estos dispositivos ayudó a mejorar la interoperabilidad para establecer una conectividad con dispositivos móviles.

⁶ USB = Universal Serial Bus

⁷ PDA = Personal Digital Assistant

⁸ HID = Human Interface Device

⁹ Headsets = Dispositivo que posee un auricular con un micrófono

¹⁰ EDR = Enhanced Data Rate, Velocidad de Datos Mejorada

2.2.3.2.4 BLUETOOTH VERSIÓN 2.1

Ibarra (2012) manifiesta

Bluetooth Core Versión especificación 2.1 + EDR es totalmente compatible con la versión 1.2, y fue adoptada por el Bluetooth SIG (Special Interest Group) el 26 de julio de 2007. La función de esta versión 2.1 es el SSP (Secure Simple Pairing), que mejora la experiencia de emparejamiento de dispositivos Bluetooth, mientras que el aumento del uso y la fuerza de seguridad; simplifica los pasos para crear la conexión entre dispositivos, además el consumo de potencia es cinco veces menor. (p. 6).

2.2.3.2.5 BLUETOOTH VERSIÓN 3.0

Ibarra (2012) manifiesta

La versión 3.0 + HS de la especificación principal Bluetooth fue aprobado por el Bluetooth SIG el 21 de abril de 2009. Permite aumentar considerablemente la velocidad de transferencia, soporta velocidades de transferencia de datos teórica de hasta 24 Mbit / s, aunque no a través del enlace Bluetooth sí mismo. La idea es que el nuevo Bluetooth trabaje con WiFi, de tal manera que sea posible lograr mayor velocidad en los Smartphone (pág. 7).

2.2.3.2.6 BLUETOOTH VERSIÓN 4.0

A finales del 2010 se lanzó esta nueva versión la cual permite disminuir drásticamente el consumo de energía, ampliando la cantidad de aplicaciones, permitiendo la incorporación de receptores y transmisores Bluetooth en dispositivos pequeños como relojes, reproductores portátiles, instrumental médico, etc. Se mejora la seguridad con encriptadores AES-¹¹128.

Ibarra (2012) manifiesta

El SIG de Bluetooth ha completado la especificación del Núcleo de Bluetooth en su versión 4.0, que incluye Bluetooth clásico, Bluetooth de alta la velocidad y protocolos Bluetooth de bajo consumo. Bluetooth de alta velocidad se basa en Wi-Fi, y Bluetooth clásico consta de protocolos Bluetooth legado.

¹¹ AES = Advanced Encryption Standard

Esta versión ha sido adoptada el 30 de junio de 2010. Bluetooth baja energía (BLE) es un subconjunto de Bluetooth v4.0 con una pila de protocolo completamente nuevo para la rápida acumulación de enlaces sencillos. Como alternativa a los protocolos estándar de Bluetooth que se introdujeron en Bluetooth versión 1.0 a versión 4.0 está dirigido a aplicaciones de potencia muy baja corriendo una célula de la moneda. Diseños de chips permiten dos tipos de implementación, de modo dual, de modo único y mejoradas versiones anteriores (pág. 7).

TABLA 3: Representación del ancho de banda y sus versiones

VERSIÓN	ANCHO DE BANDA
1.2	1 Mbps
2.0 + EDR	3 Mbps
3.0 + HS	24 Mbps
4.0	24 Mbps

Fuente: Basado en “Bluetooth”, Jonathan Ibarra Z., 2012

2.2.3.3 CARACTERÍSTICAS DE FUNCIONAMIENTO DE BLUETOOTH

Domínguez (2012)

Los dispositivos con tecnología Bluetooth, una vez que son puestos en contacto por radiofrecuencia, forman pequeñas redes de comunicación denominadas piconed, de hasta ocho dispositivos activos, de los cuales uno asume la función de maestro.

El dispositivo que ejerce como maestro es el q establece la comunicación, el resto de dispositivos se sincronizan con él, y solo el destinatario que ha recibido un paquete de datos del maestro puede transmitir una respuesta. Cada unidad activa puede recibir a su vez datos de otros dispositivos no activos, denominados picocelulares.

La protección en la trasmisión de datos se garantiza mediante una contraseña secreta de reconocimiento mutuo entre los diferentes dispositivos. (p. 107)

López (2013)

El estándar Bluetooth, del mismo modo que la tecnología WiFi, utiliza la técnica FHSS (Frequency Hopping Spread Spectrum, que en español significa Espectro ensanchado por saltos de frecuencia), que consiste en dividir la banda de frecuencia de 2.402 – 2.480 GHz en 79 canales (denominados saltos) de 1 MHz de ancho cada uno y, después, transmitir la señal utilizando una secuencia de canales que sea conocida tanto para la estación emisora como para la receptora.

Por lo tanto, al cambiar de canales con una frecuencia de 1600 veces por segundo, el estándar Bluetooth puede evitar la interferencia con otras señales de radio. El estándar Bluetooth define 3 clases de transmisores, cuyo alcance varía en función de su potencia radiada:

TABLA 4: Clases y Potencias de Bluetooth

CLASE	POTENCIA (PÉRDIDA DE SEÑAL)	ALCANCE
I	100 mW (20 dBm)	100 metros
II	2,5 mW (4 dBm)	15-20 metros
III	1 mW (0 dBm)	10 metros

Fuente: <http://arquitecturadecomputadora.wordpress.com/2013/05/23/bluetooth/>

2.2.3.4 TOPOLOGÍAS DE RED

Una topología de red prácticamente se basa en buscar la manera de interconectar el hardware o equipos de comunicación entre sí, mediante cables o vía inalámbrica con la finalidad de intercambiar información entre los dispositivos conectados. Existen tres tipos de topologías de red en donde dispositivos Bluetooth se interconectan, las cuales se mencionan a continuación:

2.2.3.4.1 TOPOLOGÍA PUNTO A PUNTO

Vásquez y Zúñiga, (2006) manifiestan: “Es una topología sencilla en donde la interconexión se realiza de “manera directa entre dos dispositivos, donde uno funciona como maestro y el otro como esclavo” (p. 12).

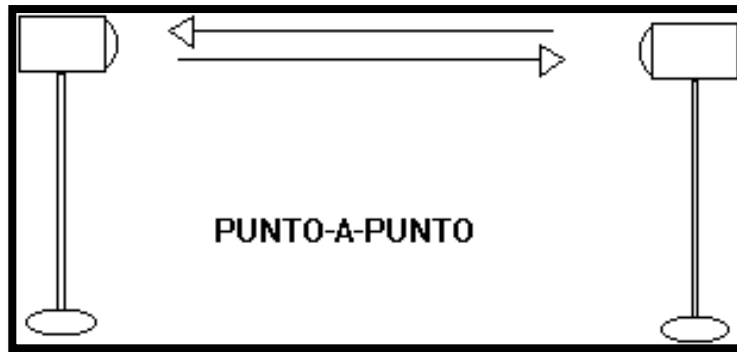


FIGURA 11: Topología Punto a punto

Fuente: <http://www.monografias.com/trabajos/redesinalam/Image73.gif>

2.2.3.4.2 TOPOLOGÍA PICONET

Vásquez y Zúñiga, (2006)

En esta topología básicamente una Piconet se forma únicamente de ocho dispositivos activos en donde los canales lógicos solamente pueden establecerse entre un dispositivo maestro y hasta siete dispositivos esclavos; los nodos esclavos no pueden formar canales lógicos entre sí, ya que estos deben necesariamente que pasar por un nodo o dispositivo maestro. “Si existiera más de ocho dispositivos dentro de una Piconet, el estado en la que se pueden encontrar sería en modo de reposo conocido como stand-by.” (p. 12).

Un mismo dispositivo puede formar parte de más de una Piconet, pero no puede ser Maestro más de una a la vez; el dispositivo Maestro que pertenezca a más de una Piconet podrá enrutar paquetes entre ambas piconets

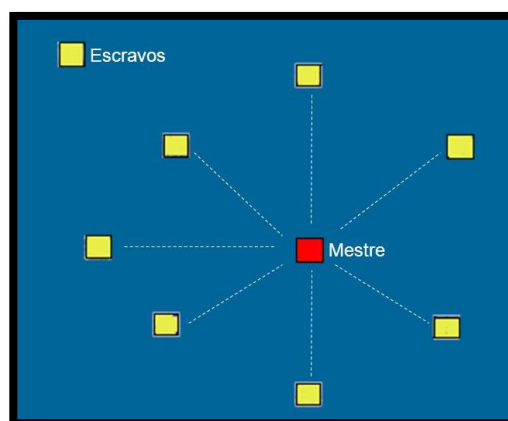


FIGURA 12: Topología Piconet

Fuente: http://images.slideplayer.com.br/1/278203/slides/slide_7.jpg

2.2.3.4.3 TOPOLOGÍA SCATTER-NET

La interconexión de varias piconets se denomina Scatter-net, este tipo de topología puede estar configurada de dos formas como:

2.2.3.4.3.1 TOPOLOGÍA MAESTRO-ESCLAVO

Vásquez y Zúñiga (2006) manifiestan: “Un dispositivo que siendo esclavo en una red puede ser Maestro para otra red distinta, pero este dispositivo no puede ser Maestro en más de una red a la vez.” (p. 13)

2.2.3.4.3.2 TOPOLOGÍA ESCLAVO-ESCLAVO

Un dispositivo puede ser esclavo de dos diferentes dispositivos Maestros como se observa en la figura 13.

En la imagen 13 se observa que existe una distribución de 10 dispositivos en una Scatter-net, en donde dos de los dispositivos se encuentran trabajando como Maestros de las piconets dos y tres, y los mismos dispositivos son esclavos de la Piconet uno

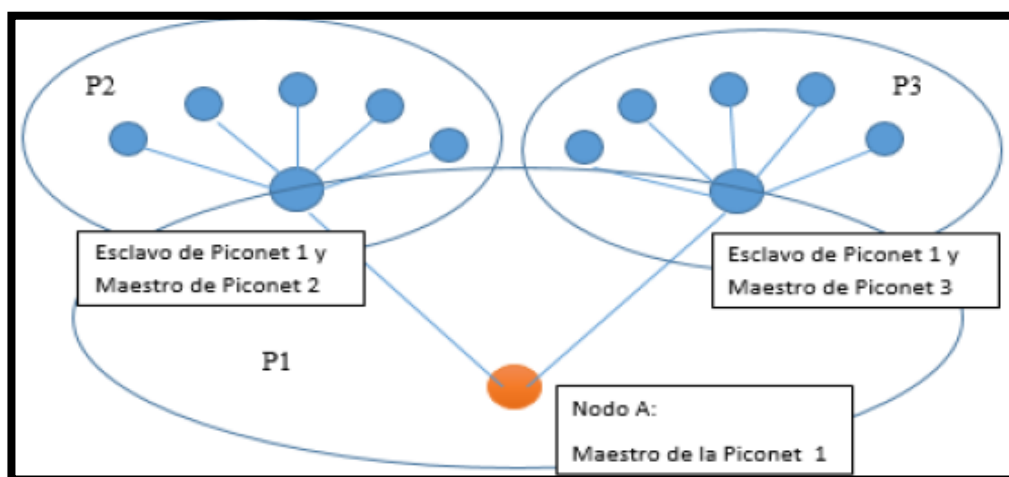


FIGURA 13: Topología Scatter-net

Fuente: <http://dspace.ups.edu.ec/bitstream/123456789/206/3/Capitulo%202.pdf>

2.2.3.5 ESTRUCTURA DE LA TRAMA DE BLUETOOTH

Andrew (2003)

Existen diversos formatos de trama, el más importante de los cuales se muestra en la figura 9 .Empieza con un código de acceso que identifica al maestro, cuyo propósito es que los esclavos que se encuentren en el rango de alcance de dos maestros sepan cuál tráfico es para ellos. A continuación se encuentra un encabezado de 54 bits que contiene campos comunes de la subcapa MAC. Luego está el campo de datos, de hasta 2744 bits (para una transmisión de cinco ranuras). Para una sola ranura de tiempo, el formato es el mismo excepto que el campo de datos es de 240 bits. (p. 21)

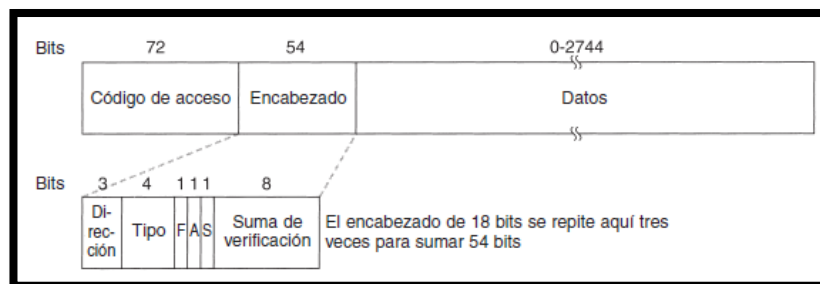


FIGURA 14: Trama de Datos Bluetooth

Fuente: <http://www.javierlongares.com/arte-en-8-bits/wp-content/uploads/2013/04/Tramas-de-datos-Zigbee.jpg>

2.2.3.6 PRINCIPIO DE COMUNICACIÓN BLUETOOTH

López (2013) manifiesta:

El estándar Bluetooth se basa en el modo de operación maestro/esclavo. El término "piconet" se utiliza para hacer referencia a la red formada por un dispositivo y todos los dispositivos que se encuentran dentro de su rango. Pueden coexistir hasta 10 piconets dentro de una sola área de cobertura. Un dispositivo maestro se puede conectar simultáneamente con hasta 7 dispositivos esclavos activos (255 cuando se encuentran en modo en espera). Los dispositivos en una piconet poseen una dirección lógica de 3 bits, para un máximo de 8 dispositivos. Los dispositivos que se encuentran en el modo en espera se sincronizan, pero no tienen su propia dirección física en la piconet.

En realidad, en un momento determinado, el dispositivo maestro sólo puede conectarse con un solo esclavo al mismo tiempo. Por lo tanto, rápidamente cambia de esclavos para que parezca que se está conectando simultáneamente con todos los dispositivos esclavos.

Bluetooth permite que dos piconets puedan conectarse entre sí para formar una red más amplia, denominada "scatternet", al utilizar ciertos dispositivos que actúan como puente entre las dos piconets.

2.2.3.7 ESTABLECIMIENTO DE CONEXIÓN BLUETOOTH

López (2013), manifiesta:

El establecimiento de una conexión entre dos dispositivos Bluetooth sigue un procedimiento relativamente complicado para garantizar un cierto grado de seguridad, como el siguiente:

1. Modo pasivo
2. Solicitud: Búsqueda de puntos de acceso
3. Paginación: Sincronización con los puntos de acceso
4. Descubrimiento del servicio del punto de acceso
5. Creación de un canal con el punto de acceso
6. Emparejamiento mediante el PIN (seguridad)
7. Utilización de la red
8. Durante el uso normal, un dispositivo funciona en "modo pasivo", es decir, que está escuchando la red.
9. El establecimiento de una conexión comienza con una fase denominada "solicitud", durante la cual el dispositivo maestro envía una solicitud a todos los dispositivos que encuentra dentro de su rango, denominados puntos de acceso. Todos los dispositivos que reciben la solicitud responden con su dirección.
10. El dispositivo maestro elige una dirección y se sincroniza con el punto de acceso mediante una técnica denominada paginación, que principalmente consiste en la sincronización de su reloj y frecuencia con el punto de acceso.
11. De esta manera se establece un enlace con el punto de acceso que le permite al dispositivo maestro ingresar a una fase de descubrimiento del servicio del punto de acceso, mediante un protocolo denominado SDP (Service Discovery Protocol, en español Protocolo de descubrimiento de servicios).

12. Cuando esta fase de descubrimiento del servicio finaliza, el dispositivo maestro está preparado para crear un canal de comunicación con el punto de acceso, mediante el protocolo L2CAP.
13. El punto de acceso puede incluir un mecanismo de seguridad denominado emparejamiento, que restringe el acceso sólo a los usuarios autorizados para brindarle a la piconet cierto grado de protección. El emparejamiento se realiza con una clave cifrada comúnmente conocida como "PIN" (PIN significa Personal Information Number [Número de identificación personal]). Para esto, el punto de acceso le envía una solicitud de emparejamiento al dispositivo maestro. La mayoría de las veces se le solicitará al usuario que ingrese el PIN del punto de acceso. Si el PIN recibido es correcto, se lleva a cabo la conexión. En el modo seguro, el PIN se enviará cifrado con una segunda clave para evitar poner en riesgo la señal.
14. Cuando el emparejamiento se activa, el dispositivo maestro puede utilizar libremente el canal de comunicación establecido. (página 45)

2.2.3.8 INTERFERENCIAS

Por el hecho de ser una tecnología basada en medios no guiados, se presenta una fuerte influencia en los problemas que este tipo de medios acarrear. Uno de estos problemas, sobretodo presente a este nivel de frecuencia, es la interferencia de la señal emitida. Una de las formas en que los dispositivos Bluetooth evitan interferir con otros sistemas es mandando señales muy débiles de 1 milivatios. En comparación, los teléfonos celulares más poderosos pueden transmitir una señal de 3 vatios. El bajo poder, limita el alcance de un dispositivo Bluetooth a unos 10m, eliminando la oportunidad de interferencias entre un sistema de computación y un teléfono inalámbrico o un televisor.

Shawn M. (2012) manifiesta:

Los dispositivos Bluetooth trabajan con la banda de radio de 2,4 GHz, que es la misma frecuencia sin licencia usada por muchos otros dispositivos inalámbricos. Si muchos dispositivos en la misma zona están utilizando el mismo tramo de ancho de banda, puede conducir a problemas de red globales, como las señales de colisión y la información tiene que ser reenviada. La señal de Bluetooth fue diseñado para cambiar su frecuencia en pequeños saltos con intervalos de tiempo muy cortos esto permite reducir las posibles interferencias; pero si suficientes dispositivos están intentando utilizar el mismo tramo pequeño de ancho de banda,

la interferencia será inevitable. Bluetooth 3.0, siendo utilizado en los dispositivos creados en el 2010, utiliza el espectro de 6.9 GHz, por lo que tendrá mucho menos los problemas de interferencias, también son capaces de transmitir a 2,4 GHz para comunicarse con las tecnologías anteriores tiene los mismos problemas con las interferencias como las tecnologías Bluetooth anteriores (párr. 4).

La interferencia dentro de entornos cerrados, se puede decir que si es posible, evitar barreras inalámbricas o cambiar la ubicación de los dispositivos Wi-Fi o Bluetooth para despejar la trayectoria de la señal.

La ubicación del dispositivo dentro de un edificio y los materiales de construcción utilizados pueden afectar a la conexión Wi-Fi y Bluetooth. En la tabla que aparece a continuación se muestran los materiales más comunes que impiden las conexiones y la posibilidad de que ocasionen interferencias.

TABLA 5: Obstrucciones absorbentes y reflectantes de radiofrecuencia.

OBSTÁCULOS	POTENCIAL DE INTERFERENCIA
MADERA	BAJO
MATERIAL SINTÉTICO	BAJO
CRISTAL	BAJO
AGUA	MEDIO
LADRILLOS	MEDIO
MÁRMOL	MEDIO
HORMIGÓN	ALTA
METAL	MUY ALTA

Fuente: Basado en “Wi-Fi y Bluetooth: Posibles Fuentes de Interferencia”, Apple, 2014

2.2.4 MÓDULOS BLUETOOTH

Los módulos de Bluetooth HC-05 y HC-06 son módulos muy populares para aplicaciones con microcontroladores PIC y Arduino. Se trata de dispositivos relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador, incluso sin realizar soldaduras.

2.2.4.1 MÓDULO BLUETOOTH HC-05

El módulo de Bluetooth HC-05 es el que ofrece una mejor relación de precio y características, ya que es un módulo Maestro-Esclavo, quiere decir que además de recibir conexiones desde una PC o tablet, también es capaz de generar conexiones hacia otros dispositivos Bluetooth. Esto nos permite por ejemplo, conectar dos módulos de Bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos. (GeekFactory, 2014)

Para comunicarnos con el módulo y configurarlo, es necesario tener acceso al módulo mediante una interfaz serial. Podemos usar un arduino con un par de cables (aprovechando el puente USB-Serial del Arduino), un kit para XBee o un simple MAX3232 en el puerto serie de la PC



FIGURA 15: Módulo HC-05 conectado a cable serial

Fuente: [https://encrypted-](https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcR0HkppFHLYLPBW6XMX7E5HWvAoPHVsH-ibTCGNYI7SscYXx8WSO)

[tbn3.gstatic.com/images?q=tbn:ANd9GcR0HkppFHLYLPBW6XMX7E5HWvAoPHVsH-ibTCGNYI7SscYXx8WSO](https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcR0HkppFHLYLPBW6XMX7E5HWvAoPHVsH-ibTCGNYI7SscYXx8WSO)

2.2.4.2 MÓDULO BLUETOOTH HC-06

“El HC-06 tiene un firmware distinto y también un funcionamiento distinto en cuanto a su modo de configuración. Para poder configurar el HC-06 es necesario que este NO este emparejado ni siendo usado por ningún dispositivo.” (GeekFactory, 2014)

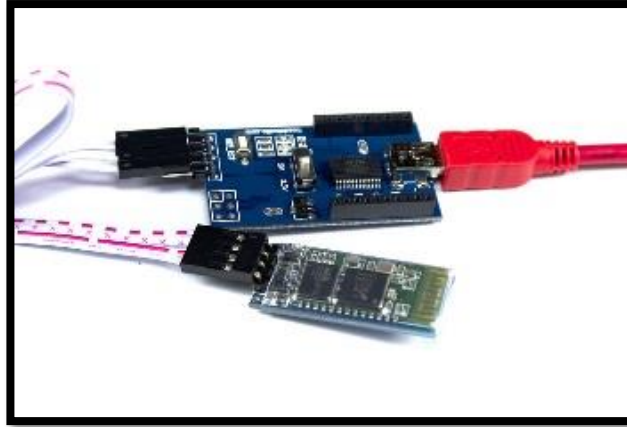


FIGURA 16: Módulo HC-06 conectado a Tarjeta Arduino

Fuente: http://g03.a.alicdn.com/kf/HTB1R_HxlXXXXXXXXZXVXXq6xXFXXu/5pcs-HC-07-Wireless-Serial-4-Pin-font-b-Bluetooth-b-font-RF-Transceiver-font-b.jpg

2.2.5 NORMAS DE TECNOLOGÍA BLUETOOTH

El estándar Bluetooth se divide en múltiples normas:

- IEEE 802.15.1 define Bluetooth 1.x, que puede alcanzar velocidades de 1 Mbps.
- IEEE 802.15.2 recomienda prácticas para utilizar la banda de frecuencia de 2.4 GHz (la frecuencia también utilizada por WiFi).
- IEEE 802.15.3 es un estándar que actualmente se está desarrollando, que ofrecerá velocidad de banda ancha (20 Mbps) con Bluetooth;
- IEEE 802.15.4 es un estándar que actualmente se está desarrollando para el uso con aplicaciones Bluetooth de baja velocidad.

2.2.6 BENEFICIOS DE COMUNICACIÓN BLUETOOTH

Domínguez (2012)

Para que dos dispositivos con tecnología Bluetooth intercambien información precisan la adopción de una contraseña o PIN mutuo. Una vez relacionada la conexión se establece la comunicación de forma automática y sin dificultad. Con una frecuencia de 2,45Ghz (frecuencia libre a nivel mundial), el alcance normal para la comunicación entre dispositivos es de hasta 10 metros. El corto alcance retribuye en sí un sistema adicional de seguridad. Los módulos Bluetooth, son pequeños y compactos, con tan solo 2 0 3 cm, e integran la etapa de control, la etapa de radiofrecuencia y la antena. (p. 107)

2.2.7 DESVENTAJAS DE COMUNICACIÓN BLUETOOTH VS ZIGBEE

“Las desventajas son que las velocidades de transmisión son muy baja, por lo que no es recomendable para transferir grandes cantidades de datos, y que el alcance es muy corto.”
(RODIL, 2010, p.127)

2.2.8 COMPARACIÓN DE TECNOLOGÍAS: WI-FI, BLUETOOTH Y ZIGBEE

En la siguiente tabla se hace referencia a una comparativa entre las distintas tecnologías inalámbricas de transmisión de información entre Bluetooth, Wi-Fi y ZigBee, tomando en cuenta los parámetros más importantes como son su ancho de banda, consumo de potencia, aplicaciones y estándares que los rigen

TABLA 6: Comparación entre tecnologías

DESCRIPCIÓN	BLUETOOTH	WI-FI	ZIGBEE
ANCHO DE BANDA	1 Mbps	De 11MBPS Hasta 54 Mbps	250 Kbps (2.4ghZ) 40 Kbps (915 MHz) 20 Kbps (868 MHz)
POTENCIA	40mA transmitiendo y 0,2 mA en reposo.	400mA transmitiendo y 20mA en reposo.	30mA transmitiendo y 3uA en reposo.
ESTÁNDAR	IEEE 802.15.1	IEEE 802.11	IEEE 802.15.4
COBERTURA	10 metros	15-100 metros	10-75 metros
SEGURIDAD	PIN (número de identificación personal)	No Obligatorio	Servicio de Cifrado y Autenticación de paquetes
NÚMERO DE DISPOSITIVOS	8 Dispositivos conforma una Piconet	32 Dispositivos	Hasta 65000 nodos por red
APLICACIÓN	Transferencia de audio y datos	Navegación por Internet, redes de ordenadores y transferencia de ficheros.	Control remoto, control médico, industrial, domótica e inmótica.
COSTO	Relativamente accesible	Relativamente caro	Relativamente barato

Fuente: Basado en http://www.teleco.com.br/es/tutoriais/es_tutorialzigbee/pagina_4.asp

2.3 SENSORES ELECTRÓNICOS

“Un sensor o captador, como prefiera llamársele, no es más que un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que seamos capaces de cuantificar y manipular.” (Molina, 2014)

De los sensores podemos obtener señales analógicas y transformarlas a señales digitales, pero cada sensor tiene su respectiva aplicación, esto dependerá de varios factores:

Magnitud a medir <ul style="list-style-type: none">— Margen de medida— Resolución— Exactitud deseada— Estabilidad— Ancho de banda— Tiempo de respuesta— Límites absolutos posibles de la magnitud a medir— Magnitudes interferentes	Características de salida <ul style="list-style-type: none">— Sensibilidad— Tipo: tensión, corriente, frecuencia— Forma señal: unipolar, flotante, diferencial— Impedancia— Destino: presentación analógica conversión digital telemedida: ¿tipo?
Características de alimentación <ul style="list-style-type: none">— Tensión— Corriente— Potencia disponible— Frecuencia (si alterna)— Estabilidad	Características ambientales <ul style="list-style-type: none">— Margen de temperaturas— Humedad— Vibraciones— Agentes químicos— ¿Atmósfera explosiva?— Entorno electromagnético
Otros factores	
<ul style="list-style-type: none">— Peso— Dimensiones— Vida media— Coste de adquisición— Disponibilidad— Tiempo de instalación	<ul style="list-style-type: none">— Longitud de cable necesaria— Tipo de conector— Situación en caso de fallo— Coste de verificación— Coste de mantenimiento— Coste de sustitución

FIGURA 17: Factores a considerar en la elección de un sensor

Fuente: <http://www.electronicosonline.com/wp-content/uploads/2012/01/215.png>

“Los sensores se basan en fenómenos físicos o químicos y en materiales donde dichos fenómenos se manifiestan de una forma útil, es decir, con sensibilidad, respetabilidad y especificidad suficientes.” (Pállas, 2004, p. 42)

2.3.1 SENSORES ANALÓGICOS

“Los sensores analógicos proporcionan una señal analógica continua, por ejemplo voltaje o corriente eléctrica. Esta señal puede ser tomada como el valor de la variable física que se mide.” (Pállas, 2004, p. 43)

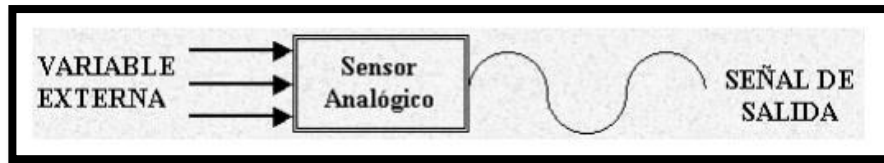


FIGURA 18: Esquema básico de un Sensor Analógico

Recuperado de: http://eie.ucr.ac.cr/uploads/file/proybach/pb_08_II/pb0811t.pdf

Chaves (2008)

El sensor analógico debe poseer ciertas propiedades indispensables como: calibración, rango de funcionamiento, confiabilidad, velocidad de respuesta, exactitud precisión, sensibilidad, linealidad entre otros. Esto con el fin de que el control de la variable que se mida, se lleve a cabo de la mejor manera y en el menor tiempo posible. (p. 9)

2.3.2 SENSORES ÓPTICOS

Bernabeu (2013)

Un sensor óptico se basa en el aprovechamiento de la interacción entre la luz y la materia para determinar las propiedades de ésta. Una mejora de los dispositivos sensores, comprende la utilización de la fibra óptica como elemento de transmisión de la luz.

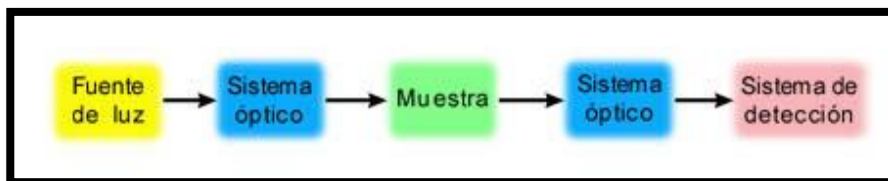


FIGURA 19: Diagrama básico de un sensor óptico

Fuente: <http://www.tecnoficio.com/optica/images/optical%20sensor.jpg>

2.3.3 PULSE SENSOR

El pulso Amped sensor es un sensor de frecuencia cardíaca plug-and-play para Arduino. Puede ser utilizado por estudiantes, artistas, deportistas, fabricantes y desarrolladores de juegos y móviles que desean incorporar fácilmente los datos de la frecuencia cardíaca en vivo en su proyectos.

2.3.3.1 CARACTERÍSTICAS DE PULSE SENSOR

Combina un sensor de frecuencia cardíaca óptica simple con amplificación y circuito de cancelación de ruido por lo que es rápido y fácil de obtener lecturas fiables de pulso. Asimismo, se utiliza solo 4 mA de corriente de 3 a 5V por lo que es ideal para aplicaciones móviles.

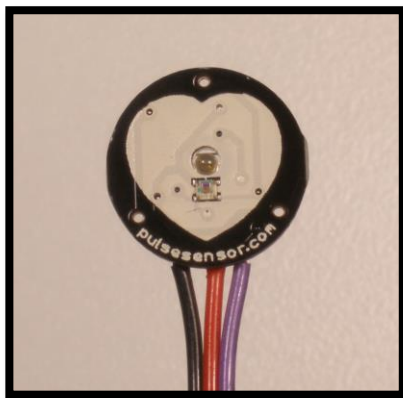


FIGURA 20: Pulse Sensor

Recuperado de: http://cdn.shopify.com/s/files/1/0100/6632/products/PulseSensorAmpedFinger-web_2_1024x1024.jpg?v=1348514131

El sensor de pulso es esencialmente un fotopletismógrafo, que es un dispositivo médico conocido utilizado para el monitoreo de la frecuencia cardíaca no invasiva. La señal de pulso del corazón que sale de un fotopletismógrafo es una fluctuación en el voltaje analógico, y tiene una forma de onda predecible como se muestra en la figura 21. La representación de la onda de pulso se denomina photoplethysmogram, o PPG.

El sensor de pulso, amplifica la señal sin procesar del sensor de pulso anterior, y normaliza la onda de pulso en torno a $V / 2$ (punto medio de la tensión). Este responde a los cambios relativos en la intensidad de la luz.

Si la cantidad de luz incidente sobre el sensor y permanece constante, el valor de la señal permanecerá en o cerca de 512 (punto medio del rango ADC). Más luz y la señal sube, menos luz, todo lo contrario. La luz del LED verde que se refleja de nuevo a los cambios de los sensores durante cada pulso.

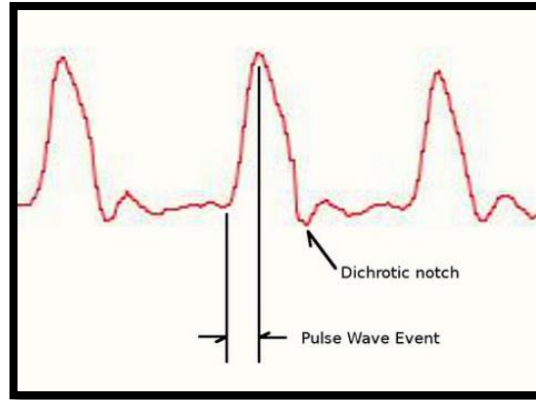


FIGURA 21: Señal de onda del corazón

Fuente:

http://cdn.shopify.com/s/files/1/0100/6632/files/pulseWaveformAmpdVersion_large.jpg?619

Tiene un cable número 24 con conectores macho estándar, así que no hay soldadura requerida. Sus tres pines de colores negro, rojo y violeta, representan tierra, Voltaje (3V o 5V) y la señal de entrada respectivamente.

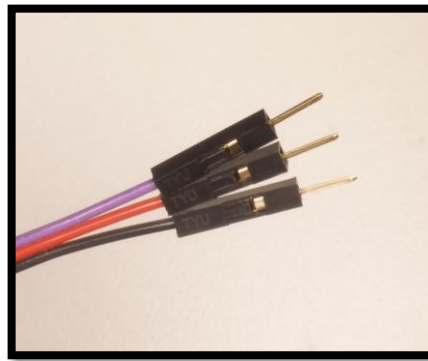


FIGURA 22: Terminales de Pulse Sensor

Fuente: http://cdn.shopify.com/s/files/1/0100/6632/products/PulseSensorAmp-Connectors_1024x1024.jpg?v=1348511609

2.3.3.2 UBICACIÓN DE PULSE SENSOR

Simplemente, sujete el sensor de pulso para el lóbulo de la oreja o un dedo de la punta y conéctelo a sus 3 o 5 voltios de Arduino y ya está listo para leer el ritmo cardíaco. (SparkFun, 2012)

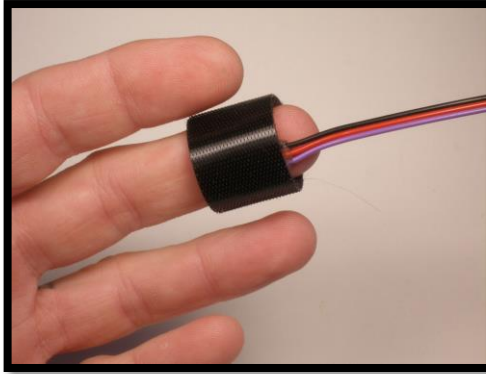


FIGURA 23: Pulse Sensor en el dedo

Recuperado de: http://cdn.shopify.com/s/files/1/0100/6632/products/Amped-Finger-Hand_1024x1024.jpg?v=1348513426

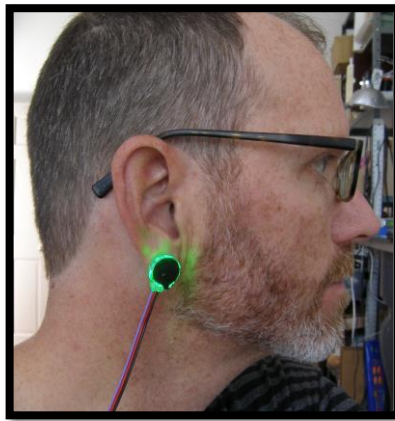


FIGURA 24: Pulse Sensor en la oreja

Recuperado de:

http://cdn.shopify.com/s/files/1/0100/6632/products/earclip_1024x1024.jpg?v=1348511818

2.4 MATERIALES ELECTRÓNICOS ADICIONALES

2.4.1 BATERÍAS LIPO

Para Andoni Días (2010)

Son baterías de Polímero de Litio (LIPO) con una excelente relación entre capacidad, peso, volumen y tensión; esta se conforma de celdas donde cada celda tiene un valor nominal de 3.7 voltios; la nomenclatura de reconocimiento de estas baterías se basa en un número de elementos o celdas seguidas de letras que indican el tipo de conexión de dichos elementos, “S” para conexión en serie y “P” para conexión en paralelo.

Por ejemplo una batería Lipo 3S, significa que contiene 3 celdas en serie por tanto tendrá un voltaje de 11.1 voltios. Estas baterías necesitan de un cargador especial. Su alta capacidad de almacenamiento de voltaje, su bajo peso y sus cortas dimensiones, hacen que este tipo de baterías se adapten de mejor manera para alimentar a prototipos electrónicos, proporcionando una larga duración de tiempo para cuando el circuito está activo.

2.4.1.1 TIEMPO DE DURACIÓN DE UNA BATERÍA

Para saber el tiempo que dura una batería teóricamente, conocida su capacidad y su velocidad de descarga y suponiendo que el consumo que le vamos a exigir es este valor máximo de descarga de forma continúa; tenemos la siguiente relación que nos da una primera aproximación:

$$\text{Tiempo (min)} = \text{Capacidad de la batería (Amp*min)} / \text{Velocidad de descarga (Amp)}$$

Tenemos una batería de 7,4 voltios con capacidad de 5000mA y 10C que es un valor específico que viene en las hojas de datos, es decir que dividiendo entre 1000 para pasar a Amperios, tenemos capacidad de 5A. Estos 5A multiplicados por 60 minutos que tiene 1 hora, nos da 300A*min, que sería la capacidad de la batería.

Por otra parte, la velocidad de descarga hemos dicho que es 10C, es decir, 10*5A que es igual a 50A. Luego el tiempo que nos dura la batería suministrando esta corriente de consumo máxima es:

$$T = 300A*min / 50A = 6 \text{ minutos}$$

Es una corriente máxima teórica que la batería es capaz de suministrar; luego está la corriente que realmente suministra la batería en cada momento en función de lo que solicitemos de ella, ya que la carga o el consumo no es constante todo el tiempo; y además sería inviable que la batería estuviera entregando de forma continua su corriente máxima en todo momento.

Por este motivo, el tiempo calculado con la expresión anterior siempre es menor que lo que realmente suele durar, y además, ese resultado no tiene en cuenta factores como por ejemplo que la batería no debe descargarse completamente por razones de seguridad y aparte existen pérdidas de potencia eléctrica “por el camino” por diferentes elementos del sistema.

Por lo tanto, para saber qué batería elegir, debemos ver primero qué carga tenemos y ver qué corriente va a necesitar durante su funcionamiento. El consumo máximo de la carga debe ser menor que la corriente máxima continua que es capaz de suministrar la batería; y aparte, deberíamos prever cierto margen de seguridad.

Por otra parte también está el tiempo que queremos que nos dure. Con el valor de la corriente anterior y el tiempo de duración estimado, podemos ver qué batería cumple ambos requisitos.

2.4.2 HILO CONDUCTOR



FIGURA 25: Hilo conductor

Fuente: <http://www.electronicaestudio.com/i/f/SP-DEV10867.jpg>

Muñoz (2014) manifiesta: “El hilo conductor es apto para la costura y el desarrollo de textiles electrónicos, con el fin de crear piezas interactivas que reaccionen a estímulos propuestos por el diseñador, ya sea al monitoreo, temperatura, entre otros. La tecnología Arduino facilita la creación de estos textiles” (párr. 3).

Este hilo conductor sirve para conectar los componentes electrónicos sobre la ropa. Puede llevar la corriente y señales. Se puede coser a la ropa y sirve especialmente para proyectos con Arduino LilyPad, adicionar sensores para múltiples usos.

En este elemento vienen 30 piés de hilo conductor hiladas a partir de fibra de acero inoxidable enrollado en una bobina de plástico. Se usa para coser todos sus proyectos textiles. Este hilo se diferencia con los demás porque se forma a partir de fibra de acero inoxidable y no tiene un núcleo de nylon, tiene una resistencia de 28 Ohms/Ft.

2.4.3 MINI INTERRUPTOR

Un interruptor eléctrico es un dispositivo que permite desviar o interrumpir el curso de una corriente eléctrica. Su expresión más sencilla consiste en dos contactos de metal inoxidable y el actuante.

Los contactos, normalmente separados, se unen mediante un actuante para permitir que la corriente circule. El actuante es la parte móvil que en una de sus posiciones hace presión sobre los contactos para mantenerlos unidos.



FIGURA 26: Mini interruptor

Fuente: <http://www.g7electronica.net/320-635-large/interruptor-mini-on-on.jpg>

2.4.4 CABLE CONDUCTOR

Marquina, Gutiérrez, Fonseca, (2012) manifiestan: “Normalmente son fabricados de un material de cobre, envueltos con un material tipo aislante o protector, el cual permite conducir el flujo de corriente eléctrica desde un punto hacia otro, estos cables son utilizados para las prácticas de laboratorio de electrónica” (p. 2).

Un cable conductor básicamente se compone de cuatro partes muy importantes que son:

- Conductor: “Elemento que conduce la corriente eléctrica y puede ser de diversos materiales metálicos como el cobre o aluminio. Puede estar formado por uno o varios hilos” (Marquina, Gutiérrez, Fonseca, 2012, p. 2).
- Aislamiento: Recubrimiento que envuelve al conductor, para evitar la circulación de corriente eléctrica fuera del mismo.
- Capa de relleno: Material aislante que envuelve a los conductores para mantener la sección circular del conjunto.
- Cubierta: Está hecha de “materiales que protejan mecánicamente al cable. Tiene como función proteger el aislamiento de los conductores de la acción de la temperatura, sol, lluvia, polvo” (Marquina, Gutiérrez, Fonseca, 2012, p. 2), entre otros.

2.5 DISPOSITIVOS MÓVILES

“Los dispositivos móviles son pequeños para poder portarse y ser fácilmente empleados durante su transporte. En muchas ocasiones pueden ser sincronizados con algún sistema de la computadora para actualizar aplicaciones y datos.” (Guevara, 2010)

Guevara (2010)

Otra característica es el que se pueda conectar a una red inalámbrica, por ejemplo, un teléfono móvil, los comunicadores de bolsillos o PDAs¹². Este tipo de dispositivos se comportan como si estuvieran directamente conectados a una red mediante un cable, dando la impresión al usuario que los datos están almacenados en el propio dispositivo.

2.5.1 SMARTPHONES (TELÉFONOS INTELIGENTES)

Rittinghouse (2010)

La definición de un teléfono inteligente no está estandarizado y varía dependiendo de a quién le pregunte. Para la mayoría de los usuarios, el consenso es que un teléfono inteligente es un dispositivo móvil que ofrece capacidades avanzadas más allá de los ofrecidos por un teléfono móvil normal. (p.236)

2.5.2 APLICACIONES EN TELÉFONOS INTELIGENTES

“Las aplicaciones de los teléfonos inteligentes pueden ser desarrolladas por el fabricante del dispositivo o por cualquier otro proveedor de terceros o desarrollador capaz de acceder al sistema operativo de código abierto.” (Rittinghouse, 2010, p.236)

En la actualidad se cuentan con dispositivos de última tecnología, los cuales son utilizados por el usuario de tal manera que puedan interactuar con el equipo móvil de una forma sencilla.

2.6 FUNDAMENTOS DE SISTEMA OPERATIVO ANDROID

Rittinghouse (2010)

¹² PDA: Personal Digital Assistant

Android es una plataforma de software y el sistema operativo para dispositivos móviles que se basa en el kernel de Linux. Fue desarrollado originalmente por Google y más tarde con el Open Handset Alliance. El sistema operativo Android es la primera plataforma móvil completa, abierta y libre. (p.238)

2.6.1 LAS APLICACIONES EN ANDROID

Rittinghouse (2010)

Android permite a los desarrolladores escribir código administrado de Java para el control de un dispositivo móvil. Los desarrolladores pueden distribuir sus aplicaciones a los usuarios de teléfonos móviles Android. Hay un mercado llamado Android Market que permite a los desarrolladores publicar y distribuir fácilmente sus aplicaciones directamente a los usuarios de teléfonos compatibles con Android. (p. 238)

Una aplicación en el sistema operativo Android, permite brindar soluciones de problemas puntuales a los usuarios. Se puede realizar una gran variedad de aplicaciones de diferentes tipos, esto dependerá de la solución que se quiera ofrecer. Cada aplicación tendrá su nivel de complejidad, y es necesario que el programador conozca el diferente software que puede utilizar para obtener procesos eficaces en la solución

La aplicación que se desarrolla aparece en el teléfono paso a paso, a medida que añaden piezas a la misma, de modo que se puede probar el trabajo mientras se elabora la misma. Una vez terminado, puede empaquetar su programa y producir una aplicación independiente para instalar.

Aunque no se disponga de un teléfono con Android, siempre se pueden construir aplicaciones utilizando el emulador de Android, el software que se ejecuta en el ordenador y se comporta igual que el teléfono respecto a la misma.

2.6.1.2 APP STORE PARA ANDROID

El surgimiento de las Apps Stores en el año de 2008 terminó de impulsar el éxito de las aplicaciones móviles y un significativo cambio en la manera en que se distribuye y comercializa el software.

Cuando la App Store abrió contaba con 500 aplicaciones y Android Market (Google Play) con 50; ahora en 2014 la App Store tiene 775.000 y Google Play 800.000 aplicaciones cada una con una función o funciones que aprovechan las características del teléfono.

Martínez (2013) manifiesta: “Hoy en la actualidad como datos se tiene que el mayor tráfico de datos registrado se obtiene de dispositivos móviles más que de computadoras, la tendencia de varias empresas migran a tener sus páginas Web adaptadas a teléfonos móviles para ser descargadas mediante una aplicación” (párr. 10).

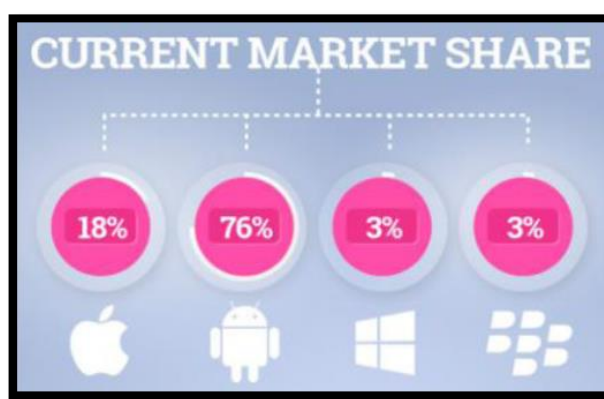


FIGURA 27: Porcentajes de uso de Sistema Operativos para aplicaciones móviles

Fuente: <http://upsasoyyo.wordpress.com/2013/09/17/aplicaciones-moviles-la-evolucion/>

Se puede decir que la mayor preferencia del mercado corresponde a los sistemas operativos Android y Apple que abarcan más del 90% del mercado de desarrollo de aplicaciones móviles.

2.6.2 CÓDIGO EN ANDROID

Rittinghouse (2010)

Google ha lanzado ahora la mayor parte del código de Android bajo la licencia Apache, un software libre y la licencia de código abierto. Los desarrolladores de Android tienen pleno acceso a la misma aplicación marco interfaces de programación (API¹³) utilizados por las aplicaciones principales. La arquitectura está diseñada para simplificar la reutilización de componentes, por lo que cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede entonces hacer uso de esas capacidades. (p. 239)

¹³ Interfaz de programación de aplicaciones

Para implementar las aplicaciones en el sistema Android, existen varios programas que ayudan a la programación del mismo, entre ellos unos de los más utilizados y de mayor importancia por su simplicidad de utilización de lenguaje es “App Inventor”

2.6.3 APP INVENTOR (PROGRAMACIÓN PARA ANDROID)

App Inventor permite a cualquier usuario crear sus propias aplicaciones para sus Smartphones de una manera sencilla.

“Inventor MIT App es una herramienta de programación basada en los bloques que permite que todos, incluso los principiantes, para iniciar la programación y construir aplicaciones totalmente funcionales para los dispositivos Android.” (MIT App Inventor, 2012)

“App Inventor funciona bajo una interfaz WYSIWYG¹⁴, que te permite ver automáticamente en pantalla los resultados de tu aplicación. Uniendo una serie de bloques el usuario podrá crear aplicaciones para Android.” (Kiosera, 2014)

“Es posible unir funcionalidades simples como un rompecabezas: botones, zonas de texto, imágenes, audio, localización, selección de contactos o números telefónicos, acceso a una mini base de datos, lector de código de barras, etc.” (Kiosera, 2014)

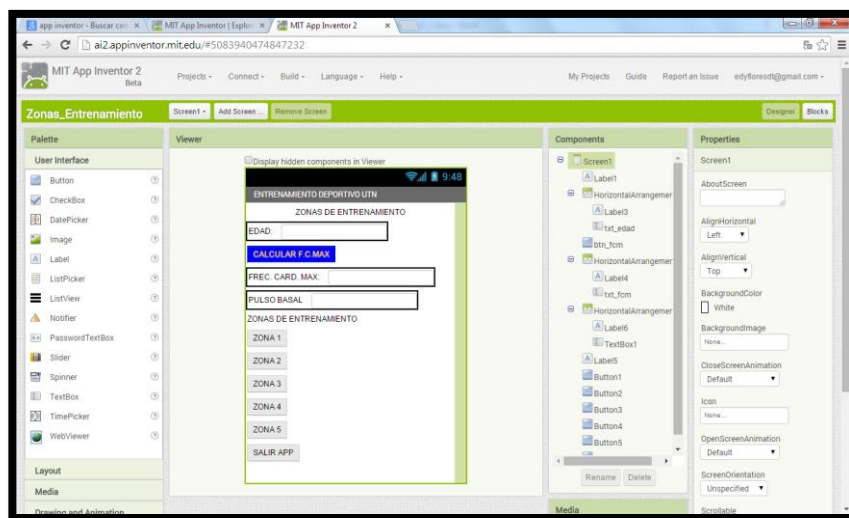


FIGURA 28: Pantalla inicial del software App Inventor

Fuente: Edison Flores

Además, este software dispone de un emulador que permite simular un Smartphone o teléfono móvil. Esto permite crear una aplicación sin la necesidad de tener el teléfono móvil a la mano.

¹⁴ What you see is what you get

2.6.3.1 PROGRAMACIÓN EN BLOQUES

Los Bloques de App Inventor (App Inventor Blocks Editor), donde se ensamblan los bloques del programa para especificar cómo deben comportarse sus componentes. Los programas se desarrollan visualmente, ensamblando piezas que se juntan encajando como las de un rompecabezas. (MIT App Inventor, 2012)

2.6.3.2 REQUISITOS DEL SISTEMA

Para utilizar la herramienta de diseño y entorno de desarrollo de aplicaciones App Inventor, un ordenador debe cumplir con los siguientes requisitos en el sistema operativo, que se observa en la siguiente tabla.

TABLA 7: Requisitos App Inventor para ordenadores

EQUIPO	SISTEMA OPERATIVO
Windows	Windows XP, Windows Vista, Windows7, y Windows 8
GNU/Linux	Versiones Ubuntu 8.0+ y Debian 5.0+
Macintosh	Mac OS X 10.5, 10.6

Fuente: Basado en

<http://s445852199.mialojamiento.es/peponees/appinventorspanish/learn/whatis/requirements.htm>

|

Para ingresar a la aplicación App Inventor es necesario tener instalado un navegador web compatible en nuestro ordenador, que se observa en la siguiente tabla.

TABLA 8: Compatibilidad App Inventor para Navegadores

NAVEGADOR	VERSIONES
Mozilla Firefox	3.6 o superior
Apple Safari	5.0 o superior
Google Chrome	4.0 o superior
Microsoft Internet Explore	7.0 o superior

Fuente: Basado en

<http://s445852199.mialojamiento.es/peponees/appinventorspanish/learn/whatis/requirements.htm>

|

Para desarrollar aplicaciones en Smartphone o tablets es necesario verificar la compatibilidad de sistema operativo Android, donde como principales fabricantes se pueden observar en la siguiente tabla.

TABLA 9: Compatibilidad App Inventor para Smartphones y tablets

FABRICANTE	VERSIONES
Samsung	Toda la gama Galaxy Ace, Duos, Plus, Alpha, Apollo, Core, Frame, Gio, Mini, Nexus, Note, Player, Pocket, S2, S3, S4, S5, Tab, Young, GT, entre otros.
Huawei	Ascend D, Astro, Express, Etisalat, G, GS, H, MediaPad, ICE, MegaFon, MTC, T-Mobile, Youth, entre otros.
HTC	Toda la gama Aria, Chacha, Butterfly, Desire, Droid DNA, Droid Eris, Nexus, My Touch, Sensation, entre otros.
TCT Alcatel	Toda la gama IDOL, POP, ONE TOUCH, Pixo, Soul, TCL, VF, EVO, Flyer, Decire, Explorer, Hero, ONE, entre otros.
Sony Ericsson	Toda la gama, Xperia, Tablet S, Tablet P, entre otros.

Fuente: Basado en

<http://s445852199.mialojamiento.es/peponees/appinventorspanish/learn/whatis/requirements.htm>

|

“Independientemente del modelo de teléfono que sea, deberá tener una tarjeta SD¹⁵ instalada o no funcionará. App Inventor utiliza la tarjeta SD para almacenar todos los archivos multimedia sonido, imagen y vídeo” (Rederjo, 2013, párr. 20); o también él es un requisito de Smartphone tener memoria RAM como mínimo de 250 MB.

¹⁵ Tarjeta SD = Tarjeta Digital Segura, es un formato de tarjeta de memoria usada para dispositivos portátiles como cámaras fotográficas, teléfonos móviles, videoconsolas, computadoras portátiles, entre otros.

2.6.3.3 FUNCIONAMIENTO

“App Inventor se basa en un lenguaje de programación por bloques” (Sánchez, 2014, p. 2), esto permite de una u otra manera facilitar el desarrollo de aplicaciones móviles, es decir no se necesita escribir ninguna línea de programación.

Rederejo (2013) manifiesta: “Se colocan bloques para construir bucles, condiciones, variables, eventos, entre otros que permiten pensar lógicamente y solucionar los problemas de forma ordenada” (párr. 7).

Permite reducir el tiempo en encontrar el punto y coma o los dos puntos que están donde no deben y producen errores de compilación o ejecución; tampoco será necesario corregir estos errores debido a que el propio programa impedirá realizar funciones no declaradas.

El diagrama de funcionamiento de App Inventor se divide en tres componentes básicos que son la parte del diseñador, el editor de bloques y el emulador o dispositivo Android.

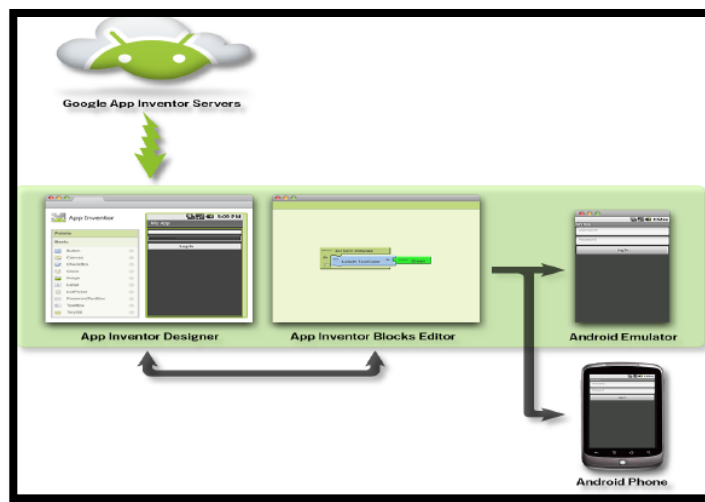


FIGURA 29: Diagrama de Funcionamiento de App Inventor

Fuente: <http://appinventor.mit.edu/explore/content/what-app-inventor.html>

2.6.3.4 DISEÑADOR

Es la pantalla principal del entorno de desarrollo App Inventor en modo gráfico, aquí el usuario podrá hacer uso de todas las opciones que a continuación se detallan.

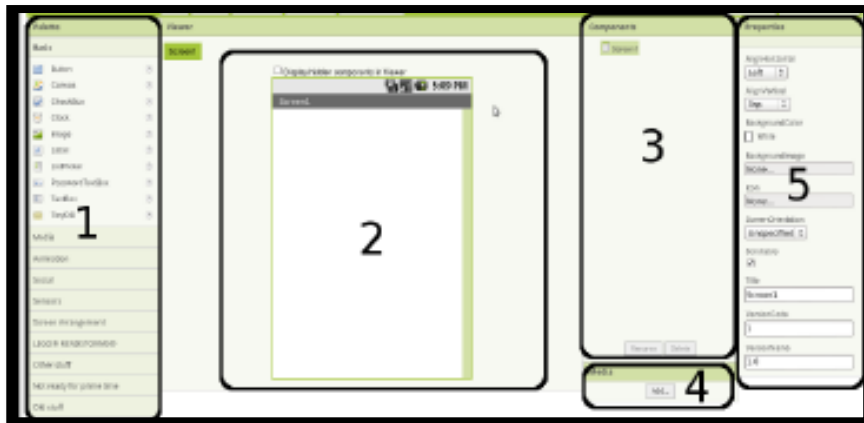


FIGURA 30: Pantalla de Diseño de la aplicación en App Inventor

Fuente: <http://recursostic.educacion.es/observatorio/web/software/programacion/1090>

1. Paleta.- Contiene todos los elementos que pueden ser arrastrados o insertados en nuestra aplicación, existen elementos gráficos como cuadros de texto, botones, dibujos; también elementos no visibles en la pantalla principal como acelerómetro, cámara de video, y bases de datos.

2. Visor de Pantalla.- Permite simular la apariencia visual que tendrá la aplicación móvil desarrollada por el usuario. Los elementos deben ser insertados desde la paleta hacia el visor de pantalla.

3. Componentes.- Muestra una lista de todos los elementos colocados desde la paleta hacia el visor de pantalla.

4. Media.- Muestra las imágenes y sonidos que el usuario cargue y estarán disponibles para usarlas en la aplicación.

5. Propiedades.- Primero hay que seleccionar los elementos arrastrados al visor de pantalla para lograr observar las propiedades de los mismos, como longitud, ancho, fuente, color, entre otros, dependiendo del tipo de elemento seleccionado.

2.6.3.5 APLICACIÓN PARA SMARTPHONE ANDROID: MIT AI2 COMPANION

Moreno (2013) manifiesta: “El Companion App Inventor cumple la función de ver el desarrollo de una aplicación a medida que el usuario va diseñándola. “MIT App Inventor oficialmente es una herramienta para desarrollo en línea por lo que requiere conexión permanente a Internet” (p. 7).

Es una aplicación que puede ser descargada mediante un Smartphone en la App Store de Android, la cual permite desarrollar aplicaciones para celulares que utilicen este sistema operativo.

Es importante saber que para desarrollar una aplicación móvil mediante Mit AI2 Companion, es necesario crear una cuenta de usuario con GMAIL¹⁶ para iniciar sesión y proceder tanto a la descarga de Mit AI2 Companion en Smartphone como del seguimiento de desarrollo de la aplicación.

Existen tres opciones que ofrece App Inventor para dar el seguimiento o también llamado “Testing en vivo” del desarrollo de aplicaciones móviles que son:

- Instalar la aplicación MIT AI2 Companion en un dispositivo móvil Android conectado a una red Internet, para comenzar la creación de aplicaciones sin necesidad de descargar ningún software en un ordenador.



FIGURA 31: Aplicación MIT y dispositivo móvil

Fuente: <http://appinventor.mit.edu/explore/sites/all/files/SetupAI2/WifiA.png>

- En caso de no tener un dispositivo móvil Android, se puede instalar el software en un ordenador para que pueda utilizar el emulador de Android en la pantalla del mismo.



FIGURA 32: Aplicación MIT y Ordenador

Fuente: <http://appinventor.mit.edu/explore/sites/all/files/SetupAI2/WifiA.png>

¹⁶ GMAIL= Llamado también Google Mail, es un servicio de correo al usuario.

- Si no se tiene una conexión inalámbrica a Internet, se tendrá que instalar “el software en el ordenador de modo que se pueda conectar a un dispositivo Android a través de un cable de datos USB”

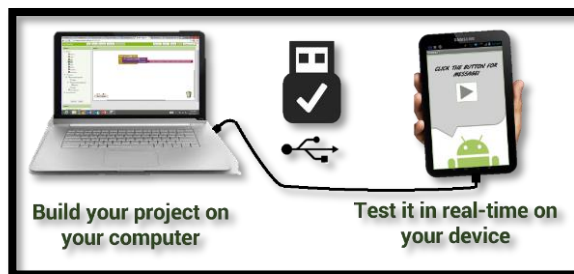


FIGURA 33: Aplicación MIT y dispositivo móvil mediante USB

Fuente:

http://www.frangoal.com/tecno/tecnologia/opcionales/informatica_4/mSCHOOLS/appinventor/PRACTICA4_files/USB.png

2.7 FUNDAMENTOS DE ENTRENAMIENTO DEPORTIVO

“El entrenamiento deportivo es un proceso que busca un grado más o menos acentuado de mejora en los objetivos de cada momento.” (Weineck, 2005, p. 15)

Carl (1985, citado en Weineck, 2005) manifiesta: “El entrenamiento deportivo es un proceso de acciones complejas cuyo propósito es incidir de forma planificada y objetiva sobre el rendimiento deportivo y sobre las capacidades de presentar de forma óptima los rendimientos en situaciones de afirmación personal. (p.15)”

“De acuerdo con las diferentes metas y niveles de desarrollo, el entrenamiento que puede llevar a la práctica dentro de diferentes tipos, como por ejemplo el entrenamiento de alto rendimiento, fitness, de rehabilitación, de técnica, juvenil o infantil.” (Weineck, 2005, p. 15)

Hohmann (2005)

El entrenamiento está abierto para todos, tanto para el principiante como para el experto y, naturalmente, para el deportista de alto rendimiento, para el escolar, para el joven, para la persona activa y para el deportista de mayor edad, para quien quiera aumentar su rendimiento, para aquel que quiera mantener su forma física y también para aquel que quiera recuperarla. (p. 16)

2.7.1 PRINCIPIOS DEL ENTRENAMIENTO DEPORTIVO

“Los principios del entrenamiento deportivo sirven para optimizar la capacidad de acción de deportistas y entrenadores.” (Weineck, 2005, p. 25)

2.7.2 PRINCIPIO DE LA CARGA INDIVIDUALIZADA

Weineck (2005)

Con el principio de la carga individualizada se busca que los estímulos de entrenamiento se correspondan con la capacidad de carga psicofísica, con la tolerancia individual y con las necesidades de cada deportista. Un estímulo de entrenamiento objetivamente igual puede suponer una exigencia escasa para un deportista y excesiva para otro. (p.26)

También llamado principio de la individualización, es uno de los más importantes en el desarrollo del entrenamiento o también en la práctica de la actividad física; este hace referencia a las diferencias fisiológicas, psíquicas, entre otras; haciendo ver a cada individuo como un ser humano único, que debe tener un entrenamiento especializado e individual.

2.7.3 PREPARACIÓN FÍSICA

“La preparación física es uno de los componentes primordiales del entrenamiento deportivo para desarrollar las cualidades motoras: fuerza, resistencia, velocidad, flexibilidad, coordinación.” (Platonov, 2001, p.10)

2.7.3.1 BENEFICIOS CARDIOVASCULARES DEL ENTRENAMIENTO FÍSICO

Salinas (2005), manifiesta:

- Disminuye la frecuencia cardíaca en reposo.
- Disminuye la presión arterial en reposo.
- Aumenta la hemoglobina total en el cuerpo y el volumen de sangre.
- Aumenta la fuerza de bombeo del corazón.
- Aumenta el tamaño del corazón (su cavidad).

- Aumenta el volumen de sangre que bombea el corazón hacia los tejidos.
- Reduce las necesidades energéticas para el corazón.
- Aumento del número y tamaño de los vasos sanguíneos en los músculos, y la extracción de más oxígeno.
- Aumenta el flujo de sangre a través de los músculos esqueléticos. (p. 183)

2.7.3.2 BENEFICIOS RESPIRATORIOS DEL ENTRENAMIENTO FÍSICO

Salinas (2005), manifiesta:

- Fortalece y agranda los pulmones.
- Mejora la capacidad del cuerpo para utilizar oxígeno.
- Se respira más profundo y se pierde menos oxígeno en el aire al exhalar.
- Aumenta el suministro de sangre hacia los pulmones. (p. 183)

2.7.3.3 BENEFICIOS ÓSEONEUROMUSCULARES DEL ENTRENAMIENTO FÍSICO

Salinas (2005), manifiesta:

- Aumenta el tamaño de las fibras musculares (hipertrofia muscular).
- Los músculos están más fuertes y tonificados.
- La resistencia muscular aumenta.
- Ayuda a prevenir la pérdida de flexibilidad a través de los años.
- Los ligamentos y tendones aumentan de grosor y se fortalecen.
- Aumenta los depósitos de calcio y otros minerales en el hueso, evitando la osteoporosis o pérdida de la masa ósea. (p. 183)

2.7.4 CAPACIDADES FÍSICAS

“Las capacidades físicas o motoras. Son capacidades necesarias para el buen desempeño físico, determinantes para el aprendizaje y la ejecución de los movimientos deportivos con sus respectivos ejercicios y se dividen en condicionales (fuerza, resistencia, velocidad, flexibilidad) y coordinativas.” (Guimares, 2002, p.62)

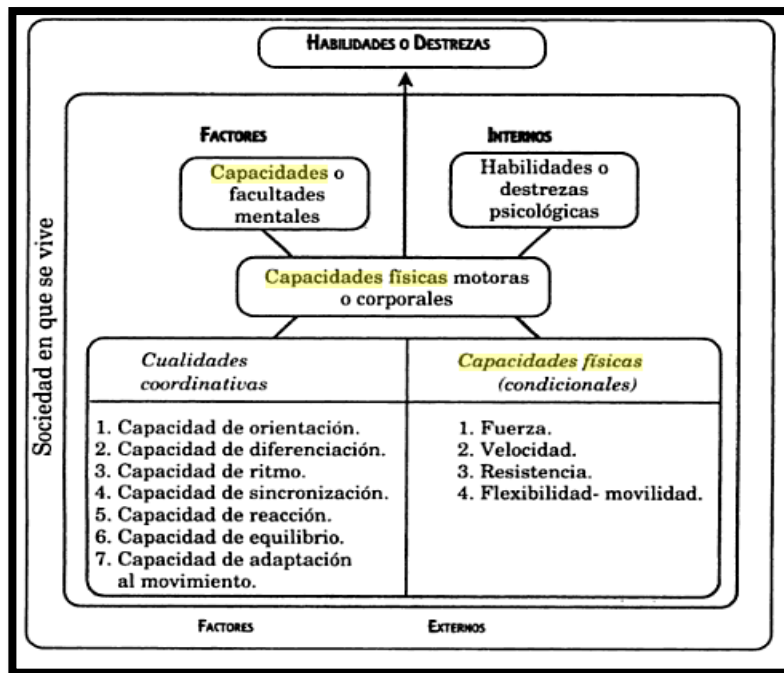


FIGURA 34: Clasificación de Capacidades Físicas

Fuente: (Guimares). Pág 62

2.7.5 CAPACIDAD FÍSICA: RESISTENCIA AERÓBICA

Bompa (2005)

La resistencia, es la capacidad de mantener actividad física durante largos periodos, es importante para deportes con una duración superior a un minuto. La resistencia no es solo una capacidad de corredores de larga duración. Una buena base de resistencia es necesaria para la mayoría de los deportistas, desde el baloncesto al fútbol o el triatlón. El principal beneficio de la resistencia para la mayoría de deportes es soportar la tensión del entrenamiento y la competición. Además un deportista con una buena base de resistencia supera la fatiga del entrenamiento, el trabajo escolar, el estilo de vida estresante más fácilmente que si no tuviera esta base. En el deporte, la fatiga es el enemigo público número uno; aquellos deportistas que no pueden superar de manera efectiva la fatiga tienen una alta probabilidad de rendir mal en el juego, carrera o partido. La fatiga afecta además a la capacidad de estar centrado, lo que da como resultado en errores técnico-tácticos y lanzamiento o disparos imprecisos. Esto explica por qué, al final del partido o competición, se hacen más visibles los errores. Si quiere mejorar su juego, debe desarrollar su capacidad física de resistencia. (p. 189)

La resistencia es una capacidad física importante en todos los deportes, permitiendo al individuo que la entrena, soportar resistencia a la fatiga, además de muchos beneficios que brinda a la salud y complemento físico para otras capacidades. Algunos preparadores físicos la catalogan muy importante para deportistas de alto rendimiento y amateurs.

2.7.5.1 BENEFICIOS DE LA RESISTENCIA AERÓBICA

Bompa (2005)

El entrenamiento de resistencia también ofrece beneficios visibles en la salud. Los sujetos implicados en actividades de resistencia tienen corazón más fuerte, y tanto frecuencias cardiacas como presiones sanguíneas más bajas. Pocos de estos sujetos experimentan enfermedades cardiorrespiratorias, y la incidencia de muertos como consecuencia de enfermedades cardiacas es rara. Los deportistas que realizan actividades de resistencia emplean más grasa como combustible, por lo tanto, parecen más flacos. Haga de las actividades de resistencia una parte de su estilo de vida. (p. 189)

La resistencia aeróbica, no solo brinda beneficios para un deportista de alto rendimiento o uno que empieza a formarse diariamente; sino para todas las personas, causando en aspectos positivos en su salud, por ejemplo, reduciendo el porcentaje de las enfermedades cardiorrespiratorias. Muchas personas ya hacen de esta capacidad física un estilo de vida diario.

2.7.5.2 OBJETIVOS DEL ENTRENAMIENTO AERÓBICO

Conde (2000)

Los objetivos del entrenamiento resistencia aeróbica son: Servirá de base para la construcción o incremento de las demás capacidades. Previene la disminución de un bajón físico en el rendimiento. Para aumentar la capacidad de recuperación de un ejercicio de alta intensidad (favorece la resistencia del lactato). Minimizar el riesgo de lesiones (ya que muchas de estas de estas están asociadas a la fatiga). Para aumentar los niveles de resistencia Psíquica (mejora de las endorfinas, mejor control del dolor). Estabilización de la salud en los individuos. Mejora de distintos parámetros fisiológicos como: Aumento del volumen sistólico, aumento de capilares, aumento de glóbulos rojos, mejor difusión de O₂ a los tejidos, aumento

de la eficiencia cardiaca, disminución de frecuencia respiratoria en reposo, mejora el metabolismo. (p. 22)

2.7.6 MÉTODOS DE ENTRENAMIENTO DE LA RESISTENCIA AERÓBICA

Para el entrenamiento de la resistencia rigen dos principios: el principio continuo y el principio fraccionado.

El principio continuo o de duración es cuando se realiza una actividad sin interrupciones, de comienzo a fin. Es decir, que se inicia el entrenamiento, se desarrolla y finaliza sin interrupciones en el mismo, sin pausas.

El principio fraccionado o intervalado es cuando el entrenamiento se divide, por partes, con pausas intermedia, entre un estímulo y otro.

2.7.6.1 MÉTODOS CONTINUOS

“Se caracterizan por la aplicación de una carga de trabajo, sin interrupciones, sin pausa o períodos de descanso durante el trabajo. Se caracteriza, por tener un inicio, un desarrollo y un final, sin intervalos.” (Mazzeo, 2009)

La duración del trabajo suele ser prolongada y el efecto del entrenamiento se basa primordialmente en ello, durante lo cual se generan constantemente adaptaciones fisiológicas a nivel cardio-circulatorio-respiratorio.

Con este método tendemos a mejorar la capacidad y/o la potencia aeróbica. Además se consiguen ejecuciones más económicas de movimiento y mejoras funcionales en los sistemas orgánicos. Dentro del método continuo se puede realizar el entrenamiento de dos formas distintas:

- Método Continuo Uniforme.
- Método Continuo Variable.

2.7.6.1.1 MÉTODO CONTINUO UNIFORME

Se caracteriza por un gran volumen de trabajo, pero sin interrupciones. Este produce mejoras en la capacidad y potencia aeróbica.

Maszzeo (2009)

Generalmente se trabaja en terrenos llanos, con pocas ondulaciones, de manera de que la intensidad sea uniforme. El terreno preferentemente llano, es para evitar modificaciones en la cadencia de la carrera, los cambios en el consumo de O₂ y en las pulsaciones. Recordamos que cuando decimos continuo, nos referimos a un trabajo que no tiene pausas.

2.7.6.1.2 MÉTODO CONTINUO VARIABLE

Maszeo (2009)

Se caracteriza por la variación de la intensidad durante la realización del trabajo, que es sin interrupciones, pero convenimos que esta variación de velocidad (o sea la intensidad del trabajo) es preestablecida y está hecha en base a la estimulación de diferentes áreas aeróbicas con predominio de una.

Las variaciones tienen que ver con un cambio en la velocidad de desplazamiento (aumentándola o disminuyéndola). A estas las podemos determinar por tiempo o por distancia de trabajo, siendo decididos estos parámetros por el entrenador. Cuando las variaciones en la intensidad no son determinadas por el entrenador, sino por los accidentes que va presentando el terreno, o bien por la voluntad del entrenado, estamos en presencia del método Fartlek.

Como se puede observar con esta simple explicación, el método Fartlek se encuentra dentro de la metodología continua variable, con la variante que depende en gran medida de las decisiones del deportista y del terreno donde se realiza el trabajo.

2.7.6.2 MÉTODOS FRACCIONADOS

En este método de trabajo para el entrenamiento aeróbico, se intercalan las fases de trabajo con las fases de descanso.

“En este método la tarea a realizar se divide en trabajo + pausa, o fases de carga y descanso. Así este método se organiza dividiendo el volumen de trabajo total a realizar en partes o fracciones, entre las cuales existen períodos de descanso.” (Maszeo, 2009)

2.7.7 CAPACIDAD FÍSICA: RESISTENCIA ANAERÓBICA

Conde (2000) manifiesta: “Es la capacidad del organismo para mantener el mayor tiempo posible una actividad de alta intensidad, y donde la producción de energía con una deuda de oxígeno muy importante.” (p. 24)

Son ejercicios de alta intensidad y de poca duración. Aquí no se necesita oxígeno porque la energía proviene de fuentes inmediatas que no necesitan ser oxidadas por el oxígeno, como son el ATP muscular, la PC o fosfocreatina y la glucosa. Ejemplos de ejercicios anaeróbicos son: hacer pesas, carreras de velocidad y ejercicios que requieran gran esfuerzo en poco tiempo.

2.7.7.1 OBJETIVOS DEL ENTRENAMIENTO ANAERÓBICO

Esta capacidad tiene por objetivo

- Aumentar la capacidad de llevar a cabo manifestaciones de fuerza y velocidad de forma continuada.
- Retrasar al máximo la aparición de factores limitantes del ejercicio, como la acidez producida por el incremento de la concentración de lactato. (Conde, 2000, p. 24)

2.7.7.2 BENEFICIOS DE LA RESISTENCIA ANAERÓBICA

Los principales beneficios que puede aportar a tu salud practicar ejercicio aeróbico son:

- Desarrolla masa muscular y fortalece los músculos.
- Mejora la capacidad para combatir la fatiga.
- Hace trabajar al corazón y al sistema circulatorio e incrementa la cantidad de oxígeno que se puede consumir durante el ejercicio y, por lo tanto, mejora el estado cardiorrespiratorio.
- Ayuda a evitar el exceso de grasa y a controlar el peso. (Conde, 2000, p. 26)

2.7.7.3 MÉTODOS DE ENTRENAMIENTO DE RESISTENCIA ANAERÓBICA

Entre los métodos de entrenamiento para el desarrollo de la resistencia anaeróbica se encuentran:

2.7.7.3.1 LA CARRERA CONTINUA

Según el objetivo que buscamos tenemos 3 ritmos:

TABLA 10: Métodos de entrenamiento de carrera continúa

INTENSIDAD	OBJETIVO	FRECUENCIA CARDÍACA
Lento	Entreno de recuperación	130 media (80 - 160)
Medio	Desarrollo básico de la resistencia	150-160
Rápido	Mejora de la potencia aeróbica	160-175

Fuente: Edison Flores

No abusar de la Carrera continua, pues un exceso de este método influirá negativamente en la velocidad del individuo (si es que este fuese el objetivo) ya que otras cosas ocasionaría modificaciones bioquímicas en los músculos. (Conde, 2000, p. 25)

2.7.7.3.2 LA CARRERA CONTINUA

Conde, (2000) manifiesta: “Este método tiene como objetivo el desarrollo de la resistencia aeróbica y anaeróbica. Muy utilizado para deportistas de alto rendimiento en pretemporadas, consiste en cambios de ritmo y de terreno” (p.26)

Sus finalidades son en función del volumen de trabajo (10min a 40 min) y la intensidad de carrera: A mayor intensidad menor volumen y mayor recuperación. Si existen fases muy altas de intensidad el trabajo debe estar entre 10min a 15 min.

2.7.7.3.3 CARRERA CON VARIACIÓN DE VELOCIDAD

Conde, (2000) manifiesta: “Método introducido por Carmelo Bosco para la mejora de la potencia aeróbica. Es muy similar al Fartlek, pero en este caso existen unos intervalos establecidos de antemano donde se alternan las máximas fases de velocidad (190 ppm¹⁷) con períodos de carrera de recuperación (140-150ppm.)” (p.27)

Este método se suele utilizar para deportistas que realicen una pretemporada previa a alguna competencia deportiva; pero como es un método bastante exigente es conveniente realizar con antelación de sesiones con métodos más suaves como la carrea continua y el Fartlek, para luego utilizarlos en períodos de trabajo no superiores a los

¹⁷ ppm: pulsaciones por minuto

8min., a continuación aumentaremos progresivamente el tiempo de ejecución llegando a un máximo de 25min. (Conde, 2000, p. 27)

2.7.7.3.4 INTERVAL TRAINING

Conde, (2000) manifiesta: “Es un método fraccionado que nos permite con la pausa la recuperación de distintos parámetros fisiológicos, que van a hacer posible el realizar más trabajo neto.” (p. 28)

Existen diferentes clasificaciones, las que voy a mencionar están en función del tiempo de ejecución.

- I.T. ¹⁸con intervalos cortos

Frecuencia Cardíaca: 220-edad

Intensidad: 90-95%

Tiempo de ejecución: 20 a 30 segundos

Numero de repeticiones: 3 a 4

Recuperación entre repeticiones: 2 a 3 minutos

Numero de series: 3 a 4

Recuperación entre series: 12 minutos. (Conde, 2000, p. 28)

- I.T. con intervalos medios de ritmo

Intensidad: 70-80%

Frecuencia Cardíaca: 160-170 ppm.

Tiempo de ejecución: de 1 min. Hasta 8 min.

Pausa de repeticiones: 4 a 5 min (Conde, 2000, p. 28)

- I.T. con intervalos de larga duración

Intensidad: 65-75%

¹⁸ I.T.: Inerval Training

Frecuencia Cardíaca: 160-165 ppm.

Numero de repeticiones: 5 a 8

Tiempo de ejecución: de 6 min. Hasta 8 min.

Recuperación entre repeticiones: situar F.C. a + - 120 ppm. (Conde, 2000, p. 28)

2.8 FRECUENCIA CARDÍACA

PEREZPLATA (2011)

Se define la frecuencia cardiaca como las veces que el corazón realiza el ciclo completo de llenado y vaciado de sus cámaras en un determinado tiempo. Por comodidad se expresa siempre en contracciones por minuto, ya que cuando nos tomamos el pulso lo que notamos es la contracción del corazón (sístole), es decir cuando expulsa la sangre hacia el resto del cuerpo. (...) El número de contracciones por minuto está en función de muchos aspectos y por esto y por la rapidez y sencillez del control de la frecuencia hace que sea de una gran utilidad, tanto para médicos, como para entrenadores y como no, para aficionados al deporte o deportistas profesionales.

2.8.1 FACTORES QUE AFECTAN A LA FRECUENCIA CARDÍACA

Existen varios factores internos y externos que afectan a la frecuencia cardíaca de cada persona, entre ellos están:

2.8.2 FACTORES INTERNOS EN LA F.C

Este tipo de factores, son producidos o tiene efectos por causas de características propias de cada individuo

2.8.2.1 LA EDAD

La frecuencia basal (la mínima) más alta la tenemos nada más nacer, desde ese momento va descendiendo con la edad. Sobre la frecuencia máxima los pre-puberales más que los adolescentes y estos menos que los adultos. La frecuencia máxima más alta se alcanza entre los 8 y 10 años. Algunos estudios afirman que la mayor diferencia entre la basal y la máxima se alcanza después de la pubertad y esta diferencia va disminuyendo con la edad.

2.8.2.1.1 LA GENÉTICA

Afecta en gran medida a todos los aspectos de las pulsaciones por minuto, afecta tanto a las pulsaciones en reposo, como a las máxima o como al rango aeróbico de funcionamiento. Estos valores son muy entrenables pero la progresión de estos también estará en gran medida dictados por la genética. También algunos aspectos dictados por la genética como la talla, o el género afectan a la frecuencia.

2.8.2.1.2 EL GÉNERO

Las mujeres por término medio tienen entre 5 y 15 pulsaciones más por minuto que los hombres.

2.8.2.1.3 SOMATOTIPO O COMPOSICIÓN CORPORAL

La persona más alta tiene las pulsaciones más bajas que los más bajos y los delgados menos que los gordos. Los musculados más que los no musculados. En este último apartado quiero señalar que me refiero a musculados de forma natural.

2.8.2.1.4 LAS PSICOLÓGICAS

Los estados que aumentan la sensación de alerta, como los nervios, la ansiedad, el miedo, el amor o la excitación sexual aumentan las pulsaciones, en algunos casos pudiendo llegar al máximo sin actividad física paralela. Y por el contrario los estados que rebajan el nivel de alerta también rebajan las pulsaciones por minuto, estos estados pueden ser el sueño, la relajación, la satisfacción o la calma.

2.8.2.1.5 LA POSTURA

Acostados es cómo podemos obtener la más baja frecuencia y bípedos (de pie) la más alta. Esta diferencia entre las pulsaciones que un sujeto tiene acostado y las que tiene de pie, es una forma rápida y fiable de ver el estado de forma de ese individuo. Cuanta más alta sea la diferencia menos preparación física tendrá el sujeto.

2.8.2.1.6 EL METABOLISMO

El metabolismo propio de cada persona afecta a su frecuencia cardiaca basal o a su frecuencia cardiaca en reposo y también a la frecuencia cardiaca máxima.

2.8.2.1.7 EL CONTROL MENTAL

Algunos maestros del yoga consiguen controlar el ritmo cardiaco mediante la concentración. Normalmente dentro de ciertos límites gracias a la relajación o al aumento del stress mediante la concentración.

2.8.2.1.8 MEDICAMENTOS

Algunos medicamentos pueden alterar las pulsaciones normales, ya sea al alza o a la baja. Normalmente psicodpresores suelen bajar las pulsaciones son medicamentos como la benzodiacepina. Al contrario los estimulantes o los llamados psicoestimulantes como ejemplo la anfetamina.

2.8.2.2 FACTORES EXTERNOS EN LA F.C

2.8.2.2.1 HORARIO DEL DÍA

Diferentes variables temporales afectan también al número de pulsaciones por minuto de cualquier individuo. Por ejemplo por la mañana tenemos menos pulsaciones que por la tarde. Después de comer, mientras hacemos la digestión y en función de la cantidad y tipo de la comida podemos tener entre un 10 y 30% más de pulsaciones que en reposo. El sueño o el cansancio disminuyen las pulsaciones. Cuando dormimos alcanzamos picos de frecuencia basal, las mínimas pulsaciones por minuto con las que podemos continuar viviendo.

2.8.2.2.2 LA TEMPERATURA

Cuanto más calor, más altas las pulsaciones y de la misma manera cuanto más frío más bajas las pulsaciones.

2.8.2.2.3 LA ALTURA

Cuanto más alto menos oxígeno tenemos en el aire que respiramos y por lo tanto el corazón tiene que bombear más para obtener el mismo oxígeno.

2.8.2.2.4 LA CONTAMINACIÓN

Algunos componentes de la contaminación como el monóxido de carbono empujan al oxígeno disminuyendo la cantidad de este en cada litro de aire. Por lo que el corazón

actúa igual que si faltara oxígeno aumentando las pulsaciones para poder mantener el consumo del oxígeno.

2.8.3 DEPORTES PARA EL CORAZÓN

Perezplata (2011)

La práctica de la actividad física suelen ser muy beneficiosas para el organismo, el cuerpo y la mente. Pero como todo tiene que hacerse con cuidado y sentido común. En función de la edad y del tiempo que lleva la persona sin hacer ejercicio y sobre todo de las enfermedades que padece, será recomendable o no, un deporte dado. Para una persona mayor, con problemas de osteoporosis quizás el más recomendable sea andar o nadar, pero si no sabe nadar quizás el más interesante sea el aquagym. En la mayoría de los casos un deporte aeróbico siempre es el más recomendable.

2.8.3.1 LA CARRERA CONTINUA

Perezplata (2011)

Es muy práctico en la relación tiempo / beneficios también por su facilidad técnica, y por no necesitar de una instalación deportiva para su práctica. (...)También conozco muchos casos de un entrenamiento inadecuado que al final no logra los beneficios que podrían lograrse con un poco de planificación. Por ejemplo el típico que corre media hora varias veces a la semana a una alta intensidad, de forma continuada y se va a casa. El organismo se adapta a ese esfuerzo corto y de alta intensidad.

Granell (2004)

La carrera continua es aquella en la que únicamente se establece el tiempo de carrera, eligiendo el individuo el lugar y demás parámetros.

También existe una carrera continua controlada (menos natural) en la que se determina el tiempo, la intensidad de la misma y el lugar. Supondrá entre un 60% y 70% de las posibilidades de trabajo máximo de un sujeto y se tendrá que ir adaptando de forma progresiva a la mejora. Primero aumentaremos la duración y posteriormente la intensidad, para finalizar combinando ambos factores. (p. 44)

2.8.3.2 LA NATACIÓN

“Es realmente el deporte aeróbico por excelencia, aconsejado desde para los bebés hasta para la tercera edad, pasando por embarazadas y personas con problemas de obesidad.” (Perezplata, 2011)

2.8.3.3 EL CICLISMO

DMedicina (2002)

El ciclismo es uno de los deportes de mayor popularidad en nuestro país. Junto a la carrera a pie y la natación, montar en bicicleta es una de las mejores formas de mejorar la capacidad aeróbica, fortaleciendo al mismo tiempo el corazón y reduciendo el riesgo de enfermedades cardiovasculares. No son necesarias unas condiciones físicas especiales para practicarlo y es recomendable entre personas mayores con problemas en las articulaciones ya que éstas no sufren sobre la bicicleta al evitarse los impactos de otras actividades como la carrera.

2.8.4 EL CORAZÓN Y EL EJERCICIO

El corazón es un músculo muy adaptable tanto a la actividad física como a una vida sedentaria. La diferencia con un músculo esquelético como puede ser bíceps, es que si tenemos un bíceps flojo y rodeado de una buena capa de grasa, esto repercutirá poco en nuestra vida diaria. Pero un corazón débil puede llevarnos a la muerte mucho antes de lo esperado. La muerte por problemas del corazón y del sistema cardiovascular, en edades intermedias, es la primera causa de muerte en los países desarrollados. Muchas de estas se podían haber evitado con una vida sana, y una vida sana es entre otras cosas realizar al menos cuarenta minutos diarios de ejercicio.

Perezplata (2011)

En las personas que se ejercitan mediante trabajos o gimnasias aeróbicas agranda las cámaras del corazón y bajan las pulsaciones por minuto además de mejorar la tensión y la capacidad del transporte sanguíneo, reducir la tensión arterial y aumentar la calidad y densidad de los glóbulos rojos.

La gente no piensa en el corazón como un músculo; que pueda crecer o fortalecerse del mismo modo que otros músculos, pero de hecho, responde como tal al entrenamiento. Un trabajo aeróbico constante y de las intensidades adecuada hace que las fibras

musculares que componen el corazón se vuelva gruesas y fuertes, lo que incrementa la masa muscular de este órgano. Cuanto mayor sea el músculo del corazón, más sangre bombeará por latido, reduciendo el trabajo final que realiza el corazón para mantenernos con vida, lo que alarga la vida de nuestro corazón, y sobre todo le permite estar más sano y fuerte durante más tiempo. (Perezplata, 2011)

2.8.5 CORAZÓN, LATIDOS Y FUNCIONAMIENTO

En el corazón se producen dos funciones específicas que son la sístole y diástole.

2.8.5.1 SÍSTOLE

Cuando nos tomamos o notamos nuestro pulso lo que notamos es el latido (sístole), es el momento que las dos válvula auriculares se cierran eméticamente y las dos válvulas de los ventrículos se abren y el corazón se contrae fuertemente expulsando por las válvulas de los ventrículos la sangre, la del lado derecho se dirige a los pulmones para oxigenar la sangre y la del lado izquierdo ya oxigenada se dirige al resto del cuerpo. (Perezplata, 2011)

2.8.5.2 DIÁSTOLE

Es cuando se produce el llegando de los ventrículos, la sangre pasa de las aurículas derecha e izquierda a los ventrículos derecho e izquierdo, en ese momento las válvulas de los ventrículos están cerradas para impedir que la sangre salga del corazón mientras los ventrículos se llenan. (Perezplata, 2011)

2.8.6 PULSACIONES

Perezplata (2011)

La sangre que bombea el corazón por las arterias al resto del cuerpo va en forma de ondas “pulsaciones”.

Estos pulsos, cada uno corresponde a un latido del corazón, se pueden medir y así conocer el ritmo al que está funcionando nuestro corazón. Las pulsaciones se pueden tomar en cualquier arteria de nuestro cuerpo ya que por todas pasan estas ondas de presión y porque son elásticas.

2.8.6.1 TOMA DE PULSACIONES

Perezplata (2011)

Se tiene dos sitios para la toma de pulsaciones, uno en el cuello “pulso carotideo” y otra en la muñeca “pulso radial”. Realmente se puede tomar en cualquier sitio donde una arteria pueda ser presionada sobre un hueso o un músculo. En el deporte se usan estas dos formas por su sencillez y su facilidad de palpación. Las pulsaciones en el cuello son mucho más fáciles de notar ya que la arteria es más grande, menos en casos médicos normalmente es la mejor zona para que los deportistas controlen su frecuencia cardiaca.



FIGURA 35: Toma de pulso en el cuello

Recuperado de: <http://www.frecuencia-cardiaca.com/quesonlaspulsaciones.php>

Las carótidas son arterias que van por los dos lados del cuello, por lo que con dos dedos el índice y el medio presionas ligeramente en uno de los lados del cuello hasta notar las pulsaciones.



FIGURA 36: Toma de pulso en la muñeca

Recuperado de: <http://www.frecuencia-cardiaca.com/quesonlaspulsaciones.php>

Con los dedos índices y medio o también con el pulgar presionas ligeramente en la muñeca casi en el inicio de la mano cerca del dedo pulgar hasta notar las pulsaciones.

2.8.6.2 TIEMPOS PARA CONTAR LAS PULSACIONES

El tiempo exacto es un minuto, ya que las pulsaciones se miden en minutos. Cuando no tenemos prisa lo normal es un minuto o tomarte las pulsaciones en treinta (30) segundos y luego multiplicar por dos para conocer las que tienes en un minuto.

Cuando se está haciendo deporte y se quiere conocer las pulsaciones durante o después de un ejercicio, ya que están bajan cuando paras, lo normal es tomarse las pulsaciones en quince (15) segundos y luego multiplicarlo por cuatro.

Para ya deportistas en los que las pulsaciones bajan a una velocidad alucinante, suelen tomarse las pulsaciones en seis segundos y luego multiplicarlo por diez. Poner un cero a la cifra.

2.8.6.3 ERRORES EN LA TOMA DE PULSACIONES

- Si nos tomamos las pulsaciones en un minuto el margen de error en ese minuto es de cero.
- Si nos tomamos las pulsaciones en treinta segundos el margen de error es desde +1 hasta -1.
- Si nos tomamos las pulsaciones en quince segundos el margen es desde +4 a -4.
- En el caso de hacerlo en seis segundos el margen es desde -9 hasta 9.

Lo ideal es un usar un buen pulsómetro para conocer a tiempo real nuestras pulsaciones.

2.8.7 EJERCICIO FÍSICO Y FRECUENCIA CARDÍACA

Desde los deportistas amateurs hasta los profesionales tienen que tener una vinculación constante entre el ejercicio físico y la frecuencia cardiaca. Cada ejercicio tiene un objetivo dentro del sistema energético de nuestro metabolismo.

Los objetivos simplificados pueden ser: aeróbicos de larga resistencia; aeróbicos de media resistencia; aeróbicos de alta intensidad; anaeróbicos lácticos o anaeróbicos alácticos.

Cada zona del sistema tiene sus propias frecuencias cardíacas óptimas, no son siempre las mismas, ya que cada persona, sobre todo los deportistas de alto nivel, tienen unas adaptaciones diferentes.

2.8.8 FRECUENCIA CARDÍACA MÁXIMA

2.8.8.1 CÁLCULO DE LA FRECUENCIA CARDÍACA MÁXIMA

“La fórmula de (Fox y Haskell), probablemente la más usada, pero la menos fiable. Esta dice que tenemos que restar a la constante 220 menos nuestra edad en años, ejemplo: $220 - 30 \text{ años} = a$ una frecuencia cardíaca máxima de 190” (Pérez, 2012, p. 4)

Pérez (2012)

Esta fórmula tiene varios inconvenientes, una es que no es lo mismo hombre o mujer, las mujeres tienen más pulsaciones, tampoco son iguales todos los hombres de treinta años, dos individuos del mismo género pueden tener distintas pulsaciones en reposo teniendo la misma edad y esto hay que considerarlo. Podría servirnos de ella con algunas matizaciones como, 220 menos la edad en hombres y 225 menos la edad para las mujeres. (p. 4)

Perezplata (2011) manifiesta: “La fórmula de Fox y Haskell, es una manera estándar y con cierto margen de error, pero si vas a empezar a entrenar el dato te servirá. Para mujeres, como suelen tener la frecuencia cardíaca más alta se recomienda restar la edad a 225.”

Ejemplo de cálculo de Frecuencia Cardíaca Máxima

- **Género:** Masculino
- **Edad:** 20
- **Frecuencia Cardíaca Máxima:** $220 - 20$
- **Frecuencia Cardíaca Máxima:** 200 ppm (pulsaciones por minuto)
- **Género:** Femenino
- **Edad:** 20
- **Frecuencia Cardíaca Máxima:** $226 - 20$
- **Frecuencia Cardíaca Máxima:** 206 ppm (pulsaciones por minuto)

2.8.8.2 OTRAS FORMAS DE CÁLCULO DE LA FRECUENCIA CARDÍACA MÁXIMA

La gente más entrenada suele tener una frecuencia basal más baja, porque el corazón es más grande, por tanto la F.C.max también será más baja, ya que el corazón con menos latidos podrá mandar más sangre y esto lo hará más eficiente. Diferenciamos entre hombres y mujeres:

Otras fórmulas más fiables son:

Frecuencia Cardíaca Máxima: $205.8 - (0,685 * \text{edad en años})$

Frecuencia Cardíaca Máxima: $206.3 - (0,771 * \text{edad en años})$

Frecuencia Cardíaca Máxima (Mujeres): $217 - (0,85 * \text{edad en años})$

Frecuencia Cardíaca Máxima (Hombres): $208 - (0,7 * \text{edad en años})$

Estas ecuaciones son el resultado de diversos estudios realizados por universidades americanas, pero todavía podemos precisar mucho más porque aunque estas se acercan bastante a la realidad no tienen en cuenta el sexo y el peso como la que a continuación se encuentran:

HOMBRES

F.C.M.: $((210 - (0,5 * \text{edad en años})) - 1\% \text{ del peso} + 4$

MUJERES

F.C.M.: $((210 - (0,5 * \text{edad en años})) - 1\% \text{ del peso}$

También se puede ir a un centro de medicina deportiva y allí te realizarán un test llamado CONCONY además chequeo médico que te permitirá empezar con garantías, sin duda esta es la opción más correcta. (Pérez, 2012, p. 5)

2.8.9 FRECUENCIA CARDÍACA DE REPOSO

Es importante tomar el pulso cuando el cuerpo está en reposo, porque en esta condición las pulsaciones y frecuencias cardíacas se encuentran a un ritmo normal. Cuando el individuo se encuentra en reposo, la frecuencia cardíaca puede estar entre 60 y 80 p.p.m.

Esta frecuencia se obtiene inmediatamente después de levantarse, que es la manera más óptima, también se la puede obtener antes de salir de la casa o antes de hacer cualquier actividad física.

Cuando el cuerpo es sometido a una actividad física requiere de un potencial energético mayor que el normal. A medida que la actividad aumenta, mayor será la necesidad de consumo de energía. Cuando un individuo altera su estado reposo a través de la actividad física, aumenta su frecuencia respiratoria, su frecuencia cardíaca, su temperatura corporal y aparece la sudoración (este último depende).

2.8.10 FRECUENCIA CARDIACA MÁXIMA DE RESERVA

La frecuencia cardíaca máxima de reserva se define como la diferencia entre la frecuencia cardíaca máxima y la frecuencia cardíaca en reposo

$$\text{Frecuencia Máxima de Reserva} = \text{Frec. Máxima} - \text{Frec. Reposo.}$$

2.9 ENTRENAMIENTO SEGÚN LA FRECUENCIA CARDÍACA

El entrenamiento según la frecuencia cardíaca, se realiza como indicador para la realización de un correcto entrenamiento en personas que realizan actividad física o algún deporte. Este tipo de entrenamiento hace referencia de manera concreta a la frecuencia cardíaca máxima de cada individuo; de esta F.C.M.¹⁹ se derivan las zonas de entrenamiento. Estas zonas de entrenamiento, son rangos de pulsaciones en donde se deberá realizar la actividad física; y se las encuentra mediante cálculos específicos

Determinar las zonas de entrenamiento (ZE) o niveles de intensidad nos ayudará a organizar y optimizar mejor nuestro entrenamiento, pues a cada ZE le corresponde una vía metabólica, unos sustratos energéticos y un porcentaje de intensidad (nosotros utilizaremos porcentajes sobre la frecuencia cardíaca máxima, y la utilización de la frecuencia cardíaca de reposo y de reserva). Así se facilitará el desarrollo de las diferentes capacidades físicas, en este caso de la resistencia aeróbica.

¹⁹ Frecuencia Cardíaca Máxima

2.9.1 ZONAS DE ENTRENAMIENTO SEGÚN LA FRECUENCIA CARDÍACA

Las zonas de entrenamiento según en la frecuencia cardiaca son 5 y cada una de ellas tiene su objetivo, intensidad y tiempo de trabajo, para lo cual la persona tiene que realizar el entrenamiento sabiendo sus características.

Zona Objetivo	Intensidad % de FC _{máx.}	Duración intervalo	Efecto/Beneficio del ejercicio.
Máxima	90-100%	0-2 minutos	✓ entona el sistema neuromuscular ✓ aumenta la máxima velocidad de sprint en carrera
Intensa	80-90%	2-10 min.	✓ incrementa la tolerancia anaeróbica ✓ mejora la resistencia a altas velocidades
Moderada	70-80%	10-40 min.	✓ aumenta la potencia aeróbica ✓ mejora la circulación sanguínea.
Suave	60-70%	40-80 min.	✓ aumenta la resistencia aeróbica ✓ prepara el cuerpo para tolerar mayor intensidad. ✓ incrementa el metabolismo
Muy suave	50-60%	20-40 min.	✓ Aumenta y acelera la recuperación después de un ejercicio intenso

FIGURA 37: Zonas de Entrenamiento

Recuperado de: [http://4.bp.blogspot.com/-](http://4.bp.blogspot.com/-AHzvw77Mmxg/UibiHCgNSYI/AAAAAAAAAJTE/XgwK4TP97HA/s1600/Cuadro_Frecuencia_Cardiac.png)

[AHzvw77Mmxg/UibiHCgNSYI/AAAAAAAAAJTE/XgwK4TP97HA/s1600/Cuadro_Frecuencia_Cardiac.png](http://4.bp.blogspot.com/-AHzvw77Mmxg/UibiHCgNSYI/AAAAAAAAAJTE/XgwK4TP97HA/s1600/Cuadro_Frecuencia_Cardiac.png)

2.9.2 ZONA 1 - MEJORAMIENTO DE LA SALUD

Se trabaja al 50% al 60% de la capacidad cardiaca. Es ideal para principiantes en el ejercicio, reduce el riesgo de enfermedades cardiovasculares pero no hay mucha mejora de rendimiento físico. No es una zona ni para control de peso ni de mejora de tu capacidad aeróbica, es una zona que se usa sólo en principiantes o personas que tengan que trabajar en su salud cardiaca. (Ramos, 2014)

2.9.2.1 CARACTERÍSTICAS DE LA ZONA 1

- Corresponde a una intensidad de esfuerzo y pulsaciones que oscila entre el 50% y el 60%.
- Solo tiene utilidad en personas mayores, con problemas cardiacos.

- Se consigue una mejora ligera de las capacidades físicas.
- Deporte aconsejado: andar, caminar, nadar.
- Método de Entrenamiento: continuo o fraccionado de larga duración. Es decir que se realice por ejemplo una caminata de una hora o se hagan cuatro series de 10 minutos nadando suave.

2.9.2.1.1 ZONA 2 – QUEMA DE GRASAS

Se trabaja del 60% al 70% de la capacidad aeróbica. Es la zona donde el mayor combustible usado por tu cuerpo es la grasa. Es usada por muchos fisicoculturistas que acompañados de una dieta específica y un programa de entrenamiento muscular muy bien diseñado, queman más grasa para poder definir más los músculos. Pero para que esto sea 100% efectivo debes permanecer en esta frecuencia (que es bastante baja) por 45 minutos, en ayunas. (Ramos, 2014)

2.9.2.2 CARACTERÍSTICAS DE LA ZONA 2

- Corresponde a una intensidad de esfuerzo y pulsaciones que oscila entre el 60% y el 70%.
- Es el rango de trabajo más indicado para adelgazar, iniciarse en el deporte aeróbico, buscar una forma física saludable.
- Tiene que mantenerse por encima de los cuarenta o cincuenta minutos de duración.
- Se consigue quemar grasa, bajar el colesterol malo, mejorar el sistema cardiovascular, el tono muscular.
- Deporte aconsejado: aerobio, spinning, natación, ciclismo, remo, fitness aeróbico.
- Método de Entrenamiento: fraccionado de corta y media duración, cambios de ritmo o Fartlek.

2.9.2.2.1 ZONA 3 - AERÓBICA

Se trabaja del 70 al 80% de la capacidad cardiaca y es la ideal para mejorar toda la forma física y la capacidad aeróbica. El número y tamaño de los vasos sanguíneos aumentan, y vas trabajando en aumentar tu nivel pulmonar y la frecuencia respiratoria. Tu corazón aumenta en fuerza para que puedas ejercitarte más tiempo antes de fatigarse.

En esta etapa todavía se están metabolizando las grasas y los hidratos de carbono en alrededor de una tasa de 50 a 50 lo que significa estar quemando en la misma proporción. (Ramos, 2014)

2.9.2.3 CARACTERÍSTICAS DE LA ZONA 3

- Corresponde a una intensidad de esfuerzo y pulsaciones que oscila entre el 70% y el 80%.
- Se mejora todos los aspectos metabólicos y orgánicos relacionados con la mejora de las capacidades físicas.
- Régimen de trabajo solo recomendado para jóvenes o personas adultas acostumbradas al deporte.
- Deporte aconsejado: deportes de equipo, spinning, natación, ciclismo, remo, fitness aeróbico.
- Método de Entrenamiento: fraccionado de corta y media duración, cambios de ritmo o Fartlek.

2.9.2.3.1 ZONA 4 - ANAERÓBICA

Esta es una zona difícil de alcanzar y más de mantener. Se trabaja del 80 al 90% de la capacidad aeróbica. Aquí se aumenta mucho tu ritmo cardíaco. En este punto, el corazón no puede bombear suficiente sangre y oxígeno para abastecer los músculos por lo que el cuerpo tiene que trabajar en el umbral anaeróbico para suplir energía.

Seguramente sólo puedes permanecer en esta zona durante un periodo de tiempo limitado, por lo general no más de una hora. Esto es porque el músculo no puede sostener trabajando anaeróbicamente (esto quiere decir, sin oxígeno suficiente) sin fatigarse.

El umbral anaeróbico, es el punto en el que el ácido láctico empieza a acumularse en los músculos, se presenta entre el 85% y el 90% de tu FCM. Es una zona más usada por atletas para mejorar su condición física, ya que favorece la quema de grasas y otros combustibles del cuerpo. Los principiantes no pueden mantenerse en esta zona. (Ramos, 2014)

2.9.2.4 CARACTERÍSTICAS DE LA ZONA 4

- El trabajo anaeróbico solo tienen utilidad para deportistas de competición o adultos que han conseguido una forma física excelente y quieren mejorar ciertas capacidades no directamente relacionadas con la mejora de la salud.
- El trabajo anaeróbico puede ser láctico o aláctico, en función de los objetivos.
- El trabajo de velocidad es un trabajo anaeróbico aláctico.
- Los trabajos intensos y cortos (1') son anaeróbicos lácticos y producen acumulación de ácido láctico en sangre. Entrenamiento
- Entrenamientos para mejorar la forma física

2.9.2.4.1 ZONA 5 – LÍNEA ROJA

Trabajas del 90 al 100% de tu capacidad cardiaca, se puede hacer sólo por pocos minutos y normalmente se usa sólo para entrenamientos de intervalos, donde varías intervalos a toda velocidad con intervalos de recuperación. (Ramos, 2014)

2.9.2.5 CARACTERÍSTICAS DE LA ZONA 5

- Más del 90% de la F.C.M.
- No recomendado para entrenamientos cardiovasculares orientados a la salud o pérdida de peso.
- Utilizado más bien en deportes de rendimiento, grandes esfuerzos o entrenamientos incontrolados.
- Este es el más alto nivel de entrenamiento de intensidad. Solo deberías trabajar en este rango si estás extremadamente entrenado.
- Mientras estés en la Zona Roja, tus músculos estarán usando más oxígeno del que tu cuerpo puede proveer.
- Mejora de la capacidad aeróbica y anaeróbica.

2.9.3 ENTRENAMIENTO: MÉTODO SIMPLIFICADO

Este método es el más simple para la obtención de las zonas de entrenamiento, pero es menos preciso. Este consiste en obtener la frecuencia cardíaca máxima de cada individuo en función de la edad biológica

$$\text{Frec. Máxima} = 220 - \text{edad}$$

Una vez obtenido este dato, la intensidad (rangos de entrenamiento) se expresa en porcentajes de este valor (Frec. Máx).

2.9.3.1 CÁLCULO DE LAS ZONAS DE ENTRENAMIENTO POR EL MÉTODO SIMPLIFICADO

Para el cálculo de las zonas de entrenamiento se realiza de la siguiente manera:

- 1) Se obtiene la frecuencia cardíaca máxima (F.C.M.)
- 2) Elegir la zona de entrenamiento
- 3) Multiplicamos la F.C.M por los porcentajes establecidos en cada zona de entrenamiento.

Ejemplo en “ZONA 1”

- **Genero** = Masculino
- **Edad** = 20
- **F.C.M.** = $220 - 20 = 200$ ppm. (Pulsaciones por minuto)
- **ZONA 1**
- **Umbral inferior** = 50% de F.C.M. = $200 * 50\% = 100$ ppm.
- **Umbral Superior** = 60 % de F.C.M. = $200 * 60\% = 120$ ppm.

Resultado: El rango de pulsaciones en la “Zona 1” deberá estar entre 100 y 120 ppm. Aquí la importancia de dichos cálculos, ya que servirán de apoyo científico y correcto al momento de realizar nuestra actividad física. De esta forma se evitarán las lesiones y los posibles problemas cardiovasculares por un entrenamiento inadecuado.

Ejemplo adicional

Edad deportista: 20

Género: Masculino

Frec. Cardíaca. Máxima: 220-20: 200 ppm.

Edad deportista: 20

Género: Femenino

Frec. Cardíaca. Máxima: 226-20: 206 ppm.

Si queremos entrenar a una intensidad, por ejemplo del 80% se realiza de la siguiente manera:

80% Frecn Cardíaca Máxima: $200 \times 80\%$

80% Frecn Cardíaca Máxima: 160 ppm.

Esta fórmula es muy conocida, pero en la actualidad no se recomienda el uso para deportistas, esta fórmula no toma en cuenta como valor inicial el estado de forma del deportista.

2.9.4 ENTRENAMIENTO: MÉTODO MEDIANTE FÓRMULA DE KARVONEN

Wilmore (2007)

Este método para saber nuestros rangos de pulsaciones (intensidades) es mucho más viable y científico; este método toma en cuenta la condición física del individuo. Se puede dar a conocer que el método tiene su validez en la siguiente relación: Una persona de 35 años que ha entrenado toda su vida, no podrá tener un mismo rango de intensidades, ya que influirá mucho su frecuencia en reposo. (p. 663)

Es una ecuación matemática que se utiliza para la dosificación de la intensidad de trabajo de acuerdo a la frecuencia cardíaca.

Algunos cálculos de frecuencia cardíaca se basan sencillamente en multiplicar el porcentaje de esfuerzo de su frecuencia cardíaca máxima. Esto no toma en cuenta el

hecho de que cada uno tiene diferente frecuencia cardíaca en reposo. El fisiólogo Karvonen, que se percató de esto, afirma que nuestra frecuencia cardíaca de reserva es igual a la frecuencia cardíaca máxima, menos nuestra frecuencia cardíaca en reposo.

Para calcular la zona de trabajo bastaría con multiplicar la frecuencia cardíaca de reserva por el porcentaje de intensidad y sumar nuestra frecuencia cardíaca en reposo a esta cifra.

$$F. .C.Ent = (F.C.Max - F.C.Rep) \times \text{intensidad del ejercicio} + F.C.Rep$$

$$F. .C.Ent = (R.F.C. \times \text{intensidad del ejercicio}) + F.C.Rep$$

F.C.Ent: Frecuencia cardíaca de entrenamiento.

F.C.Max: Frecuencia cardíaca máxima.

F.C.Rep: Frecuencia cardíaca en reposo.

R.F.C: Reserva funcional del corazón.

2.9.4.1 CÁLCULO DE LAS ZONAS DE ENTRENAMIENTO SEGÚN KARVONEN

Para el cálculo de las zonas de entrenamiento se realiza de la siguiente manera:

- 1) Se obtiene la frecuencia cardíaca máxima (F.C.M.)
- 2) Se obtiene la frecuencia cardíaca de reposo o pulso basal
- 3) Se obtiene el pulso de reserva
- 4) Se elige la zona de entrenamiento (intensidad del ejercicio)
- 5) Se multiplica el pulso de reserva por la intensidad del ejercicio y se suma la frecuencia de reposo.

Ejemplo en “ZONA 1”

- **Genero** = Masculino
- **Edad** =20

- **F.C.M.** = $220 - 20 = 200$ ppm. (Pulsaciones por minuto)
- **F.C.Rep.** = 70 (Depende de cada persona)
- **F.C. Reserva** = 130 ppm.
- **ZONA 1 (50%-60%)**
- **Umbral inferior**

$\text{F.C.entr} = (130 * 50\%) + 70$ $\text{F.C.entr} = 135 \text{ ppm}$

- **Umbral Superior**

$\text{F.C.entr} = (130 * 60\%) + 70$ $\text{F.C.entr} = 148 \text{ ppm}$

Resultado: El rango de pulsaciones en la "Zona 1" deberá estar entre 135 y 148 ppm. De esta forma se evitarán las lesiones y los posibles problemas cardiovasculares por un entrenamiento inadecuado.

CAPÍTULO III

3 DESARROLLO DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO

3.1 ANÁLISIS DE REQUERIMIENTOS DE ELEMENTOS ELECTRÓNICOS MEDIANTE LA NORMA IEEE 29148

3.1.1 INTRODUCCIÓN

3.1.1.1 PROPÓSITO DEL SISTEMA

El proyecto constituye en el diseño un sistema electrónico de monitoreo de ritmo cardíaco para personas que realizan ejercicio físico aeróbico utilizando software y hardware libre.

3.1.1.2 ALCANCE DEL SISTEMA

Para el proyecto se utilizará un sistema electrónico, el cual constará de una placa con microcontrolador con característica que sea adaptable para textiles inteligentes y se utilizará para las personas que realicen ejercicio físico aeróbico. A dicha placa estará conectado un sensor de signos vitales y en la alimentación se usará una batería de tipo Lipo. El sistema utilizará una comunicación Bluetooth, entre un módulo conectado a la placa y el Smartphone del usuario.

3.1.1.3 DESCRIPCIÓN GENERAL DEL SISTEMA

3.1.1.3.1 PERSPECTIVA DEL SISTEMA

Se realizará un análisis de las características técnicas y físicas de los elementos que se utilizarán para el sistema electrónico, los mismos que constará principalmente de una placa con microcontrolador diseñado para textiles inteligentes, que tendrá como características especiales, la impermeabilidad, facilidad y flexibilidad para las personas que realicen ejercicio físico, de desplazarse sin ninguna restricción, será ubicado en la camiseta o licra del individuo (muñequera). Dentro del sistema, también estará conectado junto a la placa especificada, un sensor de signos vitales, que servirá para captar los pulsos de frecuencia cardíaca de la persona. Además el sistema constará de una alimentación (Lipo) de corriente directa, con características específicas para conectar de manera externa a la placa.

Para este proyecto se realizará una aplicación para Sistema Operativo Android compatible con dispositivos móviles.

Posteriormente se implementará en el dispositivo móvil del usuario. Mediante la aplicación, se podrá observar la información de frecuencia cardiaca máxima (F.C.M.), a partir del ingreso de la edad y otros factores. Esta información de la FCM, servirá como punto de partida para elegir una de las cinco zonas de entrenamiento según la frecuencia cardiaca. Además, inicialmente, el usuario indicará su estado físico actual y si ha tenido huella deportiva; esto será para que la aplicación verifique los niveles de entrenamiento hasta donde el individuo pueda realizar. La aplicación constará de una alarma que indiquen si la persona se encuentra en el rango de pulsaciones adecuado y si se pasan de los mismos

Este proyecto realizará la función específica de monitorear las pulsaciones de la frecuencia cardíaca de la persona, solo cuando esta esté efectuando actividad y ejercicio físico. También, se menciona, que se utilizará tecnología inalámbrica Bluetooth, la cual servirá para la comunicación entre el Sistema Electrónico y la Aplicación.

3.1.1.3.2 FUNCIONES DEL PRODUCTO

El proyecto cumplirá con la función específica de monitorear las pulsaciones de la frecuencia cardíaca, la frecuencia cardíaca máxima, los rangos de pulsaciones en cada zona de entrenamiento, cuando las personas estén realizando ejercicio físico.

3.1.1.3.3 CARACTERÍSTICAS DEL USUARIO

El "Sistema de Monitoreo de Ritmo Cardíaco" está enfocado para su uso a todas las personas que realicen actividad y ejercicio físico aeróbico. También podrá ser utilizado por todas las personas que presenten problemas cardiacos y quieran monitorear sus pulsaciones a cualquier hora del día.

Este sistema electrónico podrá ser utilizado de una manera simple, ya que presentará una aplicación gráfica, fácil de operar para cualquier persona. Los usuarios podrán visualizar su ritmo cardíaco y su límite máximo de pulsaciones (frecuencia cardíaca máxima) y tendrán zonas de entrenamiento a elegir.

3.1.1.3.4 LIMITACIONES

El sistema electrónico tendrá condiciones de uso para su aplicación, entre las cuales se encuentran:

- Una comunicación inalámbrica de tipo WPAN, entre una placa con microcontrolador con modulo Bluetooth y el dispositivo móvil (Smartphone).
- La aplicación en el Smartphone será solo para Sistema Operativo Android.
- El sistema electrónico funcionará siempre que su alimentación con pila de tipo LIPO, este recargadas o con energía suficiente.

3.1.1.4 DEFINICIONES

- **Sistema Electrónico.-** Conjunto de dispositivos que se ubican dentro del campo de la ingeniería y la física y que se encargan de la aplicación de los circuitos electrónicos cuyo funcionamiento depende del flujo de electrones para generar, recibir, transmitir y almacenar información.
- **Entrenamiento Aeróbico.-** Son ejercicios de media o baja intensidad y de larga duración, donde el organismo necesita quemar hidratos y grasas para obtener energía y para ello necesita oxígeno.
- **Actividad Física:** Desde una perspectiva fisiológica, cualquier movimiento corporal producido por los músculos esqueléticos que den como resultado un gasto energético. Ejemplos: Trabajar, sentarse, pararse, caminar, tocar un instrumento musical, bailar, limpiar
- **Ejercicio Físico:** Es la actividad física planeada, estructurada, repetitiva y dirigida con el objetivo de mejorar o mantener uno o varios de los componentes de la aptitud física. Ejemplos: Resistencia, Fuerza, Velocidad, Flexibilidad.
- **WPAN.-** Red Inalámbrica de Área Personal o Red de área personal o Personal área network es una red de computadoras para la comunicación entre distintos dispositivos (tanto computadoras, puntos de acceso a internet, teléfonos celulares, PDA,

dispositivos de audio, impresoras) cercanos al punto de acceso. Estas redes normalmente son de unos pocos metros y para uso personal.

- **Bluetooth.-** Es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.
- **Android.-** Es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tablets; y también para relojes inteligentes, televisores y automóviles.

3.1.2 REFERENCIAS

Norma IEEE 29148.

- http://es.slideshare.net/Juan_Tapias/formato-ieee830srs-lleño
- <http://es.slideshare.net/JuAnJoShL/srs-ejemplo-sistema-tarifado-de-transito>
- BOMPA T.O. (2005), Entrenamiento para Jóvenes Deportistas, Ed. Hispano Europea, España.
- TORRENTE Oscar (2013), "Arduino: Curso práctico de información" Ed. Rc. Libros, España

3.1.3 REQUERIMIENTOS DEL SISTEMA

Se especificarán todos los requisitos de software a un nivel de detalle suficiente para permitir a los testers probar que el sistema de software cumple dichos requisitos.

3.1.3.1 INTERFACES EXTERNAS

El sistema electrónico (placa con el microcontrolador, el sensor, la alimentación eléctrica y el modulo Bluetooth), deberá interactuar con la aplicación (Android) del Smartphone.

TABLA 11: Interfaces Externas

Código.	01
Descripción	Aplicación en Smartphone.
Descripción Detallada	La aplicación se encontrará en un Smartphone con Sistema Operativo Android.
Relación con otras entras/salidas	Se comunicará con el sistema electrónico mediante comunicación Bluetooth.
Organización de Pantalla	Botones, Text Box, Label, Notificaciones
Organización de Ventana	6 Screens (independientes)
Unidades de medida	Pulsaciones por minuto

Fuente: Edison Flores

3.1.3.2 REQUERIMIENTOS FUNCIONALES

- Usuario: Persona que realice la práctica deportiva.

TABLA 12: Requerimientos Funcionales (Visualización de Pulsaciones)

Código	FR01
Descripción	Visualización de Pulsaciones por minuto
Descripción Detallada	Permite al deportista visualizar sus pulsaciones en el momento requerido.
Precondiciones	La aplicación debe conectarse previamente con el módulo Bluetooth del sistema electrónico para la visualización de los pulsos cardíacos (placa con microcontrolador, sensor cardíaco).
Entradas	Datos del sensor cardíaco
Procesos	El sensor cardíaco censa las pulsaciones de cada persona y envía la información a través de la placa Y modulo Bluetooth a la aplicación del Smartphone.
Salidas	Edad, género, frecuencia cardíaca, zona de entrenamiento, rango de pulsaciones
Postcondiciones	Se podrá elegir la zona de entrenamiento y rangos de pulsaciones
Verificaciones	Si
Prioridad del requerimiento:	Alta

Fuente: Edison Flores

TABLA 13: Requerimientos Funcionales (Visualización de FCM)

Código.	RF02
Descripción	Visualización de la Frecuencia Cardíaca Máxima.
Descripción Detallada	Permite conocer al deportista su frecuencia cardíaca máxima.
Precondiciones	Ingreso de Edad, pulso normal, género
Entradas	Datos del deportista: Edad del individuo, pulso normal, género.
Procesos	Se realizará el cálculo de la Frecuencia Cardíaca Máxima mediante la fórmula de Karvonen.
Salidas	Frecuencia Cardíaca Máxima de la persona.
Postcondiciones	Se podrá elegir las zonas de entrenamiento, tomando como referencia la F.C.M., pulso normal y pulso de reserva
Verificaciones	Si
Prioridad del requerimiento:	Alta

Fuente: Edison Flores

TABLA 14: Requerimientos Funcionales (Elección de la Z.E.)

Código.	RF03
Descripción	Elección de la zona de entrenamiento según la frecuencia cardíaca.
Descripción Detallada	El deportista podrá elegir la zona de entrenamiento, cada una tendrá sus características específicas.
Precondiciones	Haber calculado la frecuencia cardíaca máxima, el pulso de normal y el pulso de reserva.
Entradas	Frecuencia cardíaca máxima
Procesos	Se obtendrá los rangos de pulsaciones en cada zona entrenamiento, aplicando la fórmula de Karvonen.
Salidas	Rango de pulsaciones en cada zona de entrenamiento
Postcondiciones	Cuando este dentro de una zona de entrenamiento, y cuando ya pueda visualizar las mismas, podrá salir o volver al menú inicial.
Verificaciones	Si
Prioridad del requerimiento:	Alta

Fuente: Edison Flores

TABLA 15: Requerimientos Funcionales (Monitoreo Z.E.)

Código.	RF04
Descripción	Monitoreo de la zona de entrenamiento según la frecuencia cardíaca
Descripción Detallada	Se emitirá una advertencia, cuando las pulsaciones estén fuera del rango establecido en cada zona de entrenamiento.
Precondiciones	Elegir la zona de entrenamiento
Entradas	Frecuencia cardíaca máxima, Zona de entrenamiento.
Procesos	El Smartphone emitirá una advertencia cuando las pulsaciones emitidas por el sistema electrónico, estén fuera del rango establecido en cada zona de entrenamiento. Si el pulso es menor el dispositivo móvil vibrará y si el pulso es mayor sonará una alarma.
Salidas	Alarma de advertencia, vibración de dispositivo móvil
Postcondiciones	-
Verificaciones	Si
Prioridad del requerimiento:	Alta

Fuente: Edison Flores

3.1.3.3 REQUERIMIENTOS DE USABILIDAD

- La aplicación que se encontrará en el Smartphone deberá ser fiable y muy intuitiva para en la visualización de las pulsaciones y elección del entrenamiento según la frecuencia cardíaca.
- El sistema electrónico (placa con microcontrolador + modulo Bluetooth + alimentación eléctrica), deberá adaptarse a algún textil (camisa, gorra, licra, muñequera) para tener flexibilidad en su uso.

3.1.3.4 REQUERIMIENTOS DE RENDIMIENTO

- El tiempo de respuesta de los pulsos cardíacos se podrá visualizar en la aplicación cada 1000 milisegundos (1 segundo), por lo que se tiene un 100% de fiabilidad para el usuario.
- La aplicación proporcionará la información necesaria, sin emitir problemas por exceso de carga de datos
- Se espera un buen rendimiento del sistema electrónico y de la aplicación, ya que será utilizado por todas las personas que realicen ejercicio físico, en cualquier hora del día y bajo cualquier condición climática.
- Para que la comunicación Bluetooth sea correcta, la alimentación eléctrica deberá estar cargada. Así se cumplirá una comunicación sin errores.

3.1.3.5 REQUERIMIENTOS DE BASES DE DATOS

Para este sistema de Monitoreo de ritmo cardíaco no se necesita el uso de una base de datos.

3.1.3.6 RESTRICCIONES DE DISEÑO

- El sistema solo monitoreará la frecuencia cardíaca, y emitirá una señal de alarma al usuario, cuando salga de las especificaciones del rango de pulsaciones de cada zona.
- El sistema electrónico solo funcionará con la aplicación en el Sistema Android establecida por el administrador.
- No constará de una base de datos, en donde el deportista se pueda guardar sus entrenamientos.

3.1.3.7 ATRIBUTOS DEL SISTEMA DE SOFTWARE

a) Fiabilidad

- El sistema electrónico debe emitir los pulsos cardíacos en periodos cortos de tiempo y que sean confiables para el usuario.
- La aplicación en el Sistema Operativo Android debe tener una interfaz de uso intuitiva y sencilla.

b) Disponibilidad

- La disponibilidad del sistema debe ser continua con un nivel de servicio entre los 5 minutos a 60 minutos aproximadamente, necesario para cuando la persona realice su entrenamiento aeróbico.
- Puede utilizarse varias veces, solo se tendrá que ingresar la edad y elegir un tipo de zona de entrenamiento.

c) Seguridad

Garantizar la confiabilidad, la seguridad y el desempeño del sistema electrónico y de la aplicación. En este sentido la información almacenada como son los pulsos cardíacos de la persona, podrá ser visualizada y se actualizarán constantemente sin que afecte el tiempo de respuesta.

d) Capacidad de mantenimiento

La interfaz de la aplicación, debe ser de fácil interpretación, para que cualquier persona con poca experiencia en uso de aplicaciones móviles pueda ejecutar sus entrenamientos y ver su F.M. y su F.C.M.

e) Portabilidad

El sistema será implementado bajo la plataforma de software libre Arduino. Se utilizará el compilador propio de Arduino, con lenguaje de programación C. Para la aplicación en el Smartphone, será implementado con el Sistema Operativo Android. Se utilizará la programación en bloques que se incorpora en el compilador MIT App Inventor.

Las dos plataformas son de software libre y podrán ser utilizadas en cualquier otro Smartphone con sistema operativo Android y en cualquier placa Arduino

3.1.4 APÉNDICES

F.C.- Frecuencia Cardíaca

F.C.M.- Frecuencia Cardíaca Máxima

S.O.- Sistema Operativo

M.I.T.- Instituto tecnológico de Massachusetts

3.2 HARDWARE DEL SISTEMA DE MONITOREO CARDÍACO



FIGURA 38: Diseño General del Sistema Cardíaco

Fuente: Edison Flores

Para la implementación del hardware se utilizará elementos esenciales como el Arduino LilyPad, el sensor cardíaco, el módulo Bluetooth HC-05 y el Smartphone. Estos elementos estarán ubicados en una muñequera, material tipo licra y el Smartphone en el brazo.

Una vez elegidos los elementos a utilizar, además de los diferentes dispositivos, se realizará el diseño de bloques y de flujo del sistema cardíaco. De forma general, en la Figura 37 se muestra los dispositivos y elementos que intervienen en el sistema y su ubicación.

3.2.1 ELEMENTOS ELECTRÓNICOS UTILIZADOS

3.2.1.1 SENSOR CARDÍACO

Este es un elemento electrónico de mucha importancia en el sistema, deberá adaptarse a las necesidades del usuario para su toma de pulso cardíaco, en este caso en los dedos de la mano. El sensor de pulsos o sensor cardíaco recibirá la señal que emite el corazón del deportista o persona que realice AF²⁰, recibirá dicha señal en forma de voltaje analógico y transformado a rango numéricos, estos serán enviados a la placa Arduino para procesar los datos.

²⁰ Actividad Física



FIGURA 39: Pulse Sensor

Fuente: Edison Flores

TABLA 16: Características Pulse Sensor

Descripción	Características
Corriente utilizada	4 mA
Nivel de Voltaje	3,3 a 5 voltios
Uso	Dedos o lóbulo de la oreja
Compatibilidad	Arduino
Conector	Tipo Macho
Dimensiones	0.625 " de diámetro y 0.125" de espesor

Fuente: Edison Flores

3.2.1.2 ARDUINO LILYPAD ATMEGA 168

La placa Arduino Lilypad es la parte principal de todo el sistema cardíaco. Este receptorá los datos emitidos por el sensor de pulsos, los procesará y transformará en información confiable del ritmo cardíaco de cada persona. También procesará la información que enviará la aplicación que se encuentra en el Smartphone.

Existen una gran variedad de placas Arduino, sin embargo, las características que necesita el proyecto, es que la placa electrónica se pueda adaptar a la ropa o prenda textil para convertirse en textil inteligente. Por tal motivo se optó por utilizar la placa Arduino Lilypad ATmega 168.

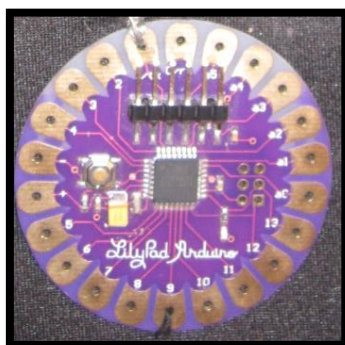


FIGURA 40: Placa Arduino LilyPad ATmega 168

Fuente: Edison Flores

TABLA 17: Características ATmega168

Número de pines	Descripción
0	Comunicación Serial (Rx)
1	Comunicación Serial (Tx)
2 -4 -7 -8 -12 -13	Pines Digitales
a0 -a1 -a2 -a3 -a4 -a5	Pines Análogos
3 -5 -6 -9 -10 -11	Pines Digitales (PWM)
Negativo (-)	GND o Tierra
Positivo (+)	Vcc o Fuente

Fuente: <http://arduino.cc/en/Main/ArduinoBoardLilyPadSimple>

El microcontrolador que utiliza el LilyPad Arduino es del fabricante ATMEGA de la familia AVR, donde el microcontrolador usado es un ATMEGA 168 0 328, tiene una distribución de 28 pines.

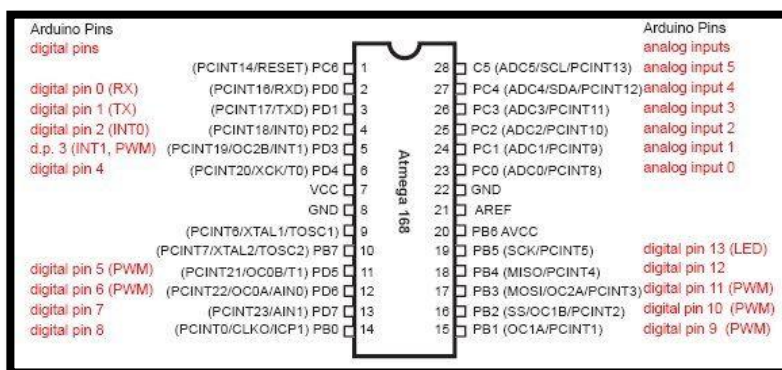


FIGURA 41: Diagrama de pines de ATmega168

Fuente: <http://cdn.instructables.com/FSP/UX40/FVDBZTDS/FSPUX40FVDBZTDS.LARGE.jpg>

Para el sensor cardíaco se utiliza cualquiera de los pines análogos, en este caso se utilizará el pin a0. Además para el módulo Bluetooth se utilizara los pines 0 y 1 que son de transmisión y recepción.

3.2.1.3 MÓDULO BLUETOOTH HC- 05

El uso de la tecnología Bluetooth es la mejor elección para la comunicación inalámbrica con entre la placa Arduino y el Smartphone. Cabe mencionar que la mayoría de dispositivos móviles cuentan con tecnología Bluetooth incorporada.

El módulo estará conectado al Arduino Lilypad y cumplirá la función de transmitir y recibir la información y datos hacia el módulo Bluetooth del Smartphone. El área de comunicación que se necesita el sistema de monitoreo cardíaco es de 1 metro, por lo que no existiría problemas de distancia.

Se ha elegido un módulo de comunicación Bluetooth HC-05 ya que esto dependió mucho principalmente de la disponibilidad del dispositivo que existe en el mercado nacional, su fácil adquisición en cualquier tienda electrónica y su bajo valores económicos pueden variar entre los 15 a 20 dólares.



FIGURA 42: Modulo Bluetooth HC-05

Fuente: Edison Flores

El módulo Bluetooth HC-05 cuenta con pines para la transmisión y recepción de datos en serie, ventaja por la cual fue seleccionado para ser conectado con los pines de comunicación serial de la placa electrónica Arduino Lilypad ATmega168.

TABLA 18: Distribución de pines de Módulo Bluetooth HC-05

DISTRIBUCIÓN DE PINES	DESCRIPCIÓN
ENABLE	En nivel alto entra al modo de configuración
VCC	Alimentación del módulo entre 3.6-6 voltios
GND	Tierra del módulo
TXD	Transmisión de datos
RXD	Recepción de datos a 3.3 voltios
STATE	Conectar un LED para visualizar la transmisión de datos.

Fuente: <http://diymakers.es/arduino-bluetooth/>

Además se puede mencionar características adicionales, para mayor especificación del módulo.

TABLA 19: Características adicionales de Módulo Bluetooth HC-05

CARACTERÍSTICAS	DESCRIPCIÓN
Chipset	CSR BC417143
Versión	V2.0 + EDR
Tensión de Alimentación	3.3 – 6 voltios
Consumo de corriente máximo	40mA
Consumo de corriente mínimo	8mA
Frecuencia	2.4 GHz en las bandas ISM sin licencia
Modulación	GFSK (Modulación por Desplazamiento de Frecuencia Gausiana)
Velocidad de Transmisión de Datos	2 – 3 Mbps teórico, y 1Mbps en la práctica
Potencia de Salida	Clase 2 (2.5 mW – 4dBm)
Cobertura	De 15 a 20 metros teóricamente, y 10 metros máximo en la práctica.
Soporte	Comandos AT para su configuración mediante un puerto serie.
Configuración	Mediante puertos de comunicación a una velocidad de 9600 baudios.
Modo de Trabajo	Configurados por fábrica como esclavos, pero pueden ser cambiados a modo maestro.
Dimensiones	26.9 mm x 13 mm x 2.2 mm
Temperatura de Trabajo	-20 °C a + 75 °C

Fuente: Basado en <http://www.aquihayapuntes.com/indice-practicas-pic-en-c/bluetooth-hc-05.html>

3.2.1.4 MÓDULO BLUETOOTH DEL SMARTPHONE

Enviará los datos e información que emita la aplicación del dispositivo móvil y también recibirá los datos desde el modulo Bluetooth conectado a la placa Arduino. Este dispositivo electrónico viene incorporado en los Smartphone actuales.



FIGURA 43: Modulo Samsung S4

Fuente: Edison Flores

3.2.1.5 SMARTPHONE (TELÉFONO INTELIGENTE)

Este elemento contendrá la aplicación desarrollada por el administrador del sistema, la cual constituirá de varios factores para el cálculo de la F.C.M. y para el cálculo de las zonas de entrenamiento de cada persona. Además en la aplicación se podrá visualizar la Frecuencia Cardíaca al momento de hacer el ejercicio físico. Para este proyecto se utilizó un Smartphone “Samsung S3”.

Se prefiere el uso de Smartphone de gama alta, ya que la aplicación no tendrá ningún problema, y tienen q ser necesariamente con sistema operativo Android.

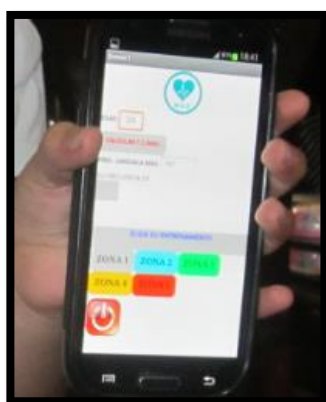


FIGURA 44: Smartphone Samsung S3

Fuente: Edison Flores

3.3 DISEÑO DE BLOQUES DEL SISTEMA

El diagrama de bloques indicará una representación gráfica de las partes fundamentales para el diseño del sistema de monitoreo de ritmo cardíaco para personas que realizan ejercicio físico aeróbico.

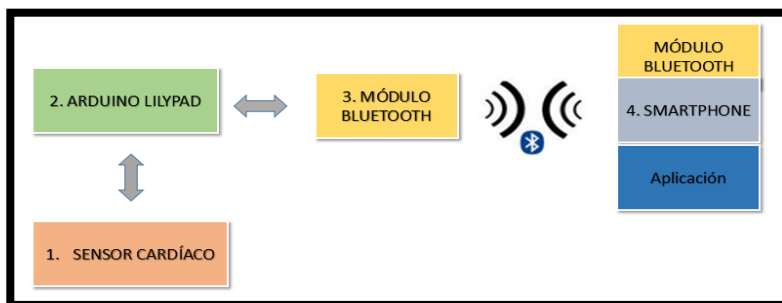


FIGURA 45: Diseño de bloques del Sistema de monitoreo de ritmo cardíaco

Fuente: Edison Flores

Como se observa en el diagrama de bloques, el sistema consta de 4 partes importantes que se detallarán a continuación:

1. Sensor Cardíaco.- Este sensor será ubicado en uno de los dedos de la mano (derecha o izquierda) del deportista. De preferencia será ubicado en el dedo pulgar o dedo índice. Este sensor permitirá adquirir los datos, en este caso el pulso cardíaco de cada persona, para posteriormente analizar esos datos y procesarlos según sea correspondiente el caso.
2. Arduino Lilypad.- Se utilizará una placa Arduino Lilypad, porque el sistema necesita características para textiles inteligentes y esta placa electrónica se acopla a ellas. Esta placa será cocida dentro de una muñequera, la placa debe adaptarse a las características de la muñequera como su tamaño y textura.
3. Modulo Bluetooth HC-05.- Este módulo se adapta a las condiciones del sistema, una comunicación inalámbrica de corto alcance y que sirve de manera correcta para la utilización conjunta con los Smartphone. Por su pequeño tamaño, se adecua cómodamente dentro de una muñequera.
4. Smartphone - Aplicación.- Se hará uso de cualquier dispositivo móvil inteligente, con Sistema Operativo Android, sugiere con versiones mayores a 4.1.2 en adelante.

Este elemento servirá para comunicarnos entre el módulo Bluetooth que está conectado a la placa Arduino con el dispositivo móvil. Aquí se alojará la aplicación, la misma que servirá para la interacción del usuario, como ingreso de edad, pulso normal, entre otros.

3.4 DIAGRAMA DE PINES DEL SISTEMA

Para facilidad de programación en la placa Arduino LilyPad y en el módulo Bluetooth HC-05, fue necesario realizar un diagrama de los pines, que represente su respectiva utilidad para poder configurarlos adecuadamente. Esto ayudará también a tener la idea para elegir un camino correcto al momento de coser con el hilo conductor.

3.4.1 DIAGRAMA DE PINES - ARDUINO LILYPAD

Para la placa Arduino LilyPad utilizaremos los pines de transmisión y recepción, estos tienen su propia identificación dentro de la placa electrónica. Los pines están escritos “Rx” como recepción y “Tx” como transmisión. Estos se encuentran bajo los espadines tipo macho.

También se utilizó los pines de Gnd (-) y Vcc (+), esto servirán para alimentar eléctricamente al sensor cardíaco. De ellos se obtiene 3,3 voltios, suficientes para encender y para un correcto funcionamiento del sensor. Para el sensor cardíaco se utilizó el pin analógico 0, este emite una corriente de 40mA, suficientes ya que el “Pulse Sensor” utiliza solo 4mA.

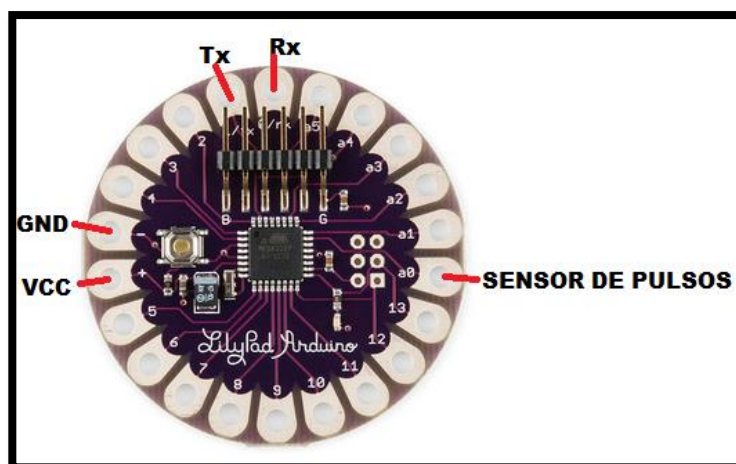


FIGURA 46: Diagrama de pines de la placa Arduino LilyPad

Fuente: Edison Flores

3.4.2 DIAGRAMA DE PINES - MÓDULO BLUETOOTH HC-05

Para el módulo Bluetooth, se utilizó 4 de las 6 pines, Se utilizó Gnd (-), Vcc (+), pin de transmisión (Tx) y pin de recepción (Rx). Los pines tipo macho, vienen soldados a la placa del módulo.

Se debe hacer un cableado cruzado entre transmisión del módulo y recepción de la placa y recepción del módulo y transmisión de la placa. Esto se debe hacer para que exista una comunicación correcta.

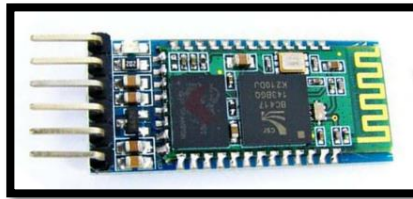


FIGURA 47: Módulo Bluetooth HC-05 (vista frontal)

Fuente: Edison Flores

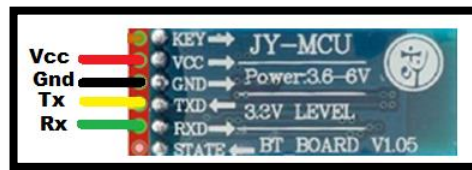


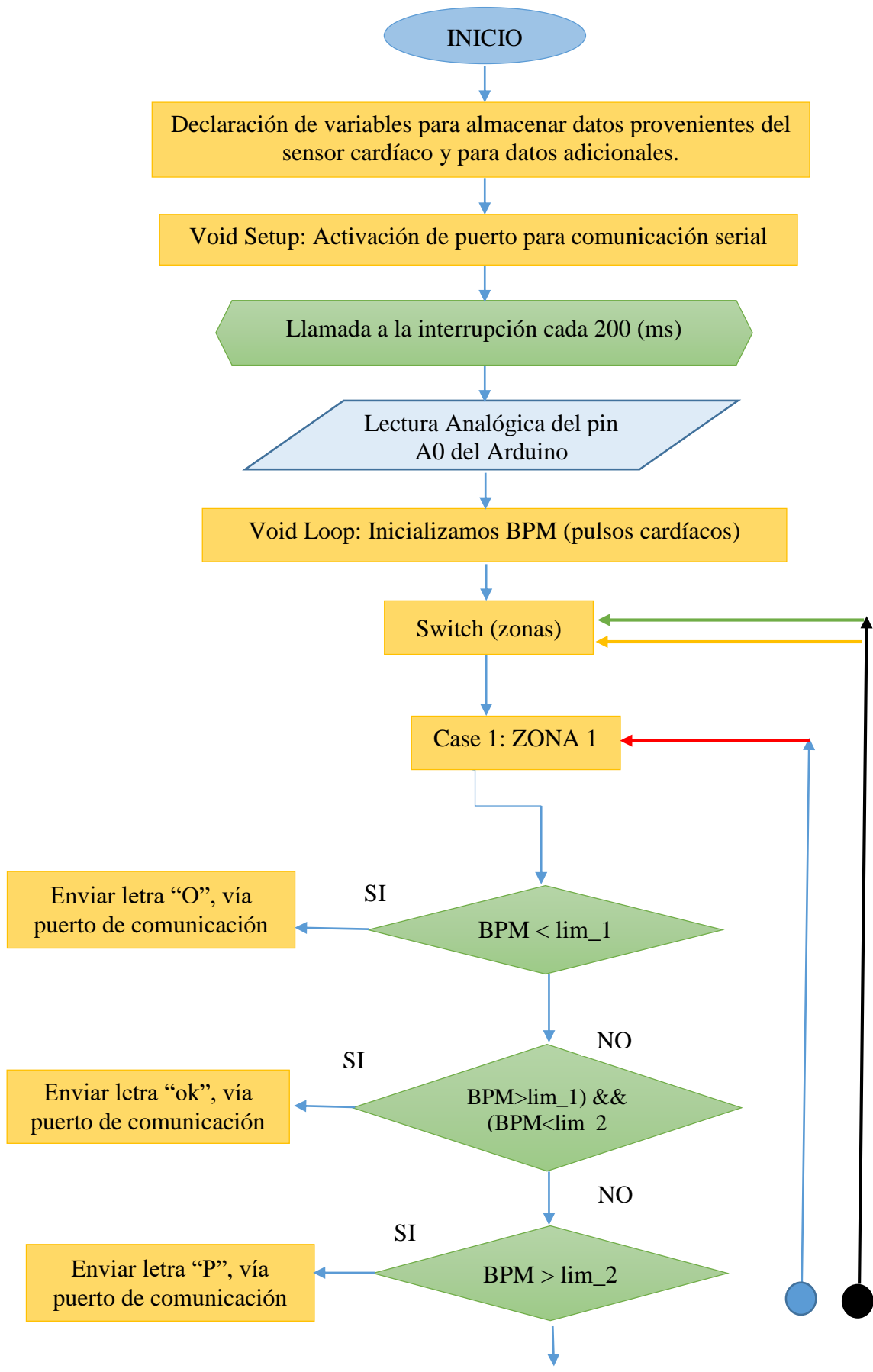
FIGURA 48: Diagrama de pines del módulo Bluetooth HC-05

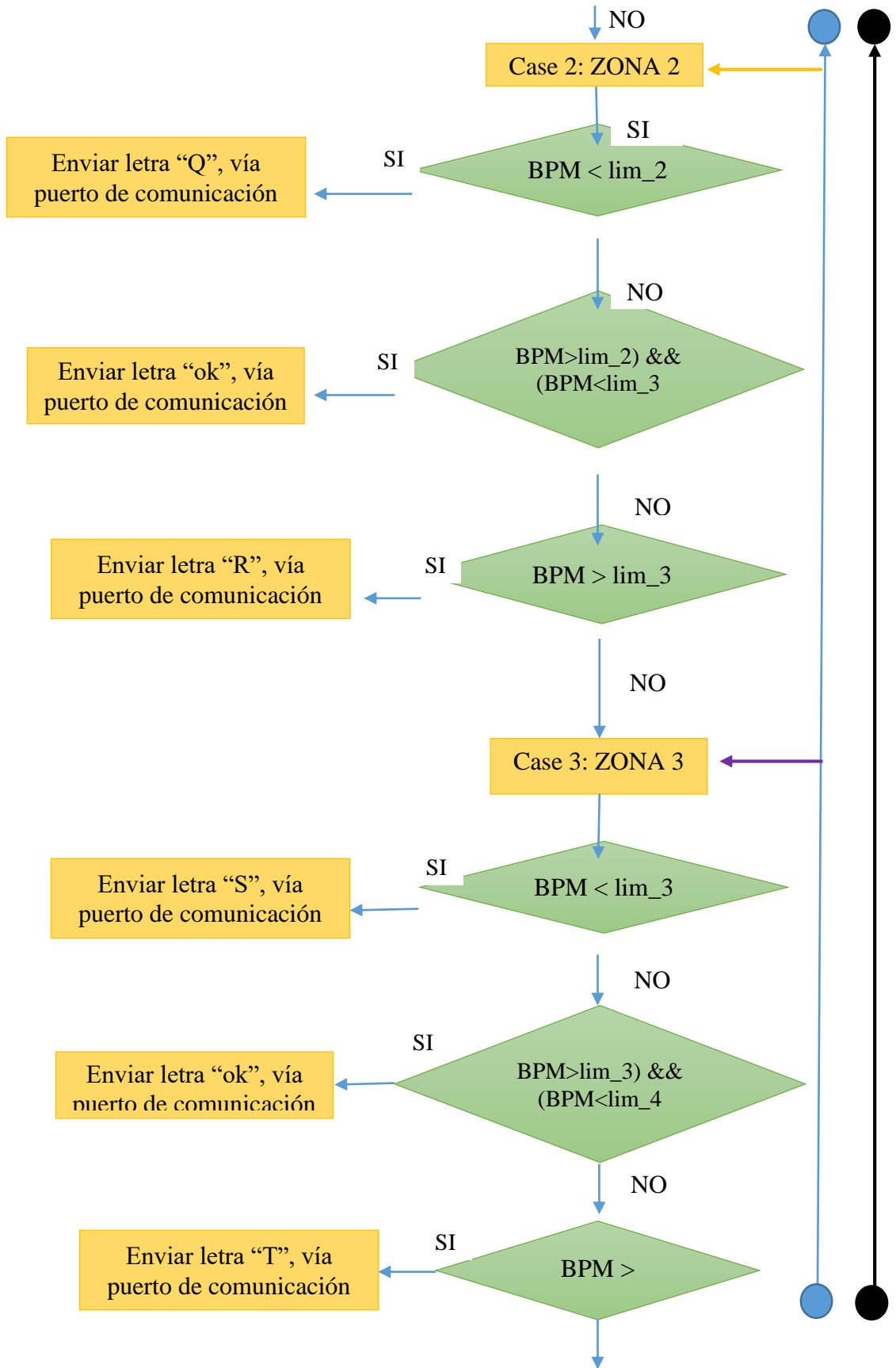
Fuente: Edison Flores

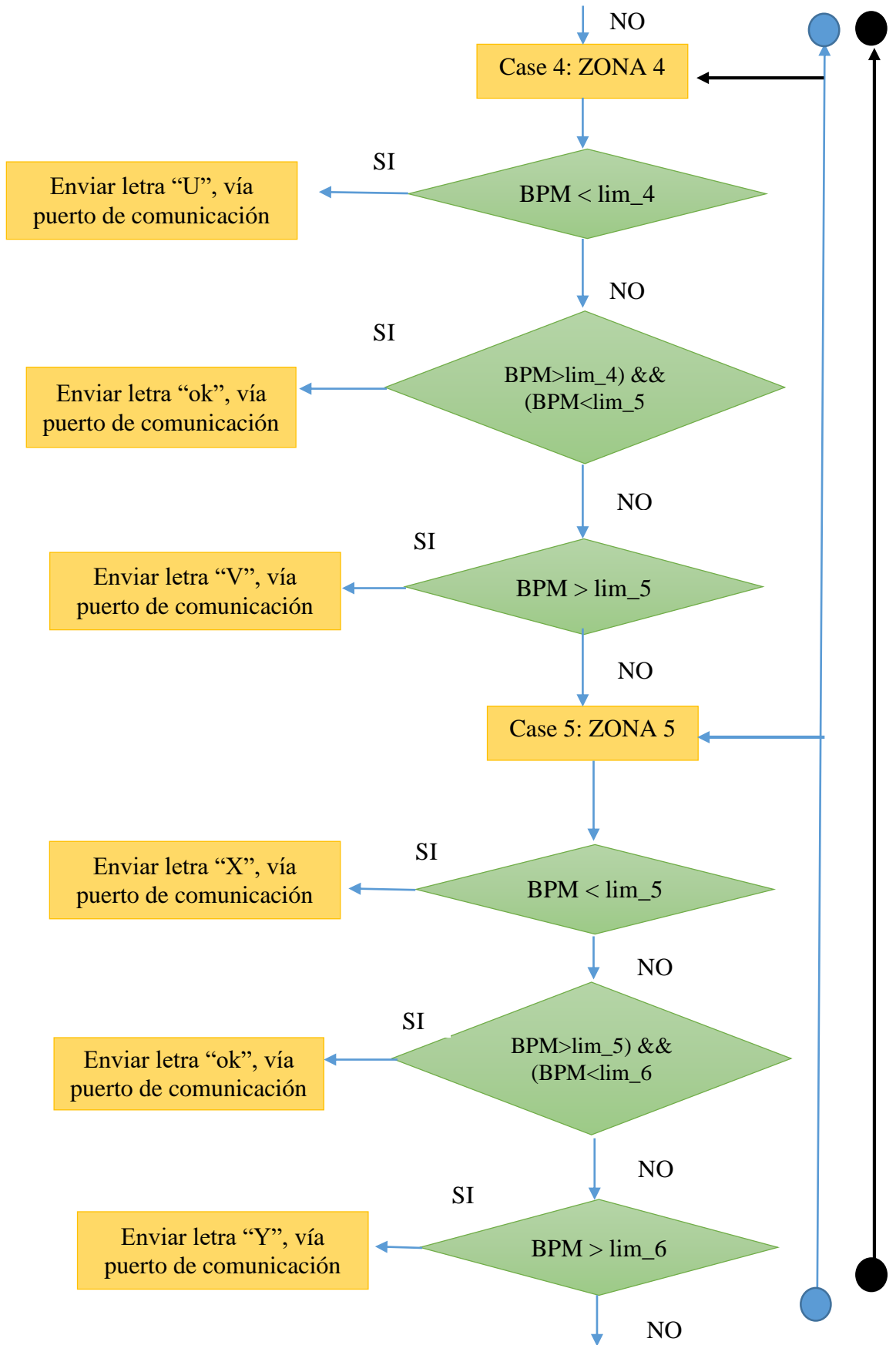
3.5 DIAGRAMAS DE FLUJO DEL SISTEMA

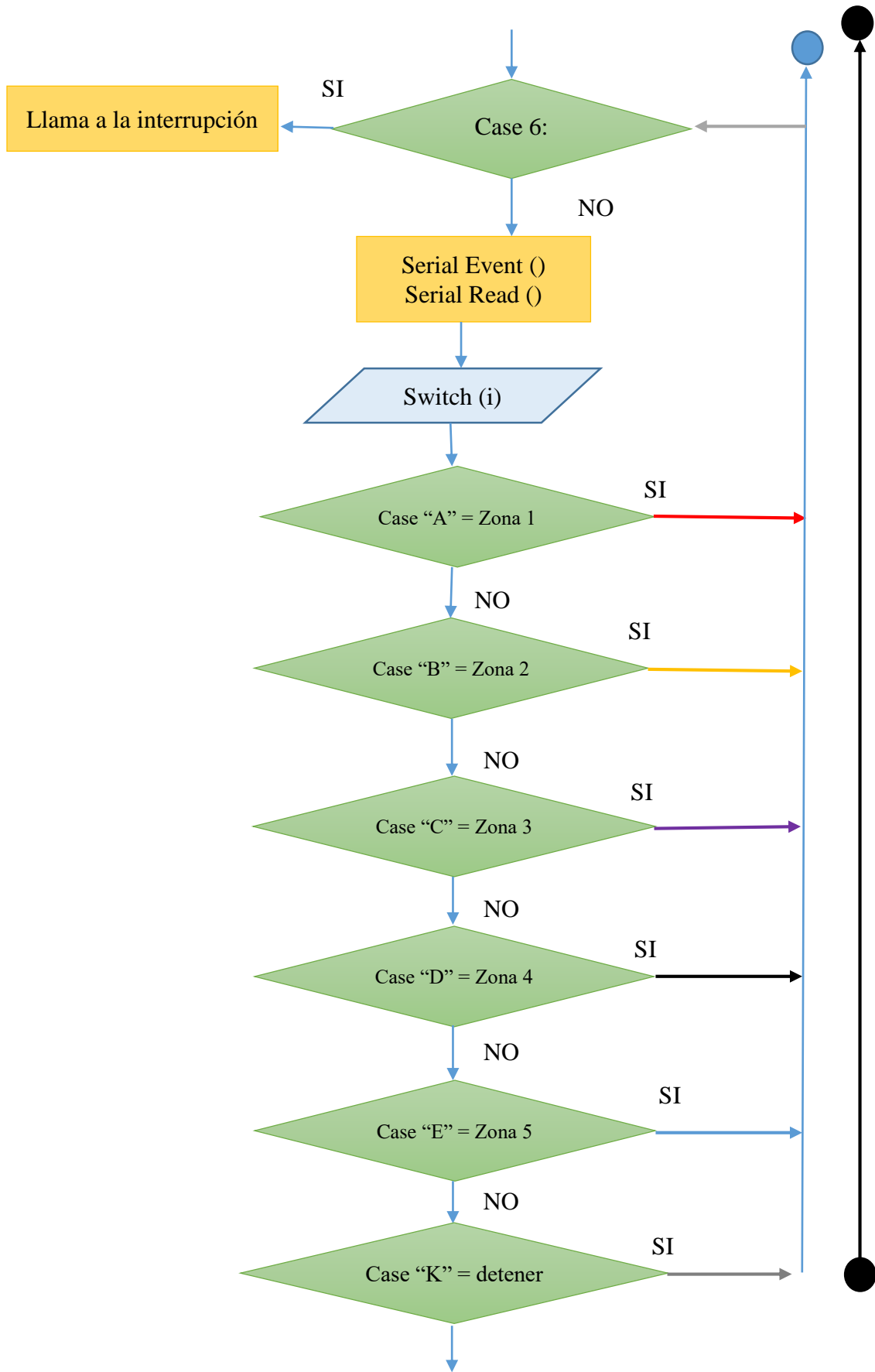
Como punto de partida se deberá establecer las variables, las cuales permitirán almacenar datos enteros, volátiles de tipo float. Estas variables almacenarán la edad, pulso normal, frecuencia cardíaca máxima, entre otros. Cuando los datos son almacenados y censados correctamente, se podrá comprobar mediante comunicación serial, o directamente en la aplicación.

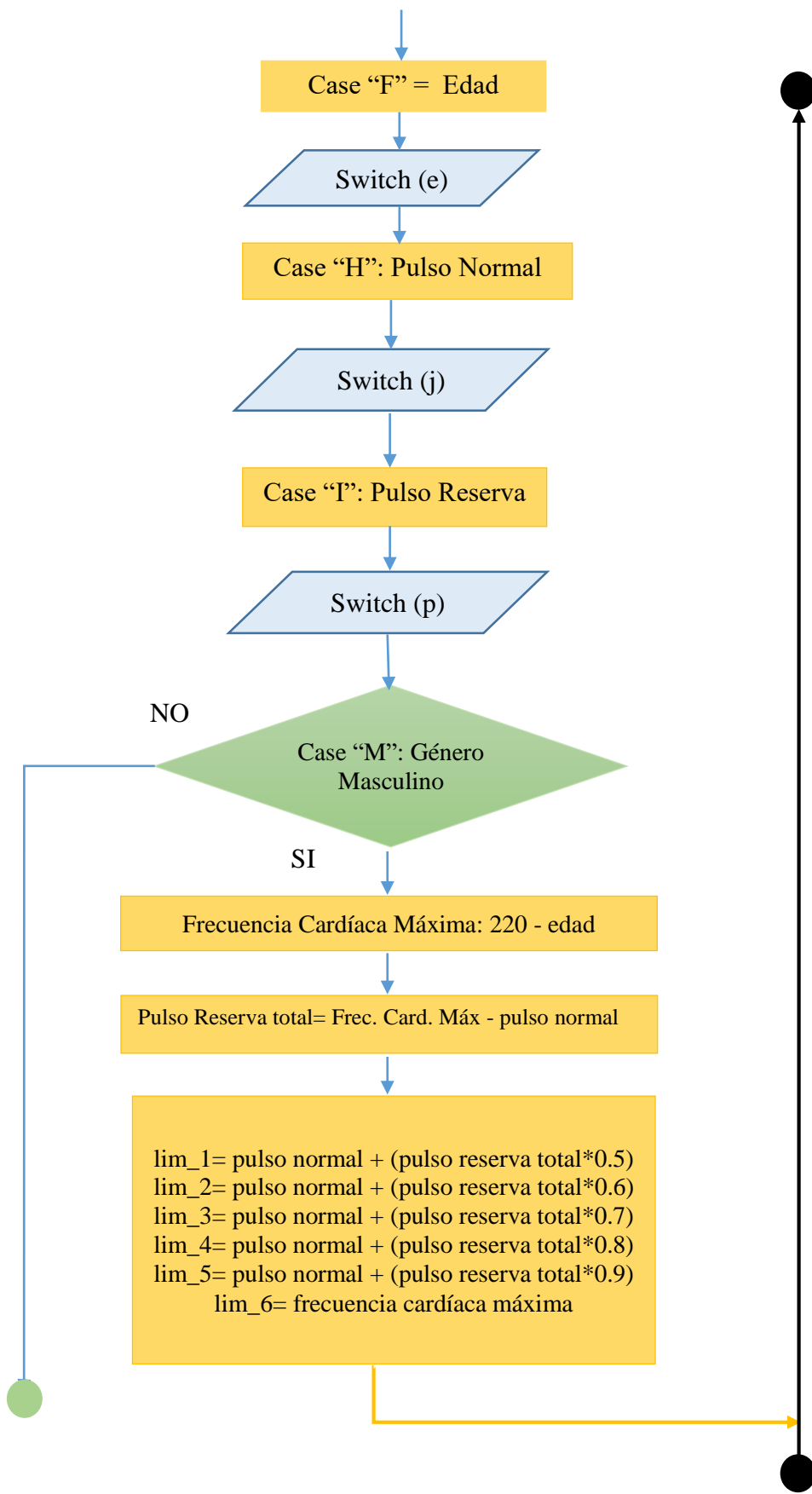
Para la programación de Arduino se cuenta con dos diagramas de flujo, uno como parte específica del funcionamiento de las zonas de entrenamiento, adquisición de datos como edad, pulsos, etc. El segundo diagrama de flujo hace referencia a la interrupción que se produce cada 2 (ms), en esta se configura para adquirir el pulso cardíaco del individuo.

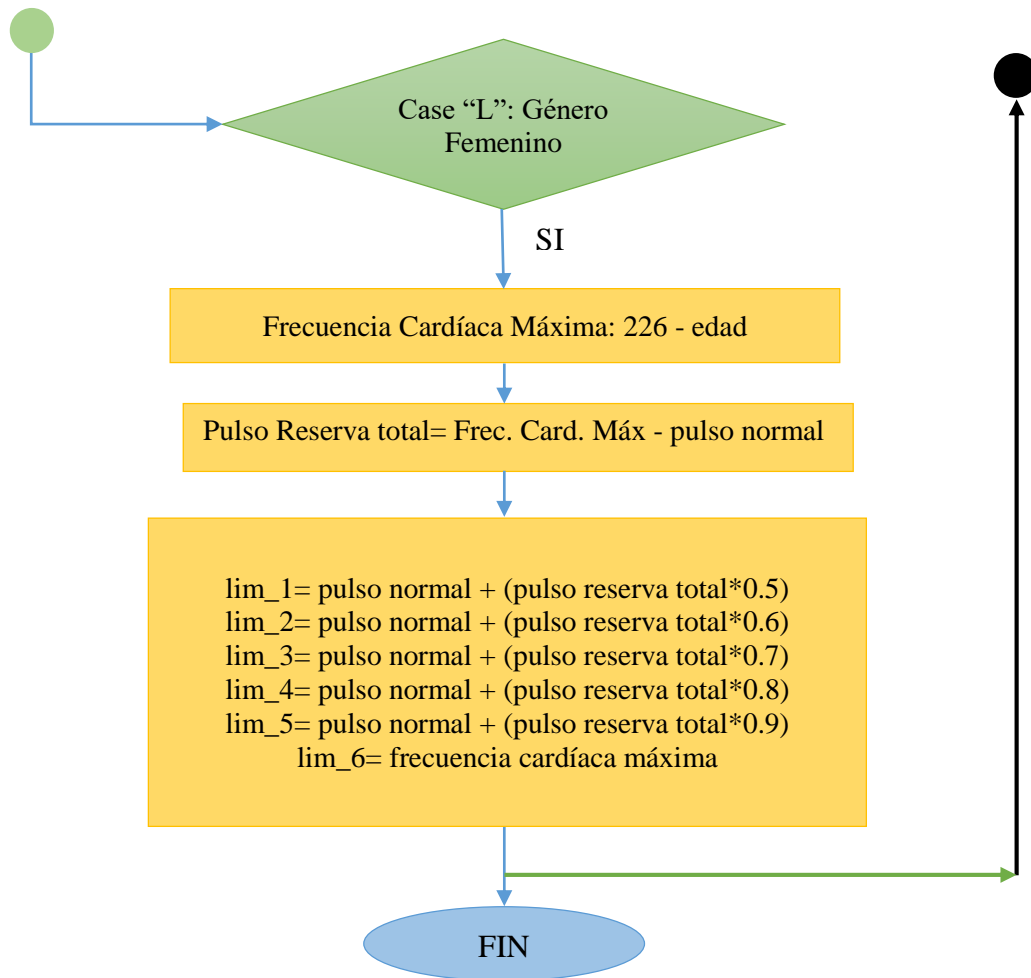








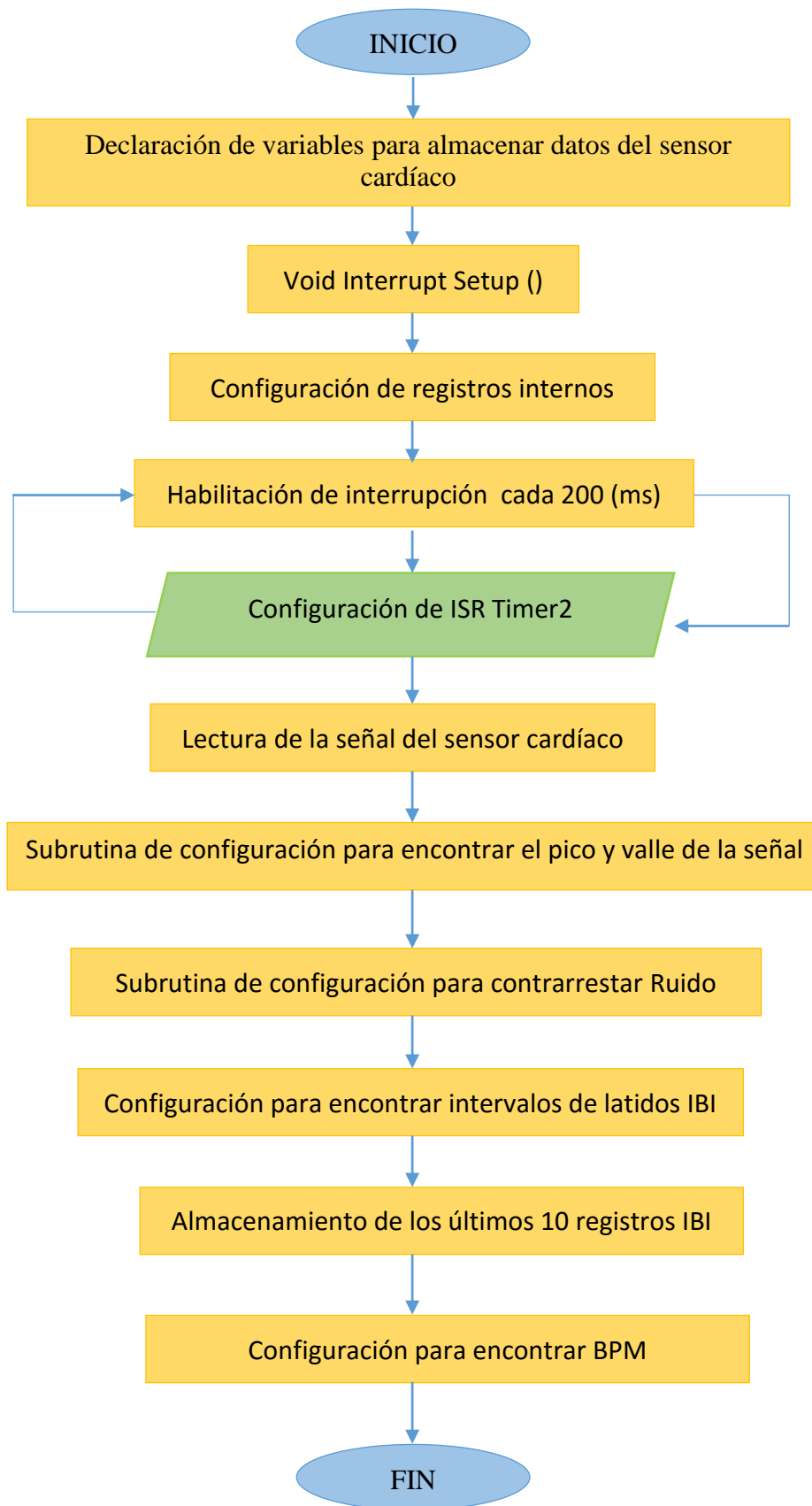




A continuación se presenta el diagrama de flujo que se utiliza en la interrupción del sistema de monitoreo de ritmo cardíaco

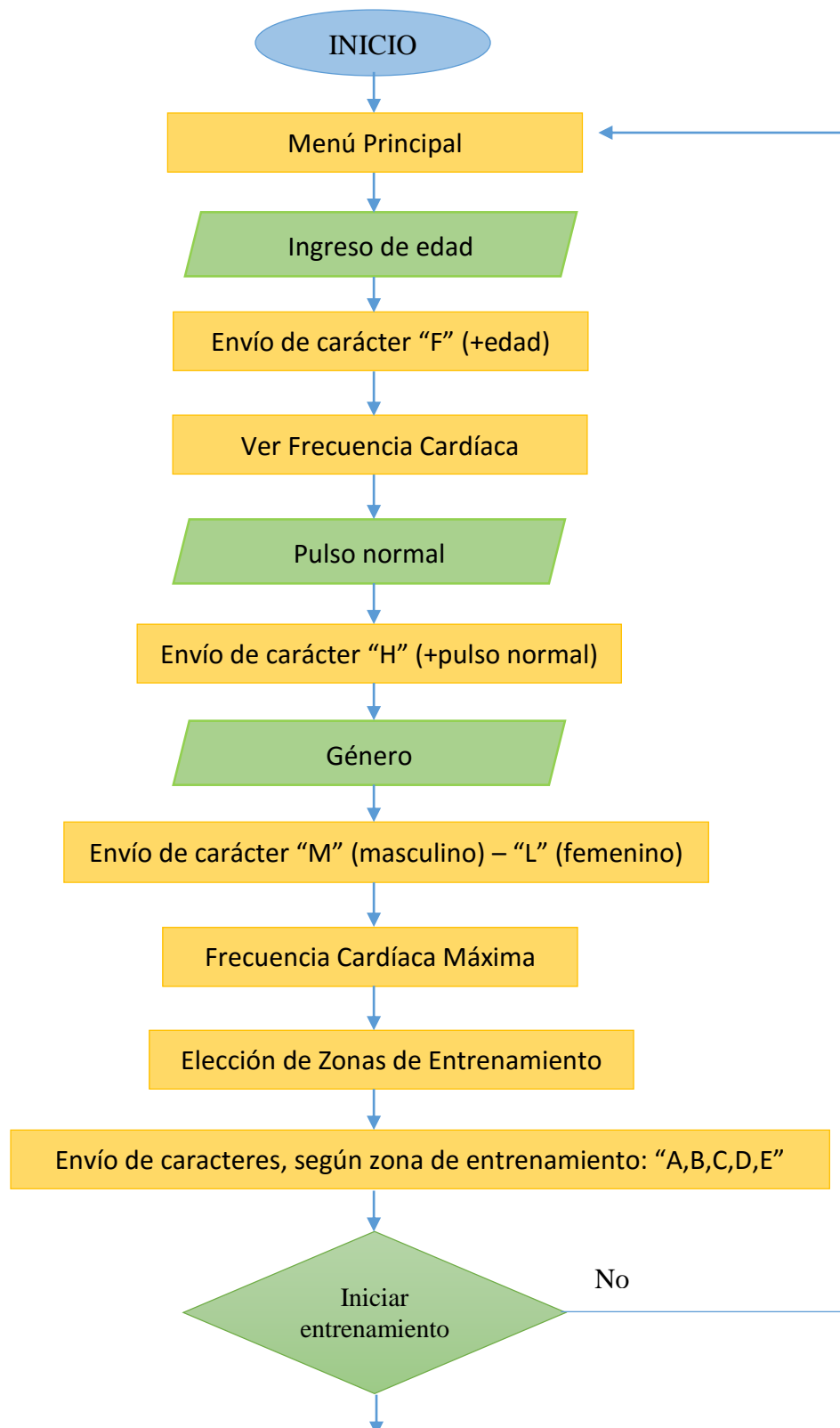
. Esta interrupción tiene como función específica leer el voltaje analógico que envía el "Pulse Sensor" y transformarlos a pulsos cardíacos directamente. Se utiliza una subrutina para el ruido, y se configura para almacenar los pulsos.

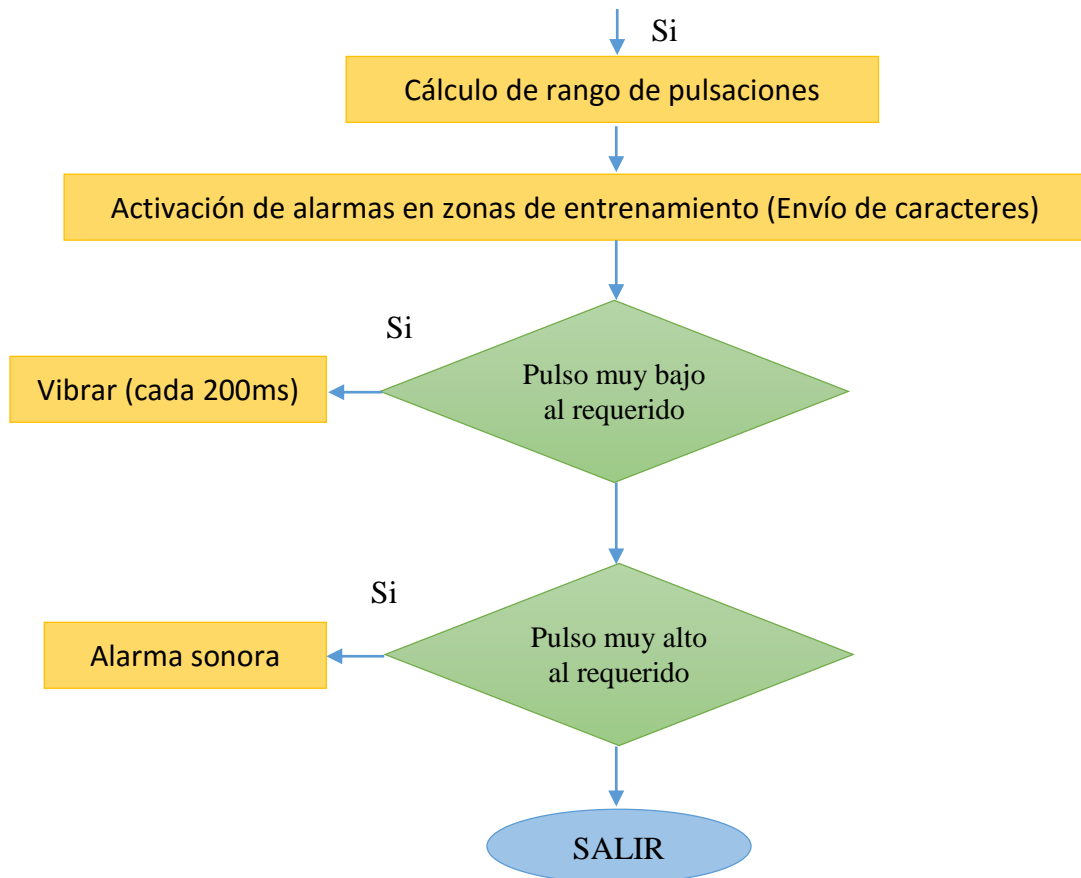
La interrupción fue configurada en 200 ms., fue realizada en base a prueba y error y resultado de tiempos y configuraciones por hardware en varias tomas de pulsos realizados en diferentes ocasiones y dando como resultados correctamente con el tiempo antes mencionado.



3.5.1 DIAGRAMA DE FLUJO DE LA APLICACIÓN EN ANDROID

Para la aplicación que utilizará el sistema de monitoreo cardíaco se realizó el siguiente diagrama de flujo.





Este diagrama de flujo, muestra como es la estructura para el diseño de la aplicación, que se encontrará en el Smartphone. Esta tendrá una interacción con la programación que se realizó en la placa Arduino Lilypad.

En cada zona de entrenamiento específica, tendrán un carácter único que servirá para reconocerla. Además si el pulso, es mayor o menor a los rangos de pulsaciones requeridos se emitirán las alarmas correspondientes. Se aclara que cada alarma en cada zona tiene un carácter único.

3.6 CARACTERIZACIÓN DEL SISTEMA

3.6.1 DATOS DE COMUNICACIÓN ARDUINO LILYPAD

Se asignó una letra, como datos de comunicación, para cada zona de entrenamiento, para el pulso cardíaco normal, pulso de reserva, edad, género, entre otros. Estos datos servirán para que la aplicación realizada en Android reconozca los límites de las zonas y procese la información según sea conveniente. A continuación se mostrará una tabla con los caracteres de comunicación que contiene la placa Arduino Lilypad.

TABLA 20: Caracteres Arduino Lilypad

CARÁCTER		
ENVÍA	RECIBE	SIGNIFICADO
	“A”	Entrenamiento en Zona 1
	“B”	Entrenamiento en Zona 2
	“C”	Entrenamiento en Zona 3
	“D”	Entrenamiento en Zona 4
	“E”	Entrenamiento en Zona 5
	“F”	Edad del deportista
	“K”	Detener el entrenamiento
	“H”	Pulso Normal
	“I”	Pulso de Reserva
	“M”	Género: Masculino
	“L”	Género: Femenino
“O”		Pulsaciones bajas en Zona 1
“P”		Pulsaciones altas en Zona 1
“Q”		Pulsaciones bajas en Zona 2
“R”		Pulsaciones altas en Zona 2
“S”		Pulsaciones bajas en Zona 3
“T”		Pulsaciones altas en Zona 3
“U”		Pulsaciones bajas en Zona 4
“V”		Pulsaciones altas en Zona 4
“X”		Pulsaciones bajas en Zona 5
“Y”		Pulsaciones altas en Zona 6

Fuente: Edison Flores

3.6.2 DATOS COMUNICACIÓN APLICACIÓN ANDROID

Los caracteres que utilizará la aplicación Android, que está alojada en el Smartphone son los que a continuación se presentan en la siguiente tabla:

TABLA 21: Caracteres de App Android

CARÁCTER		SIGNIFICADO
RECIBE	ENVIA	
	“A”	Entrenamiento en Zona 1
	“B”	Entrenamiento en Zona 2
	“C”	Entrenamiento en Zona 3
	“D”	Entrenamiento en Zona 4
	“E”	Entrenamiento en Zona 5
	“F”	Edad del deportista
	“K”	Detener el entrenamiento
	“H”	Pulso Normal
	“I”	Pulso de Reserva
	“M”	Género: Masculino
	“L”	Género: Femenino
“O”		Pulsaciones bajas en Zona 1
“P”		Pulsaciones altas en Zona 1
“Q”		Pulsaciones bajas en Zona 2
“R”		Pulsaciones altas en Zona 2
“S”		Pulsaciones bajas en Zona 3
“T”		Pulsaciones altas en Zona 3
“U”		Pulsaciones bajas en Zona 4
“V”		Pulsaciones altas en Zona 4
“X”		Pulsaciones bajas en Zona 5
“Y”		Pulsaciones altas en Zona 6

Fuente: Edison Flores

3.7 FIRMWARE DEL SISTEMA

Para la configuración del Firmware del sistema en Arduino, se inicia con la configuración de las variables, tanto en el programa principal como en la interrupción.

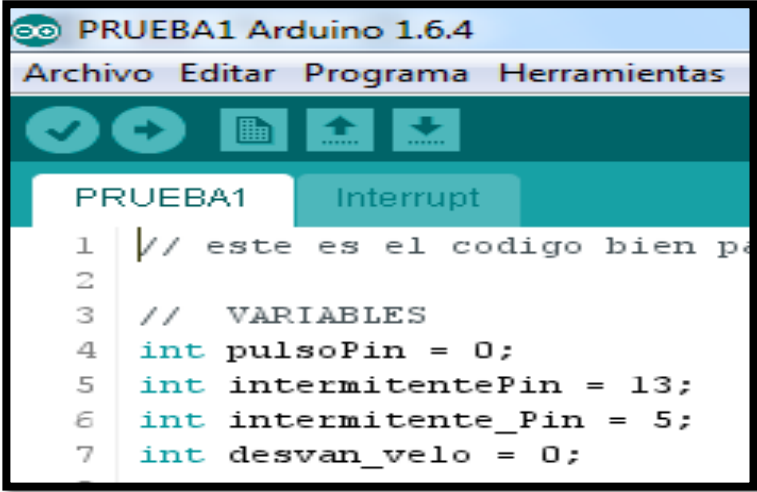
A continuación de esto se detallaran las configuraciones principales para obtener el sistema de monitoreo cardíaco listo para su utilización.

3.7.1 FIRMWARE DE ARDUINO LILYPAD

Para el sistema de monitoreo de ritmo cardíaco se utilizó el IDE de Arduino, con sus respectivos bloques de instrucciones, se utilizó variables, eventos seriales, switch, entre otros. Se detallará a continuación la programación de configuración del sistema de manera muy general. Las líneas completas de programación las podremos encontrar en el anexo1, con sus respectivos comentarios.

3.7.1.1 VARIABLES

Las variables sirven para nombrar y almacenar un valor para su uso posterior por el programa, tales como los datos de un sensor o un valor intermedio utilizado en un cálculo. Para el sistema de monitoreo de ritmo cardíaco se utilizó las variables para el sensor analógico, llamado "Pulsopin" el cual será activado en el pin A0, además variables para indicar el ritmo cardiaco mediante el encendido y apagado de un led (LED 13, incorporado en la placa Arduino lilypad). (Ver Anexo Nro. 1)

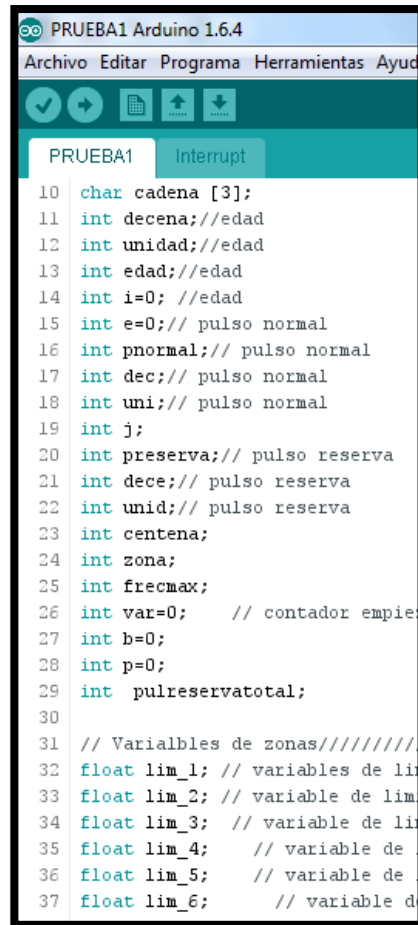


```
PRUEBA1 Arduino 1.6.4
Archivo Editar Programa Herramientas
PRUEBA1 Interrupt
1 // este es el codigo bien pa
2
3 // VARIABLES
4 int pulsoPin = 0;
5 int intermitentePin = 13;
6 int intermitente_Pin = 5;
7 int desvan_velo = 0;
```

FIGURA 49: Declaración de variables

Fuente: Edison Flores

Además se declararon variables adicionales para el cálculo de la edad, variables de tipo float para establecer los límites de cada zona de entrenamiento.

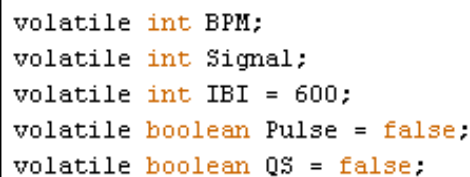


```
PRUEBA1 Arduino 1.6.4
Archivo Editar Programa Herramientas Ayuda
PRUEBA1 Interrupt
10 char cadena [3];
11 int decena;//edad
12 int unidad;//edad
13 int edad;//edad
14 int i=0; //edad
15 int e=0;// pulso normal
16 int pnormal;// pulso normal
17 int dec;// pulso normal
18 int uni;// pulso normal
19 int j;
20 int preserva;// pulso reserva
21 int dece;// pulso reserva
22 int unid;// pulso reserva
23 int centena;
24 int zona;
25 int frecmax;
26 int var=0; // contador empie
27 int b=0;
28 int p=0;
29 int pulreservatotal;
30
31 // Variables de zonas/////////
32 float lim_1; // variables de li
33 float lim_2; // variable de lim
34 float lim_3; // variable de li
35 float lim_4; // variable de
36 float lim_5; // variable de
37 float lim_6; // variable d
```

FIGURA 50: Declaración de variables tipo int y float

Fuente: Edison Flores

También se creó otras variables volátiles que se utilizan para encontrar el ritmo cardíaco y sus componentes.



```
volatile int BPM;
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;
```

FIGURA 51: Variables tipo "volatile"

Fuente: Edison Flores

3.7.1.1.1 DESCRIPCIÓN DE VARIABLES

Se mostrará en la tabla 21, las variables que se utilizó en la programación para el sistema de monitoreo de ritmo cardíaco. A continuación se mostrará las variables del programa principal.

TABLA 22: Descripción de Variables

Variable	Pin	Tipo	Descripción
pulsoPin	0	int	Pin de pulso (conexión del sensor, cable violeta)
intermitentePin	13	int	Led intermitente según el pulso
intermitente_Pin	5	int	Led de parpadeo (opcional)
desvan_velo	-	int	Velocidad de desvanecimiento (pin 5)
edad	-	int	Edad del Usuario
pnormal	-	int	Pulso normal del usuario
frecmax	-	int	Cálculo de frecuencia cardíaca máxima
preserva	-	int	Alamacena el pulso de reserva
pnormal	-	int	Almacena el pulso normal
pulreservatotal	-	int	Almacena la diferencia entre la F.C.M y el pulso normal.
cadena	-	char	Almacenamiento
decena	-	int	Decena de edad
unidad	-	int	Unidad de la edad
dec	-	int	Decena de pulso normal
uni	-	int	Unidad de pulso normal
centena	-	int	Centena de pulso de reserva
dece	-	int	Decena de pulso reserva
unid	-	int	Unidad de pulso reserva
i	-	int	(Variable).- Switch para la adquisición de datos de edad
j	-	Int	(Variable).- Switch para la adquisición de datos del pulso de reserva
e	-	int	(Variable).- Switch para la adquisición de datos para el pulso normal
p	-	int	(Variable).- Switch para la adquisición de datos para el género: masculino o femenino
zona	-	int	Variable para elección de zona de entrenamiento
lim_1	-	float	Almacenamiento del 50% de la F.C.M.
lim_2	-	float	Almacenamiento del 60% de la F.C.M.
lim_3	-	float	Almacenamiento del 70% de la F.C.M.
lim_4	-	float	Almacenamiento del 80% de la F.C.M.
lim_5	-	float	Almacenamiento del 90% de la F.C.M.
lim_6	-	float	Almacenamiento del 1000% de la F.C.M.
BPM	-	int	Latidos del Corazón (Ritmo cardíaco)
Signal	-	int	Datos de la señal de entrada
IBI	-	int	Determina el tiempo entre latidos para tomar la muestra
Pulse	-	boolean	Onda de pulso
OP	-	boolean	Verificación de Onda de pulso

Fuente: Edison Flores

3.7.1.2 DESCRIPCIÓN DEL PROGRAMA PRINCIPAL

Para el cálculo de la Frecuencia Cardíaca Máxima, existen varias fórmulas como se ha mencionado en el fundamento teórico del capítulo II, la forma tradicional, la cual es la más usada pero menos exacta por todos los Pulsómetros. Esta saca los rangos de pulsaciones de las zonas de entrenamiento a partir de la frecuencia cardíaca máxima (220-edad). Esta fórmula no toma en cuenta la forma deportiva actual del individuo, lo cual la hace inestable.

Para el sistema de monitoreo cardíaco se utilizará la Fórmula de Karvonen. En esta fórmula a diferencia de la tradicional, se añade el uso del pulso normal (opción para ver la forma deportiva actual del individuo), un pulso de reserva y la especificación del género.

Para la descripción del programa principal, lo estructuraremos por el orden de cada requerimiento específico para encontrar nuestra frecuencia cardíaca máxima, nuestro pulso normal y de reserva, entre otros.

3.7.1.2.1 EDAD

Se necesita la edad del usuario como primer parámetro, esta se ingresará en la aplicación que está en el Smartphone llamada "Pulsómetro_V1.2", y se enviará la edad al Arduino Lilypad, como carácter "F". Para el cálculo de la edad, el carácter "F", que está en la programación de Arduino, se encuentra configurado en el programa en un caso de un Serial Event conjuntamente con un swith (i).

```
245
246 ////////////////////////////////////////////////////////////////////EDAD//////////////////////////////////////////////////////////////////
247 case 'F':
248     cadena[1]=' ';
249     cadena[2]=' ';
250     Serial.println("EDAD"); // // Para comprobar
251     i++;
252     break;
253 }
254 break;
255
256 case 1:
257     cadena[i]=inChar;
258     decena=atoi(&cadena[i]);
259     i++;
260     break;
261
262 case 2:
263     cadena[i]=inChar;
264     unidad=atoi(&cadena[i]);
265     edad=(decena*10)+unidad; // edad
266     Serial.println (edad);
267
268     i=0;
269     break;
```

FIGURA 52: Configuración de la edad

Fuente: Edison Flores

3.7.1.2.2 FRECUENCIA CARDÍACA

La configuración de la adquisición de los pulsos cardíacos del individuo se obtiene de la interrupción, esta configuración se mostrará más adelante; pero dentro del programa principal se llama a la interrupción mediante la siguiente línea de código que se encuentra dentro del void setup ()

```
45
46 void setup(){
47 |
48   Serial.begin(9600);           // comunicacion serial velocidad----
49   interruptSetup();             //establece para leer pulsos de señal del sensor cada 2mS
50 }
```

FIGURA 53: Llamado a la interrupción desde programa principal

Fuente: Edison Flores

3.7.1.2.3 PULSO NORMAL

Para la adquisición del valor del pulso normal, la aplicación enviará este valor mediante un carácter "H" a la placa Arduino. Se encuentra dentro del serial event (), dentro del switch (e) con un case H.

```
PRUEBA1 $ Interrupt
272 //////////////////////////////////////////////////////////////////// PULSO NORMAL//////////////////////////////////////////////////////////////////
273 switch(e)
274 {
275   case 0:
276     cadena[e]=inChar;
277     switch(cadena[e])
278     {
279
280       case 'H':
281         cadena[1]=' ';
282         cadena[2]=' ';
283         Serial.println("Pnormal"); // // Para comprobar
284         e++;
285         break;
286     }
287     break;
288
289   case 1:
290     cadena[e]=inChar;
291     dec=atoi(&cadena[e]);
292     e++;
293     break;
294
295   case 2:
296     cadena[e]=inChar;
297     uni=atoi(&cadena[e]);
298     pnormal=(dec*10)+uni; // pulso normal
299     Serial.println (pnormal);
300     e=0;
301     break;
```

FIGURA 54: Configuración del pulso Normal

Fuente: Edison Flores

3.7.1.2.4 PULSO DE RESERVA

El pulso de reserva, se puede configurar o no en el Arduino Lilypad, la explicación es la siguiente: Para conseguir este valor se necesita la frecuencia cardiaca máxima y el pulso normal, estos son valores ya son obtenidos previamente, por lo tanto no serviría ponerlas las líneas de código. El pulso de reserva sale de la diferencia entre la frecuencia cardiaca máxima y el pulso normal.

NOTA: .Estas líneas de código solo servirán si el administrador del sistema, activa el cuadro de texto donde se calcula el pulso de reserva; caso contrario no es necesario

```
PRUEBA1 $ Interrupt
304 //////////////////////////////////////////////////// PULSO RES
305 switch(j)
306 {
307 case 0:
308   cadena[j]=inChar;
309   switch(cadena[j])
310   {
311   case 'I':
312     cadena[1]=' ';
313     cadena[2]=' ';
314     Serial.println("PRESERVA");
315     j++;
316     break;
317   }
318   break;
319 case 1:
320   cadena[j]=inChar;
321   centena=atoi(&cadena[j]);
322   j++;
323   break;
324 case 2:
325   cadena[j]=inChar;
326   dece=atoi(&cadena[j]);
327   j++;
328   break;
329 case 3:
330   cadena[j]=inChar;
331   unid=atoi(&cadena[j]);
332   preserva=(centena*100)+(dece*10)+ unid;
333   Serial.println(preserva);
```

FIGURA 55: Configuración Pulso de Reserva

Fuente: Edison Flores

3.7.1.2.5 GÉNERO Y LÍMITES

Para la elección del género, el usuario debe elegir en la aplicación. Cuando esta es elegida, envía un carácter a la placa Arduino, dependiendo si es masculino o femenino. Este parámetro es muy importante porque de esto depende una variación de la fórmula de la F.C.M. y por ende los rangos de pulsaciones. (Masculino: 220-edad, femenino: 226 –edad)

Cuando se elige el género, se generan todos los rangos de pulsaciones, es decir, este es el último paso dentro de la configuración. Aquí se generan todas las fórmulas y rangos. Estas variables estarán listas para su utilización cuando se ingrese a las diferentes zonas de entrenamiento. Se encuentra en el Switch (p). Su configuración es la siguiente:

```
PRUEBA1 $ Interrupt
343 }
344
345 //////////////// GENERO Y FÓRMULAS -.------
346
347 switch(p)
348 {
349 case 0:
350 cadena[p]=inChar;
351 switch(cadena[p])
352 {
353 case 'M':
354 Serial.println("masculino"); // Para comprobar
355
356 frecmax = 220 - edad;
357 pulreservatotal=frecmax-pnormal;
358 lim_1= pnormal+(pulreservatotal*0.5);
359
360 lim_2= pnormal+(pulreservatotal*0.6);
361
362 lim_3= pnormal+(pulreservatotal*0.7);
363
364 lim_4= pnormal+(pulreservatotal*0.8);
365
366 lim_5= pnormal+(pulreservatotal*0.9);
367
368 lim_6=pnormal+(pulreservatotal*1);
369
370
371 break;
```

FIGURA 56: Configuración de género masculino + fórmulas

Fuente: Edison Flores

```
PRUEBA1 $ Interrupt
383 break;
384
385
386 case 'L':
387 Serial.println("femenino"); // Para comprobar
388
389 frecmax = 226 - edad;
390 Serial.println(frecmax);
391 pulreservatotal=frecmax-pnormal;
392 lim_1= pnormal+(pulreservatotal*0.5);
393
394
395 lim_2= pnormal+(pulreservatotal*0.6);
396
397 lim_3= pnormal+(pulreservatotal*0.7);
398
399 lim_4= pnormal+(pulreservatotal*0.8);
400
401 lim_5= pnormal+(pulreservatotal*0.9);
402
403 lim_6=pnormal+(pulreservatotal*1);
404
405
406 break;
407
```

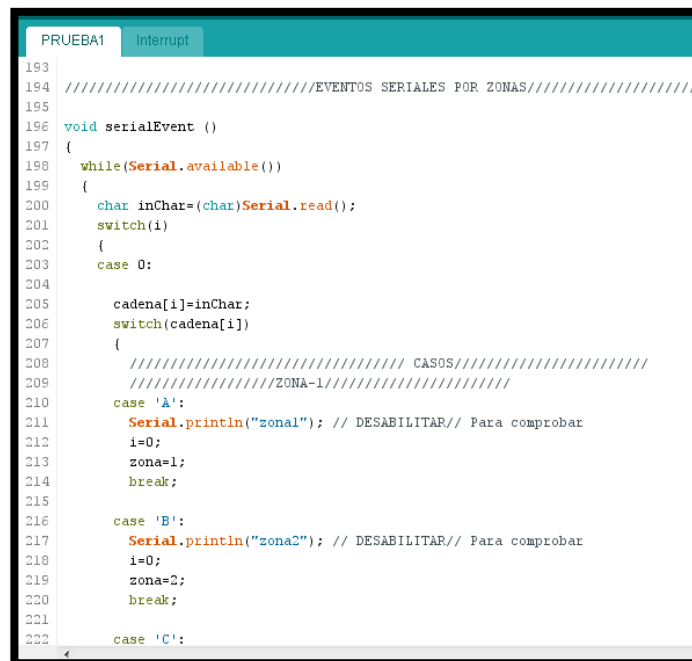
FIGURA 57: Configuración de género femenino + fórmulas

Fuente: Edison Flores

3.7.1.2.6 ZONAS DE ENTRENAMIENTO

Una vez que tenemos los límites o rangos de pulsaciones, debemos enmarcar a estos rangos dentro de las 5 zonas de entrenamiento; esto se logra cuando el usuario elige una zona determinada y son asignadas a la placa Arduino mediante caracteres. (Ver tabla 19)

Para esto, se realizó un Serial Event en donde utilizaremos “casos” para diferenciar una zona de otra.



```
PRUEBA1 Interrupt
193
194 //////////////////////////////////////////////////EVENTOS SERIALES POR ZONAS////////////////////////////////////
195
196 void serialEvent ()
197 {
198   while(Serial.available())
199   {
200     char inChar=(char)Serial.read();
201     switch(i)
202     {
203     case 0:
204
205       cadena[i]=inChar;
206       switch(cadena[i])
207       {
208         ////////////////////////////////// CASOS////////////////////////////////////
209         //////////////////////////////////ZONA-1////////////////////////////////////
210         case 'A':
211           Serial.println("zonal"); // DESABILITAR// Para comprobar
212           i=0;
213           zona=1;
214           break;
215
216         case 'B':
217           Serial.println("zona2"); // DESABILITAR// Para comprobar
218           i=0;
219           zona=2;
220           break;
221
222         case 'C':
```

FIGURA 58: Configuración zonas de entrenamiento

Fuente: Edison Flores

Cada zona de entrenamiento tiene rangos de pulsaciones establecidos, con características diferentes en el organismo humano al ejecutarse. Estos rangos deben ser respetados para cumplir un correcto entrenamiento en cada zona. Para que esto se cumpla, se crearon alarmas que emitirá el Smartphone conjuntamente con la aplicación cuando estemos por encima o debajo de los límites. Estas alarmas provienen cuando se superan los rangos inferiores y superiores en cada zona de entrenamiento. Para que se emita una alarma, se debe superar o estar por debajo del límite, cuando esto sucede se envía un carácter a la aplicación, esta reconoce el carácter y cumple una función específica de información para alertar al usuario que está por encima o debajo de los límites.

Cada límite es previamente calculado por la placa Arduino y se envían a la aplicación en forma de caracteres. (Ver caracterización del sistema). Para realizar la configuración se utilizó un Switch y casos dentro de él. Aquí se compara el BPM (latidos del corazón) con los límites.

```

PRUEBA1 Interrupt
54 void loop(){
55   Serial.println(BPM);          // IMPRIME LOS Datos DE BPM (pulso cardiaco)
56   delay (200);
57   //////////////////////////////////// SWITCH ZONAS-----
58   switch(zona){
59   case 1:////////////////////////////////// ZONA 1-----
60     led_latidos();
61     if (BPM<lim_1)
62     {
63       Serial.print("0");
64     }
65   }
66   if (
67     (BPM>lim_1)&&(BPM<lim_2))
68   {
69     Serial.print("ok");
70   }
71   if (BPM>lim_2)
72   {
73     Serial.print("P");
74   }
75   }
76   }
77   }
78   }
79   break;
80

```

FIGURA 59: Configuraciones de límites

Fuente: Edison Flores

3.7.1.3 INTERRUPCIÓN EN EL SISTEMA

El programa, tiene una interrupción llamada “Interrupt”, la cual tiene la función de obtener una frecuencia de muestreo para determinar la señal y numero de pulsaciones de cada individuo. Esta interrupción transforman las fluctuaciones de voltaje analógico a valores reales de pulsos cardíacos. Una de las funciones de las interrupción es encontrar los picos y valles de la onda, encontrar un punto medio, encontrar la amplitud y numero de pulsaciones (BPM) que se emiten por minuto.

Utiliza un temporizador configurado por hardware, el cual es de 8 bits. Esta interrupción funciona cada 2 milisegundos (este tiempo puede ser configurado por el administrador), con ellos obteniendo una frecuencia de muestreo de 500 Hz.

$$F = \frac{1}{T}$$

$$F = \frac{1}{2ms} = 0,0005 Mhz$$

$$F = 500 Hz$$

3.7.1.3.1 VARIABLES DE LA INTERRUPCIÓN

En esta parte de la programación se utilizará variables que ayudaran a encontrar la señal del pulso cardíaco, almacenar valores, temporizar pulsos, encontrar el pico y valle de la onda, entre otras. (Ver en ANEXO 2)

```

1
2 volatile int almacenar [10];
3 volatile unsigned long cont_muestras = 0;
4 volatile unsigned long intervalo_latido = 0;
5 volatile int P = 512;
6 volatile int T = 512;
7 volatile int tiem_latido = 512;
8 volatile int amplitud = 100;
9 volatile boolean primer_latido = true;
10 volatile boolean segundo_latido = true;
11

```

FIGURA 60: Inicialización de variables en la interrupción

Fuente: Edison Flores

3.7.1.3.2 DESCRIPCIÓN DE VARIABLES EN LA INTERRUPCIÓN

Para la interrupción del programa principal, se utilizaron las siguientes variables que se presentan en la tabla 24.

TABLA 23: Variables de la interrupción

Variable	Tipo	Descripción
almacenar	int	Almacena últimos diez valores IBI
cont_muestras	unsigned long	Utilizado para determinar temporización de los impulsos
intervalo_latido	unsigned long	Utilizado para determinar el intervalo entre latidos
P	int	Utilizado para encontrar pico en la onda de pulso
T	int	Utilizado para encontrar mínimo en la onda de pulso
tiem_latido	int	Utilizado para determinar momento instantáneo de los latidos del corazón
amplitud	int	Utilizado para mantener la amplitud de la forma de onda de pulso
primer_latido	boolean	Porcentaje de inicio de BPM
segundo_latido	boolean	Porcentaje de inicio de BPM

Fuente: Edison Flores

3.7.1.3.3 CONFIGURACIÓN DE VOID INTERRUPTSETUP ()

En esta parte se inicializa un “Timer2”, configurando los registros internos, para posteriormente utilizarlos cada 2 ms. Este “Timer2” es un temporizador configurado por hardware, es de 8 bits. Se lo configura para ser utilizado de manera global por todo el programa con la instrucción “sei”.

Los timers tienen números específicos, pueden configurarse y generar frecuencias por hardware, los cuales son:

TABLA 24: Lista de Timers

Timer#	Interrumpe en
Timer0	2 KHz
Timer1	1 Hz
Timer2	8 KHz
Timer3	64 KHz
Timer4	256 KHz

Fuente: Edison Flores

En el sistema se utilizó el Timer número 2. Los registros configurados en nuestro sistema de monitoreo fueron:

TCCR2A = 0x02;

TCCR2B = 0x05;

OCR2A = 0X7C;

TIMSK2 = 0x02;

Se probó los registros con sus configuraciones en un “Arduino MEGA 2500” que tiene un microcontrolador “ATMEGA 2560” y oscilador de 16MHZ, aquí las configuraciones fueron las siguientes:

TCCR2A = 0x02;

TCCR2B = 0x06;

OCR2A = 0X7C;

```
TIMSK2 = 0x02;
```

Nota: Cuando se utilice el firmware para una placa Arduino con microcontrolador "ATMEGA 168" y oscilador a 8MHz, se debe tener en cuenta que la configuración del registro TCCR2B debe ser igual a 0x05. Hay q tener en cuenta que cuando se utiliza el timer2, se habilita el PWM²¹ y los pines 3 y 11 no funcionarán.

```
13 void interruptSetup(){
14     // Inicializa Timer2
15     TCCR2A = 0x02;    //
16     TCCR2B = 0x05;    //
17     OCR2A = 0x7C;    //
18     TIMSK2 = 0x02;    //
19     sei();           //
20
```

FIGURA 61: Configuración de Registros

Fuente: Edison Flores

Esta configuración en el control de registro "B" se hace que el Arduino se establezca con un pre-escaler a 128 divisiones (TCCR2B = 0x05). Si mantengo a TCCR2B = 0x06 el divisor es de 256.

Con la línea de código OCR2A = 0x7C, se ajusta la configuración a una frecuencia de muestreo de 124 a 500 Hz

3.7.1.3.4 CONFIGURACIÓN DEL ISR (TIMER2_COMPA_VECT)

Se configurará un vector llamado ISR²², en donde se almacenará la señal se, se encontrará los picos y valles con los que se procesará la señal. Cuando el Arduino Lilypad, está encendido y funcionando con el Sensor de Pulsos conectado a pin analógico 0, constantemente se leerá el valor cada 2 ms y buscará el latido del corazón.

```
ISR(TIMER2_COMPA_vect){
cli();
Signal = analogRead(pulsoPin);
cont_muestras += 2;
```

²¹ PWM (Pulse-width modulation).- Modulación por ancho de pulso

²² ISR (Interrupt Service Routines).- Rutinas de servicio de interrupción

```
int Ruido = cont_muestras - intervalo_latido;
```

```
ISR(TIMER2_COMPA_vect){
  cli();
  Signal = analogRead(pulsoPin);
  cont_muestras += 2;
  int Ruido = cont_muestras - intervalo_latido;
```

FIGURA 62: Configuración del vector ISR

Fuente: Edison Flores

Esta función se llama cada 2 milisegundos. Lo primero que debe hacer es tomar una lectura analógica del sensor de pulso. A continuación, incrementamos el variable “cont_muestras”. La variable “cont_muestras” es lo que usamos para mantener la referencia del tiempo. La variable “int Ruido” ayudará a evitar el ruido después.

A continuación, hacemos un seguimiento de los valores más altos y más bajos de la onda, para obtener una medida exacta de la amplitud.

```
32   if(Signal < tiem_latido && Ruido > (IBI/5)*3){
33       if (Signal < T){
34           T = Signal;
35       }
36   }
37
38   if(Signal > tiem_latido && Signal > P){
39       P = Signal;
40   }
41
```

FIGURA 63: Configuración de punto bajo y alto de la onda

Fuente: Edison Flores

Aquí la variable P “Pico” y T “Valle” esperan y valores. La variable se inicializa con un umbral de 512 (en el centro del rango analógico) y los cambios durante el tiempo de ejecución para el seguimiento de un punto en el 50% de la amplitud. Hay un período de tiempo de 3/5 IBI que debe transcurrir antes de “T” se actualiza como una forma de evitar el ruido y los falsos positivos del corte dicroico en el sensor.

Después se emiten unas configuraciones para evitar el ruido a altas frecuencias y para activar “intermitentePin” (pin13) (opcional, para activar led en la placa)

```
44 if (Ruido > 250){
45   if ( (Signal > tiem_latido) && (Pulse == false) && (Ruido > (IBI/5)*3) )
46   {
47     Pulse = true;
48     digitalWrite(intermitentePin,HIGH);
49     IBI = cont_muestras - intervalo_latido;
50     intervalo_latido = cont_muestras;
51
```

FIGURA 64: Configuración para evitar ruido a altas frecuencias

Fuente: Edison Flores

La explicación de la figura 60 es que antes de considerar la búsqueda de un latido del corazón, una cantidad mínima de tiempo tiene que pasar. Esto ayudará a evitar el ruido de alta frecuencia. En nuestro sistema se configuro con un tiempo de 250 milisegundos a la variable “Ruido”.

Luego calculamos el tiempo entre latidos, haciendo una diferencia entre “cont_muestras” y el “intervalo_latido”, nos dará un tiempo en milisegundos. Posteriormente actualizaremos el valor de “intervalo_latido”.

La siguiente configuración se utilizará para asegurarse de que comencemos con un valor de BPM²³ real en el arranque.

```
if(primer_latido){
    primer_latido = false;
    return;
}
if(segundo_latido){
    segundo_latido = false;
    for(int i=0; i<=9; i++){
        almacenar[i] = IBI;
    }
}
```

FIGURA 65: Configuración de valor real de BPM

Fuente: Edison Flores

La variable “primer_latido” se inicializa como verdadera y la variable “segundo_latido” se inicializa como falsa en el posteriormente, por lo que la primera vez que nos encontramos con un ritmo cardíaco y en vector ISR, emiten un retorno condicional de la variable “primer_latido”. Eso va a terminar emitiendo el primer IBI²⁴, la cual significará como un pulso erróneo o mal tomado.

Cuando se encuentre por segunda vez un ritmo cardíaco, podemos confiar en el IBI, y será utilizado para crear el vector con el fin de comenzar con un BPM más preciso. El BPM se deriva de un promedio de los últimos 10 valores del IBI, de ahí la necesidad de crear el vector.

²³ BPM (beat per minute).- Pulsaciones por minuto

²⁴ IBI (inter beat interval).- Intervalo entre latidos

Ahora calcularemos el BPM de los últimos 10 valores IBI

```
64   word runningTotal = 0;
65 |
66   for(int i=0; i<=8; i++){
67       almacenar[i] = almacenar[i+1];
68       runningTotal += almacenar[i];
69   }
70
71   almacenar[9] = IBI;
72   runningTotal += almacenar[9];
73   runningTotal /= 10;
74   BPM = 60000/runningTotal;
75   OP = true;
76
77 }
```

FIGURA 66: Configuración de RunningTotal

Fuente: Edison Flores

Después configuramos la variable “runningTotal”, para tomar datos de IBI, entonces el contenido del vector se desplaza un espacio y añade a “runningTotal”. El datos más antiguo del IBI (hace 11 latidos) se cae de la posición 0, y el nuevo dato IBI se pone en la posición 9. Entonces es un proceso sencillo para promediar la matriz y el cálculo de BPM. Lo último que hay que hacer es establecer el indicador a la bandera “OP”. Esta bandera se activa cuando se encuentra un pulso y el BPM se está actualizando.

Cuando la variable “pulse” fue declarada como verdadera, durante la subida de la señal del sensor de pulso, encontramos el ritmo, por encima, y cuando la señal cruza umbral inferior, podemos averiguar que el pulso se ha terminado.

A continuación, se mide la amplitud de la onda que acaba de pasar, y “tiem_latido” se actualiza con la nueva marca de 50%. Luego “P” y “T” se restablecen al nuevo umbral. La configuración está lista para encontrar el siguiente latido.

```
80
81   if (Signal < tiem_latido && Pulse == true){
82       digitalWrite(intermitentePin,LOW);
83       Pulse = false;
84       amplitud = P - T;
85       tiem_latido = amplitud/2 + T;
86       P = tiem_latido;
87       T =tiem_latido;
88   }
89
```

FIGURA 67: Configuración de variable pulse

Fuente: Edison Flores

Por último se configura, cuando no existan latidos. El sensor seguirá funcionando, pero dará valores en los umbrales superior e inferior

```
if (Ruido > 2500){
    tiem_latido = 512;
    P = 512;
    T = 512;
    intervalo_latido = cont_muestras;
    primer_latido = true;
    segundo_latido = true;
}
```

FIGURA 68: Configuración de "sin latidos"

Fuente: Edison Flores

Si no hay un latido en 2,5 segundos las variables utilizadas para encontrar los latidos del corazón se reinician a los valores por defecto en la configuración. Una especie de reset. Aquí finaliza el ISR.

La programación completa tanto del programa principal como de la interrupción se encuentra en los anexos 1 y 2 respectivamente, Aquí se podrá leer con detalle por cada línea de código

3.7.2 FIRMWARE DE ANDROID

La aplicación se realizó en MIT App Inventor2, está diseñada para cualquier Smartphone con Sistema Operativo Android desde la versión 4.1.2 (esta es una de las versiones más antiguas, así que no habrá problema en cualquier Smartphone actual). La aplicación se realizó con una programación por bloques.

A continuación se dará a conocer la configuración de los bloques de programación que se usaron. Se detallará la configuración por cada "Screen", hay que mencionar que el "Screen1", es el "Menú Principal" y del "Screen 2 al 6" la configuración es repetitiva. En estos últimos "Screen" mencionados, cambia los caracteres al ingresar a cada zona y los caracteres para emitir las alarmas por los límites de rangos de pulsaciones.

3.7.2.1 CONFIGURACIÓN EN EL SCREEN 1 "MENÚ PRINCIPAL"

En este "Screen 1", que es el Menú principal del Sistema, es una pantalla donde el usuario que va a realizar la práctica de ejercicio físico; antes de realizarlo, ingresa su edad, visualiza su frecuencia cardíaca en ese momento y en base a esto ingresa su pulso

normal. También hace la elección de su género y puede calcular su frecuencia cardíaca máxima. Después aceptará el mensaje de tiempos de calentamiento y luego elegirá el de entrenamiento que desee.



FIGURA 69: Diseño del menú principal

Fuente: Edison Flores

Dentro de esta primera pantalla, la configuración es la siguiente:

3.7.2.1.1 CONFIGURACIÓN DE CONEXIÓN AL MÓDULO BLUETOOTH – HC05

Se utiliza un “Bluetooth client1” para activar dicha comunicación.



FIGURA 70: Llamada a Bluetooth client

Fuente: Edison Flores

Para la conexión inalámbrica entre el módulo HC05 y Smartphone, se programa en la aplicación para que se conecte a la MAC del módulo, la cual es la siguiente:

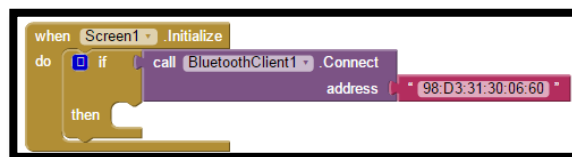


FIGURA 71: Configuración para la conexión a la MAC del módulo HC-05

Fuente: Edison Flores

Al ingresar a este screen, se conectará automáticamente al módulo Bluetooth, es decir, que cuando habrá la aplicación, automáticamente los pulsos cardíacos aparecerán en la pantalla.

3.7.2.1.2 CONFIGURACIÓN DE LOS TIMER

Se configuraron dos timer que sirven para mostrar y borrar los datos de interacción con la placa Arduino, es decir, los pulsos cardíacos. El timer1 llamado “Clock1_dato” analiza si existen datos que sean mayores a 0 que se estén transfiriendo. Si hay estos datos se almacenan en un cuadro de texto, que será el “TextBox4_fc”. Está configurado cada 1000 (ms).

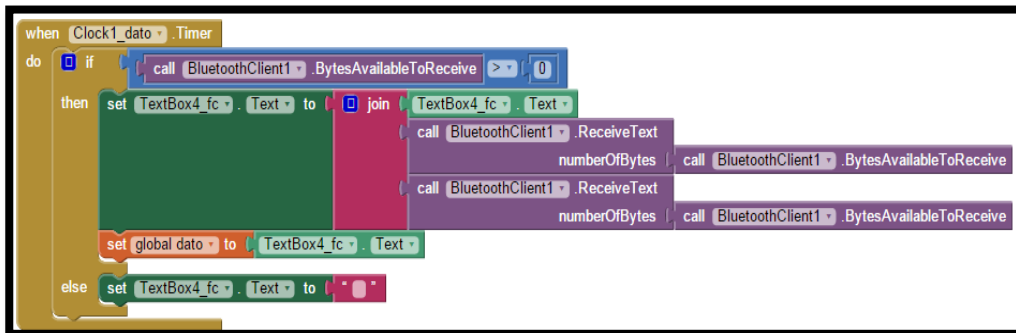


FIGURA 72: Configuración Timer1 en aplicación

Fuente: Edison Flores

Además se configuró para que el pulso se observe en el TextBox4_fc” y a su vez que se guarde en un variable global.



FIGURA 73: Configuración de variable global en menú principal (aplicación)

Fuente: Edison Flores

En el timer 2 llamado “Clock1_borrar” se borra el dato constantemente del “TextBox4_fc” cada 1300 (ms). Esto se realiza para que el dato este constantemente visualizándose, ya que de lo contrario el cuadro de texto se llena de datos y no se puede visualizar los demás pulsos.

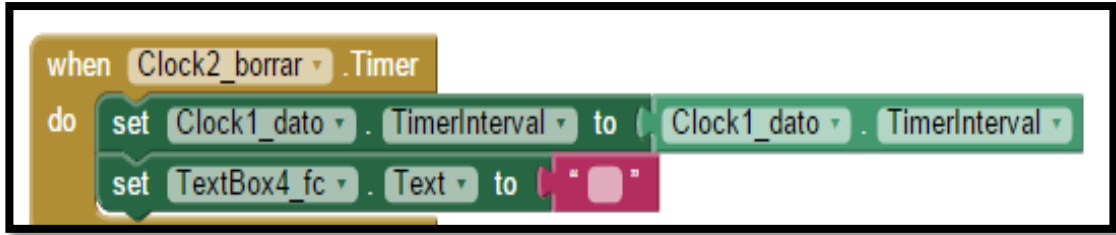


FIGURA 74: Configuración Timer2 en aplicación

Fuente: Edison Flores

3.7.2.1.3 INGRESO DE LA EDAD

El usuario podrá ingresar la edad fácilmente su edad, se configuró para que cuando el usuario haga clic sobre el cuadro de texto de la edad aparecerá un teclado numérico, lo que facilita el ingreso e interacción.



FIGURA 75: Teclado numérico para ingreso de edad

Fuente: Edison Flores

Esta es la opción que se debe habilitar para que nos aparezca el teclado numérico. Se encuentra en las propiedades del cuadro de texto

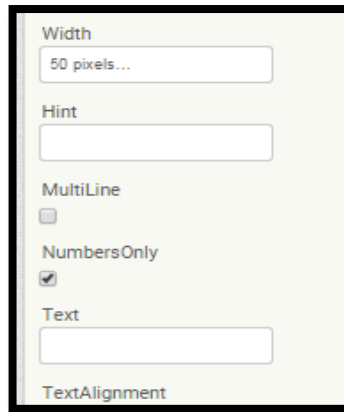


FIGURA 76: Configuración para habilitar teclado numérico

Fuente: Edison Flores

Posteriormente de ingresar la edad, se debe hacer clic sobre el botón “Aceptar”, esto es para indicar que se ingresó la edad correctamente y también para enviar, a la paca Arduino, la edad junto con un carácter. Este carácter indica específicamente que es la edad del usuario. La letra que se envía es la “F”.

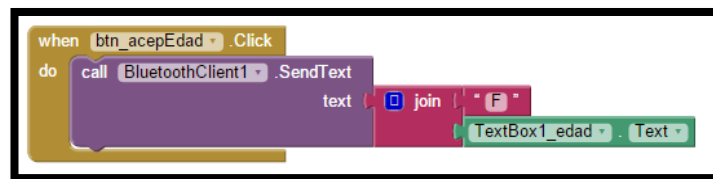


FIGURA 77: Configuración de edad en la aplicación

Fuente: Edison Flores

3.7.2.1.4 VISUALIZACIÓN DE LA FRECUENCIA CARDÍACA

La frecuencia cardíaca se podrá observar en un cuadro de texto constantemente, mientras se permanezca en el menú principal. En cuanto a la configuración, se la realizó en los timer 1 y 2; cuando se configura los timers, se habilita para que el pulso pueda ser observado en el “TextBox4_fc” o cuadro de texto de la frecuencia cardíaca.



FIGURA 78: Visualización de la Frecuencia Cardiac

Fuente: Edison Flores

3.7.2.1.5 INGRESO DEL PULSO NORMAL

Como se dijo en el capítulo II, el pulso normal no es nada más que cuando el individuo se encuentra en reposo, puede ser previo a la actividad física o lo más óptimo es ver este valor inmediatamente después de levantarnos.

Ahora para nuestro sistema vamos a observar el pulso que se tiene “antes” de realizar el ejercicio físico. El usuario visualizará el pulso que tiene en ese momento y lo ingresará en el cuadro de texto del pulso normal.

Para ingresar el pulso normal se configuró para que aparezca un teclado numérico al igual que para la edad. Adicionalmente, cuando se ingresa y se hace clic sobre el botón “OK”, este valida la información y es enviada a la placa Arduino para que procese la información. Se envía junto con el carácter “H”, este indica de manera específica a la placa electrónica que se hace referencia al pulso normal.

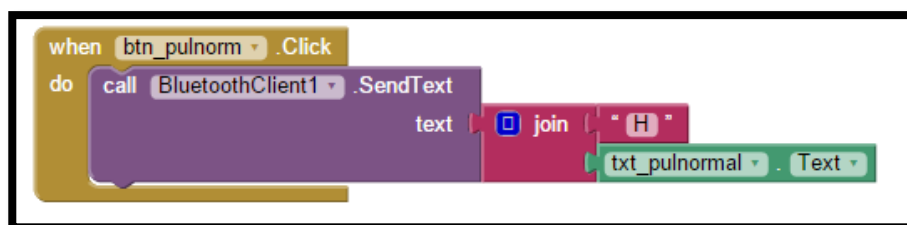


FIGURA 79: Configuración del pulso normal en la aplicación

Fuente: Edison Flores

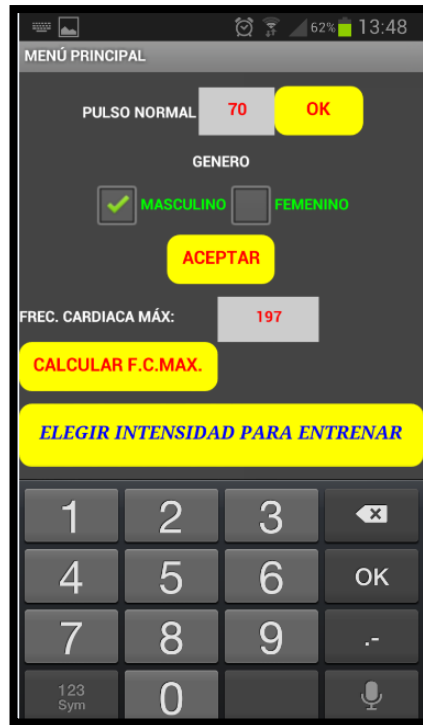


FIGURA 80: Visualización de ingreso del pulso normal

Fuente: Edison Flores

3.7.2.1.6 ELECCIÓN DEL GÉNERO

La elección del género, es otro ítem importante para la obtención de la fórmula de Karvonen. Se configuró para que el usuario puede seleccionar con solo hacer un contacto sobre la pantalla, en una casilla de verificación. También en la programación por bloques se configuró para que se envíe un carácter, dependiendo el género, si es masculino, envía la letra “M”, y si es femenino se envía la letra “L”. Estos caracteres se envían al caer clic sobre el botón aceptar.

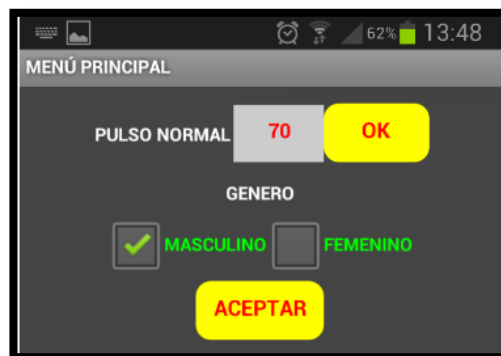


FIGURA 81: Visualización de elección del género en la aplicación

Fuente: Edison Flores

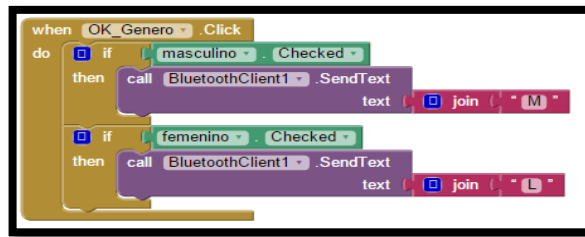


FIGURA 82: Configuración para elección del género

Fuente: Edison Flores

3.7.2.1.7 CÁLCULO DE LA F.C.M

Se asignará el cálculo de la F.C.M. a la variable `TextBox2_fcmax`, la cual se deriva de la resta de $220 - \text{edad}$ para género masculino y $226 - \text{edad}$ para género femenino. La edad fue previamente ingresada por el usuario en un cuadro de texto. La F.C.M. se observará cuando se haga clic en el botón "Calcular F.C.MAX".

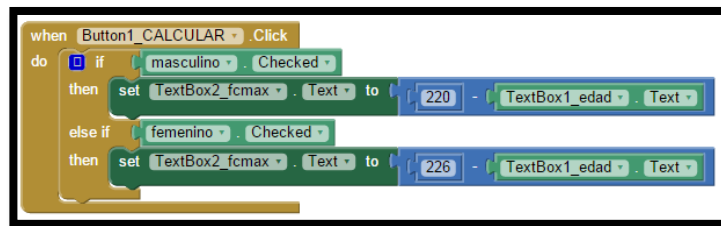


FIGURA 83: Configuración para cálculo de la F.C.M.

Fuente: Edison Flores

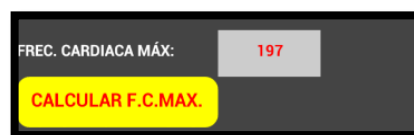


FIGURA 84: Visualización para la F.C.M.

Fuente: Edison Flores

3.7.2.1.8 ELECCIÓN DE UNA ZONA DE ENTRENAMIENTO

Después de haber ingresado los datos del deportista, se elegirá la zona de entrenamiento. Se diseñó un botón "Elegir intensidad para entrenar", al hacer clic sobre este, aparecerá una notificación que nos dirá que antes de realizar el ejercicio físico debemos realizar un calentamiento previo.

Primero debemos elegir el icono de notificación y posteriormente te configurarla en la programación por bloques.



FIGURA 85: Visualización de Llamada de notificación

Fuente: Edison Flores

Conjuntamente se configuró cuando haga clic en el botón antes mencionado, aparecerá la notificación y las opciones para elegir las zonas de entrenamiento. También se configuró para que se guarde en un cuadro de texto (oculto a la vista del usuario) el pulso de reserva (este es la diferencia entre la frecuencia cardíaca máxima y el pulso normal). Este pulso de reserva se utilizará en los screen posteriores para cálculos de los rangos.



FIGURA 86: Configuración de "Elegir Entrenamiento"

Fuente: Edison Flores

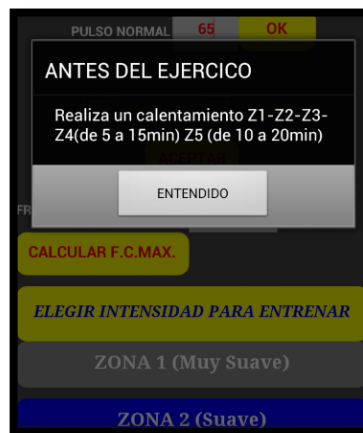


FIGURA 87: Visualización de la notificación

Fuente: Edison Flores

Una vez que se hace clic en el botón “ENTENDIDO”, aparecerá las zonas de entrenamiento para elegir al gusto del usuario.



FIGURA 88: Visualización de las zonas de entrenamiento

Fuente: Edison Flores

3.7.2.1.9 BOTONES ZONAS DE ENTRENAMIENTO

La persona que realice ejercicio físico podrá elegir una de las 5 zonas de entrenamiento, cada una con la característica principal de intensidad diferente. Cuando hagamos clic en una de las zonas de entrenamiento (botones) se dirigirá al Screen de la zona específica, llevando en una especie de vector, los valores del pulso normal y el pulso de reserva al Screen seleccionado. También se desconectará la conexión del Bluetooth.

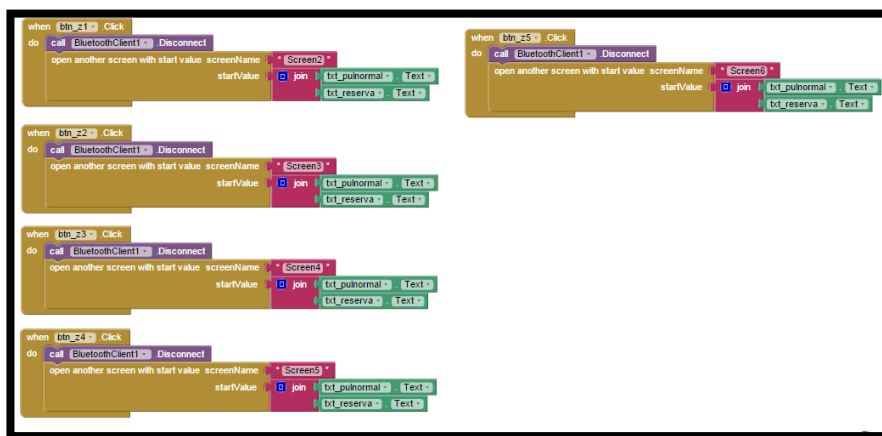


FIGURA 89: Configuración de ingreso a zonas de entrenamiento en la aplicación

Fuente: Edison Flores

3.7.2.1.10 Salir de la aplicación

Cuando el usuario quiera cerrar la aplicación hará clic en el botón “SALIR”. En la programación esto se lo hace con el mando “close application”.

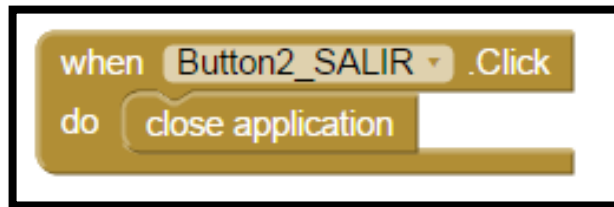


FIGURA 90: Configuración para salir de la aplicación

Fuente: Edison Flores

3.7.2.1.10 OPCIÓN RECONECTAR

Se configuró también una opción de reconectar, esta solo aparecerá cuando estemos dentro de cualquier zona y regresamos al menú principal. Lo que hace este botón es volver a conectar al módulo Bluetooth.

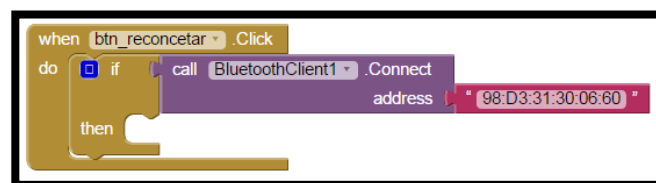


FIGURA 91: Configuración de reconectar

Fuente: Edison Flores

3.7.2.2 CONFIGURACIÓN DEL “SCREEN 2” AL “SCREEN 6”

Desde los Screen 2 al 6 se encuentran las zonas de entrenamiento desde las 1 hasta la 5 respectivamente. En estas pantallas se dará a conocer características fisiológicas del entrenamiento acorde a cada zona de entrenamiento. También se dará a conocer los rangos de pulsaciones al hacer clic en el botón de Iniciar entrenamiento.

Se mencionó anteriormente que las configuraciones son repetitivas para estos screen, lo único que cambia son los caracteres enviados y los porcentajes para las zonas de pulsaciones. Estos rangos y los caracteres se hacen referencia en el capítulo 2 y en la tabla 20 respectivamente.

3.7.2.2.1 BOTÓN CONTINUAR

Una vez que se ingresa al screen de cualquier zona aparecerá la pantalla inicial, en donde estas las características fisiológicas de cada zona de entrenamiento. Se configuró para que el usuario pueda primero leer las características primero, si es lo que desea entrenar tendrá que hacer clic en el botón “Continuar”, caso contrario podrá volver al menú principal.

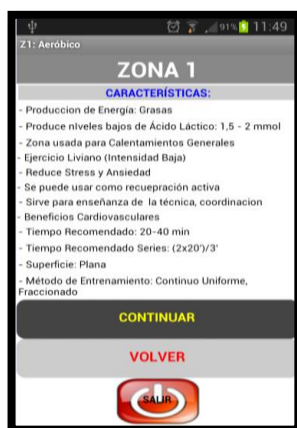


FIGURA 92: Visualización de la zona 1 en la aplicación

Fuente: Edison Flores

Si se hace clic en “Continuar”, se habilitarán otras opciones para iniciar el entrenamiento. Se tendrá que hacer clic en el botón “Iniciar entrenamiento” para que inicie y se visualice los rangos de pulsaciones. En esta parte de la configuración, también se hizo la extracción de los valores del pulso normal y pulsos de reserva que venía en un vector desde el screen de menú principal. Estos valores fueron asignados a variables globales independientes para que puedan ser utilizados en todo el screen (en cada screen de cada zona).

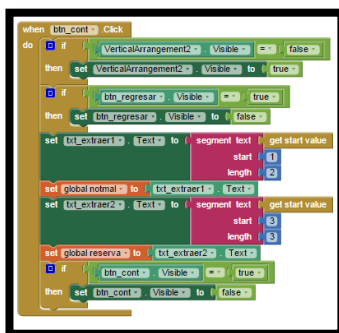


FIGURA 93: Configuración botón continuar en la aplicación

Fuente: Edison Flores

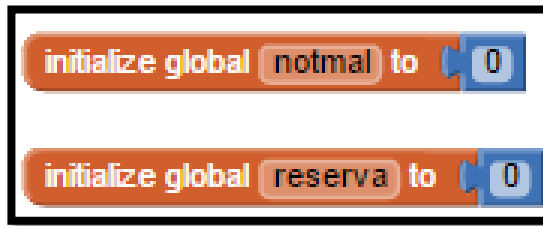


FIGURA 94: Configuración de variables globales (aplicación)

Fuente: Edison Flores

3.7.2.2.2 BOTÓN VOLVER

Este botón se configuró para que cierre el screen actual, y vuelva al menú principal.



FIGURA 95: Configuración de botón volver (aplicación)

Fuente: Edison Flores

3.7.2.2.3 BOTÓN INICIAR ENTRENAMIENTO

Una vez que se hace clic en continuar, se configuró para que este botón desaparezca y aparece la nueva opción de iniciar entrenamiento.

Dentro de la configuración de este botón también se realizó los cálculos para obtener los rangos de pulsaciones y se los asignó a un cuadro de texto. Estos valores deberán cambiar de acuerdo a zona de entrenamiento (ver figura 36)

Además, cuando se haga clic aquí, se habilitará un timer llamado "Clock1_dato", este timer también tiene su configuración específica la cual cumple con la función de conectarse al módulo Bluetooth HC-05 (fue desconectado en el screen de menú principal cuando se eligió la zona preferida) e interactuar con la programación de la palca Arduino para que se emitan las alertas de pulsos altos o bajos.

Dentro de la programación de bloques en este timer, está también la emisión de las alarmas, que son dos. Una de ellas, si el pulso está muy bajo del requerido se emite una alarma que será una vibración de 200 (ms); y la otra alarma si el pulso está por encima de los pulsos requeridos se emite una alarma sonora.

La función del timer también es enviar el carácter de la zona específica. Cada zona tiene un carácter diferente. (Ver tabla 20)

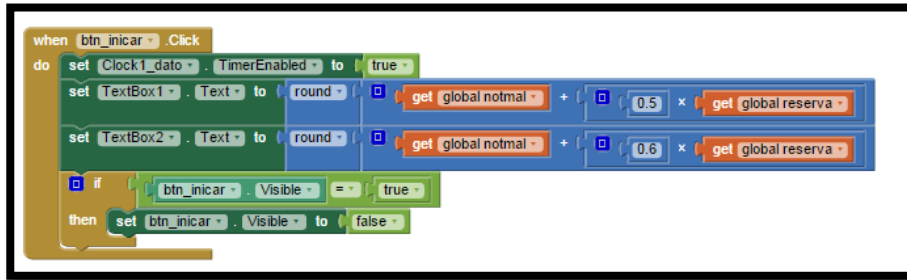


FIGURA 96: Configuración de botón iniciar entrenamiento en la aplicación
Fuente: Edison Flores

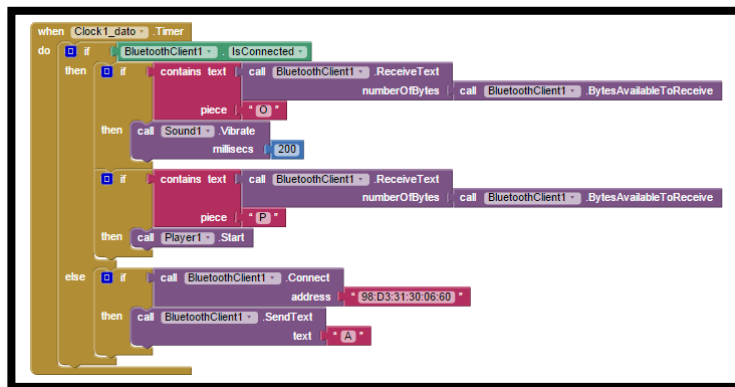


FIGURA 97: Configuración de timer en la zona de entrenamiento específica dentro de la aplicación
Fuente: Edison Flores

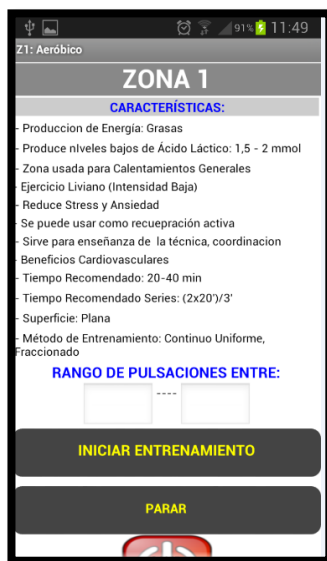


FIGURA 98: Visualización de iniciar entrenamiento en la aplicación
Fuente: Edison Flores

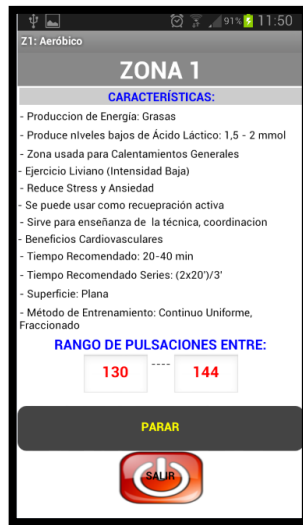


FIGURA 99: Visualización de los rangos de pulsaciones

Fuente: Edison Flores

3.7.2.2.4 OPCIÓN PARAR

En este botón se configuró para que se detenga el entrenamiento que se estaba realizando, es decir, si se estaba emitiendo las alertas que está por encima o por debajo de las pulsaciones establecidas, estas dejaran de funcionar, ya que se ha detenido el entrenamiento. Cuando esto suceda el usuario tendrá como única opción salir de la aplicación.

En la programación por bloques, se realizó el envío del carácter “k”, este carácter se envía a la placa Arduino para ordenará que detenga el entrenamiento. Se siguen emitiendo los pulsos pero sin alertas. Adicionalmente el Smartphone emitirá un mensaje de “Entrenamiento Detenido” para que el usuario lo sepa auditivamente.

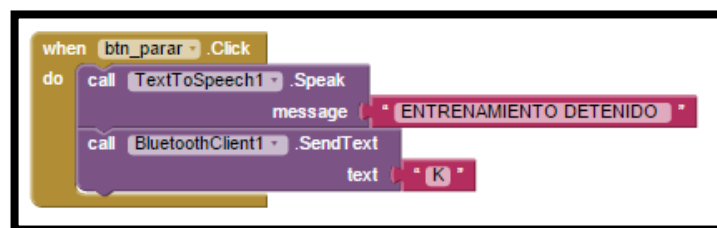


FIGURA 100: Configuración botón de parar (aplicación)

Fuente: Edison Flores

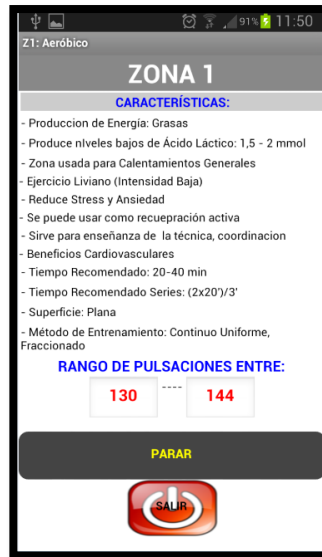


FIGURA 101: Visualización de botón parar en la aplicación

Fuente: Edison Flores

3.7.2.2.5 SALIR DE LA APLICACIÓN

Para salir de la aplicación, se configuró desde cada Screen un botón para cerrar la misma. Dentro de la configuración esto se hizo con un “close application”

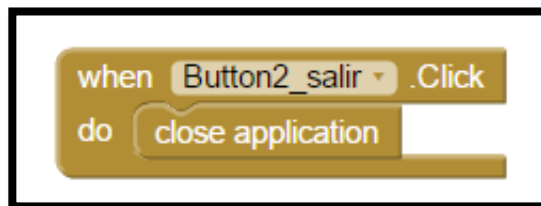


FIGURA 102: Configuración para cerrar screen de zona de entrenamiento (aplicación)

Fuente: Edison Flores

3.8 PROCESO DE IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO – PROTOTIPO #1

Una vez que se realizó el diagrama de bloques, los diagramas de flujo, y el firmware tanto en Arduino, como en Android; se procedió al primer prototipo para la implementación del sistema de ritmo cardíaco. Este fue solo a manera de prueba, para verificar la programación establecida previamente.

Se procedió a implementar un prototipo que me permita verificar la programación establecida, por lo tanto se realizó las pruebas en un “ARDUINO MEGA 2560”.

En este primer prototipo no se hizo el uso del hilo conductor, aquí solo se realizó las conexiones con los cables normales de conexión que se usa en un protoboard.

3.8.1 MATERIALES UTILIZADOS

Para este prototipo #1 se utilizó los siguientes materiales.

TABLA 25: Materiales utilizados en prototipo#1.

ELEMENTOS	CARACTERÍSTICA	UNIDADES
Módulo Arduino	Mega 2560	1
Sensor de Pulsos	3 - 5 V	1
Módulo Bluetooth HC-05	3,3 – 5V	1
Cables para protoboard	Tipo macho	-

Fuente: Edison Flores

3.8.2 FIRMWARE DEL PROTOTITPO

Para el Arduino Mega se utiliza la programación establecida anteriormente en la sección 3.7.1. (Firmware de Arduino Lilypad), solo se debe cambiar una línea de código, en los registros de la interrupción. La línea que se cambia y su justificación, fue explicada anteriormente (ver página 123)

3.8.2.1 DISEÑO DE LA APLICACIÓN EN APP INVENTOR

Para la aplicación de Android, de igual manera, se debe utilizar la misma programación establecida en la sección 3.7.2. Aunque para pruebas iniciales para el prototipo #1 se tenía estructurado partes básicas de la aplicación completa, como la edad, la F.C.M., el pulso cardíaco y las zonas de entrenamiento.

Con esta aplicación básica, derivada de la App completa se procedió a realizar las pruebas y tomas de pulso en las personas. Se menciona que esta aplicación en cuanto a estructura de programación era totalmente igual a la versión completa, es por ello que no se detalla los bloques de programación. A continuación se detalla el diseño de la interfaz gráfica de la aplicación, para que se observe las modificaciones que se realizaron antes de finalizar la App completa en el prototipo final.



FIGURA 103: Aplicación Demo para prototipo#1

Fuente: Edison Flores

En el “Screen 1” se tiene las funciones para:

- Ingreso de Edad
- Botón para el cálculo de la frecuencia cardíaca máxima
- Indicador de la F.C.M. de la persona.
- Visualización del número de pulsaciones de la persona
- Botones para la elección de la zona de entrenamiento
- Botón para Salir de la Aplicación

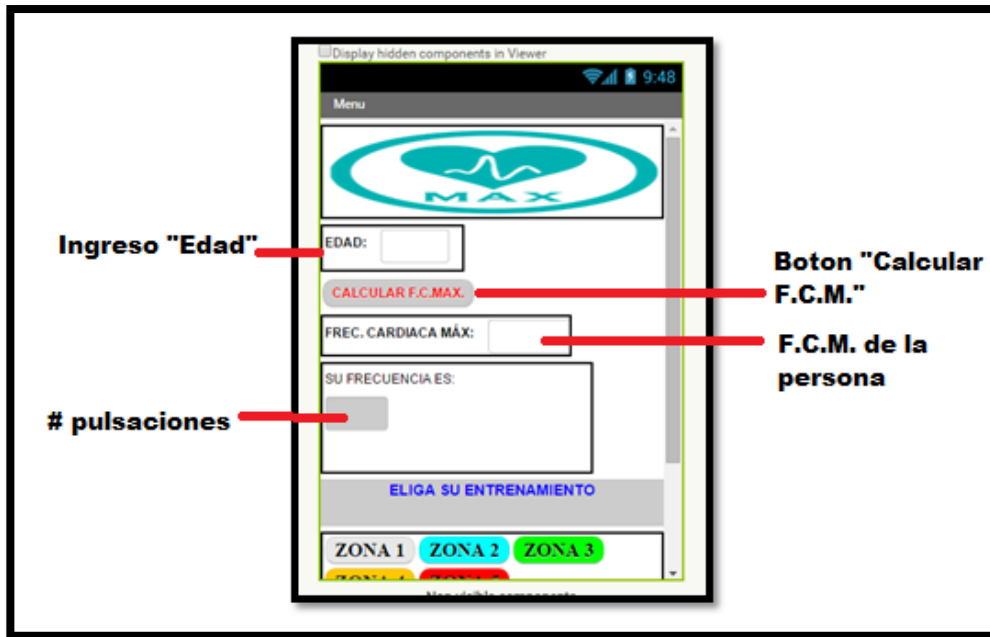


FIGURA 104: Menú principal de la aplicación (Versión Demo)

Fuente: Edison Flores

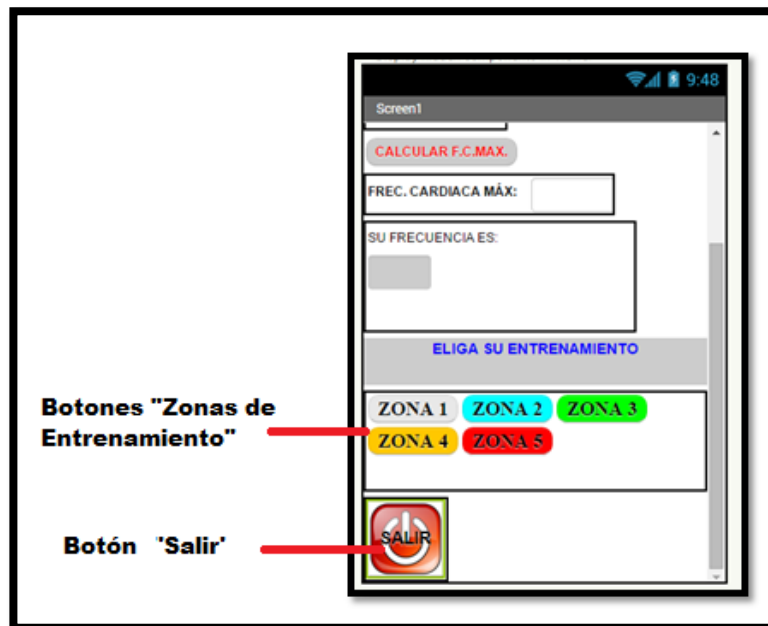


FIGURA 105: Elección de zonas de entrenamiento en la aplicación (versión Demo)

Fuente: Edison Flores

En los “Screen 2 al 6” que son donde se encuentran las 5 zonas de entrenamiento, tienen la misma estructura, pero cada una con adaptaciones específicas para cada zona en la que este entrenando el usuario. Las características que cuentan estas pantallas son:

- Nombre de la zona de entrenamiento.
- Características de la zona.
- Rango de pulsaciones. (Depende del entrenamiento elegido)
- Botón de “Inicio” de entrenamiento.
- Botón para “Parar” el entrenamiento.
- Botón para “Regresar” al Menú principal
- Botón para “Salir” de la Aplicación

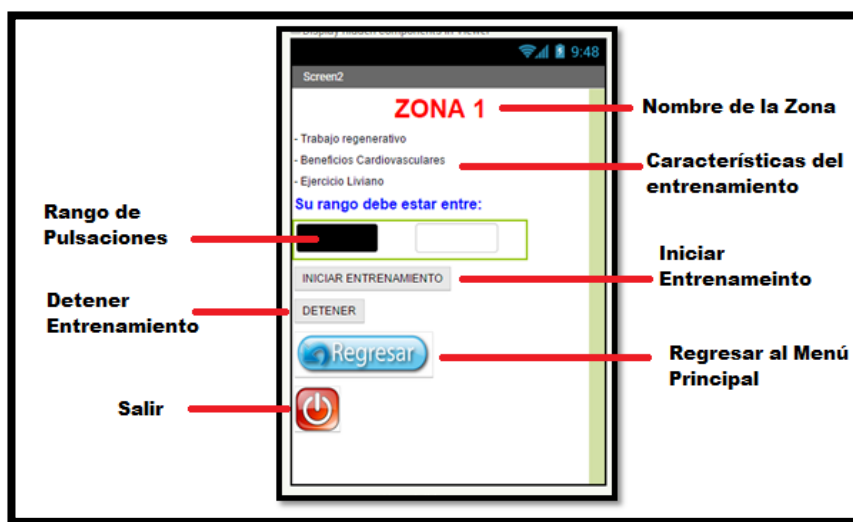


FIGURA 106: Visualización de Zona 1 en la aplicación (Versión Demo)

Fuente: Edison Flores

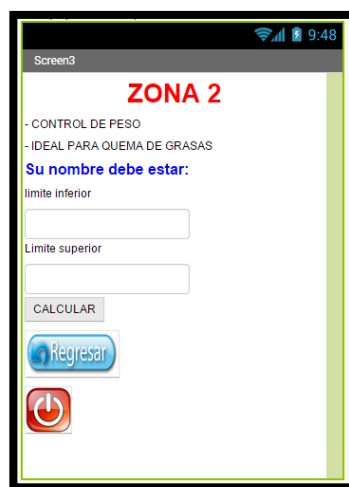


FIGURA 107: Visualización de Zona 2 en la aplicación (Versión Demo)

Fuente: Edison Flores

3.8.2.1.1 INSTALACIÓN DE LA APLICACIÓN

Para instalar la aplicación diseñada y configurada en el programa App Inventor debemos realizar lo siguiente:

- Una vez terminada la aplicación, tenemos que ir al Menú en la opción “Build”, aparecerán dos opciones, y elegiremos la opción “App (provide QR code for .apk)”. Esta opción sirve para que instalemos la aplicación, permanentemente a nuestro dispositivo móvil, mediante un código “QR²⁵”.

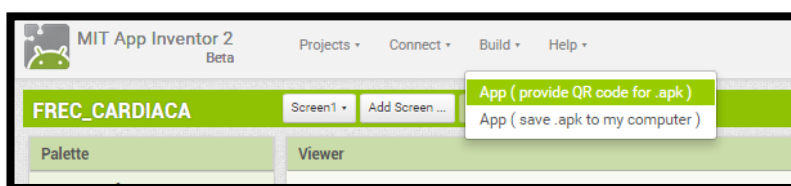


FIGURA 108: Instalación de la aplicación mediante código QR

Fuente: Edison Flores

- Después espero un momento hasta que se compile el programa y se genere el código QR.

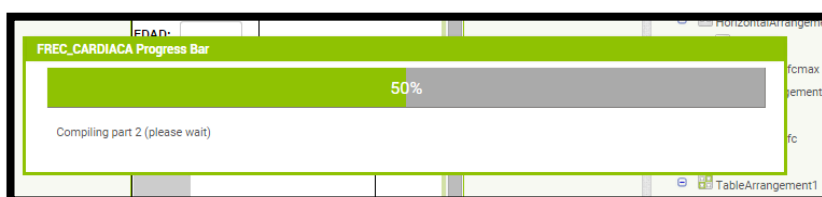


FIGURA 109: Compilación de la aplicación

Fuente: Edison Flores

- Aparecerá el código QR en la pantalla, estará listo para que la cámara del Smartphone lo codifique.

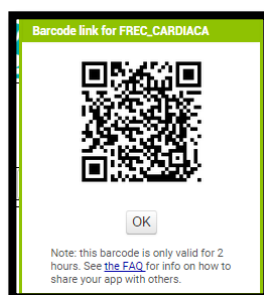


FIGURA 110: Generación de código QR para instalar la aplicación

Fuente: Edison Flores

²⁵ Quick Response.- Respuesta rápida

- Para que el Smartphone codifique el código que genera la aplicación de App inventor, se necesita de un programa que esté instalado en el dispositivo móvil inteligente, este se llama “MIT AI2 Companion”.

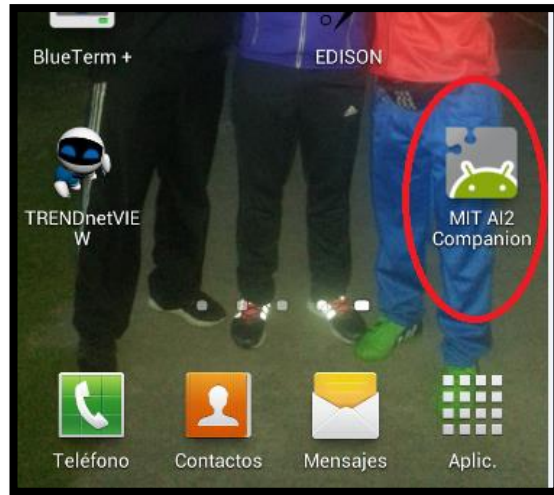


FIGURA 111: Icono de MIT AI2 Companion

Fuente: Edison Flores

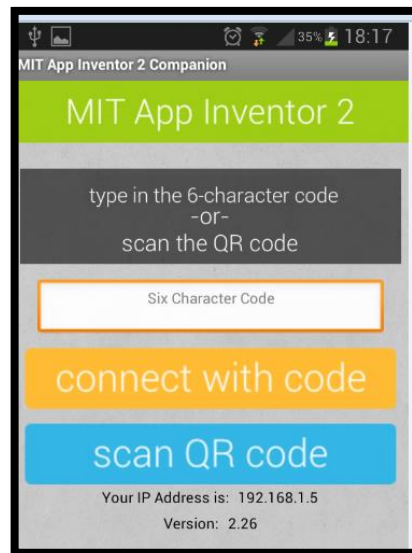


FIGURA 112: Aplicación MIT AI2 Companion

Fuente: Edison Flores

- Cuando haya capturado con la cámara del Smartphone la imagen del código QR, le aparecerá una enlace en el cuadro de texto blanco, a continuación se deberá hacer clic en conectar.

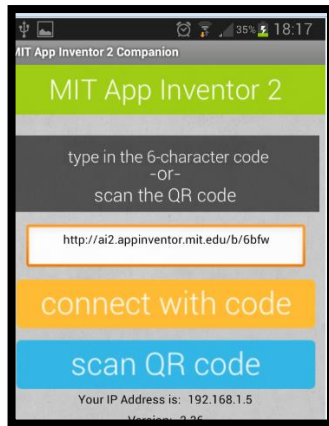


FIGURA 113: Enlace para descargar aplicación al Smartphone

Fuente: Edison Flores

- Aparecerá un instalador donde se deberá elegir “Instalador de paquete”.

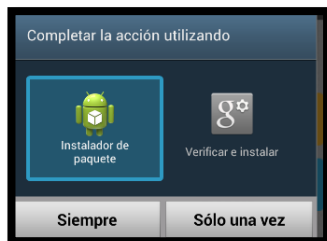


FIGURA 114: Instalador de paquetes de la aplicación

Fuente: Edison Flores

- Aparecerá una pantalla para leer las características de la aplicación y se dará clic en “Instalar”.

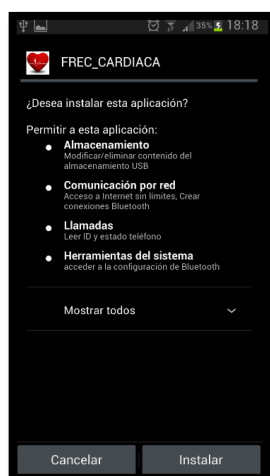


FIGURA 115: Instalación de la aplicación en el Smartphone

Fuente: Edison Flores

- Si no hubo problemas al instalar, aparecerá una pantalla con el mensaje “Aplicación instalada” y se podrá abrir directamente

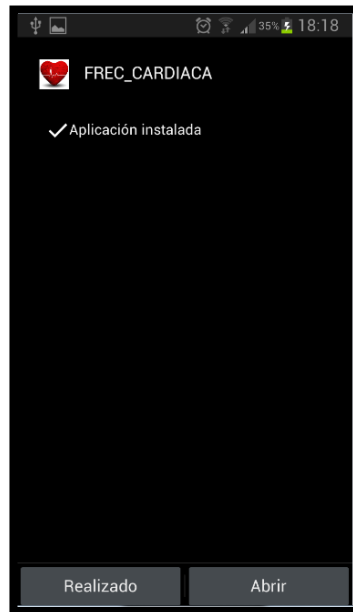


FIGURA 116: Pantalla de aplicación instalada en el Smartphone

Fuente: Edison Flores

3.8.2.1.2 APLICACIÓN EN EL SMARTPHONE

El icono de esta aplicación demo tiene como figura un corazón rojo con una señal cardiaca, el usuario podrá abrirla solo con tocar el icono con el dedo.

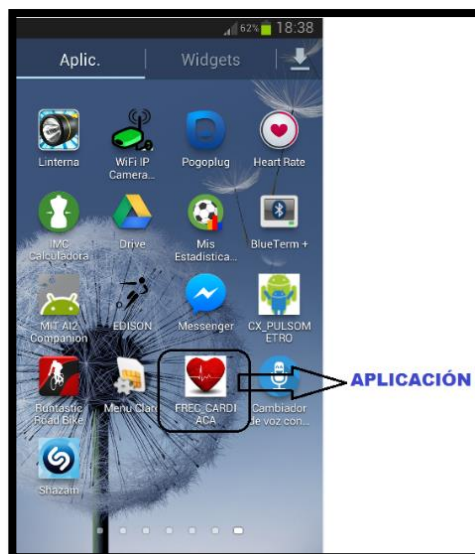


FIGURA 117: Icono de la aplicación Demo en el Smartphone

Fuente: Edison Flores

3.8.3 CONEXIONES DEL PROTOTIPO #1

Para las conexiones se procedió a conectar de la siguiente manera

- Conexión del pulse sensor al Arduino Mega 2560
- Conexión del módulo Bluetooth HC-05 al Arduino Mega 2560
- Conexión de alimentación eléctrica desde una laptop al módulo Arduino Mega 2560 mediante USB.

Se conectó directamente los pines del “Pulse sensor” al módulo Arduino. Como se ha mencionado anteriormente el sensor solo necesita de mínimo 3 voltios para su funcionamiento y de corriente 4 mA. En el módulo Arduino mega 2560, viene incorporado fuentes de conexión de voltaje para estos casos, de 3,3 voltios y de 5 voltios; es por eso que se conecta directamente sin ningún problema. También en cada pin entrega una corriente de 40 mA, que es más que suficiente para su uso.

El módulo Bluetooth se conectó un protoboard, para un manejo más fácil en sus conexiones hacia el Arduino Mega. De igual manera las conexiones eléctricas fueron tomadas de los pines de la placa Arduino.

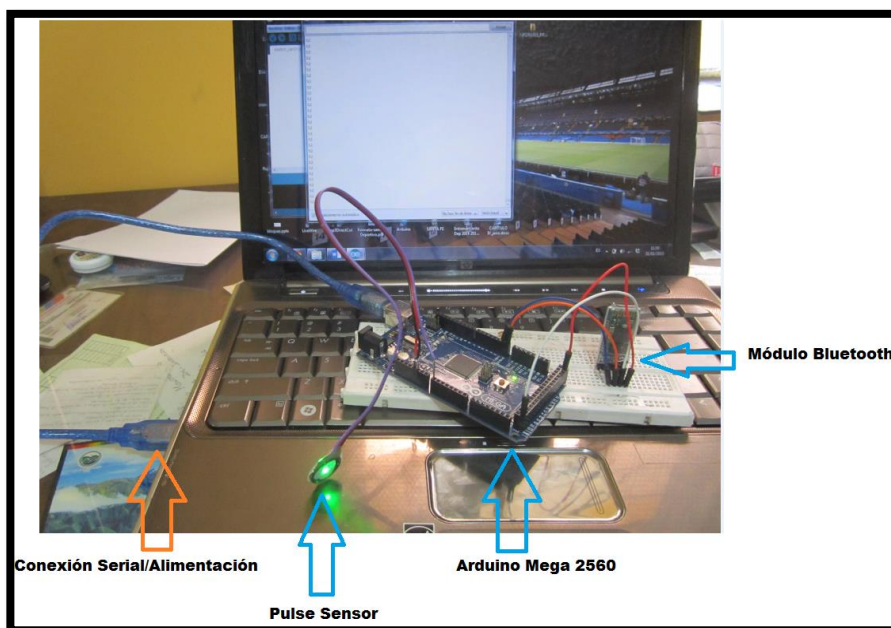


FIGURA 118: Conexiones del prototipo#1

Fuente: Edison Flores

3.8.4 PRUEBAS Y RESULTADOS DEL PROTOTIPO #1

En el prototipo #1 se realizaron pruebas para verificar que la programación de Arduino y Android se encuentre correctamente. Como muestra la figura 104, se procedió a verificar el correcto funcionamiento de toma de pulsos de sensor. Se menciona que se dio positivamente el reconocimiento de pulsos, se observaba rangos normales en estado de reposo, es decir entre los 60 y 75 ppm.

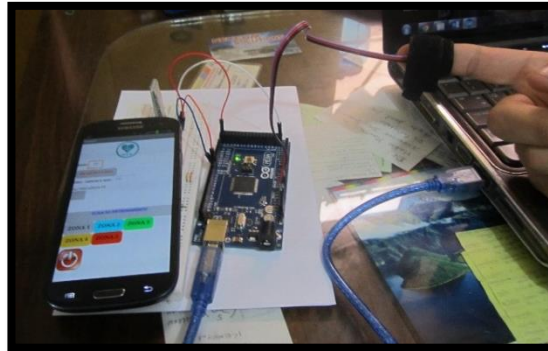


FIGURA 119: Verificación de funcionamiento de pulse sensor

Fuente: Edison Flores

Estos pulsos se los pudo verificar de tres formas. Una de ellas fue en el monitor serial de Arduino, es decir observamos los pulsos que se emiten, en la pantalla de la computadora

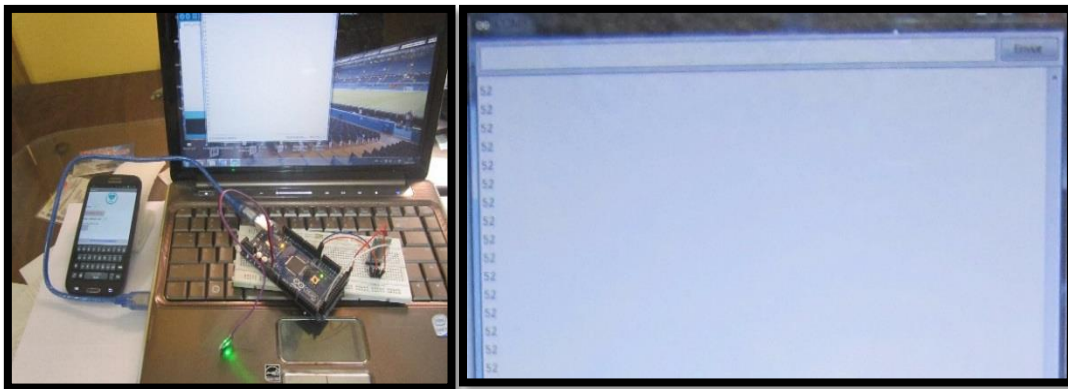


FIGURA 120: Verificación de pulsos en monitor serial de Arduino

Fuente: Edison Flores

La otra forma era observando a través de un programa instalado en el Smartphone, llamado "Blue Term+". Este programa mediante comunicación Bluetooth, se conectaba entre el módulo HC-05 y esta aplicación. De esta forma se podía observar los pulsos cardíacos que se estaban emitiendo.

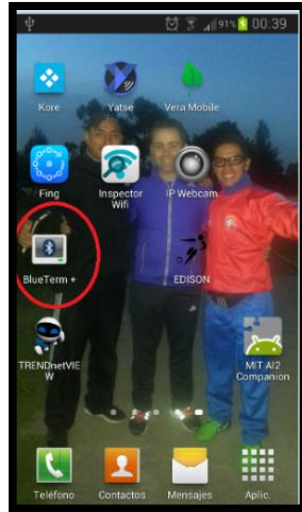


FIGURA 121: Icono de Blue Term+

Fuente: Edison Flores

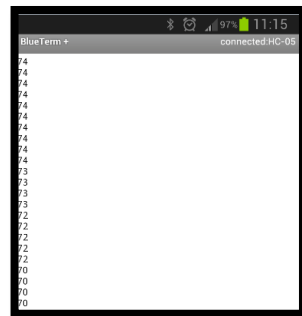


FIGURA 122: Visualización de pulsos a través de Blue Term+

Fuente: Edison Flores

La tercera forma de observar los pulsos cardíacos es en la aplicación instalada en el Smartphone. Aquí de igual manera se comunica mediante Bluetooth. Los pulsos se observa en un cuadro de texto de la aplicación.

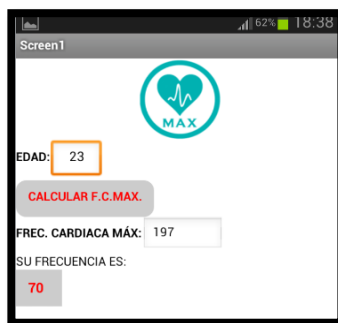


FIGURA 123: Visualización de pulsos a través de la aplicación

Fuente: Edison Flores

Este prototipo #1 se lo diseño específicamente solo para verificar la programación tanto de Android y Arduino, es decir para pruebas de la programación. En este diseño, el usuario no tenía flexibilidad para su uso, al estar conectado a la computadora, solo se podría realizar las pruebas en el mismo sitio de la Pc. Es así que el usuario era dependiente del lugar donde se encuentre el sistema de monitoreo de ritmo cardíaco.

3.9 PROCESO DE IMPLEMENTACIÓN DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO – PROTOTIPO #2

En el segundo prototipo, se propuso incorporar los elementos electrónicos a una muñequera, para que el uso sea más flexible para el usuario (movilidad de usos). Esta muñequera tiene que adaptarse a la movilidad de la persona que realiza ejercicio físico, así como también a los elementos electrónicos que irán dentro de ella. Se propuso convertir una muñequera en un textil inteligente.

En este prototipo se utilizará la placa Arduino LilyPad y se dejara de usar el módulo Arduino Mega 2560. El proceso de comunicación será totalmente inalámbrico, es decir a diferencia del Prototipo#1 que se podía ver los pulsos inalámbricamente y alámbricamente por puerto serial; este prototipo #2 para flexibilidad de uso del usuario, será ya de manera inalámbrica propiamente.

3.9.1 MATERIALES UTILIZADOS

Los materiales utilizados en el prototipo #2 fueron los siguientes:

TABLA 26: Materiales Utilizados

ELEMENTOS	CARACTERÍSTICA	UNIDADES
Arduino LilyPad	3,7 – 5 V	1
Batería Lipo	5V – 200mA	1
Sensor de Pulsos	3- 5 V	1
Módulo Bluetooth HC-05	3,3 – 5V	1
Hilo Conductor	28 Ohms	1
Estaño	--	1
Espadines Macho-Hembra	--	3
Muñequera	--	1
Ajuga	--	1

Fuente: Edison Flores

3.9.2 DISEÑO PARA LA IMPLEMENTACIÓN DEL PROTOTIPO #2

Para la implementación se realizó un diseño previo en un gráfico, esto para distribuir donde se iban a ubicar los elementos electrónicos, en la persona que realiza ejercicio físico.

En el diseño de implementación en el prototipo #2 se consideró lo siguiente:

- En el segmento electrónico, la Placa Arduino Lilypad, se buscó que sea cosida directamente a una muñequera (por dentro). El modulo Bluetooth no requiere ser cosido, pero necesitaba ir por dentro de la muñequera, al igual que la baterías lipo.
- Para el sensor cardíaco, se necesitaba que los pines de conexión estén por dentro de la muñequera, y la parte de censo de pulsos se encuentre en uno de los dedos del usuario.
- Para este sistema se utilizó una batería Lipo para todos los componentes electrónicos.
- La muñequera se ubicó entre el antebrazo y la mano, y el dispositivo Android, si ubicó en el brazo, mediante una banda de brazo para celular.

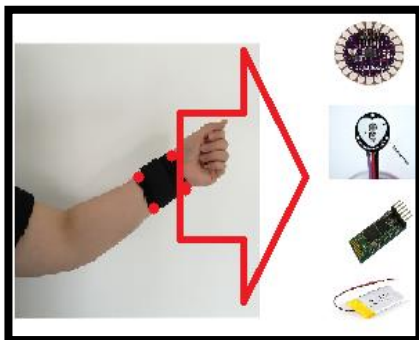


FIGURA 124: Diseño para implementación del prototipo #2

Fuente: Edison Flores



FIGURA 125: Banda de brazo para celular

Fuente: <http://img.clasf.co.ve/2015/01/07/Banda-De-Brazo-Porta-Celular-Jetion-20150107112923.jpg>

De esta forma, el usuario podrá sacarse y ponerse nuevamente la muñequera y así también su sistema de monitoreo cardiaco.

3.9.3 CARACTERÍSTICAS DE LA MUÑEQUERA

Para este prototipo #2 utilizó una muñequera "Nike Premier" de color negro con blanco, en ella se realizó las siguientes modificaciones.

- Se utilizó una muñequera fabricada "Nike", la cual se descosió y se la abrió, para poder ingresar y coser dentro de ella los elementos electrónicos.
- Como era un muñequera de fábrica, se realizó las modificaciones a medida que se iba realizando el proyecto
- Su talla era única, por lo que debimos adaptarnos a esas dimensiones, de largo 10 cm y de diámetro 5 cm.
- Esta muñequera tenía fibras elásticas, lo que hacía que se pueda estirar y adaptarse a las medidas del usuario.



FIGURA 126: Muñequera "Nike Premier"

Fuente: [http://ecx.images-
amazon.com/images/I/51AoqBAIRL._SL246_SX190_CR0,0,190,246_.jpg](http://ecx.images-
amazon.com/images/I/51AoqBAIRL._SL246_SX190_CR0,0,190,246_.jpg)

3.9.4 COSIDO DE ARDUINO LILYPAD A LA MUÑEQUERA

La placa Arduino Lilypad se cosió en la parte interior de la muñequera, para que se sujete, se utilizó los pines que no se estaba utilizando, es decir con hilo normal, se podía sujetar la placa. La placa se fijó aún más, al coser los pines utilizados para las conexiones de los elementos, ya que se tuvo que pasar el hilo conductor, por lo menos tres veces, por los pines (agujeros donde pasa el hilo). Esto aseguró que el hilo conductor logrará una correcta conexión eléctrica con los elementos.

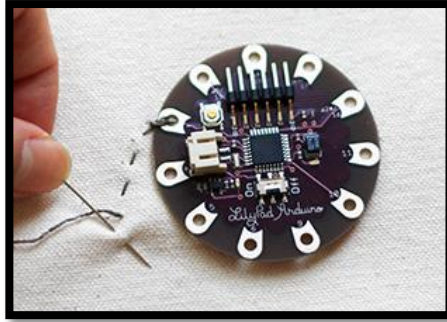


FIGURA 127: Primeros pasos cosido de Arduino Lilypad a la muñequera (Prototipo#2)

Fuente: Edison Flores

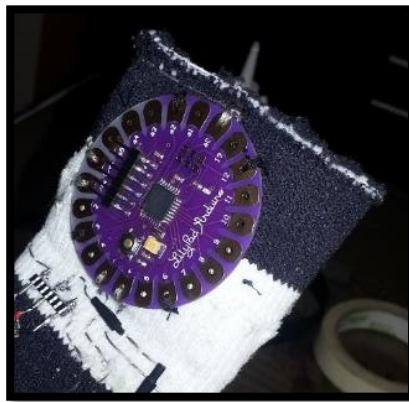


FIGURA 128: Arduino Lilypad cosida totalmente a la muñequera (Prototipo#2)

Fuente: Edison Flores

Las conexiones entre la placa Arduino Lilypad y los demás elementos están establecidas en la figura 129. Las conexiones debían respetar la distribución de los pines y la programación de los mismos, que se estableció previamente.

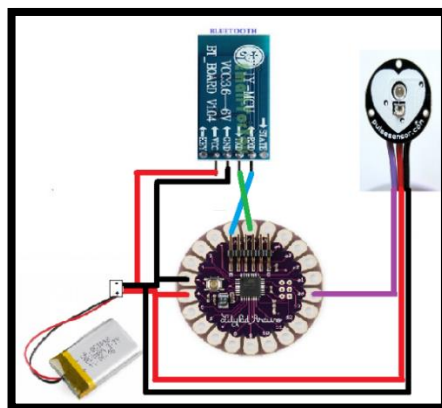


FIGURA 129: Diagrama de conexión (prototipo#2)

Fuente: Edison Flores

3.9.5 CONEXIÓN DEL MÓDULO BLUETOOTH

El módulo Bluetooth fue ubicado en dentro de la muñequera., fijado con hilo normal para que no se mueva. En este prototipo #2 se hizo la conexión entre la placa Arduino y el módulo mediante el hilo conductor, es decir desde tenían unión directa por medio del hilo conductor. Esto dificultó la conexión en los pines del módulo, ya que si existía un hilo chocando levemente con el otro, podía producir un cortocircuito. Por lo tanto, en el módulo Bluetooth, se aisló en cada pin para que no se unan con el hilo conductor. Esto se lo pudo hacer con pedazos de palillos, para separar los pines y el hilo.

Este elemento electrónico tenía que conectarse a dos elementos, uno era la placa Arduino y el otro era a la batería para su alimentación eléctrica.



FIGURA 130: Conexión del módulo Bluetooth en la muñequera (prototipo#2)

Fuente: Edison Flores

3.9.6 CONEXIÓN DEL SENSOR

El sensor cardíaco, tiene tres terminales, uno de ellos que es de color morado, es el que envía la señal para encontrar los pulsos cardíacos, este se cosió al "pin 0" del LilyPad y los otros dos, el de color negro (Ground) y de color rojo (Vcc) tendrán conexión a la batería Lipo. La parte específica del sensor tiene que estar por fuera de la muñequera, porque será para ubicarlo en el dedo del usuario en donde captará los pulsos cardíacos.

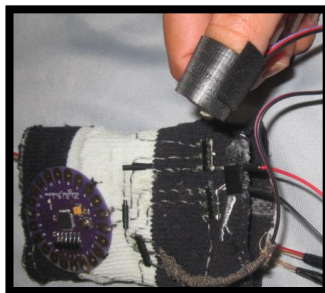


FIGURA 131: Conexión del sensor cardíaco (prototipo#2)

Fuente: Edison Flores

3.9.7 ALIMENTACIÓN ELÉCTRICA DEL SISTEMA

Para alimentar el sistema, se conectó los positivos y negativos de la placa Arduino Lilypad, del módulo Bluetooth y del sensor cardíaco a los terminales de positivo y negativo de la batería Lipo. La batería lipo que se utilizó era de 3,7 voltios a 200 mA.

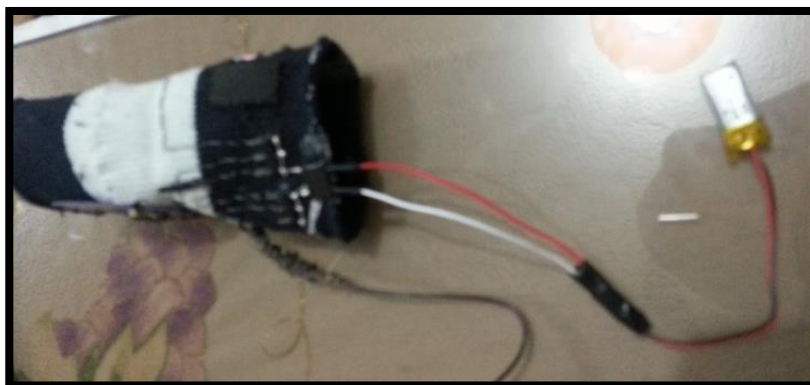


FIGURA 132: Conexión a batería lipo (prototipo#2)

Fuente: Edison Flores

3.9.8 PRUEBAS Y RESULTADOS DEL PROTOTIPO #2

A diferencia del prototipo #1, en este se realizó pruebas en varios aspectos, pruebas del sensor cardíaco, pruebas de envío de caracteres, pruebas de alimentación eléctrica y también pruebas de comodidad del uso de la muñequera.

Como resultado del prototipo #2 se obtuvo la siguiente muñequera, convertida en textil inteligente. En apariencia se nota una muñequera más el sensor cardíaco por fuera de ella.



FIGURA 133: Prototipo#2 (Muñequera)

Fuente: Edison Flores

Por otro lado el Smartphone podrá ubicarlo en una banda de brazo que es lo más óptimo.

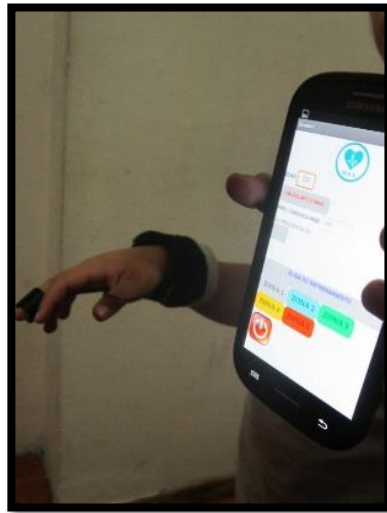


FIGURA 134: Smartphone con la aplicación (prototipo#2)

Fuente: Edison Flores

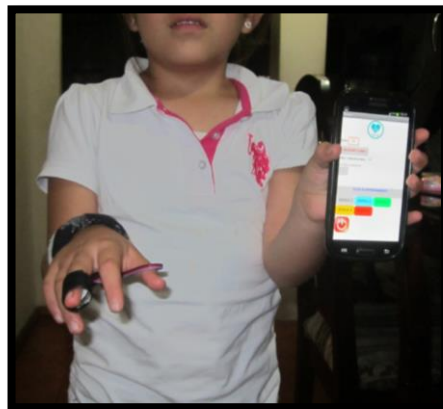


FIGURA 135: Vista frontal del sistema (prototipo#2).

Fuente: Edison Flores

3.9.8.1 PRUEBAS DE COMODIDAD DE LA MUÑEQUERA

En este prototipo el usuario se coloca la muñequera, ingresando la abertura de esta, por la en frente de los dedos de la mano, hasta terminar en la muñeca y el sensor se lo ubicará en cualquier dedo de la mano. Con el diseño de esta muñequera en el prototipo #2 se pudo observar y apreciar que, por lo pequeña que era, aunque muy elástica; fue muy incómodo al momento de ingresarla por la mano hasta llegar a ubicarla entre el

antebrazo y la mano. Los elementos electrónicos se podían palpar claramente su relieve haciendo molesto para el usuario que se la pone, y al no haber mucho espacio para las conexiones con el hilo conductor; en varios puntos que se unían o se topaba, se debió poner cinta aislante en ciertas secciones de hilo para no provocar cortocircuito. (Ver figura 135)



FIGURA 136: Ubicación de la muñequera y el sensor (prototipo#2)

Fuente: Edison Flores

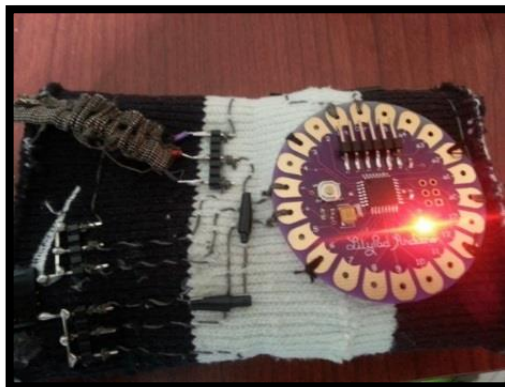


FIGURA 137: Puesta de cinta aislante en partes del hilo conductor

Fuente: Edison Flores

3.9.8.2 PRUEBAS DEL SENSOR CARDIACO

Para la verificación de que el sensor cardiaco y la programación no se vieran afectadas por las conexiones con el hilo conductor se procedió a evidenciar si los pulsos emitidos eran correctos, primero se comprobó dichos pulsos en la aplicación “BlueTerm+”.



FIGURA 138: Verificación de pulsos cardíacos en prototipo#2

Fuente: Edison Flores

Se pudo evidenciar que los pulsos cardíacos eran emitidos correctamente, es decir que por la implementación del hilo conductor no se vio afectada la programación, ni el funcionamiento de los elementos.

3.9.8.3 PRUEBA DE ENVÍO DE CARACTERES

Se realizó pruebas del funcionamiento para verificar las configuraciones de edad, zonas de entrenamiento, límites inferiores y superiores de las zonas. Esto se realizó para analizar la comunicación inalámbrica y el envío de caracteres entre la aplicación en Android y la placa Arduino Lilypad.

En la aplicación Android, el usuario ingresa con primer parámetro su edad. Esto es muy importante, porque con esta edad se hará el cálculo de la Frecuencia Cardíaca Máxima. Este dato se envía desde la Aplicación a la Placa Lilypad. Esta verificación se realizó utilizando el programa "BlueTerm+".

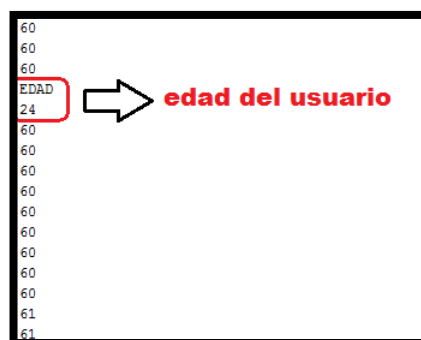


FIGURA 139: Verificación de ingreso de edad

Fuente: Edison Flores

Otro parámetro a verificar es cuando el usuario elija la zona de entrenamiento desde la Aplicación, la placa Lilypad reconocerá la zona, imprimiendo "Zona#"

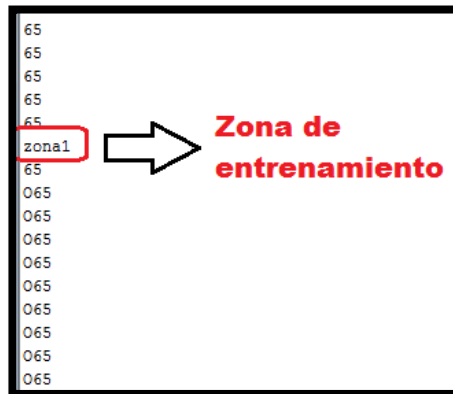


FIGURA 140: Verificación de elección de zona de entrenamiento 1

Fuente: Edison Flores

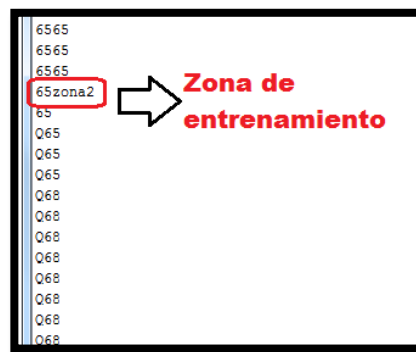


FIGURA 141: Verificación de elección de zona de entrenamiento 2

Fuente: Edison Flores

Dentro de cada zona de entrenamiento, existen límites inferiores y superiores de pulsaciones cardiacas.

Cada límite en cada zona, tiene un carácter asociado a ella. (Como se especificó anteriormente en las tablas 19 y 20). Este carácter emite la placa Arduino hacia la Aplicación, para que emita la alarma correspondiente.

```
097
097
097
097
097
097
097
097
097
097
097
097
097
097
097
```

FIGURA 142: Envío de carácter de límite inferior en la zona 1

Fuente: Edison Flores

```
O43
O43
O43
O43
O43
O43
O43
O43
O43
P160
P163
P168
P168
P173
P178
P184
```

FIGURA 143: Envío de carácter de límite superior en la zona 1

Fuente: Edison Flores

3.9.8.4 PRUEBAS DE ALIMENTACIÓN ELÉCTRICA

Para la pruebas de alimentación eléctrica en este prototipo se realizó de dos formas. Una de ellas era con una batería tipo Lipo de 3,7 voltios a 200 mA., y la otra forma fue mediante una fuente de 5 voltios a 1 Amperio (transformador de 110v a 5v).

Con el uso de la batería tipo Lipo, los elementos del sistema encendieron correctamente, el sensor cardíaco, la placa Arduino y el módulo Bluetooth. Se verificó la trasmisión de datos emitidos, pulsos, caracteres enviados entre placa y la aplicación; todo se presentó de manera correcta hasta los 30 segundos después del encendido.

Después de este tiempo, todos los elementos seguían encendidos sin ningún problema, pero el inconveniente con una sola batería y de corriente muy baja, fue que en el módulo Bluetooth, ya no se enviaban las tramas con los datos correspondientes. Se pudo

observar en la aplicación que después del tiempo mencionado, se dejaba de recibir los pulsos cardíacos y por ende las alarmas en cada zona, edad y demás datos. Esto afectaba a la funcionalidad del sistema, ya que no se podía establecer de manera definida la comunicación inalámbrica entre placa Arduino y Aplicación.

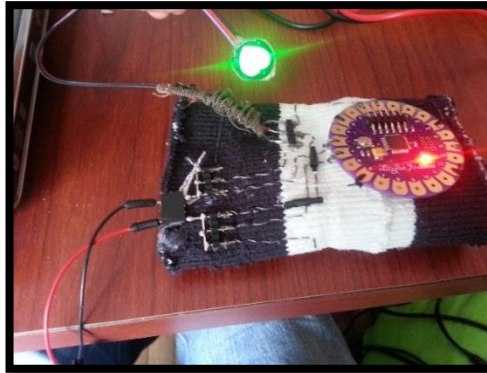


FIGURA 144: Pruebas de Alimentación eléctrica mediante batería lipo

Fuente: Edison Flores

Con el uso de un transformador de 110v a 5v y de 1 amperio como fuente de alimentación eléctrica para el sistema, el funcionamiento de este fue correcto todo el tiempo, es decir, se aumentó la corriente para el sistema de 200 mA. a 1A. y también el voltaje de 3,7v a 5v. Se observó en la aplicación que los datos eran enviados correctamente y constantemente todo el tiempo. Pero con esta fuente se le quitaba flexibilidad de uso al usuario, ya que el sistema se hacía dependiente del lugar donde se encuentre la fuente.

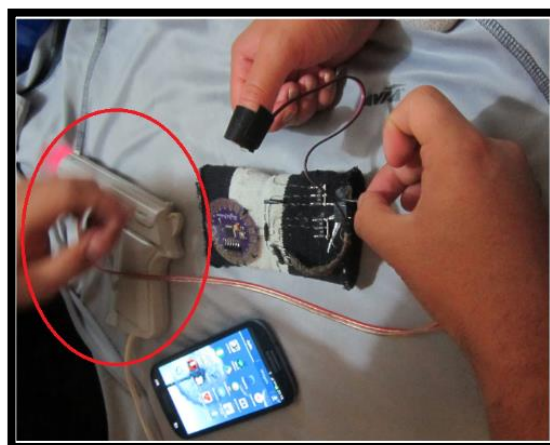


FIGURA 145: Fuente utilizada para alimentar eléctricamente al prototipo#2

Fuente: Edison Flores

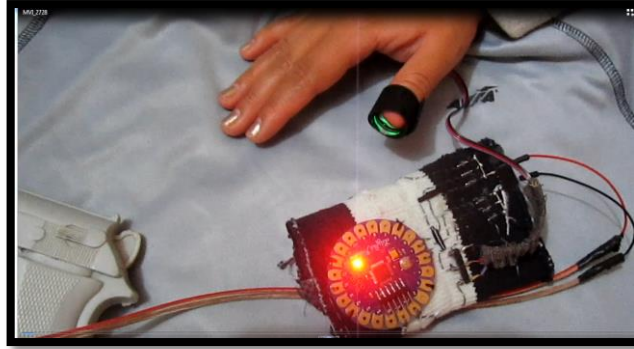


FIGURA 146: Pruebas de alimentación eléctrica mediante fuente

Fuente: Edison Flores

3.9.9 ANÁLISIS DE RESULTADOS DEL PROTOTIPO #2

Realizadas las diferentes pruebas en el prototipo#2, se pudo observar que el sistema de monitoreo cardíaco fue totalmente exitoso en cuanto a la programación realizada tanto en Arduino, como en Android.

En cuanto a la parte de alimentación eléctrica hubo inconvenientes con la batería tipo lipo de 3,7v a 200mA. Ya que duraba un tiempo muy corto de comunicación inalámbrica entre la placa Arduino y la aplicación. Se tendrá que tener en cuenta este parámetro para el siguiente diseño.

En lo referente al diseño de la muñequera se notó que hubo incomodidad al momento de ingresarla por la mano, ya que se palpaba los elementos y resulto muy pequeña, pese a su elasticidad.

Se observó que se pueden presentar problemas como cortocircuitos, si el hilo conductor se topa entre sí, y por el reducido tamaño de la muñequera la canalización del circuito está muy unida. Se puede tomar en cuenta para mejorar este inconveniente en un nuevo prototipo.

Se abre la brecha para poder realizar posibles cambios para mejorar el diseño de la muñequera y para la alimentación eléctrica en el sistema. Las pruebas fueron realizadas tratando de simular la realidad con condiciones cotidianas de uso, además de observar las necesidades prioritarias para orientar al buen uso del sistema.

3.10 PROCESO DE IMPLEMENTACIÓN DEL SISTEMA CARDIACO – PROTOTIPO #3

Partiendo de experiencias de diseño y configuraciones del prototipo #1 y #2 tanto en software y hardware, se procedió al diseño e implementación de un nuevo prototipo#3, el cual tiene como objetivo mejorar las características y dar una mejor flexibilidad y comodidad en cuanto a su uso.

Para este prototipo no se realizó ningún cambio en la programación en Arduino y Android. Se utilizarán los mismos elementos electrónicos del prototipo #2 y las mismas conexiones; solo se realizó un cambio de diseño en la muñequera, es decir, este prototipo será realizado para mejorar el diseño del textil inteligente.

3.10.1 APLICACIÓN EN EL SMARTPHONE

La aplicación que se utilizó para este prototipo fue la misma utilizada en el prototipo #1 y #2; esta sirvió para monitorear los pulsos y elegir el entrenamiento deseado. No se realizaron cambios ya que contenía los elementos necesarios para el funcionamiento correcto.

3.10.2 CONFIGURACIONES DE ARDUINO LILYPAD

El software desarrollado anteriormente, será utilizado de igual manera en este nuevo diseño. La programación será la misma para ejecutar el sistema de monitoreo cardiaco.

3.10.3 MATERIALES UTILIZADOS

Para este diseño, se empleó casi los mismos materiales que se utilizó previamente. Los materiales fueron los siguientes

TABLA 27: Materiales utilizados en el Prototipo#3

ELEMENTOS	CARACTERÍSTICA	UNIDADES
Arduino Lilypad	3,7 – 5 V	1
Batería Lipo	5V – 200mA	1
Sensor de Pulsos	3- 5 V	1
Módulo Bluetooth HC-05	3,3 – 5V	1
Hilo Conductor	28 Ohms	1
Espadines Macho-Hembra	--	3
Estaño	--	1
Muñequera	--	1
Ajuga	--	1

Fuente: Edison Flores

3.10.4 DISEÑO DE LA MUÑEQUERA EN EL PROTOTIPO #3

En este diseño se utilizó una nueva muñequera confeccionada de manera propia, el material que la recubre era de tela tipo toalla y por dentro se utilizó tela normal para que se puedan coser los elementos. Esta no era de ningún fabricante en especial y se la diseño con características para presentar mayor comodidad de uso para el usuario. Esta muñequera tiene un sector amplio donde se pudieron distribuir los elementos y realizar el cocido de los mismos sin ningún problema.



FIGURA 147: Diseño de muñequera para prototipo#3

Fuente: Edison Flores

En esta muñequera se ubicaron cintas de velcro para una mejor sujeción en el antebrazo del individuo. También ayudará para que los dispositivos electrónicos y el sensor cardíaco puedan estar fijos, sin mucho movimiento y contacto entre ellos. En las conexiones de los pines del sensor cardíaco, estarán mucho más fijos que el prototipo#2, esto ayudará a q no se produzcan valores alterados en los pulsos.

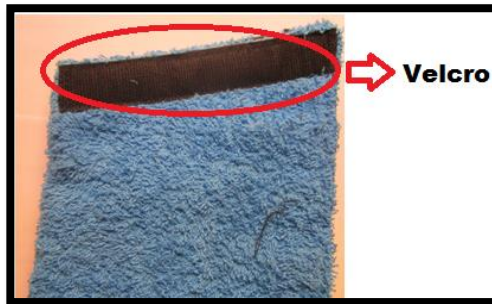


FIGURA 148: Cintas de velcro en la muñequera del prototipo #3

Fuente: Edison Flores

La muñequera tuvo dos doblajes, el primero servirá para cubrir los elementos electrónicos, proteger a los elementos de impurezas que puedan afectar su funcionamiento, del posible sudor producido en la zona, así como también tiene la ventaja de que cuando el usuario desee, puede revisar si los elementos están encendidos o no, y si existe algún problema visible en los elementos.



FIGURA 149: Primer doblaje de la muñequera en el prototipo #3

Fuente: Edison Flores

El segundo doblaje sirvió para darle precisión de ajuste a la muñequera en el antebrazo de cada persona. Ayudó a que la muñequera no se mueva y quede fija en el antebrazo de las personas, cuando esté realizando ejercicio físico. Presenta una cinta velcro para el ajuste.



FIGURA 150: Segundo Doblaje de la muñequera en el prototipo #3

Fuente: Edison Flores

Sus dimensiones eran las siguientes:



FIGURA 151: Dimensiones de la muñequera completa (prototipo#3)

Fuente: Edison Flores

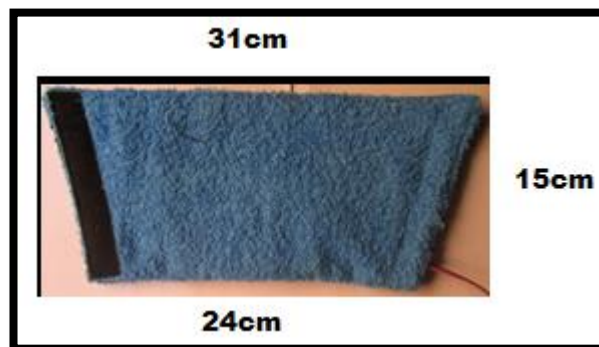


FIGURA 152: Dimensiones de la muñequera con 1 doblaje (prototipo#3)

Fuente: Edison Flores

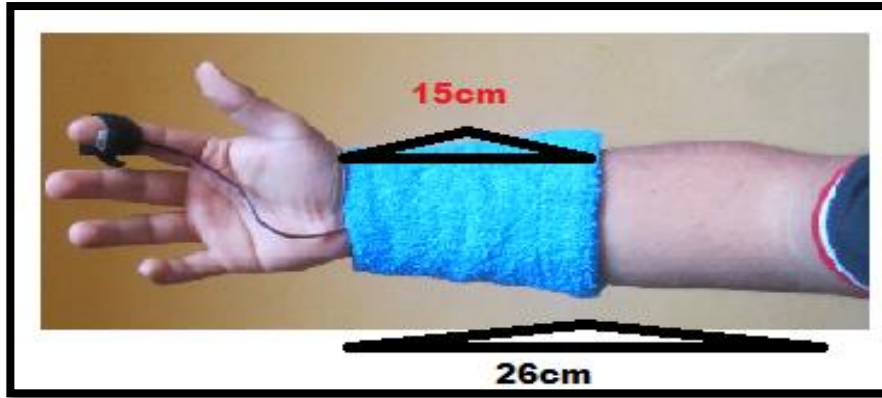


FIGURA 153: Dimensiones de la muñequera y el antebrazo (prototipo#3)

Fuente: Edison Flores

3.10.5 CONEXIONES DE LOS ELEMENTOS

Como se mencionó anteriormente, las conexiones en este prototipo serán iguales a las del prototipo#2, es decir se procederá a realizar de igual forma el cosido de la placa Arduino Lilypad, se realizará las conexiones del módulo Bluetooth, del sensor cardíaco y de la batería.

3.10.5.1 COSIDO Y CONEXIÓN DE LOS ELEMENTOS ELECTRÓNICOS

El cosido se lo realizo con el mismo hilo conductor, se empezó cosiendo la placa Arduino Lilypad, posteriormente el módulo Bluetooth y finalmente el sensor cardíaco y la batería. En esta muñequera se tuvo la facilidad de tener más espacio en donde se pueda organizar el recorrido del hilo conductor. Esto ayudó en gran parte a prevenir un posible cortocircuito.

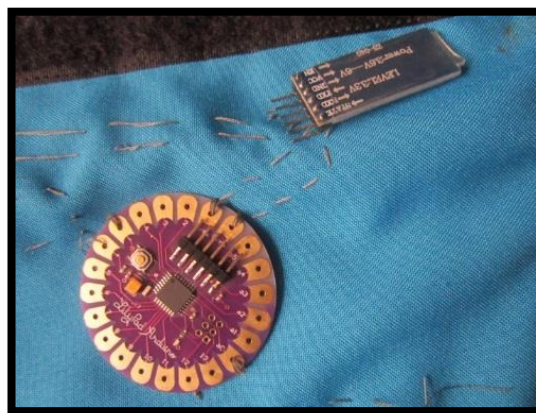


FIGURA 154: Cosido y conexión de elementos en la muñequera (prototipo#3)

Fuente: Edison Flores

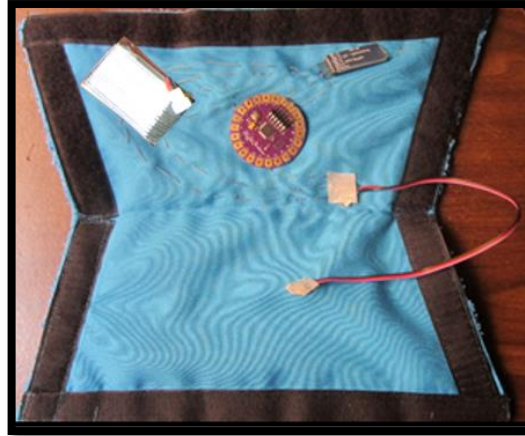


FIGURA 155: Conexión de los elementos (prototipo#3)

Fuente: Edison Flores

3.10.6 PRUEBAS Y RESULTADOS DEL PROTOTIPO #3

Las configuraciones en la placa Arduino y en la aplicación siguen siendo las mismas; es por esto que las pruebas del sensor cardíaco y del envío de caracteres produjeron los mismos resultados del prototipo #2. Estos resultados fueron correctos en dicho prototipo.

3.10.6.1 PRUEBAS DE COMODIDAD DE LA MUÑEQUERA

En este prototipo lo importante era realizar las pruebas de comodidad en el uso de la muñequera, aquí se podía observar y sentir claramente su flexibilidad y confort de uso, el usuario se sentirá más cómodo con este nuevo diseño de muñequera, será mucho más fácil ponerse, sacarse y utilizarla al momento de realizar ejercicio físico.

Se concluyó claramente que este prototipo fue mejor que el anterior, no en cuanto a sus configuraciones de software ya que no cambio, sino en cuanto a una mejor facilidad de uso, es lo que la hizo mucho más factible.



FIGURA 156: Pruebas de comodidad de muñequera (prototipo#3)

Fuente: Edison Flores



FIGURA 157: Vista lateral de muñequera (prototipo#3)

Fuente: Edison Flores

3.10.7 ANÁLISIS DE RESULTADOS DEL PROTOTIPO #3

Las pruebas realizadas en el prototipo#3, fueron realizadas de manera positiva, presentando éxito en cada una de ellas. En cuanto a las configuraciones no se obtuvo ningún problema, cabe mencionar que estas configuraciones son las mismas del prototipo #1 y #2 y anteriormente no hubo problemas con ello. Se obtuvieron los mismos resultados positivos.

En cuanto a la alimentación eléctrica se vino arrastrando el mismo inconveniente que el prototipo #2, como se utilizó la misma batería, ocurrió lo mismo, que dicho prototipo, la comunicación inalámbrica se establecía por poco tiempo. Pero si se cambiaba a la fuente con mayor voltaje y corriente, era un éxito el sistema, pero con la desventaja de la movilidad.

La muñequera al ser más grande, presentó ventajas a la hora de coser con el hilo conductor, ahora se tuvo más espacio para canalizar el hilo conductor, sin tener problemas de posibles cortocircuitos. Además, con respecto al prototipo#2, este nuevo, presentó mayor facilidad para ponerse y sacarse. En el prototipo #2 la muñequera era muy ajustada para cualquier usuario y los elementos electrónicos se podían palpar, y posiblemente pudo estorbar a algunos usuarios.

Ahora en el nuevo prototipo cuenta con cinta velcro que ayuda a la sujeción perfecta de la muñequera, así como también al tener dos doblajes, hace que sea más resistente, más robusta y esponjosa para no palpar los elementos electrónicos en la piel del usuario.

Como parámetro a mejorar en este diseño fue que los velcros utilizados eran muy gruesos y anchos y la tela que la recubría que era tipo toalla también, cuando se realizaba los doblajes estos ocupaban mucho espacio y se hacía muy gruesa la muñequera.

3.11 PROCESO DE IMPLEMENTACIÓN DEL SISTEMA CARDIACO – DISEÑO FINAL

Para este diseño final, se partió de configuraciones y diseños realizados en prototipos previos. En este diseño se tomaron las siguientes consideraciones:

- El uso de un switch para el encendido y apagado del sistema.
- Se utilizó dos baterías para alimentar eléctricamente al sistema.
- Se cargó en el Smartphone la aplicación completa con todos los parámetros para que el entrenamiento del usuario sea personalizado.
- Se mejoró el diseño de la muñequera.

3.11.1 MATERIALES UTILIZADOS

Los materiales que se utilizaron en este último diseño fueron:

TABLA 28: Materiales utilizados en el diseño final del sistema

ELEMENTOS	CARACTERÍSTICA	UNIDADES
Arduino Lilypad	3,7 – 5 V	1
Batería Lipo	5V – 200mA	2
Sensor de Pulsos	3 - 5 V	1
Módulo Bluetooth HC-05	3,3 – 5V	1
Hilo Conductor	28 Ohms	1
Cables para protoboard	Hembra-macho	12
Switch pequeño		2
Estaño	-	1
Muñequera	-	1
Banda de Brazo para celular	-	1
Ajuga	-	1

Fuente: Edison Flores

3.11.2 DISEÑO PARA LA IMPLEMENTACIÓN

En este diseño final se añadió más elementos, como es el uso de dos baterías y los switch, el objetivo fue dar un encendido y apagado al sistema. El diseño fue el siguiente:

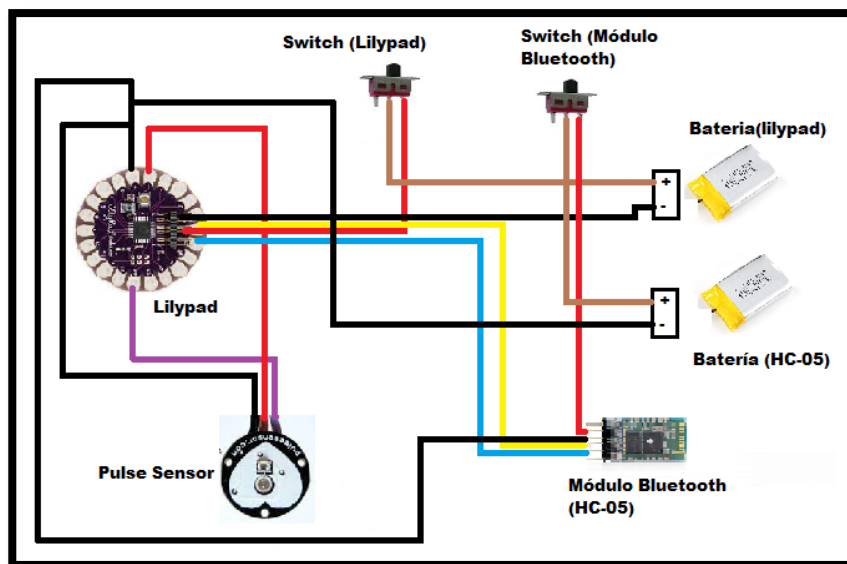


FIGURA 158: Esquema previo a la implementación del sistema

Fuente: Edison Flores

En la siguiente tabla, se explica la representación de conexiones de la figura 158.

TABLA 29: Funciones de cableado del diseño final

Elemento	Color Cable	Representación
Arduino Lilypad	Rojo	Positivo (Vcc)
Arduino Lilypad	Negro	Negativo (Gnd)
Pulse Sensor	Violeta	Señal de pulso
Pulse Sensor	Rojo	Positivo (Vcc)
Pulse Sensor	Negro	Negativo (Gnd)
Módulo Bluetooth HC-05	Amarillo	Transmisión (Tx)
Módulo Bluetooth HC-05	Azul	Recepción (Rx)
Módulo Bluetooth HC-05	Rojo	Positivo (Vcc)
Módulo Bluetooth HC-05	Negro	Negativo (Gnd)
Switch para Arduino Lilypad	Rojo	Positivo (Vcc)

Switch para Arduino Lilypad	Café	Retorno
Switch para Módulo Bluetooth	Rojo	Positivo (Vcc)
Switch para Módulo Bluetooth	Café	Retorno

Fuente: Edison Flores

3.11.3 FIRMWARE PARA ARDUINO LILYPAD

Para este diseño se utilizó el firmware establecido en la sección 3.7.1. No se realizó ningún cambio adicional.

3.11.4 APLICACIÓN

Para este diseño final, se utilizó la misma programación en bloques y el mismo diseño que se encuentra en la sección 3.7.2. No se realizó ningún cambio adicional

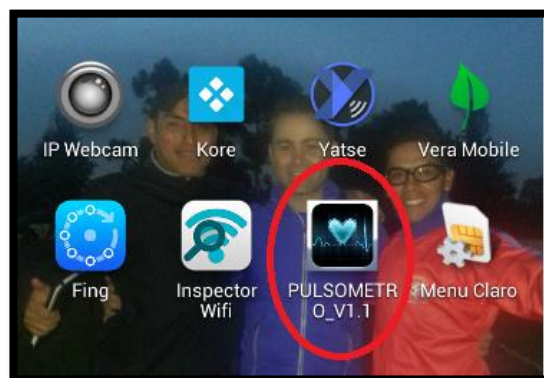


FIGURA 159: Icono de la aplicación (diseño final)

Fuente: Edison Flores

3.11.5 DISEÑO DE LA MUÑEQUERA

Para el diseño de la muñequera se cumplió con los siguientes requerimientos:

- El color utilizado fue negro.
- Se confeccionó con tela tipo licra.
- Se utilizó velcros más delgados (0,5cm).
- Se utilizó almohadillas delgadas para hacerla más robusta y cómoda.

- Se añadió pequeñas tiras dentro de la muñequera para asegurar los elementos.
- Se utilizó tiras de telas sobre el cosido del hilo conductor, para prevenir cortocircuitos
- Se realizó una pequeña abertura en la muñequera para que pasen los cables del sensor cardíaco.
- Tiene dos doblajes, como la del prototipo #3.

Se mejoró el diseño teniendo en cuenta, los resultados presentados en diseños anteriores, con esta muñequera, el usuario pudo obtener más comodidad y flexibilidad de uso.



FIGURA 160: Muñequera (Diseño final)

Fuente: Edison Flores



FIGURA 161: Zonas de velcro en la muñequera (Diseño final)

Fuente: Edison Flores



FIGURA 162: Ubicación de la muñequera (diseño final)

Fuente: Edison Flores



FIGURA 163: Muñequera junto con sensor cardíaco (Diseño final)

Fuente: Edison Flores

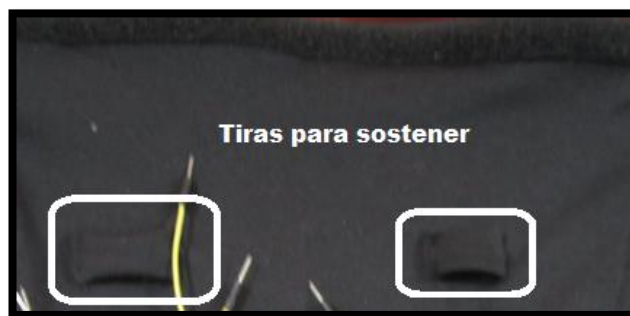


FIGURA 164: Tiras para sostener elementos en la muñequera

Fuente: Edison Flores

3.11.5.1 DIMENSIONES DE LA MUÑEQUERA

Las dimensiones fueron casi similares a las del prototipo #3, en este último diseño se puede apreciar las medidas en la figura 164.

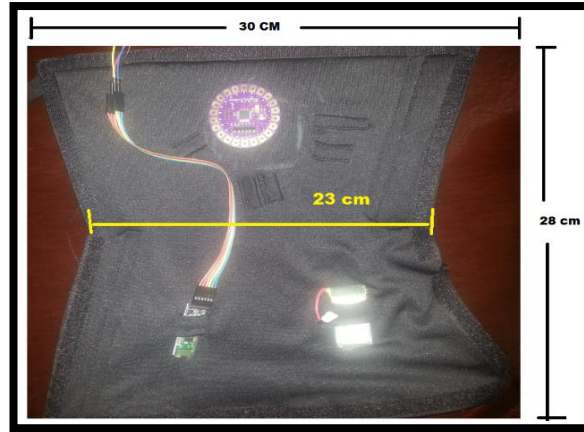


FIGURA 165: Dimensiones de la muñequera (diseño final)

Fuente: Edison Flores

3.11.6 COCIDO DE ARDUINO LILYPAD A LA MUÑEQUERA

Se fijó la placa Arduino lilypad a la tela tipo licra, esta tela era muy delgada, por lo que no hubo inconvenientes al momento de cocer con el hilo conductor. Se fue cociendo cada pin, para cada elemento.

En los orificios de la placa electrónica es decir los pines, se sugiere que a la hora de coser, el hilo pase tres veces por el orificio, esto es para que exista continuidad de voltaje y corriente. Una vez realizado este paso, se lleve el hilo hasta un punto específico para unir a los cables flexibles de protoboard.

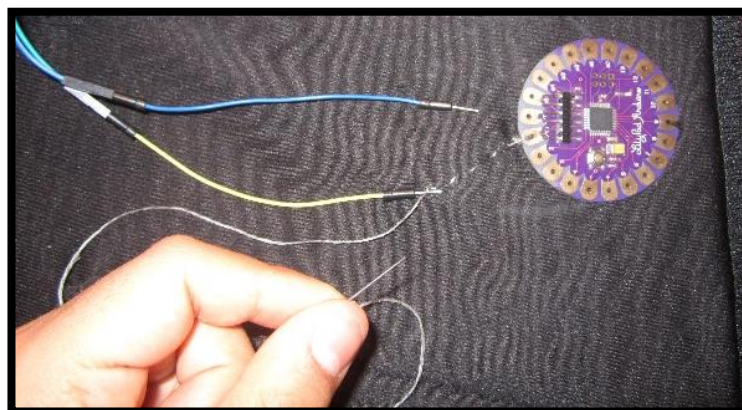


FIGURA 166: Cosido de la placa Arduino Lilypad (diseño final)

Fuente: Edison Flores

Todos estos cables que se unen con el hilo conductor son tipo macho

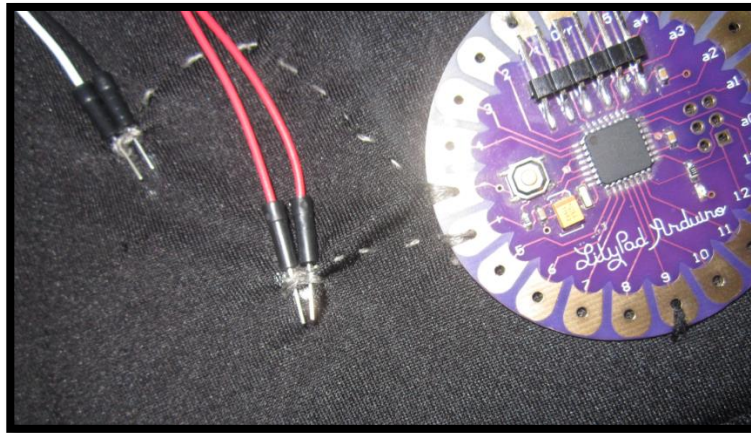


FIGURA 167: Intersección de cables con hilo conductor (diseño final)

Fuente: Edison Flores

Se cosen los pines generando caminos ordenadamente, es primordial que ningún hilo se tope entre sí, ya que podría causar un cortocircuito. Adicionalmente se colocó cinta adhesiva sobre los cables flexibles para que no se muevan cuando se estaba realizando el cocido.

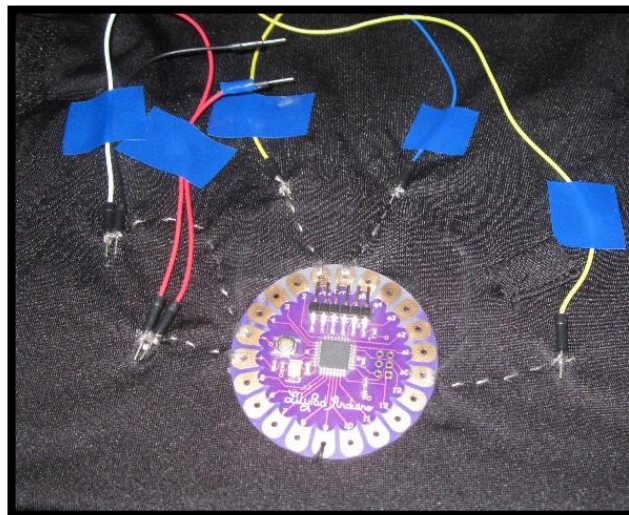


FIGURA 168: Caminos de cosido del hilo conductor

Fuente: Edison Flores

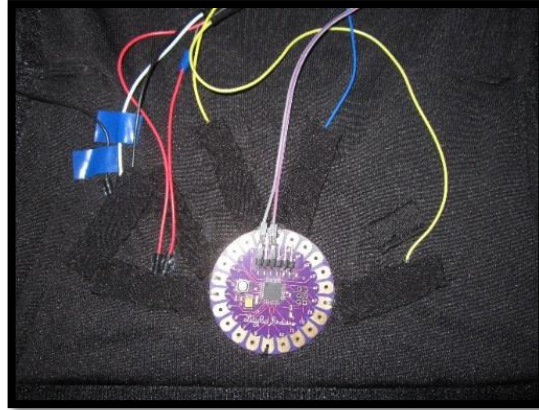


FIGURA 169: Protección al hilo conductor

Fuente: Edison Flores

3.11.7 CONEXIÓN DEL MÓDULO BLUETOOTH

Para las conexiones del módulo Bluetooth, se realizó lo siguiente

- Para los pines “Tx” y “Rx” del módulo que son de tipo macho se conectó cable tipo hembra- hembra. Esto se realizó para sacar una extensión de conexión de estos pines para que se luego se conecte al hilo conductor.
- Para los pines “Vcc” y “Gnd” se conectó directamente a una batería lipo, este no necesito de hilo conductor, solo se utilizó el cable flexible de protoboard.



FIGURA 170: Conexión del Módulo Bluetooth (Diseño Final)

Fuente: Edison Flores

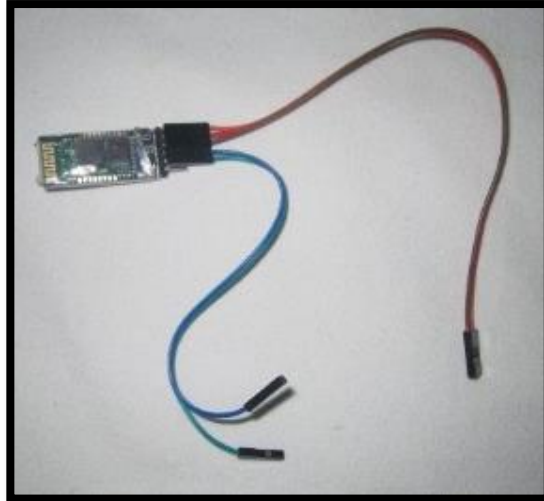


FIGURA 171: Modulo Bluetooth más extensiones de cable para conexión

Fuente: Edison Flores

A continuación se muestra un diagrama de conexión entre el módulo Bluetooth Hc05 y la placa Arduino, para los pines de transmisión y recepción de datos.

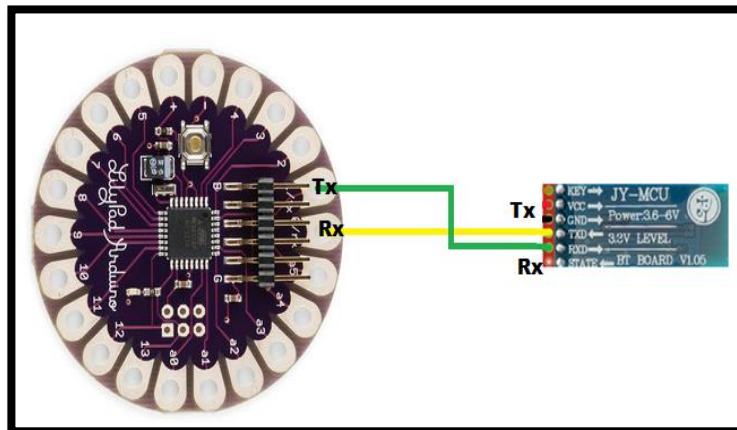


FIGURA 172: Conexión de pines Tx y Rx del módulo Bluetooth

Fuente: Edison Flores

Como se observa en la figura 169, los pines de “Tx” (transmisión) y “Rx” (recepción) del módulo Bluetooth se conectan directamente a los pines de “Tx” y “Rx” de la placa Arduino, pero de manera invertida, es decir, “Tx” del módulo Bluetooth a “Rx” de la placa Arduino y “Rx” del módulo Bluetooth a “Tx” de la placa Arduino.

3.11.7.1 CONFIGURACIÓN DEL MÓDULO BLUETOOTH

El módulo viene con configuraciones por defecto, pero se configuró el módulo para ayudar al usuario, al momento de conectarse por primera vez. Se configuró el nombre del Módulo, y se verificó que la velocidad de transmisión sea la correcta (9600 baudios).

Para la configuración del módulo Bluetooth, se necesitó de un elemento electrónico, como es el USB-TTL, con la finalidad de poder entrar en su interfaz de configuración de comandos de control o también llamados comandos AT (atención).



FIGURA 173: Conexión del USB-TTL al módulo Bluetooth

Fuente: Edison Flores

La distribución de conexión de pines tanto del módulo Bluetooth HC-05 como del cable adaptador USB-TTL es la siguiente:

TABLA 30: Distribución de pines de Modulo Bluetooth HC-05 y TTL

MODULO BLUETOOTH HC-05	CABLE ADAPTADOR USB-TTL
Pin ENABLE	Pin 3.3 Voltios
Pin VCC	Pin 5.0 Voltios
Pin GND	Pin GND
Pin TXD	Pin RXD
Pin RXD	Pin TXD

Fuente: Edison Flores

Una vez realizada la conexión respectiva, se procede a reconocer el puerto de comunicación por medio del administrado de dispositivos donde se obtuvo que el puerto COM5 es el seleccionado para la configuración del módulo Bluetooth HC-05.

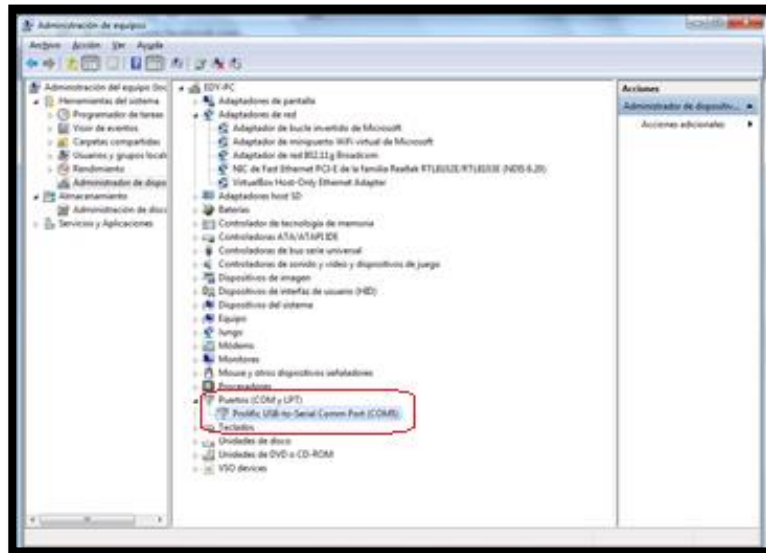


FIGURA 174: Puerto de conexión para el módulo Bluetooth

Fuente: Edison Flores

Una vez obtenido el puerto de comunicación, se procede a establecer una conexión mediante el programa ejecutable en el sistema operativo Windows llamado HyperTerminal, en donde se tiene la descripción de la conexión. Se coloca el nombre de la conexión, para nuestro caso será "Bluetooth Pulsómetro".

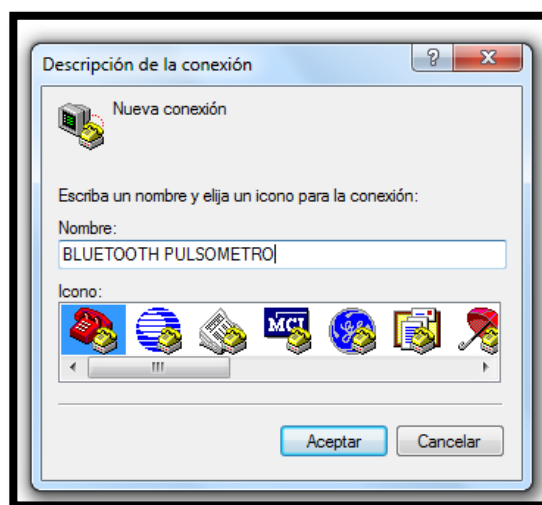


FIGURA 175: Descripción de la conexión del módulo Bluetooth

Fuente: Edison Flores

A continuación se elige el puerto de comunicación, en nuestro caso era el COM5.

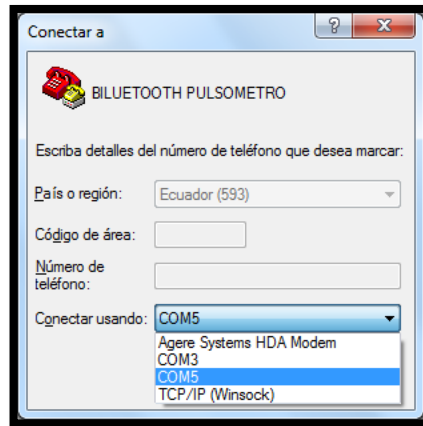


FIGURA 176: Elección del puerto de comunicación para configurar el módulo Bluetooth

Fuente: Edison Flores

Una vez elegido el puerto COM5, configuramos las propiedades de comunicación donde, los módulos Bluetooth HC-05 vienen configurados por defecto a una velocidad de conexión física de 38400 bps, 8 bits de datos, ninguna paridad, con 1 bit de parada y ningún control de flujo.

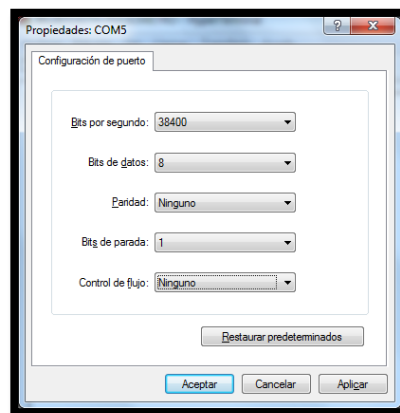


FIGURA 177: Propiedades de comunicación del módulo Bluetooth

Fuente: Edison Flores

Una vez definido el puerto y las propiedades de comunicación, procedemos a elegir en la ventana de comunicaciones la opción “Archivo” y luego “Propiedades”, luego elegimos la opción “Configuración” y finalmente optamos por la opción “Configuración ASCII”. En esta opción activamos tres primeras casillas.

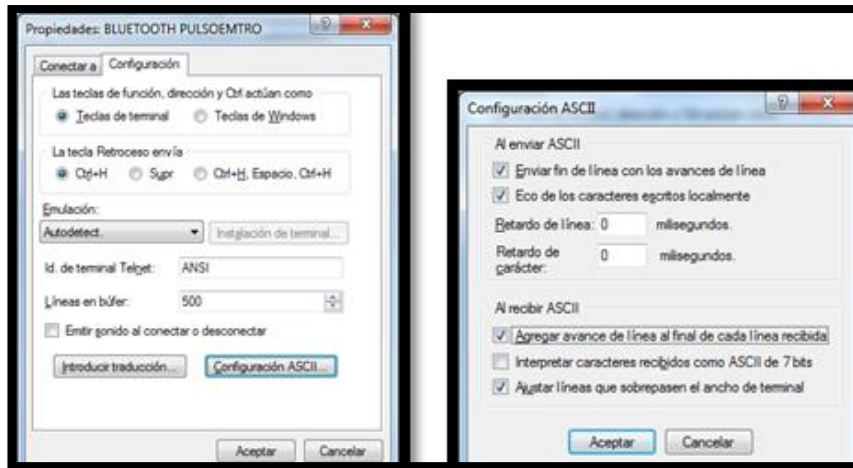


FIGURA 178: Propiedades de configuración ASCII

Fuente: Edison Flores

Se inicia la configuración del módulo Bluetooth HC-05 por medio de los comandos de AT, donde se procedió a configurar el nombre, y la velocidad, la contraseña la dejaremos por default que es “1234”, así como también el modo esclavo.

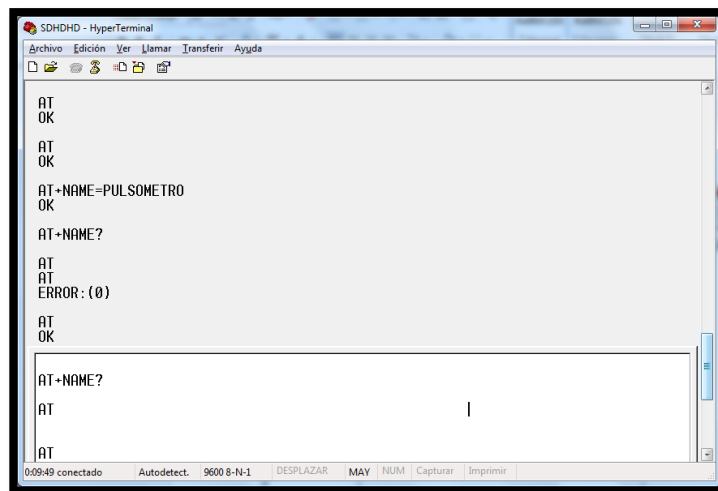


FIGURA 179: Configuración del nombre para el módulo Bluetooth HC-05

Fuente: Edison Flores

Se debe tener en cuenta que la velocidad de transmisión de la tanto en Arduino Lilypad y en el módulo Bluetooth HC-05, deben ser las mismas para tener compatibilidad de envío de datos, en nuestro caso fue a 9600 baudios.

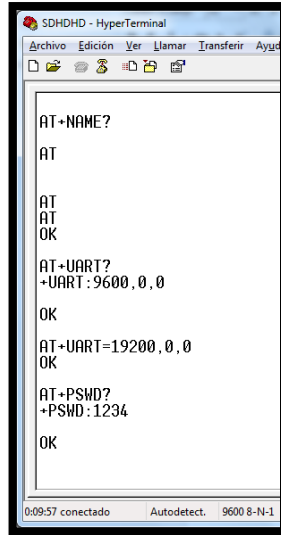


FIGURA 180: Configuración para verificación de velocidad y Contraseña del módulo Bluetooth HC-05

Fuente: Edison Flores

En la siguiente tabla se observa la descripción de cada comando utilizado para la configuración del módulo Bluetooth HC-05.

TABLA 31: Comandos AT para configurar módulo Bluetooth HC-05

COMANDOS AT	DESCRIPCIÓN
AT	Mensaje de inicialización, muestra un ERROR en caso de que la conexión ha fallado, o un OK en caso de que la conexión sea exitosa.
AT+PSWD=	Configura una clave de emparejamiento para el dispositivo.
AT+NAME=	Configura el nombre del dispositivo.
AT+ROLE=	Un 0 significa modo esclavo. Un 1 significa modo maestro.
AT+UART?	Permite conocer la velocidad en baudios de la conexión para la transmisión de datos.
AT+VERSION?	Permite conocer la versión del módulo Bluetooth HC-05.
AT+ADDR?	Permite conocer la dirección MAC de módulo Bluetooth HC-05.

Fuente: Edison Flores

3.11.8 CONEXIÓN DEL SENSOR CARDÍACO

Para la conexión del sensor cardíaco se procedió a conectar los 3 pines del sensor, al hilo conductor. Se añadió una extensión de cable para tener más flexibilidad en su uso. El sensor utilizó un pin para transmisión de señales, y los otros dos para "Vcc" y "Gnd" respectivamente. Además se cubrió el cable del sensor con una tira de tela, para protección del mismo.

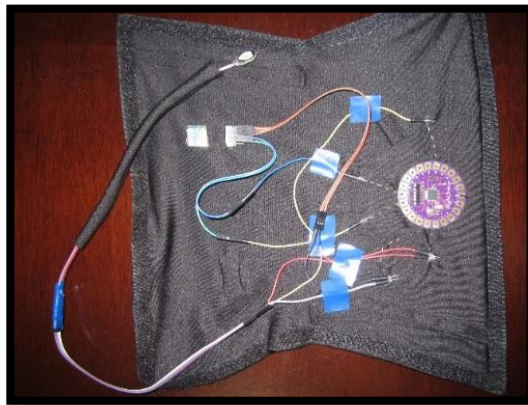


FIGURA 181: Conexión del Sensor Cardíaco

Fuente: Edison Flores

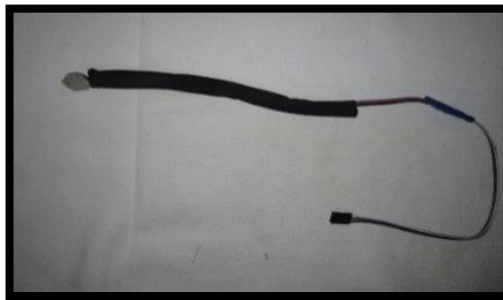


FIGURA 182: Recubrimiento para el cable del sensor

Fuente: Edison Flores

3.11.9 ALIMENTACIÓN ELÉCTRICA DEL SISTEMA

Para la alimentación eléctrica en este diseño final, se utilizó dos baterías lipo, cada una de 3,7 voltios a 680 mA. El hecho de utilizar dos baterías fue porque en diseños anteriores se probó con una sola y se tenía problemas de transmisión de datos con el módulo Bluetooth.

Ahora con dos baterías, una para la placa Arduino Lilypad y el sensor cardíaco, y la otra para el módulo Bluetooth; no hubo inconvenientes en ningún aspecto. Para ver las conexiones de las baterías con los elementos, se puede observar en la figura 158.

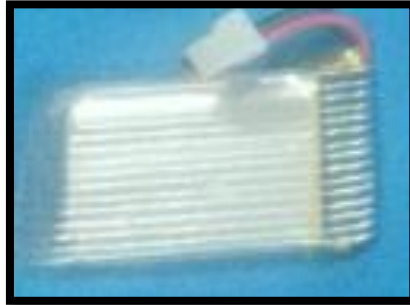


FIGURA 183: Batería Lipo para sistema de monitoreo

Fuente: Edison Flores

Se verifico el voltaje proveniente de la batería lipo, aunque la batería era teóricamente de 3,7 voltios, al momento de medir su voltaje por medio de un multímetro, este fue de 4,09 voltios. Este voltaje no afecta a ningún elemento ya que el rango el rango de funcionamiento de todos estaba entre los 3,3 a 5 voltios.

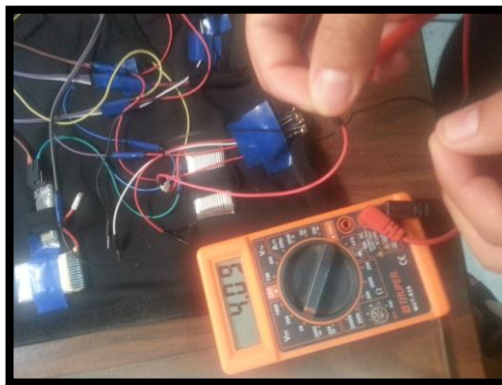


FIGURA 184: Medición de voltaje de batería lipo por medio de multímetro

Fuente: Edison Flores

Para alimentar a la placa Arduino se conecta en los espadines que sobresalen en la parte superior de la placa. Se conecta al primer pin para "Gnd" y al tercer pin para "Vcc". Luego esta placa tiene pines para coser que alimentan a otros dispositivos, en nuestro caso el sensor cardíaco.

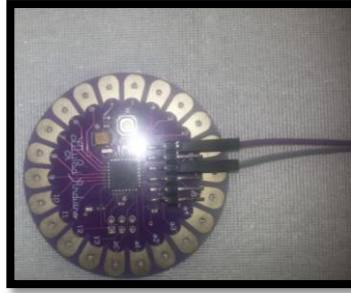


FIGURA 185: Alimentación de entrada para la placa Arduino Lilypad

Fuente: Edison Flores

Podemos observar en la figura 186, que los pines de “voltaje out” de la placa Lilypad, entregan 4,08 voltios, este voltaje es correcto para conectar nuestro sensor cardíaco.



FIGURA 186: Voltaje out de placa Lilypad

Fuente: Edison Flores

3.11.9.1 CONSUMO DE ENERGÍA DEL MÓDULO BLUETOOTH HC-05

En la siguiente tabla se muestra el consumo teórico de corriente del Bluetooth HC-05, pero en la práctica el consumo de corriente tanto en su conexión como en la transmisión de datos va en los 8 mA. Por tanto se puede decir que su rango máximo de consumo de corriente es de 40 mA, y el consumo mínimo de corriente es de 8 mA.

TABLA 32: Consumo de energía del módulo Bluetooth

MODO DE FUNCIONAMIENTO	CONSUMO DE CORRIENTE
En reposo a la escucha de peticiones de conexión	25mA
Detección de dispositivos y establecimiento de la conexión	Varia de 30mA a 40mA
Una vez establecida la conexión para la transmisión y recepción de datos	8mA

Fuente: Edison Flores

3.11.9.2 CONSUMO DE ENERGÍA DE LA PLACA ELECTRÓNICA LILYPAD ARDUINO

La placa Arduino Lilypad funciona con un voltaje que puede ir entre 2,7 a 5,5 voltios. La batería lipo es suficiente para alimentar a esta placa electrónica.

Teóricamente, “Lilypad Arduino permite un consumo de corriente para sus pines de entrada y salida de hasta 40 mA, mientras que para los pines de alimentación y tierra el consumo de corriente es de 200 mA” (Arduino, 2014, p.1).

En la siguiente tabla se muestran los límites del consumo que no se deben superar, por la suma de corriente, en los pines que se encuentran en los siguientes puertos.

TABLA 33: Consumo de corriente límite en pines

DISTRIBUCIÓN DE PINES	CONSUMO DE CORRIENTE
Pines Análogos a0, a1, a2, a3, a4 y a5	150 mA
Pin digital TX y RX	150 mA
Pines Digitales 3, 7, 8, 12, 13	150 mA
Pines Digitales PWM 3, 5, 6, 9, 10, 11	150 mA

Fuente: Edison Flores

3.11.9.3 CONSUMO DE ENERGÍA DEL SENSOR DE PULSOS CARDÍACOS

El sensor cardíaco utiliza un voltaje de funcionamiento entre 3 a 5 voltios. Como se observa en la figura 186 el voltaje que otorga la placa Arduino para el sensor es de 4,08 voltios, más que suficiente para su correcto funcionamiento. Además utiliza 4 mA de corriente y el pin de la placa Arduino Lilypad provee hasta 40mA, por lo que no hubo ningún problema.

3.11.9.4 CONSUMO DE ENERGÍA DE LA BATERÍA LIPO

Las especificaciones que muestra una batería LIPO normalmente vienen dados por el fabricante; donde los datos característicos se muestran en la siguiente tabla.

TABLA 34: Consumo de energía de batería Lipo

DESCRIPCIÓN	VALORES
Voltaje	3.7 voltios
Amperaje o Capacidad de la Batería	680 mAH (mili amperios hora)
Velocidad de Descarga Máxima	20C

Fuente: Edison Flores

Una de las características de esta batería LIPO es que el momento de su carga completa ofrece 4,1 voltios, mientras que cuando se encuentra en un estado de descarga su voltaje no baja de los 3 voltios. Esta batería se compone por una sola celda, y cada celda permite obtener un voltaje de 3.7 voltios.

3.11.9.5 TIEMPO DE DURACIÓN DE LA BATERÍA LIPO

Para saber el tiempo de duración que puede proporcionar la batería LIPO es necesario saber sus datos que provee el fabricante como se puede observar en la tabla 33. A continuación se presenta una fórmula que nos permite saber el tiempo de duración que posee una batería LIPO en función del tiempo.

$$\text{Tiempo de duración (min)} = \frac{\text{Capacidad de la batería (Amp * min)}}{\text{Velocidad de descarga maxima (Amp)}}$$

La capacidad es un parámetro que nos indica la cantidad de energía que puede llegar a almacenar nuestra batería, y se mide en miliamperios hora (mAH). En nuestro caso la batería LIPO elegida tiene la capacidad de 680 mAH, pasando a Amperios por minutos, tenemos 0.68 amperios y que multiplicado por 60 minutos se obtienen 40.8 Amp*min.

La velocidad de descarga, se define como la rapidez con la que la batería se puede descargar de forma segura, es decir, la cantidad de amperios que la batería nos puede suministrar durante una hora de forma continua, y que normalmente viene expresada en referencia a su capacidad 20C. El valor de la velocidad de descarga máxima se obtiene de la multiplicación del amperaje medido en amperios por el valor de la velocidad de descarga como se muestra a continuación.

$$\text{Velocidad de descarga máxima (A)} = 20 \times 0.68 \text{A} = 13.6 \text{ Amperios}$$

Una vez obtenidos los datos se reemplaza en la fórmula del tiempo de duración, obteniendo los siguientes resultados.

$$\text{T tiempo de duración (min)} = \frac{40.8 \text{ (Amp * min)}}{13.6 \text{ (Amp)}} = 3 \text{ minutos}$$

El resultado mostrado nos dice que con una corriente máxima teóricamente utilizada de 680 mA que la batería puede suministrar, su tiempo de duración de descarga máxima será de 3 minutos es decir en constante funcionamiento. Cabe aclarar que este es un resultado teórico, ya que el consumo que demanda el circuito depende de los componentes electrónicos que lo compongan por lo tanto el tiempo de duración aumentara.

Para la placa electrónica LilyPad Arduino el consumo de corriente por cada pin es de 40mA, y si se tiene una fuente de alimentación externa que en este caso es una batería LIPO que suministra una capacidad máxima de 680 mA; dividiendo la capacidad de amperaje que presenta la batería LIPO para el consumo de corriente de la placa electrónica, el tiempo de duración de la batería esta alrededor de las 17 horas.

Es importante saber que el voltaje máximo que suministra esta batería LIPO este es de 4.09 voltios, como se observa en la imagen 184.

3.11.10 PRUEBAS Y RESULTADOS DEL DISEÑO FINAL

En este diseño final se realizó las respectivas pruebas de funcionamiento, tanto en envío de datos, de alimentación eléctrica, pruebas de conexión y pruebas de comodidad de la muñequera. Los resultados fueron los siguientes.

3.11.10.1 PRUEBA DE ENVÍO DE CARACTERES CORRECTOS

Se realizó la prueba de envío de caracteres, de dos formas, una fue mediante una comunicación serial, donde se podían observar en el monitor serial del IDE de Arduino los datos enviados.

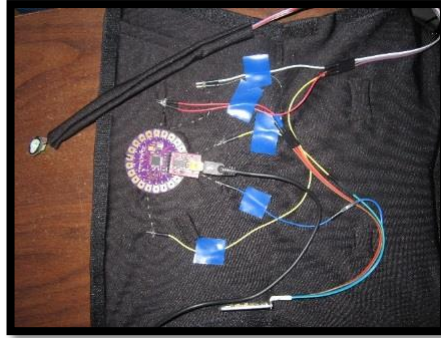


FIGURA 187: Conexión serial para verificación de datos con el IDE de Arduino

Fuente: Edison Flores

Se realizó la verificación de las diferentes variables del sistema de monitoreo, a través de la comunicación serial. Como resultado se obtuvo que todos los datos fueron transmitidos correctamente. A utilizar el mismo código, no hubo problema con este y las conexiones tampoco alteraron el mismo.

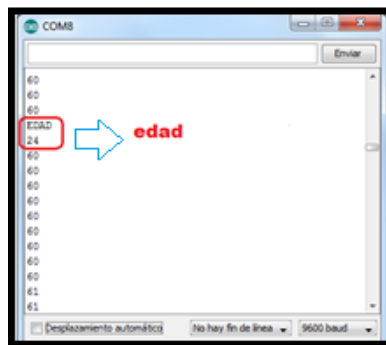


FIGURA 188: Verificación de edad a través de monitor serial de Arduino (diseño final)

Fuente: Edison Flores

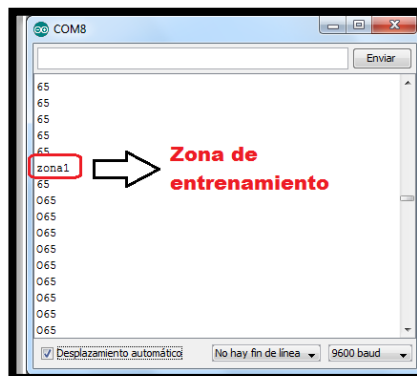


FIGURA 189: Verificación de zonas a través de monitor serial de Arduino (diseño final)

Fuente: Edison Flores

3.11.10.2 PRUEBAS DE ALIMENTACIÓN ELÉCTRICA

Para las pruebas en el aspecto de la alimentación eléctrica, se probó primero con la utilización de una sola batería lipo de 3,7 voltios a 680 mA. El sistema encendió correctamente, pero al igual que en el diseño #2 y #3, se tuvo problemas en la comunicación de datos, ya que se envía por un momento y se cortaba la transmisión de datos.

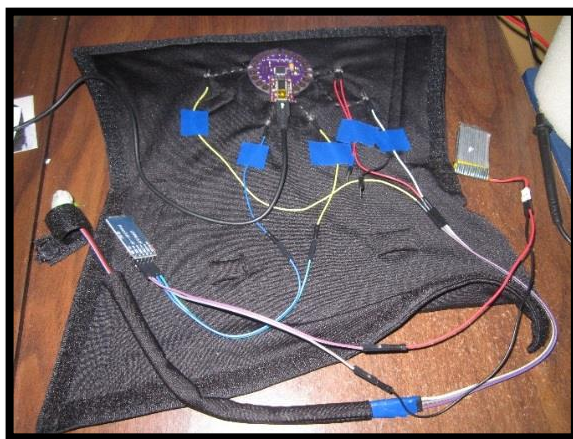


FIGURA 190: Conexión eléctrica con una batería en el diseño final

Fuente: Edison Flores

Al ver este resultado negativo, se procedió a la utilización de otra batería. Cuando se probó la conexión con las dos baterías del mismo voltaje y corriente, la conexión tuvo éxito, ya que no se volvió al problema de transmisión de datos. Con esto se solucionó el problema, y también hizo fiable al sistema, al transmitir los pulsos cardíacos constantemente.

Adicionalmente en la parte eléctrica, en este diseño final se incluyó la utilización de dos switch, uno por cada batería, para un mejor uso del sistema en el encendido y apagado. La implementación de los switch se probó primero en un protoboard para posteriormente unirlos a la muñequera. El usuario sabrá que el sistema está encendido, cuando la palanca del switch, este junto al punto azul que se encuentra en el mismo.

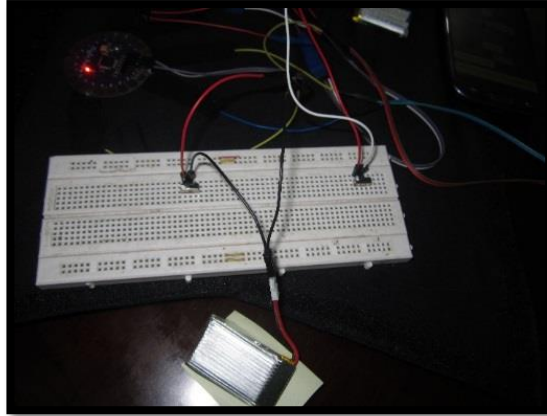


FIGURA 191: Pruebas de switch en protoboard (diseño final)

Fuente: Edison Flores

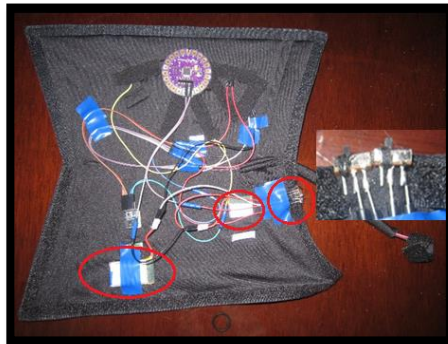


FIGURA 192: Sistema con dos baterías y switch incorporado

Fuente: Edison Flores



FIGURA 193: Ubicación de los switch en la muñequera

Fuente: Edison Flores

3.11.10.3 PRUEBAS DE COMODIDAD DE LA MUÑEQUERA

En este diseño final, al presentar mejoras en su diseño, las pruebas de comodidad fueron muy positivas para el uso de la persona que realice ejercicio físico. La muñequera de tela tipo licra, de textura suave para la piel, junto con unas delgadas almohadillas (esponjas), hizo que está presente mucho confort para su uso.



FIGURA 194: Vista aérea de la muñequera (diseño final)

Fuente: Edison Flores



FIGURA 195: Vista lateral de la muñequera (diseño final)

Fuente: Edison Flores

Además se probó también el sistema en conjunto con la banda en el brazo, esta sirvió para dar comodidad al usuario cuando lleve el celular, ya que este, es el que emite las alarmas, y se necesita que estén cerca de nuestro oído; una alarma fue por medio de vibración y la otra por sonido.



FIGURA 196: Muñequera más banda de brazo

Fuente: Edison Flores

3.11.11 ANÁLISIS DE RESULTADOS DEL DISEÑO FINAL

Realizadas las diferentes pruebas en el diseño final, se pudo observar que el sistema de monitoreo cardíaco fue totalmente exitoso en cuanto a la programación realizada tanto en Arduino, como en Android. Como no se cambió de programación, y al usarla en prototipos anteriores, es evidente que en el diseño final no se presentaría ningún problema.

En cuanto a la parte de alimentación eléctrica, con la utilización de las dos baterías de 3,7 voltios 680mA, se pudo verificar que el sistema funcionó correctamente y de manera perfecta en el diseño final. Además también se probó junto con la conexión de los switch. Este parámetro fue corregido, ya que en prototipos anteriores se tenía problemas.

En lo pertinente al diseño de la muñequera, se presentaron ventajas en el momento de la costura con el hilo conductor, se tuvo suficiente espacio para canalizar el hilo. Además de los recubrimientos sobre el mismo fueron parte fundamental para que el usuario este tranquilo de que no se producirá un cortocircuito por la unión de este hilo. Se menciona que con respecto a diseños anteriores, en este diseño final, la muñequera tiene dos medidas de ajuste, es robusta y cómoda; lo que lo hace muy versátil para su uso

CAPÍTULO IV

4 ANÁLISIS ECONÓMICO

En este capítulo se realiza el análisis económico de los materiales usados con respecto al diseño del sistema de monitoreo cardíaco y de los componentes electrónicos implementados en el mismo, así como de la muñequera y del Smartphone, el cual sirvió de plataforma para el desarrollo de la aplicación móvil.

4.1 PRESUPUESTO PARA EL DISEÑO DE LA MUÑEQUERA

Para el diseño estético de la muñequera que se utilizó en el sistema de monitoreo de frecuencia cardíaca en el diseño final, en la tabla 34 se puede observar los costos que se emplearon, así como de los demás materiales que se utilizaron en la construcción de la misma.

TABLA 35: Presupuesto del sistema de monitoreo cardíaco

MATERIAL	COSTO UNITARIO	UNIDADES	COSTO TOTAL
¼ de tela tipo licra	\$ 5	1	\$ 5
Esponja	\$ 1	1	\$ 1
Hilo normal	\$ 0,50	2	\$ 1
Velcro (25 cm)	\$ 0,30	9	\$ 2,70
Mano de obra	\$ 15		\$ 15
TOTAL			\$ 24,70

Fuente: Edison Flores

Como se observa en la tabla 34, el costo total de todos los materiales utilizados, incluyendo el rediseño y mano de obra, se valora en un total de 24,70 dólares americanos.

4.2 PRESUPUESTO DEL HARDWARE UTILIZADO

A continuación se presentan los costos generales de los elementos electrónicos y demás elementos que ayudaron a la implementación desde el diseño del primer prototipo hasta el diseño final del mismo, como se observa en la siguiente tabla 35.

TABLA 36: Presupuesto de componentes electrónicos implementados

MATERIAL	COSTO UNITARIO	CANTIDAD	COSTO TOTAL
Módulo Bluetooth HC-05	\$ 18.00	1	\$ 18.00
Arduino LilyPad Atmega 128	\$ 20.00	1	\$ 20.00
Sensor de Pulsos Cardíacos	\$ 35.00	1	\$ 35.00
Batería Lipo de 3,7v a 680mA	\$ 16.00	2	\$ 32.00
Bobina de Hilo Conductor (9m)	\$ 6.0	1	\$ 6.00
Cables de protoboard	\$0,10	20	\$ 2.00
Switch	\$ 0.25	2	\$ 0.50
Espadines tipo Hembra	\$ 1.10	1	\$ 1.10
Espadines tipo Macho	\$ 1.10	1	\$ 1.10
Estaño (1m)	\$ 0.35	1	\$ 0.35
Cautín	\$ 5.00	1	\$ 5.00
Brujita	\$ 0.50	1	\$ 0.50
Acido	\$ 1,50	1	\$ 1,50
Pistola de Silicona	\$ 5.00	1	\$ 5.00
Tubos de Silicona	\$ 1.50	1	\$ 1.50
Diodos LED	\$ 0.50	2	\$ 1.00
TOTAL			\$ 130,55

Fuente: Edison Flores

Los costos de los elementos electrónicos utilizados solo para el diseño final, son los que se muestran a continuación en la tabla 36.

TABLA 37: Costos del diseño final

MATERIAL	COSTO UNITARIO	CANTIDAD	COSTO TOTAL
Módulo Bluetooth HC-05	\$ 18.00	1	\$ 18.00
Arduino Lilypad Atmega 128	\$ 20.00	1	\$ 20.00
Sensor de Pulsos Cardíacos	\$ 20.00	1	\$ 20.00
Batería Lipo de 3,7v a 680mA	\$ 16.00	2	\$ 32.00
Bobina de Hilo Conductor (9m)	\$ 6 .0	1	\$ 6.00
Cables de protoboard	\$0,10	20	\$ 2.00
Switch	\$ 0.25	2	\$ 0.50
TOTAL			\$ 98,50

Fuente: Edison Flores

A continuación en la tabal 37 se muestra el costo total del dispositivo móvil inteligente, con sistema operativo Android, el cual sirvió de plataforma para el desarrollo de la aplicación móvil en App Inventor.

TABLA 38: Presupuesto del dispositivo móvil inteligente

MATERIAL	COSTO UNITARIO	CANTIDAD	COSTO TOTAL
Samsung Galaxy III	\$ 285.00	1	\$ 285.00
TOTAL			\$ 285.00

Fuente: Edison Flores

Es importante mencionar que el costo para la adquisición de un dispositivo móvil no es fijo, es decir, que se puede utilizar otro dispositivo móvil pero que tenga las características mínimas, como en su sistema operativo Android (versión mínima 4.1.2), memoria interna de 512MB (memoria interna mínima que requiere) y 250MB de memoria RAM (memoria RAM mínima que requiere).

Existen teléfonos inteligentes que se pueden adquirir desde los 150 dólares americanos hasta los 1000 dólares americanos que cumplen con los requerimientos de funcionamiento, dependiendo de las condiciones económicas de cada usuario que adquiera el sistema de monitoreo cardiaco

4.3 PRESUPUESTO DEL SOFTWARE UTILIZADO

El software que se utilizó tanto en el desarrollo de la aplicación móvil en App Inventor y en la programación realizada en el IDE de Arduino son bajo la arquitectura de Open Source y de descarga libre, se detallan los costos en la tabla 38.

TABLA 39: Presupuesto del software

DESCRIPCIÓN	CANTIDAD	VALOR TOTAL
IDE de Arduino	1	\$ 0
App Inventor	1	\$ 0
MIT AI2 Companion	1	\$ 0
TOTAL		\$ 0

Fuente: Edison Flores

4.4 ANÁLISIS COSTO BENEFICIO

En este análisis se detallan todos los costos empleados en el diseño final del sistema de monitoreo de ritmo cardíaco, tomando en cuenta que los costos deben ser tangibles, es decir que se puedan medir en alguna unidad económica, que en este caso son dólares americanos; mientras que los beneficios son determinados de manera personal ya que estos permitirán obtener beneficios para la salud y el buen vivir de las persona que realizan o no ejercicio físico.

En la tabla 39 se observan los valores de los costos que implicaron obtener el desarrollo del proyecto.

TABLA 40: Costos del proyecto

DESCRIPCIÓN	VALOR
Costos de la muñequera del sistema de monitoreo de ritmo cardíaco	\$ 24,70
Costos de elementos electrónicos	\$ 98.50
Costo del dispositivo móvil inteligente	\$ 285.00
Costo del software	\$ 0
TOTAL COSTOS	\$ 408,2

Fuente: Edison Flores

Los beneficios que se obtienen al finalizar la implementación del sistema de monitoreo de ritmo cardíaco, es el permitir a todas las personas realizar actividad física de manera planificada y controlada, ayudándoles así, al cumplimiento de una actividad deportiva que favorecerá a su salud, en la prevención de lesiones, posibles taquicardias, entre otras. Ayudará a las personas, sean deportistas profesionales o no, a adaptar el ejercicio a los objetivos que se haya propuesto, controlando la intensidad del ejercicio e informando sobre las mejoras de la condición física, a través del ritmo cardíaco en reposo

Cumplirá uno de los objetivos del “Plan Nacional del Buen Vivir” que es mejorar la calidad de vida de la población, garantizando la salud desde la generación de prácticas saludables; además de motivar a la práctica deportiva permanente o eventual y de manera saludable.

Se menciona también que el deporte es uno de los mejores métodos para unir a distintas personas y mejorar así la capacidad de socialización de cada persona.

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Con el desarrollo del Sistema de Monitoreo de Ritmo Cardíaco (S.M.R.C.) se permitirá a todas las personas, sean aficionados o profesionales en la práctica deportiva, a realizar ejercicio físico de manera planificada y controlada; siendo la clave para obtener beneficios como el fortalecimiento del corazón, activar zonas inactivas del cerebro, mantener fuertes los huesos y músculos, así como también será el modo correcto de llevar una vida saludable con la prevención de lesiones musculares.
- El entrenamiento según la frecuencia cardíaca, mediante la aplicación de la fórmula de Karvonen, es el método más usado en la actualidad por los entrenadores, para realizar de manera correcta la práctica del ejercicio aeróbico. Los rangos de pulsaciones establecidos en este procedimiento, dependen de características fisiológicas y cardiovasculares propias de cada persona, teniendo como resultado el cumplimiento del principio de la individualización en el entrenamiento deportivo.
- El sistema de monitoreo de ritmo cardíaco ayudará a las personas que desconozcan de la utilización de un Pulsómetro, este sistema proveerá de una aplicación gráfica muy intuitiva para cualquiera, así como también las personas tendrán la confianza de que cuando realicen el ejercicio, el sistema le avisará los límites exactos en los que deberá entrenar, aspecto importante que solo los Pulsómetros de gama alta lo hacen.
- Se investigó las características de cada elemento electrónico que fue implementado en el sistema, donde el sensor cardíaco se adaptó de la mejor manera al mismo, para el senso del pulso cardíaco en las personas, también se empleó la placa electrónica LilyPad Arduino ATmega128 y sus ventajas de uso junto con el hilo conductor cumplieron con los requerimientos tanto de sujeción a la muñequera como del uso de pines necesarios para las respectivas conexiones; además de la utilización de un módulo Bluetooth HC-05 que permitió establecer y mantener una comunicación inalámbrica de corto alcance.

- La realización de la aplicación llamada Pulsómetro, con desarrollo de programación en bloques en el programa App Inventor bajo sistema Operativo Android, que en conjunto con un Smartphone, cumplieron con la función de informar al usuario las alarmas emitidas por salirse de un rango de pulsaciones establecidos, una vibrando y la otra auditivamente. Además por medio de este dispositivo sirvió para la comunicación inalámbrica mediante tecnología Bluetooth.

5.2 RECOMENDACIONES

- El presente proyecto da una pauta inicial para el desarrollo de futuros sistemas electrónicos que estén relacionadas al beneficio de la salud y del entrenamiento deportivo en sus diferentes ámbitos.
- Utilizar las hojas de datos (data sheets) de los elementos electrónicos, estas permiten una correcta configuración y conexión de los mismos.
- Al momento de utilizar el hilo conductor, es decir a la hora de coser, se debe tener cuidado en el recorrido del mismo, por que si el hilo se topa entre sí, causará un cortocircuito, pudiendo quemar y alterar el desempeño de todos los elementos electrónicos conectados. Es por ello se aconseja que antes de coser se debe tener un esquemático de guía del cosido, para organizar los caminos del hilo conductor.
- Se recomienda el uso de software libre, como es el programa App Inventor para el desarrollo de una aplicación móvil e IDE de Arduino, ya que estos permiten reducir los costos en el desarrollo de proyectos.
- Se recomienda la aplicación del manual de usuario adjunto en el proyecto, antes de usarlo, con la finalidad de ver el correcto funcionamiento y evitar posibles daños en el sistema.
- Se recomienda para futuros proyectos el desarrollar de un sistema de monitoreo de frecuencia cardíaca, para el ámbito de la medicina, relacionadas con patologías de arritmias; que permita la verificación en tiempo real de pulsos cardíacos y alertas, en una plataforma en la internet.

5.3 BIBLIOGRAFÍA

- WILMORE J. (2007), *Fisiología del esfuerzo y del deporte*, Editorial Paidotribo, España.
- BOMPA T. (2005), *Entrenamiento para jóvenes deportistas*, Ed. Hispano Europea, España.
- CONDE M. (2000), *Organización del Entrenamiento de la Resistencia*, Ed. Instituto Monsa, España.
- TORRENTE O. (2013), *Arduino:Curso práctico de información* Ed. Rc. Libros,
- DOMÍNGUEZ E, (2013), *Redes de comunicación de datos (Circuitos eléctricos auxiliares del vehículo)*, Ed. Editex
- JIMENEZ I. (2010), *Operaciones auxiliares con tecnologías de la información y la comunicación*, Ed. Paraninfo.
- ANDREW, (2003), *Redes de computadoras*, Editorial Pearson
- PALLÁS R. (2004), *Sensores y acondicionadores de señal*, Editorial Marcombo.
- WEINECK J. (2005), *“Entrenamiento Total”*, Editorial Paidotribo, España.
- RITTINGHOUSE, J, (2010), *Cloud Computing, Implementation, Management And Security*, Editorial CRC Press.
- HOHMANN A. (2005), *Introducción a las ciencias del entrenamiento*, Editorial Paidotribo, España.
- PLATONOV V. (2001), *La Preparación Física*, Editorial Paidotribo, España.
- SALINAS N. (2005), *Manual para técnico de sala de Fitness*, Editorial Paidotribo, España.
- GUIMARES T. (2002), *El Entrenamiento Deportivo. Capacidades Físicas*, Editorial UENED, Costa Rica.
- GRANELL J. (2004), *Las técnicas de atletismo. Manual práctico de enseñanza*, Editorial Paidotribo, 17/5/2004
- PÉREZ P. (2012), *Yo soy tu entrenador personal*, Editorial Lulu.com

5.4 LINKOGRAFÍA

- PEREZPLATA N. (2011), *La Frecuencia Cardíaca*, Recuperado de:
<http://www.frecuencia-cardiaca.com/>
- Enríquez R. (2009), *Guía del usuario Arduino*, Recuperado de:
http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf
- ARDUINO (2015), *LilyPad Arduino USB*, Recuperado de:
<https://www.arduino.cc/en/Main/ArduinoBoardLilyPadUSB>
- ARDUINO (2015), *LilyPad Arduino Simple*, Recuperado de:
<https://www.arduino.cc/en/Main/ArduinoBoardLilyPadSimple>
- LÓPEZ C. (2013), *Bluetooth*, Recuperado de:
<http://arquitecturadecomputadora.wordpress.com/2013/05/23/bluetooth/>
- LÓPEZ C. (2013), *Cómo funciona Bluetooth*, Recuperado de:
<http://es.kioskea.net/contents/69-como-funciona-bluetooth>
- GEEKFACTORY (2012), *Bluetooth HC-05 y HC-06 Tutorial de Configuración*, Recuperado de: <http://www.geekfactory.mx/radio/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>
- MOLINA J. (2014), *Sensores*. Recuperado de:
<http://www.profesormolina.com.ar/index.htm>
- BERNABEU E. (2013), *Sensores Ópticos*, Recuperado de:
http://pendientedemigracion.ucm.es/info/otri/complutecno/fichas/tec_ebernabeu2.htm

- KIOSERA (2014), *App Inventor*, Recuperado de:
<http://es.kioskea.net/faq/4873-app-inventor-crea-tus-propias-aplicaciones-para-tu-smartphone>
- GARCIA J. (2005), *Textiles Inteligentes*, Recuperado de:
http://www.solbru.cat/pdf/induyco_estudio.pdf
- GÓMEZ J. (2011), “Ropa Inteligente”, Recuperado de:
<http://ropainteligente.blogspot.com/2011/05/secretos-invisibles.html>
- CHAVES P. (2008), *Sensores Analógicos*, Recuperado de:
http://eie.ucr.ac.cr/uploads/file/proybach/pb_08_II/pb0811t.pdf
- RAMOS C. (2014), *Zonas de Entrenamiento*, Recuperado de:
<http://sudandola.com/2014/10/27/como-entrenar-segun-la-frecuencia-cardiaca/>
- LÓPEZ F. (2011), *Programación en bloques*, Recuperado de:
<http://tel.unir.net/~flopez/macprog/bloques.pdf>
- MUÑOZ, A. (2014). *Textiles Electrónicos*. Recuperado de:
<http://www.corriendocontijeras.com/los-textiles-electronicos-de-amor-munoz/>
- GUTIÉRREZ, J. & Fonseca, E. & Marquina, R. (2012). *Conductores Eléctricos*. Recuperado de: <http://es.slideshare.net/ronyjmv73/conductores-electricos-15073033>
- PLAZA, E. (2011). *Historia del nacimiento de Bluetooth*. Recuperado de:
<http://www.wayerless.com/2011/09/la-historia-del-nacimiento-de-bluetooth/>
- LORENZO C. (2010), *Redes y Bluetooth*, Recuperado de:
<http://www.luiskano.net/blog/2009/11/04/usar-tu-celular-como-camara-web/>
- IBARRA J. (2012), *Versiones Bluetooth*, Recuperado de:
<https://sena11a.files.wordpress.com/2012/10/quc3a9-es-bluetooth.docx>

- VÁSQUEZ, B. & Zúñiga, F. (2006). *Tecnologías Inalámbricas de corto alcance Bluetooth y ZigBEE*. Recuperado de:

<http://dspace.ups.edu.ec/handle/123456789/206>

- SHAWN M. (2012), Limitaciones Bluetooth, Recuperado de:

http://www.ehowenespanol.com/cuales-son-limitaciones-del-bluetooth-lista_118026/

- MARTÍNEZ (2013), “App Store”, Recuperado de: <http://creiserds.com/la-web-se-adapta-la-era-post-pc/>

- REDERJO, J. (2013). *Uso de AppInventor en la materia de Tecnologías de la Información y Comunicación*. Recuperado de: Salgado, I. (2012). *ZigBee y sus aplicaciones*. Recuperado de:

<http://www.dea.icaei.upco.es/sadot/Comunicaciones/avanzadas/Zigbee%20y%20sus%20aplicaciones.pdf>

- MORENO (2013), *Desarrollo Android usando MIT App Inventor*, <http://www.etnassoft.com/biblioteca/desarrollo-android-usando-mit-app-inventor/>

REVISTAS DIGITALES

- GUEVARA A. (2010), *Seguridad de Redes*, Recuperado de: <http://revista.seguridad.unam.mx/numero-07/dispositivos-m%C3%B3viles>, publicado el Vie, 06/08/2010.

- MAZZEO E. (2009), *El entrenamiento de la Resistencia Aeróbica*, Recuperado de:

http://www.portalfitness.com/7162_el-entrenamiento-de-la-resistencia-aerobica-pagina-iii.aspx

- DIARIO MÉDICO, (2015), *El ciclismo, un deporte*, Recuperado de: [saludablehttp://www.dmedicina.com/vida-sana/salud-y-deporte/el-ciclismo-un-deporte-saludable](http://www.dmedicina.com/vida-sana/salud-y-deporte/el-ciclismo-un-deporte-saludable), Editorial Revistas, S.L.U

5.5 ANEXOS

ANEXO 1: CÓDIGO PRINCIPAL DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO

```
// VARIABLES
int pulsoPin = 0;           // Sensor de Pulso cable violeta conectado al pin analógico 0
int intermitentePin = 13;  // Pin para LED intermitente en latidos (1)
int intermitente_Pin = 5;  // Pin parpadeo en cada latido (opcional)
int desvan_velo = 0;      // Utilizado a desvanecerse LED con PWM en pin5
char cadena [3];         // almacenamiento de datos en vector
int decena;              //decena de edad
int unidad;              //unidad de edad
int edad;                //almacenamiento de edad
int i=0;                 //switch para el cálculo de edad
int e=0;                 // swirh para pulso normal
int j=0;                 // SWITCH PARA PULSO DE RESERVA, le aumento igual a 0
int p=0;                 //Switch para género: masculino o femenino
int pnormal;            // pulso normal
int dec;                 // decena de pulso normal
int uni;                 // unidad de pulso normal
int preserva;           // pulso reserva
int centena;            //centena de pulso de reserva
int dece;                // decena de pulso reserva
int unid;                // unidad de pulso reserva
int zona;                // variable para switch zona
int frecmax;            // frecuencia cardíaca máxima

int pulreservatotal;     // almacenamiento. F.C.M - Pnormal

// Variables de zonas//////////
float lim_1;             // variables de límites de zona 1
float lim_2;             // variable de límites de zona 1-2
float lim_3;             // variable de límites de zona 2-3
float lim_4;             // variable de límites de zona 3-4
float lim_5;             // variable de límite de zona 4-5
float lim_6;             // variable de límites de zona 5

// Variables volátiles usada durante la rutina
volatile int BPM;        // Se utiliza para mantener la frecuencia del pulso
volatile int Signal;     // Mantiene los datos en bruto de entrada
volatile int IBI = 600;  // Mantiene el tiempo entre los latidos, inter-Intervalo
volatile boolean Pulse = false; // verdadero cuando la onda de pulso es elevado, false cuando es
baja
volatile boolean OP = false; // Se convierte en realidad cuando Arduino encuentra un latido

void setup(){
  pinMode(intermitentePin,OUTPUT); // pin para el led intermitencia (opcional)
  pinMode(intermitente_Pin,OUTPUT); // pin para el led de parpadeo (opcional)

  Serial.begin(9600); // velocidad de comunicación serial en baudios
  interruptSetup(); //establece para leer pulsos de señal del sensor cada 2mS, ingreso de la
  interrupción}

void loop(){
  Serial.println(BPM); // imprime los BPM
  delay (200);
```

```

////////// -----SWITCH ZONAS-----
switch(zona){
case 1:////////// ZONA 1-----
  led_latidos();
  if (BPM<lim_1)
  {
    Serial.print("O"); // PULSO BAJO
  }
  if (
                                (BPM>lim_1)&&(BPM<lim_2))
  {
    Serial.print("ok");
  }
  if (BPM>lim_2)
  {
    Serial.print("P"); // PULSO ALTO
  }
  break;

case 2:////////// ZONA 2----- ENVIO DE CARACTERES A LA APLICACION MOVIL

  led_latidos();
  if (BPM<lim_2)
  {
    Serial.print("Q"); // PULSO BAJO
  }
  if (
(BPM>lim_2)&&(BPM<lim_3))
  {
    Serial.print("ok");
  }
  if (BPM>lim_3)
  {
    Serial.print("R"); // PULSO ALTO
  }
  break;

case 3:////////// ZONA 3-----ENVIO DE CARACTERES A LA APLICACION
MOVIL

  led_latidos();
  if (BPM<lim_3)
  {
    Serial.print("S"); // PULSO BAJO
  }
  if (
(BPM>lim_3)&&(BPM<lim_4))
  {
    Serial.print("ok");
  }
  if (BPM>lim_4)
  {
    Serial.print("T"); // PULSO ALTO
  }
  break;

```

```
case 4:////////// ZONA 4-----ENVIO DE CARACTERES A LA APLICACION MOVIL
```

```
led_latidos();  
if (BPM<lim_4)  
{  
  Serial.print("U"); // PULSO BAJO  
}  
if (  
(BPM>lim_4)&&(BPM<lim_5))  
{  
  Serial.print("ok");  
}  
if (BPM>lim_5)  
{  
  Serial.print("V"); // PULSO ALTO  
}  
  
break;
```

```
case 5:////////// ZONA 5-----ENVIO DE CARACTERES A LA APLICACION MOVIL
```

```
led_latidos();  
if (BPM<lim_5)  
{  
  Serial.print("X"); // PULSO BAJO  
}  
if (  
(BPM>lim_5)&&(BPM<lim_6))  
{  
  Serial.print("ok");  
}  
if (BPM>lim_6)  
{  
  Serial.print("Y"); // PULSO ALTO  
}  
break;
```

```
////////// DETENER//////////
```

```
case 6:
```

```
interruptSetup();  
break;  
}  
  
} // fin de void loop()
```

```

void led_latidos(){
  if (OP == true){          // BANDERA automática, para verificar si hay pulsos
  desvan_velo = 255;       // Desvanecer el led de parpadeo

    OP = false;           // Restablecer el indicador(bandera)
  }
  desvan_velo -= 15;       // Conf. para desvanecer LED
  desvan_velo = constrain(desvan_velo,0,255); // Mantener el valor de desvanecimiento LED
  analogWrite(intermitente_Pin,desvan_velo); // producir intermitencia
  delay(10); // tiempo mínimo
}

////////////////////////////////EVENTOS SERIALES POR ZONAS////////////////////////////////

void serialEvent ()
{
  while(Serial.available())
  {
    char inChar=(char)Serial.read();          // Se activa el serial read, para leer los datos
    switch(i)                                 // inicio del switch (i)
    {
      case 0:

        cadena[i]=inChar;
        switch(cadena[i])
        {
          ////////////////////////////////// CASOS////////////////////////////////
          //////////////////////////////////ZONAS////////////////////////////////
          case 'A':
            Serial.println("zona1"); //Para comprobar (opcional)
            i=0;
            zona=1;
            break;

          case 'B':
            Serial.println("zona2"); //Para comprobar (opcional)
            i=0;
            zona=2;
            break;

          case 'C':
            Serial.println("zona3"); //Para comprobar (opcional)
            i=0;
            zona=3;
            break;

          case 'D':
            Serial.println("zona4"); //Para comprobar (opcional)
            i=0;
            zona=4;
            break;

          case 'E':
            Serial.println("zona5"); //Para comprobar (opcional)
            i=0;
            zona=5;
            break;
        }
      }
    }
  }
}

```

```

case 'K': /////////////////////////////////////////////////// CASA PARA DETENER EL ENTRENAMIENTO////////////////////////////////////
    Serial.println("detener"); //Para comprobar (opcional)
    i=0;
    zona=6;
    break;

//////////////////////////////////EDAD////////////////////////////////////

    case 'F':
        cadena[1]=' ';
        cadena[2]=' ';
        Serial.println("EDAD"); //Para comprobar (opcional)
        i++;
        break;
    }
    break;

case 1:
    cadena[i]=inChar;
    decena=atoi(&cadena[i]);
    i++;
    break;

case 2:
    cadena[i]=inChar;
    unidad=atoi(&cadena[i]);
    edad=(decena*10)+unidad; // retorno de edad
    Serial.println (edad); //Para comprobar (opcional)

    i=0;
    break;

} // fin de switch (i)

////////////////////////////////// PULSO NORMAL////////////////////////////////////

switch(e) //inicio del switch (e)
{
case 0:
    cadena[e]=inChar;
    switch(cadena[e])
    {

case 'H':
        cadena[1]=' ';
        cadena[2]=' ';
        Serial.println("Pnormal"); // Para comprobar
        e++;
        break;
    }
    break;

case 1:
    cadena[e]=inChar;
    dec=atoi(&cadena[e]);
    e++;
    break;
}

```

```

case 2:
  cadena[e]=inChar;
  uni=atoi(&cadena[e]);
  pnormal=(dec*10)+uni;      // retorno del pulso normal
  Serial.println (pnormal);
  e=0;
  break;

} // fin switcj (e)

//////// PULSO RESERVA//////// (opcional, ya q no se utiliza, este pulso se lo obtiene de funciones
matemáticas)
switch(j)                    //inicio del switch (j)
{
case 0:
  cadena[j]=inChar;
  switch(cadena[j])
  {

case 'I':
  cadena[1]=' ';
  cadena[2]=' ';
  Serial.println("PRESERVA");
  j++;
  break;
}
break;

case 1:
  cadena[j]=inChar;
  centena=atoi(&cadena[j]);
  j++;
  break;

case 2:
  cadena[j]=inChar;
  dece=atoi(&cadena[j]);
  j++;
  break;

case 3:
  cadena[j]=inChar;
  unid=atoi(&cadena[j]);

  preserva=(centena*100)+(dece*10)+ unid;    // retorno de pulso de reserva
  Serial.println(preserva);                 //Para comprobar (opcional)

  j=0;
  break;

} // fin switcj (j)

```



```
//////////////// GÉNERO Y FÓRMULAS////////////////
```

```
switch(p)          //inicio del switch (p)
{
case 0:
cadena[p]=inChar;
switch(cadena[p])
{
case 'M':
Serial.println("masculino"); //Para comprobar (opcional)

frecmax = 220 - edad;      // frec. max, masculino
Serial.println(edad);     //Para comprobar (opcional)
Serial.println(frecmax);  //Para comprobar (opcional)
pulreservatotal=frecmax-pnormal; // pulso de reserva
Serial.println(pulreservatotal); //Para comprobar (opcional)
lim_1= pnormal+(pulreservatotal*0.5); // configuración del límite cardiaco
Serial.println ("lim 1"); //Para comprobar (opcional)
Serial.println (lim_1); //Para comprobar (opcional)

lim_2= pnormal+(pulreservatotal*0.6); // configuración del límite cardiaco
Serial.println ("lim 2"); //Para comprobar (opcional)
Serial.println (lim_2); //Para comprobar (opcional)
lim_3= pnormal+(pulreservatotal*0.7); // configuración del límite cardiaco
Serial.println ("lim 3"); //Para comprobar (opcional)
Serial.println (lim_3); //Para comprobar (opcional)

lim_4= pnormal+(pulreservatotal*0.8); // configuración del límite cardiaco
Serial.println ("lim 4"); //Para comprobar (opcional)
Serial.println (lim_4); //Para comprobar (opcional)
lim_5= pnormal+(pulreservatotal*0.9); // configuración del límite cardiaco
Serial.println ("lim 5"); //Para comprobar (opcional)
Serial.println (lim_5); //Para comprobar (opcional)
lim_6=pnormal+(pulreservatotal*1); // configuración del límite cardiaco
Serial.println ("lim 6"); //Para comprobar (opcional)
Serial.println (lim_6); //Para comprobar (opcional)

break;
```

```

case 'L':

    Serial.println("femenino"); //Para comprobar (opcional)

    frecmax = 226 - edad;          // frec. max femenino
    Serial.println(frecmax);      // configuración del límite cardiaco
    pulreservatotal=frecmax-pnormal; // pulso de reserva
    lim_1= pnormal+(pulreservatotal*0.5); // configuración del límite cardiaco
    Serial.println ("lim 1");    //Para comprobar (opcional)
    Serial.println (lim_1);      // Para comprobar

    lim_2= pnormal+(pulreservatotal*0.6); // configuración del límite cardiaco
    Serial.println ("lim 2");    //Para comprobar (opcional)
    Serial.println (lim_2);      //Para comprobar (opcional)
    lim_3= pnormal+(pulreservatotal*0.7); // configuración del límite cardiaco
    Serial.println ("lim 3");    //Para comprobar (opcional)
    Serial.println (lim_3);      //Para comprobar (opcional)
    lim_4= pnormal+(pulreservatotal*0.8); // configuración del límite cardiaco
    Serial.println ("lim 4");    //Para comprobar (opcional)
    Serial.println (lim_4);      //Para comprobar (opcional)
    lim_5= pnormal+(pulreservatotal*0.9); // configuración del límite cardiaco
    Serial.println ("lim 5");    // //Para comprobar (opcional)
    Serial.println (lim_5);      //Para comprobar (opcional)
    lim_6=pnormal+(pulreservatotal*1); // configuración del límite cardiaco
    Serial.println ("lim 6");    //Para comprobar (opcional)
    Serial.println (lim_6);      //Para comprobar (opcional)

    break;

}

} // fin switch (p)
} // fin while serial. available ()
} //fin serial event

```

ANEXO 2: CÓDIGO DE LA INTERRUPCIÓN DEL SISTEMA DE MONITOREO DE RITMO CARDÍACO

```
volatile int almacenar [10];           // Utiliza para mantener últimos diez valores IBI
volatile unsigned long cont_muestras = 0; // Utilizado para determinar temporización de los
impulsos // Este tipo se usa comúnmente para almacenar el resultado de la función millis(), la cual
retorna el tiempo que el código actual ha estado corriendo, en milisegundos.
volatile unsigned long intervalo_latido = 0; // Utilizado para determinar el intervalo entre
latidos//
volatile int P =512;                   // Utilizado para encontrar pico en la onda de pulso
volatile int T = 512;                 // Utilizado para encontrar mínimo en la onda de pulso
volatile int tiem_latido = 512;       // Utilizado para determinar momento instantáneo de los
latidos del corazón
volatile int amplitud = 100;          // Se utiliza para mantener la amplitud de la forma de onda de
pulso
volatile boolean primer_latido = true; // Utilizado para sembrar variedad tasa por lo que inicio
con BPM razonable valor lógico verdadero o falso
volatile boolean segundo_latido = true; //Utilizado para sembrar variedad tasa por lo que inicio
con razonable BPM

void interruptSetup(){
// Inicializa Timer2 para lanzar una interrupción cada 2mS.
TCCR2A = 0x02; // PWM DESACTIVAR EN DIGITAL pines 3 y 11, y entra en MODO CTC ,
establezco registro , además timer2 trabajo a 8Khz // control de registro A
TCCR2B = 0x05; // No fuerce COMPARAR, 256 prescaler // precontrolador se establece en
128 / control de registro b
OCR2A = 0x7C; // AJUSTAR EL TOP DEL RECUENTO A 124 DE 500 Hz FRECUENCIA DE
MUESTREO , Frec está en 124
TIMSK2 = 0x02; // HABILITACIÓN DE INTERRUPCIÓN EN PARTIDO ENTRE TIMER2 Y
OCR2A , habilita un timer de desbordamiento de interrupción establecido a 8 khz,
sei(); // HABILITA INTERRUPCION GLOBAL
}

// ESTE ES EL TIEMPO DE SERVICIO 2 Rutina de interrupción.
// Temporizador 2 se asegura de que tomamos una lectura cada 2 milisegundos
ISR(TIMER2_COMPA_vect){ // desencadenó cuando Timer2 cuenta con 124
cli(); //Interrupciones desactivar mientras hacemos esto
Signal = analogRead(pulsoPin); // Leer el sensor de pulso
cont_muestras += 2; // No perder de vista el tiempo en ms con esta variable
int Ruido = cont_muestras - intervalo_latido; // Controlar el tiempo transcurrido desde el último
latido para evitar el ruido

//Encontrar el máximo y mínimo de la onda de pulso
if(Signal < tiem_latido && Ruido > (IBI/5)*3){ // evitar el ruido dicrótico esperando 3.5 de IBI
última
if (Signal < T){ // T es valle, punto más bajo
T = Signal; // No perder de vista el punto más bajo de la onda de pulso
}
}

if(Signal > tiem_latido && Signal > P){ // tiempo de latido, condición ayuda a evitar el ruido
P = Signal; // P es el pico
} // No perder de vista el punto más alto de la onda de pulso
```

```

// PASOS PARA BUSCAR EL RITMO DEL CORAZÓN
// Señal resurge de valor cada vez que se produce un pulso
if (Ruido > 250){ // evitar el ruido de alta frecuencia Noise 250
  if ( (Signal > tiem_latido) && (Pulse == false) && (Ruido > (IBI/5)*3) )
  {
    Pulse = true; // Establecer el indicador de pulso cuando pensamos
que hay un pulso
    digitalWrite(intermitentePin,HIGH); // Enciende en el pin 13 LED (opcional
puedo quitar si no utilizo led)
    IBI = cont_muestras - intervalo_latido; // Medir el tiempo entre latidos en mS
    intervalo_latido = cont_muestras; // Un seguimiento del tiempo para el
próximo pulso

    if(primer_latido){ // si es la primera vez que encontramos el ritmo,
si primer latido == TRUE
      primer_latido = false; // Bandera primer latido
      return; // IBI no es fiable por lo que descartarlo, la primera
vez
    }
    if(segundo_latido){ // si este es el segundo tiempo, si segundo
latido == TRUE
      segundo_latido = false; // Bandera secondBeat clara
      for(int i=0; i<=9; i++){ // total acumulado para obtener un BPM real
en el arranque
        almacenar[i] = IBI;
      }
    }
  }

// Mantener un total acumulado de los últimos 10 valores IBI
word runningTotal = 0; //variable RunningTotal inicializada en 0//número sin signo entre,
0 y 65535.

for(int i=0; i<=8; i++){ // desplazamiento de datos de la matriz de tasa
  almacenar[i] = almacenar[i+1]; // Y colocar el valor más antiguo IBI
  runningTotal += almacenar[i]; // Sumar los 9 valores más antiguos IBI
}

almacenar[9] = IBI; // Añadir la última IBI a la matriz de tasa
runningTotal += almacenar[9]; // Añadir la última IBI a RunningTota
runningTotal /= 10; // Promedio de los últimos 10 valores IBI
BPM = 60000/runningTotal; // Cuántos golpes puede haber en un minuto? eso es BPM!
OP = true; // Establecer la bandera automaticamente OP
// OP FLAG no se borra EN ESTA ISR
} // TERMINA EL 1 IF
} // termina configuración de ruido

```

```

if (Signal < tiem_latido && Pulse == true){ // cuando los valores están bajando, el ritmo es más
digitalWrite(intermitentePin,LOW); // Desactivar pin 13 LED
Pulse = false; // Restablecer el indicador de impulso para que podamos hacerlo de
nuevo
    amplitud = P - T; // Obtener amplitud de la onda del pulso
    tiem_latido = amplitud/2 + T; // Establecer umbral al 50% de la amplitud
    P = tiem_latido; // Restablecer estos para la próxima vez
    T =tiem_latido;
}

if (Ruido > 2500){ // si 2,5 segundos pasan sin un latido 2500
    tiem_latido = 512; // Conjunto predeterminado umbral
    P = 512; // Conjunto P defecto
    T = 512; // Conjunto T defecto
    intervalo_latido = cont_muestras; // Llevar el lastBeatTime al día
    primer_latido = true; // Establecer estos para evitar el ruido
    segundo_latido = true; // Cuando conseguimos el latido del corazón de vuelta
}

sei(); // Habilitar las interrupciones cuando haya terminado
} // fin del vector isr

```

ANEXO 3: HOJA DE DATOS DE ARDUINO LILYPAD

ARDUINO LILYPAD ATEMGA 128

Información General

El LilyPad Arduino es una placa electrónica diseñada para wearables y e-textiles. Se puede coser a la tela y se monta de manera similar a fuentes de alimentación, sensores y actuadores con hilo conductor. El consejo se basa en la ATmega168V (la versión de bajo consumo del ATmega168) (ficha técnica) o la ATmega328V (ficha técnica). El LilyPad Arduino fue diseñado y desarrollado por Leah Buechley y Sparkfun Electronics.

Resumen

Advertencia: No alimentar el LilyPad Arduino con más de 5,5 voltios, o enchufe el poder al revés: te matas.

Microcontroladores	ATmega168V o ATmega328V
Tensión De Funcionamiento	2.7 a 5.5 V
Voltaje De Entrada	2.7 a 5.5 V
Digital pines I / O	14 (de las cuales 6 proporcionan salida PWM)
Botones de entrada analógica	6
Corriente DC por E / S Pin	40 mA
Memoria Flash	16 KB (de los cuales 2 KB utilizado por el gestor de arranque)
SRAM	1 KB
EEPROM	512 bytes
Velocidad De Reloj	8 MHz

Programación

El LilyPad Arduino se puede programar con el software de Arduino. * Nota *, el LilyPad Arduino sólo debe ser programado con las versiones de software 0010 o superior. Usted puede programar con versiones anteriores, pero todas las funciones de tiempo relacionadas estarán apagadas (el doble de lento como debe ser).

El ATmega168V o ATmega328V en la Arduino LilyPad viene precargado con un gestor de arranque que le permite cargar nuevo código para con el software de Arduino. También puede pasar por alto el gestor de arranque y el programa de la ATmega través del ICSP (In-Circuit Serial Programming) cabeza; consulte estas instrucciones para más detalles.

Poder

El LilyPad Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa.

Si se utiliza una fuente de alimentación externa, se debe proporcionar entre 2,7 y 5,5 voltios. Esto puede venir con un adaptador de CA a CC (pared-verruga) o la batería. Una vez más, no alimentar el LilyPad Arduino con más de 5,5 voltios, o enchufe la fuente de al revés: se le matas.

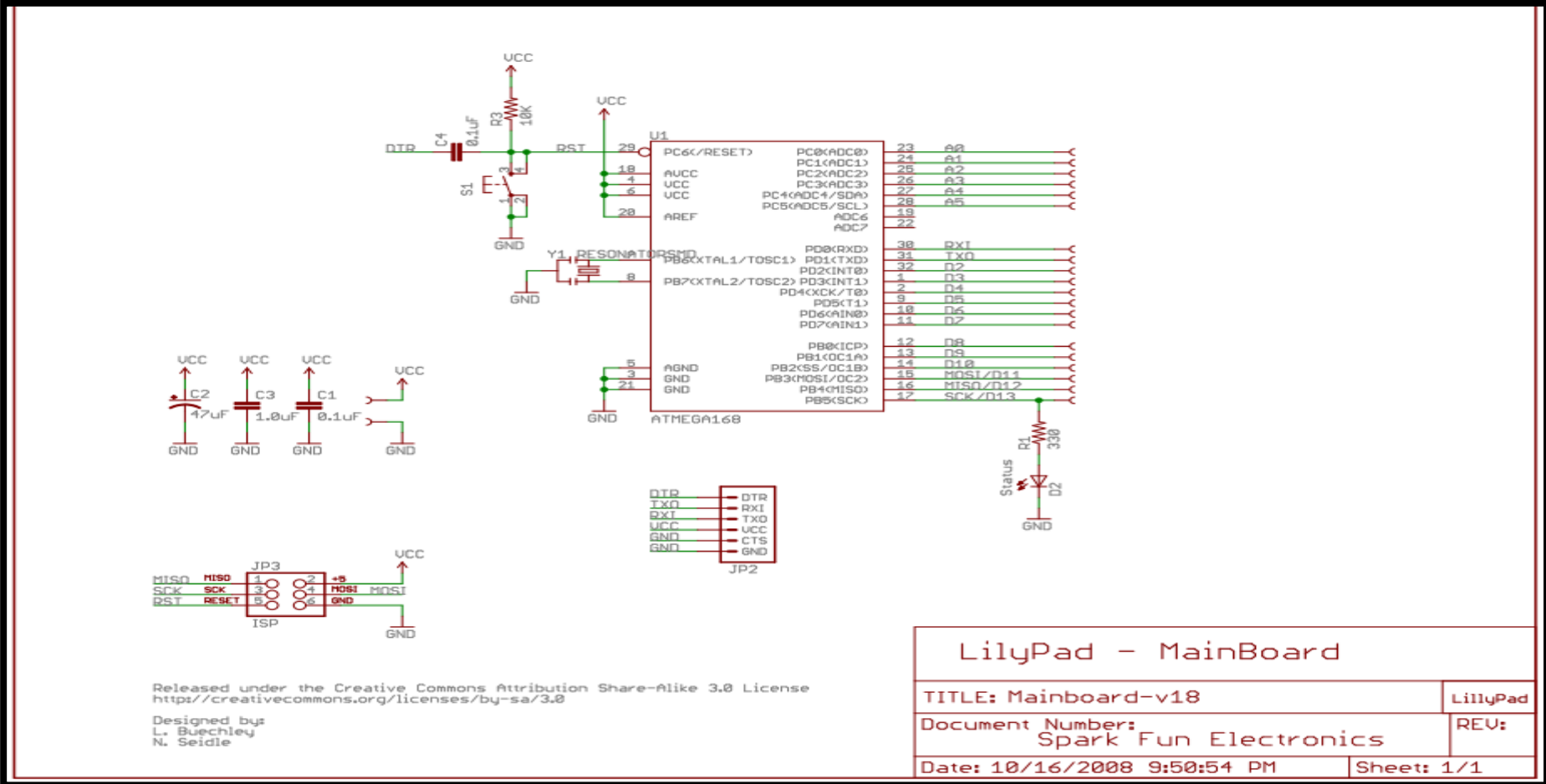
Características Físicas

El LilyPad Arduino es un círculo, de aproximadamente 50 mm (2 ") de diámetro. El propio Consejo es .8mm (1/32") de espesor (aproximadamente 3 mm (1/8 ")), donde se unen la electrónica).

Lavabilidad

Lavado a su propio riesgo que hacemos. Recomendamos proyectos lavado a mano con un detergente

Lilypad Mainboard



Released under the Creative Commons Attribution Share-Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0>

Designed by:
 L. Buechley
 N. Seidle

LilyPad - MainBoard	
TITLE: Mainboard-v18	LillyPad
Document Number: Spark Fun Electronics	REV:
Date: 10/16/2008 9:50:54 PM	Sheet: 1/1

ANEXO 4: HOJA DE DATOS DE PULSE SENSOR

PULSE SENSOR



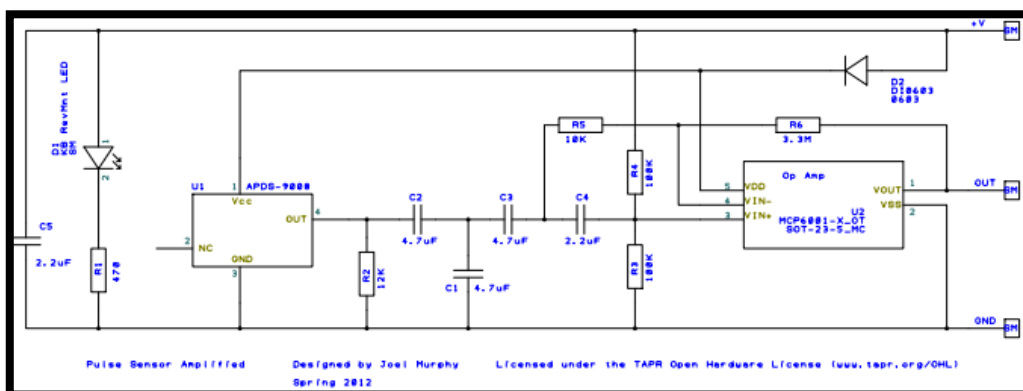
Descripción: datos de frecuencia cardíaca puede ser muy útil si usted está diseñando una rutina de ejercicios, estudiar sus niveles de actividad o de ansiedad o simplemente quiere que su camisa a parpadear con el latido de su corazón. El problema es que la frecuencia cardíaca puede ser difícil de medir. Por suerte, el Sensor de Pulso Amped puede resolver ese problema!

El pulso Amped sensor es un sensor de frecuencia cardíaca plug-and-play para Arduino. Puede ser utilizado por estudiantes, artistas, deportistas, fabricantes y desarrolladores de juegos y móviles que quieren incorporar fácilmente los datos de frecuencia cardíaca en vivo en su projects. It esencialmente combina un sensor de frecuencia cardíaca óptica simple con amplificación y circuito de cancelación de ruido por lo que es rápido y fácil de obtener lecturas de pulso fiables. Además, sorbe el poder con sólo 4 mA de corriente a 5 V por lo que es ideal para aplicaciones móviles.

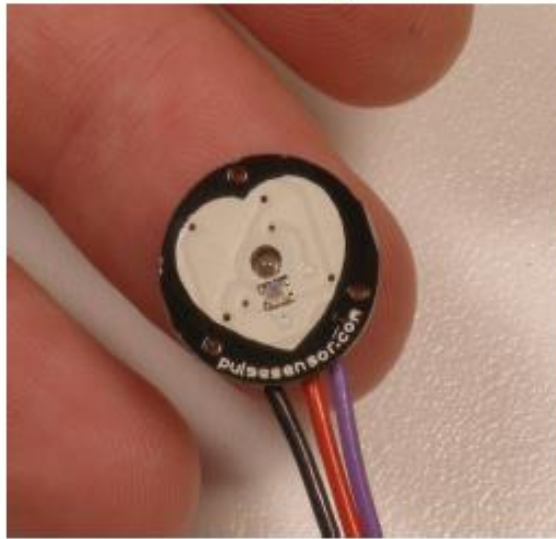
Simplemente sujete el sensor de pulso para el lóbulo de la oreja o un dedo de la punta y conéctelo a su 3 o 5 voltios Arduino y ya está listo para leer la frecuencia cardíaca! El "cable 24 en el sensor de pulso está terminado con conectores macho estándar así que no hay soldadura requerida. Por supuesto código de ejemplo Arduino está disponible, así como un esbozo de Procesamiento para la visualización de datos de frecuencia cardíaca.

Dimensiones: 0.625 "de diámetro y 0.125" de espesor

Esquemático



Pulse Sensor Getting Started Guide



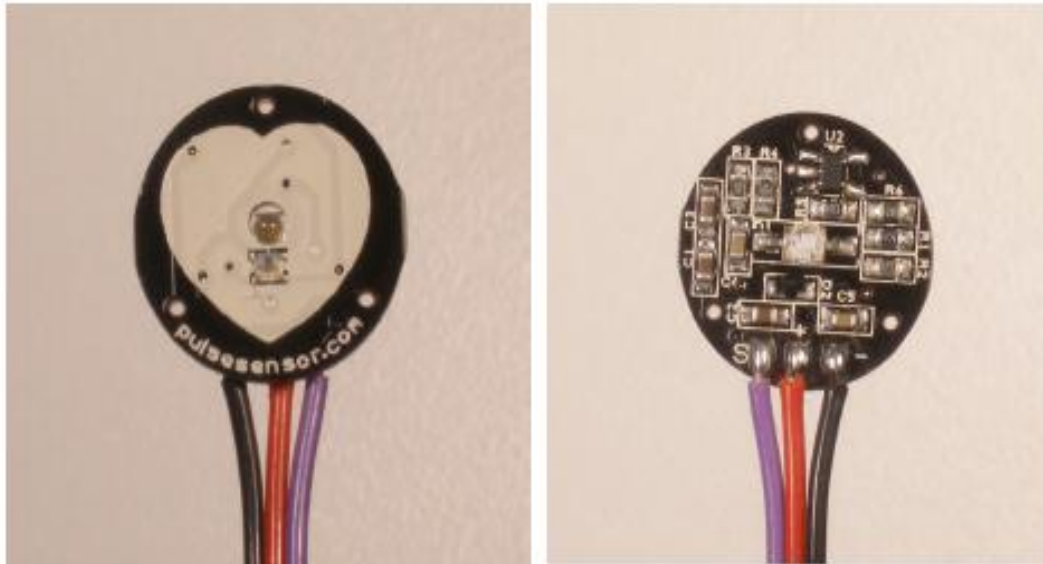
Introduction:

Pulse Sensor is a well-designed plug-and-play heart-rate sensor for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects. The sensor clips onto a fingertip or earlobe and plugs right into Arduino with some jumper cables. It also includes an open-source monitoring app that graphs your pulse in real time.

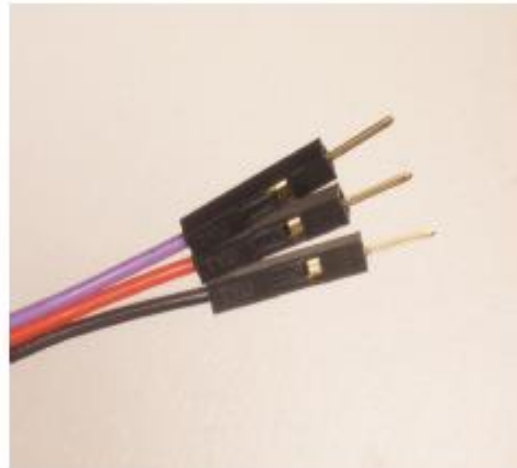


The Pulse Sensor Kit includes:

- 1) A 24-inch Color-Coded Cable, with (male) header connectors. You'll find this makes it easy to embed the sensor into your project, and connect to an Arduino. No soldering is required.
- 2) An Ear Clip, perfectly sized to the sensor. We searched many places to find just the right clip. It can be hot-glued to the back of the sensor and easily worn on the earlobe.
- 3) 2 Velcro Dots. These are 'hook' side and are also perfectly sized to the sensor. You'll find these velcro dots very useful if you want to make a velcro (or fabric) strap to wrap around a finger tip.
- 4) Velcro strap to wrap the Pulse Sensor around your finger.
- 4) 3 Transparent Stickers. These are used on the front of the Pulse Sensor to protect it from oily fingers and sweaty earlobes.
- 5) The Pulse Sensor has 3 holes around the outside edge which make it easy to sew it into almost anything.



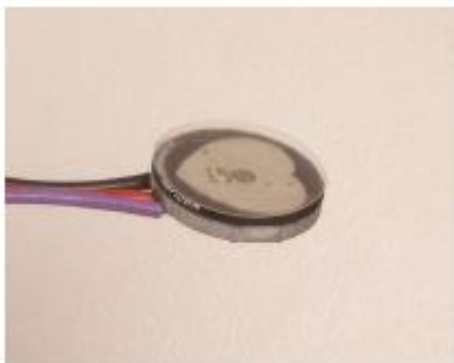
RED wire = +3V to +5V
BLACK wire = GND
PURPLE wire = Signal



The Pulse Sensor can be connected to arduino, or plugged into a breadboard. Before we get it up and running, we need to protect the exposed circuitry so you can get a reliable heart beat signal.

Preparing the Pulse Sensor

Before you really start using the sensor you want to insulate the board from your (naturally) sweaty/oily fingers. The Pulse Sensor is an exposed circuit board, and if you touch the solder points, you could short the board, or introduce unwanted signal noise. We will use a thin film of vinyl to seal the sensor side. Find the small page of four clear round stickers in your kit, and peel one off. Then center it on the Pulse Sensor. It should fit perfectly.



When you are happy with the way it's lined up, squeeze it onto the face all at once! The sticker (made of vinyl) will kind of stretch over the sensor and give it a nice close fit. If you get a wrinkle, don't worry, just press it down really hard and it should stick. We gave you 4, so you can replace it if necessary.

That takes care of the front side. The vinyl sticker offers very good protection for the underlying circuit, and we rate it 'water resistant'. meaning: it can stand to get splashed on, but don't throw it in the pool!

If this is your first time working with Pulse Sensor, you're probably eager to get started, and not sure if you want to use the ear-clip or finger-strap (or other thing). The back of the Pulse Sensor has even more exposed contacts than the front, so you need to make sure that you don't let it touch anything conductive or wet.

The easiest and quickest way to protect the back side from undesirable shorts or noise is to simply stick a velcro dot there for now. The dot will keep your parts away from the Pulse Sensor parts enough for you to get a good feel for the sensor and decide how you want to mount it. You'll find that the velcro dot comes off easily, and stores back on the little strip of plastic next to the other one.

Notice that the electrical connections are still exposed! We only recommend this as a temporary setup so you can get started. We show you how to better seal the Pulse Sensor later in this document.



Running The Pulse Sensor Code

Get the latest Arduino and Processing Pulse Sensor software here <http://pulsesensor.com/downloads/>

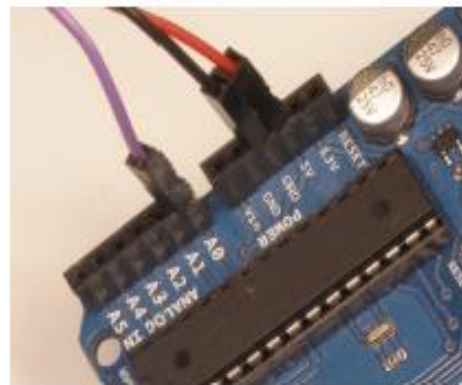
Arduino code is called "PulseSensorAmped_Arduino-xx"

The Processing code is called "PulseSensorAmped_Processing-xx"

We strongly advise that you DO NOT connect the Pulse Sensor to your body while your computer or arduino is being powered from the mains AC line. That goes for charging laptops and DC power supplies. Please be safe and isolate yourself from from the power grid, or work under battery power.

Connect the Pulse Sensor to: +V (red), Ground (black), and Analog Pin 0 (purple) on your favorite Arduino, or Arduino compatible device, and upload the 'PulseSensorAmped_Arduino-xx' sketch.

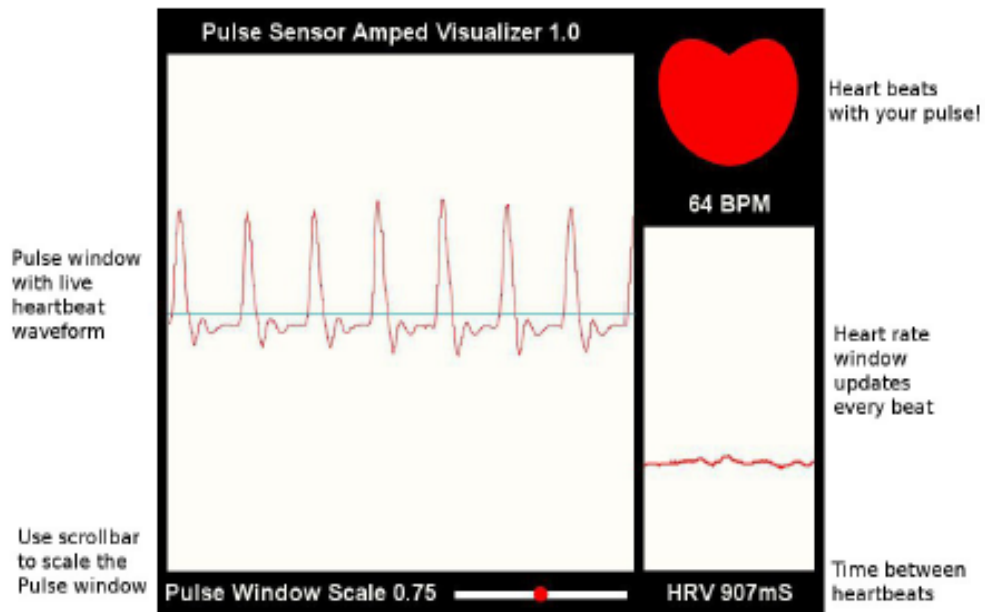
note: If you want to power Pulse Sensor Amped with low voltage (3.3V for example), make sure you have this line of code in the setup()
`analogReference(EXTERNAL);`
Also, make sure that you apply the lower voltage to the Arduino Aref pin (next to pin 13).



After it's done uploading, you should see Arduino pin 13 blink in time with your heartbeat when you hold the sensor on your fingertip. If you grip the sensor too hard, you will squeeze all the blood out of your fingertip and there will be no signal! If you hold it too lightly, you will invite noise from movement and ambient light. Sweet

Spot pressure on the Pulse Sensor will give a nice clean signal. You may need to play around and try different parts of your body and pressures. If you see an intermittent blink, or no blink, you might be a zombie or a robot.

To view the heartbeat waveform and check your heart rate, you can use the Processing sketch that we made. Start up Processing on your computer and run the 'PulseSensorAmped_Processing-xx' sketch. This is our data visualization software, and it looks like this.



note: If you get an error when starting this code, you may need to make sure you are selecting the right serial port. Check the Troubleshooting section below..

The large main window shows a graph of raw sensor data over time. The Pulse Sensor Data Window can be scaled using the scrollbar at the bottom if you have a very large or very small signal. At the right of the screen, a smaller data window graphs heart rate over time. This graph advances every pulse, and the Beats Per Minute is updated every pulse as a running average of the last ten pulses. The big red heart in the upper right also pulses to the time of your heartbeat. When you hold the Pulse Sensor to your fingertip or earlobe or (fill in body part here) you should see a nice heartbeat waveform like the one above. If you don't, and you're sure you're not a zombie, try the sensor on different parts of your body that have capillary tissue. We've had good results on the side of the nose, middle of the forehead, palm, and lower lip. We're all different, original organisms. Play around and find the best spot on you and your friends. As you are testing and getting used to the sensor. You may find that some fingers or parts of fingers are better than others. For example, I find that when I position the sensor so that the edge of the PCB is at the bottom edge of my earlobe I get an awesome signal. Also, people with cold hands or poor circulation may have a harder time reading the pulse. Run your hands under warm water, or do some jumping-jacks!

Arduino and Processing programming environments available for download here:
www.arduino.cc www.processing.org

Sealing the Back Side of Pulse Sensor

Basic protection for the back of the Pulse Sensor
and prep for attaching Velcro Dot.

It's really important to protect the exposed Pulse Sensor circuitry so the sweat of your fingertips or earlobe (or wherever) doesn't cause signal noise or a short circuit. This How-To uses hot glue, which can be removed easily or re-worked if you want to change where you've stuck your Pulse Sensor. We love hot glue!

The other things you'll need are:

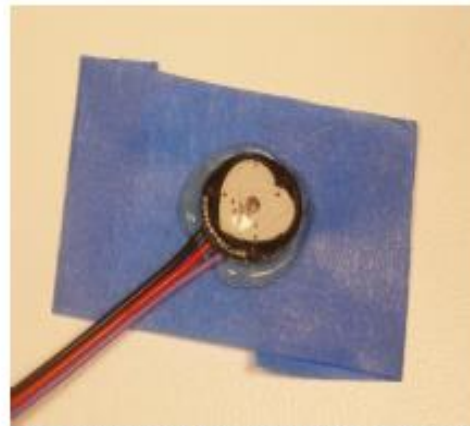
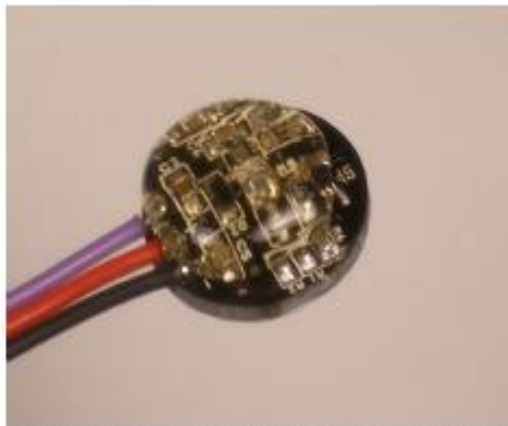
Hot Glue Gun

Blue Tape (any tape should do ok)

Nail Trimmers (or your favorite trimming device. Flush-cut wire snips work well too)

Read these instructions all the way through before you start!

First, attach the clear vinyl sticker to the front of your Pulse Sensor, as shown above. Then put a blob of hot glue on the back, right over the circuit. Size can be difficult to judge sometimes. What I meant was put a hot glue blob about the size of a kidney bean on the back side of the Pulse Sensor.

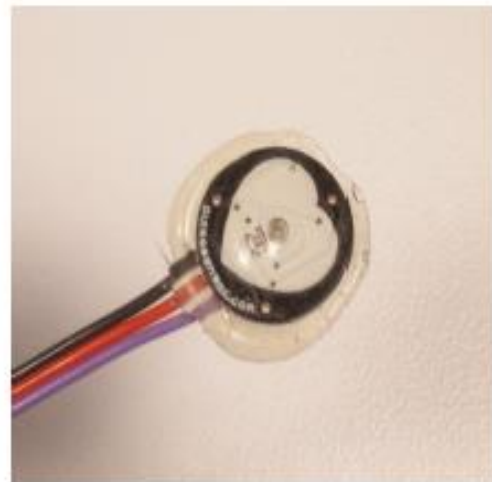
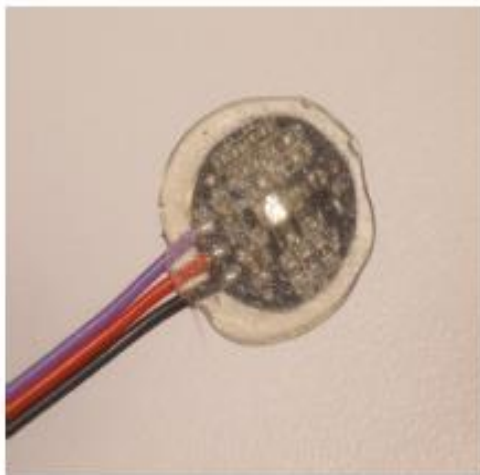
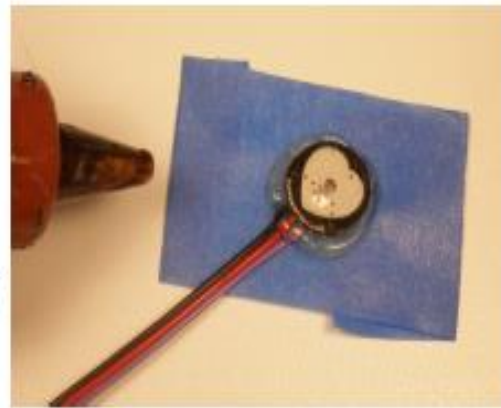


Then, while the glue is still very hot, press the Pulse Sensor glue-side-down onto the sticky side of a piece of blue tape (I believe that blue tape has magical properties, but if you don't have blue tape other kinds of tape will work just as well).

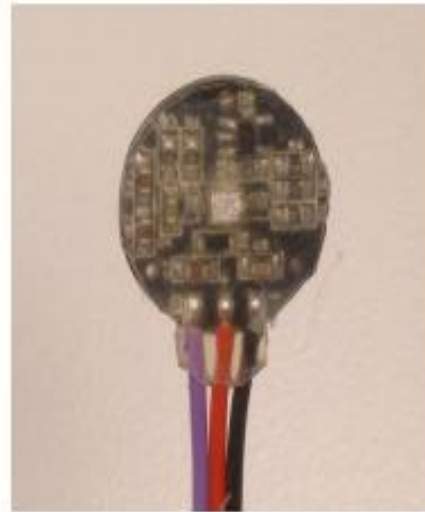


note: The tallest thing on the back of the Pulse Sensor is the green LED housing right in the middle. Use it to make the hot-glue seal thin and strong. When you press evenly until the back of the LED touches, all the conductive parts will be covered with hot glue. If the glue doesn't ooze out all around, let it cool down, then peel it from the Pulse Sensor and try again. Once the glue has cooled down and has some body, you can peel it off easily. Here's some pics of hot glue 'impressions' that I took during the making of this guide.

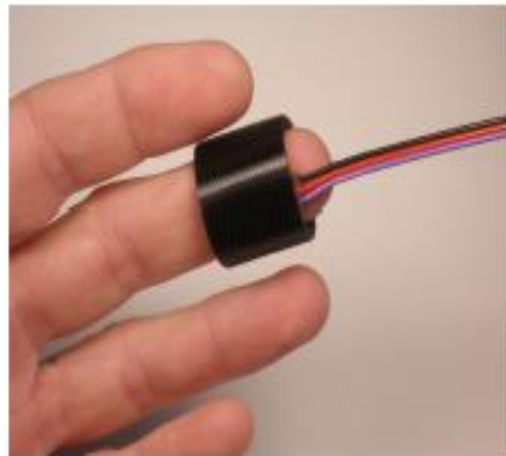
Next put a small dab of hot glue on the front of the cables, where they attach to the Pulse Sensor circuit board. This will bond to the other glue that's there and act as a strain-relief for the cable connection. This is important because the cable connection can wear out over time.



Once the hot glue has cooled (wait for it!) the blue tape will peel off very easily. Check your work to make sure that there are not exposed electrical connections! Next is trimming. I find the easiest way is to use nail clippers. Flush cutting wire snips work too. Take care not to clip the wires!!! And leave enough around the cable to act as a good strain-relief



This is the basic Pulse Sensor Hot Glue Seal. It's also got the clear vinyl sticker on the front face. We're calling this 'Water Resistant', ready to be handled and passed around from fingers to earlobes or whatever. It is not advised to submerge or soak the Pulse Sensor with this basic seal.
Now you can stick on the velcro dot (included) and make a finger strap with the velcro tape (included)!



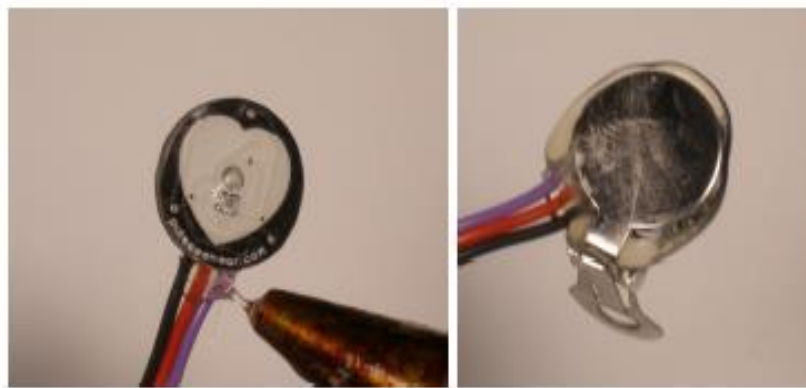
Attaching the Ear Clip

We looked all over, and were lucky enough to find an ear clip that fits the Pulse Sensor perfectly. The earlobe is a great place to attach Pulse Sensor. Here's some instruction on how to do it.

It is important to apply some strain relief to the cable connection where it meets the Pulse Sensor PCB. The little wire connections can wear out and break (or short on something) over time. We can do this with hot glue, like we did in the previous example.

First, attach a clear vinyl sticker to the front of the Pulse Sensor if you have not already. Then, put a small dab of hot glue on the front of the cables right where they meet the PCB. Get some on the edge of the PCB too, that will help. Remember, if you don't like the blob you've made for any reason, it's easy to remove once it cools down.

Next place the Pulse Sensor face down, and put a dab of glue about the size of a kidney bean on the back as illustrated above. Center the round part of the ear clip on the sensor and press it into the hot glue. The tallest component on the back is the plastic body of the reverse mount LED, and if you press it evenly it will help keep the metal of the ear clip from contacting any of the component connections.



Allow the hot glue to ooze out around the ear clip. That will ensure good coverage. Take care not to let the hot glue cover around the ear clip hinge, as that could get in the way of it working. Trimming is easy with nail clippers (as above) or your trimming tool of choice. Don't trim the wires by mistake!!!



Take a good look at your work with a magnifying glass to be sure that you haven't made contact with any of the solder joints, then plug it in and test it. Hot glue is also great because it is easy to remove or re-work if you need to.

Troubleshooting:

Processing Sketch gives me a COM port error at startup.

Make sure you are plugged into an Arduino board, that it is working correctly, and running our firmware.

Check to see if you have the right serial port. The code underlined in red should match the correct port number in the terminal window at the bottom of Processing IDE.

```
// GO FIND THE ARDUINO
println(Serial.list()); // print a list of available serial ports
// choose the number between the [] that is connected to the Arduino
port = new Serial(this, Serial.list()[0], 115200); // make sure Arduino is ta
port.clear(); // flush buffer
port.bufferUntil('\n'); // set buffer full flag on receipt of carriage return
}

Public Library
Native lib Version = SXDE-2.1-7
Java lib Version = SXDE-2.1-7
[0] "COM30"
```

Processing gives an RXTX mismatch warning, then nothing happens

The RXTX mismatch problem can be resolved by making sure you are running the latest version of Processing and Arduino.

If you're still having trouble, go to <http://processing.org/reference/libraries/serial/index.html> for more info.

DATASHEET BLUETOOTH TO SERIAL PORT MODULE HC05



Overview

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the

footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

Hardware features

- Typical -80dBm sensitivity.
- Up to +4dBm RF transmit power.
- Low Power 1.8V Operation, 3.3 to 5 V I/O.
- PIO control.
- UART interface with programmable baud rate.
- With integrated antenna.
- With edge connector.

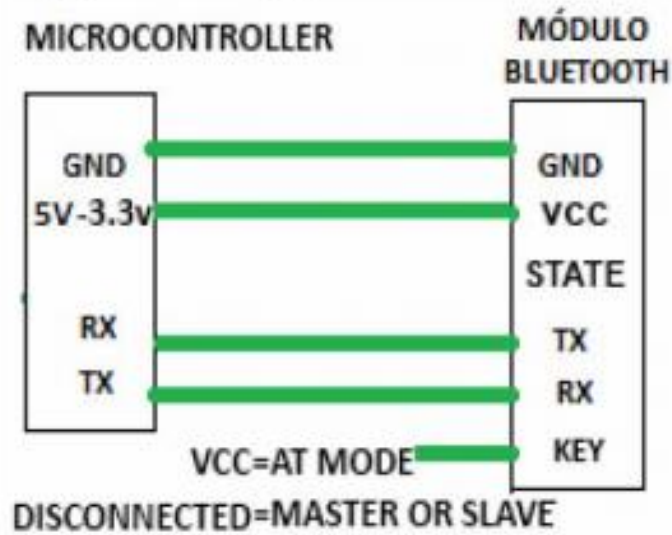
Software features

- Slave default Baud rate: 9600, Data bits:8, Stop bit:1,Parity:No parity.
- PIO9 and PIO8 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing **PINCODE:"1234"** as default.
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Pin out configuration



Typical Application Circuit



After connect the Bluetooth module, scan for new devices from the PC and you will find the module with the device name "HC-05", after that, click to connect, if some message appears asking about "Pairing code" just put "1234" as default code.

BLUE LED = ACTIVE (Blinking 500ms period inactive connection, change 1seg with active connection)

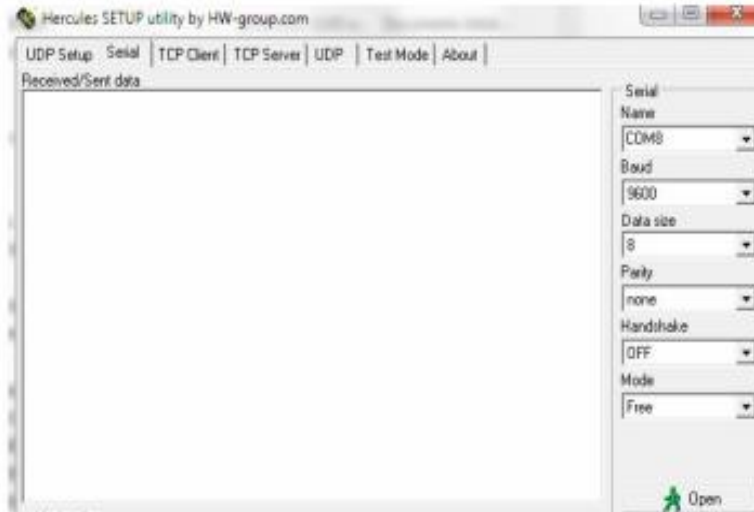


Open a serial terminal and select the serial COM x port number that assigned Windows to Bluetooth Module.

Configure the serial terminal with these parameters:

- Baud rate: 9600.
- Data bits:8.
- Stop bit:1.
- Parity: No parity.

Open connection and you will be ready to send and receive data from module Bluetooth like Serial Port COM



AT COMMANDS

How to get to AT COMMAND mode

- 1: Connect KEY pin to VCC.
- 2: Supply power to module. Then the module will enter into AT MODE. In this mode you have to use baud rate at 38400. In this way, user should change the baud rate for SLAVE AND MASTER mode.

How to set this module as "Master - Host" role

- 1: Input high level to KEY.
 - 2: Supply power to the module. And the module will enter to AT COMMAND.
 - 3: Set the parameters of the hyper terminal or the other serial tools (baud rate: 38400, data bit:8, stop bit:1, no parity bit, no Flow Control).
 - 4: Sent the characters "AT+ROLE=1\r\n" through serial, then receive the characters "OK\r\n". Here, "\r\n" is the CRLF.
 - 5: Sent the characters "AT+CMODE=1\r\n" through serial, then receive the characters "OK\r\n". Here, "\r\n" is the CRLF.
 - 6: Default factory password passkey is: 1243, this must be the same in the Bluetooth slave module if you want to pair it.
- To read passkey use this command: "AT+PSWD?".
- To Reset the password command sent the characters "AT+PSWD=XXXX".
- The password must be 4-bits.

7: Leave free KEY, and supply power to the module again. Then this module will become master role and search the other module (slave role) automatically to build the connection (baud rate:9600, data bit:8, stop bit:1, no parity bit, no Flow Control).

How to set this module be the “Slave - Device” role

1: Input high level to KEY.

2: Supply power to the module. And the module will enter to AT COMMAND.

3: Set the parameters of the super terminal or the other serial tools (baud rate:

38400, data bit:8, stop bit:1, no parity bit, no Flow Control).

4: Sent the characters “AT+ROLE=0\r\n” through serial, then receive the characters “OK\r\n”. Here, “\r\n” is the CRLF.

5: Sent the characters “AT+CMODE=0\r\n” through serial, then receive the characters “OK\r\n”. Here, “\r\n” is the CRLF.

6: Default factory password passkey is: 1243, this must be the same in the Bluetooth master module if you want to pair it.

To read passkey sent the characters “AT+PSWD?”.

To Reset the password command sent the characters “AT+PSWD=XXXX”.

The password must be 4-bits.

7: Leave free KEY, and supply power to the module again. Then this module will become slave role and wait to be discover it by the other module (master role) automatically to build the connection (baud rate:38400, data bit:8, stop bit:1, no parity bit, no Flow Control).

How to get to the standard communication mode

1: Leave free KEY, don't connect it to VDD neither GND.

2: Supply power to the module. Then the module will enter to communication mode. It can be used for pairing.

Notes

(1) HC-05's command should end up with “\r\n”. It means when you finish programming, you should add terminator (“ENTER” or “0x0d 0x0a”) to the program.

(2) The most common commands for HC-05 are: AT+ROLE (set master–slave), AT+CMODE(set address pairing) , AT+PSWD (set password).

If you want the master module has the function of remembering slave module,the most simply way is: First, set AT+CMODE=1. Make the master module pair with the slave module. Second, set AT+CMODE=0. Then the master module just can make pair with that specified slave module.

www.electronica60norte.com

electronica60norte@hotmail.com

ANEXO 6: MANUAL DE USUARIO PARA EL USO CORRECTO DEL SISTEMA DE MONITOREO CARDÍACO

MANUAL DE USUARIO

Antes de usar el sistema de monitoreo de ritmo cardíaco, leer las instrucciones.

On/Off del sistema

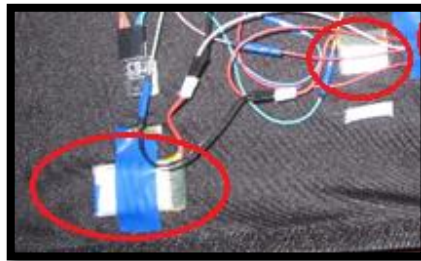
Para el encendido y apagado del sistema, bastará con mover la palanca del pequeño interruptor que se encuentra en la muñequera. Si la palanca del interruptor se encuentra junto al punto azul, significará que el sistema está encendido; caso contrario apagado.



Baterías

El sistema debe contar siempre de dos baterías Lipo de 3,7 voltios con corriente de 680mA. Una batería es para la placa Arduino y el Sensor Cardíaco y la otra es para el módulo Bluetooth.

Nota: El sistema soportará baterías tipo Lipo con un rango de voltaje entre 3,7 a 5 voltios y una corriente de 680mA a 1A. Si se supera estos límites, el sistema podrá presentar daños electrónicos en los elementos.



Funciones

El presente sistema cumple con las siguientes funciones:

- Conexión inalámbrica automática al abrir la aplicación del sistema.
- Ingreso de datos personales: Edad, Pulso normal, Género.
- Muestra los pulsos cardíacos de cada persona.
- Muestra la F.C.M. de cada persona.
- Muestra tiempos de calentamiento por zonas, previo al entrenamiento.

Elección de 5 zonas de entrenamiento.

- Muestra beneficios y características fisiológicas en cada zona de entrenamiento.
- Muestra tiempos recomendados para realizar el entrenamiento en cada zona.
- Muestra métodos de entrenamiento para realizar ejercicio físico.
- Define límites superior e inferior en cada zona de entrenamiento.
- Emisión de alarmas cuando se incumple los rangos de pulsaciones en cada zona.
- Botón de detener entrenamiento.

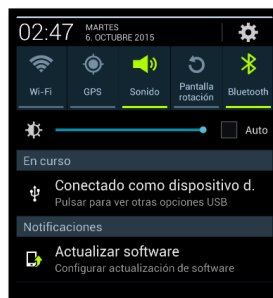
Compatibilidad con Smartphone

El sistema es compatible con cualquier dispositivo móvil inteligente con sistema Operativo Android que tenga las siguientes características.

- Sistema Operativo Android versión 4.1.2 o superior.
- Memoria interna de 512 MB o superior.
- Memoria RAM de 250MB o superior.

Conexión Bluetooth

1. Se debe de encender el Bluetooth del dispositivo móvil, para el uso del sistema.



2. La primera vez que se use el sistema, se deberá realizar un emparejamiento entre el sistema y el dispositivo móvil. Desde este último mencionado, en las configuraciones Bluetooth, seleccionar el nombre del equipo Bluetooth al que se conectará llamado “PULSOMETRO”. Posteriormente ingresar la contraseña “1234”

Nota: Esto solo se realizará la primera vez que se utilice, los siguientes usos se omitirá el paso #2 de la conexión Bluetooth.

En usos posteriores, cumpliendo inicialmente el paso 1 y después ingresando a la aplicación móvil, el dispositivo móvil se conectará automáticamente al sistema cardíaco.

Recomendaciones para el uso de la muñequera

- Ubicarla siempre en el lugar indicado, entre la mano y el brazo.
- Ajustar bien los velcros antes de realizar el ejercicio físico
- No abrir el segundo doblaje, ya que dentro de este se encuentran los componentes electrónicos. (Solo el administrador del sistema podrá hacerlo)

Nota: Si se requiere lavar la muñequera, hacerlo con agua tibia, sin detergente y muy despacio; retirando todos los elementos a excepción de la placa Arduino Lilypad y el cosido del hilo conductor (estos no presentan problema si se mojan, los demás sí).

Recomendaciones para el uso del sensor

- Cubrir el sensor junto con el velcro en cualquier dedo de la mano (preferencia dedo índice).
- No ajustar muy fuerte el velcro al dedo, hacer un ajuste normal.
- No mojar el sensor cardíaco.
- No jalar demasiado la manga del sensor cardíaco. (Solo lo necesario para que llegue el sensor al dedo elegido)

Uso de la Aplicación Móvil

La aplicación es muy intuitiva, lo que facilita su uso, a continuación se muestra lo que el usuario deberá hacer.

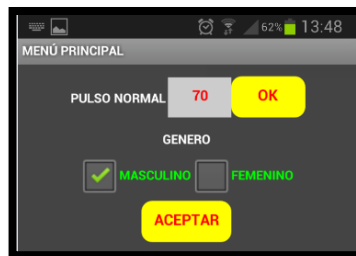
1. Ingresar su edad y clic en el botón “ACEPTAR”.



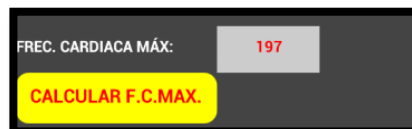
- Ingresar su pulso normal y clic en el botón “OK” (Para el pulso normal se deberá ingresar el pulso cardíaco que se muestra en el cuadro de texto Frecuencia Cardíaca, previo a realizar el ejercicio físico, es decir en reposo.)



- Seleccionar el Género y clic en el botón “ACEPTAR”.



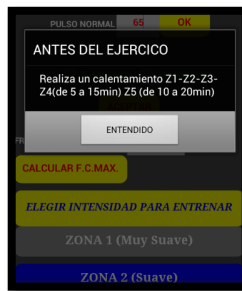
- Hacer clic en el botón “CALCULAR F.C.Máx”, para ver la frecuencia cardíaca máxima.



- Hacer clic en el botón “Elegir Intensidad para Entrenar”



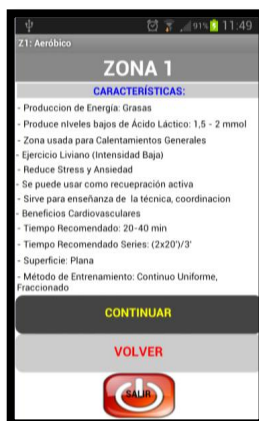
6. Hacer clic en el botón “ENTENDIDO” en la pantalla donde se presentan los tiempos de calentamiento.



7. Elegir la zona deseada de entrenamiento.



8. Leer detenidamente las características del ejercicio en la zona elegida, antes de iniciar el entrenamiento



9. Para iniciar, hacer clic en el botón “INICIAR ENTRENAMIENTO”.



10. La aplicación le emitirá automáticamente dos alarmas, vibración cuando se esté por debajo de límite inferior de pulsaciones y una alarma sonora cuando se esté por encima del límite superior, en cada zona.

Recomendaciones de uso en General

Seguir las instrucciones mencionadas anteriormente, para un correcto funcionamiento del sistema

Nota: Si en la aplicación móvil no se observa el pulso cardíaco, es posible que cualquiera de las dos baterías este agotada. Se recomendará recargarla para un nuevo uso.