



**INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ**

---

---

**INGENIERÍA ELECTRÓNICA**

**RESIDENCIA PROFESIONAL**

**PRESENTA:**

**Roger Ulises Pérez Ruiz**

**TEMA:**

Sistema de monitoreo y regulación soportado en android de la temperatura de una incubadora de aves

**REPORTE FINAL DE RESIDENCIA**

<b>Asesor interno</b>	<b>M en C. Aldo Esteban Aguilar Castillejos</b>
<b>Asesor externo</b>	<b>Dr. Rubén Herrera Galicia</b>
<b>Revisor</b>	<b>M en C. Raúl Moreno Rincón</b>
<b>Revisor</b>	<b>M en C. Álvaro Hernández Sol</b>

**TUXTLA GUTIÉRREZ, CHIAPAS.**

**ENERO DE 2014**

## INDICE

### CAPITULO I

GENERALIDADES.....	5
1.1 Resumen.....	6
1.2 palabras calves .....	7
1.3 Introducción.....	7
1.4 Justificación.....	8
1.5 Objetivos.....	9
1.5.1 Objetivo general.....	9
1.5.2 Objetivo Especifico.....	9

### CAPITULO II

MARCO TEÓRICO.....	10
2.1 Importancia de la temperatura en la incubación.....	11
2.1.1 Temperatura de la incubación y del embrión.....	12
2.1.2 Temperatura ideal de la cascara .....	13
2.1.3 Temperatura ideal del embrión.....	14
2.1.4 Necesidad de temperatura del embrión.....	15
2.1.5 Eventos de la diferenciación.....	15
2.1.6 Temperatura durante la primera semana.....	17
2.2 App Inventor.....	18
2.2.1 Historia.....	19
2.2.2 Características.....	19
2.3 Bluetooth.....	19
2.3.1 Nombre y logotipo.....	19
2.3.2 Usos y Aplicaciones.....	20
2.3.3 Aplicaciones.....	21
2.3.4 Especificaciones.....	22
3.3.4.1 Bluetooth V1.0 y V1.0b .....	22

# Residencia profesional ingeniería electrónica

---

3.3.4.2 Bluetooth V.1.1 (2002).....	22
3.3.4.3 Bluetooth v1.2 (2003).....	23
3.3.4.4 Bluetooth V2.0+ EDR (2004) .....	23
3.3.4.5 Bluetooth V2.1+EDR (2007).....	23
3.3.4.6 Bluetooth V3.0+Hs (2009).....	24
3.3.4.7 Bluetooth V4.0 (2010).....	24
2.3.5 Información técnica.....	25
2.3.6 Arquitectura Hardware.....	26
2.3.7 Dispositivo de radio bluetooth genérico.....	27
2.3.7.1 usos del bluetooth.....	27
2.3.7.2 bluetooth contra WIFI.....	28
2.4 Sistema operativo android.....	28
2.4.1 Aplicaciones.....	29
2.4.2 Marco de trabajo de Aplicación.....	29
2.4.4 Biblioteca.....	29
2.4.3 Rutime de android.....	29
2.4.5 Núcleo Linux.....	30
2.4.6 Aspectos Histórico de Android.....	30
<b>CAPITULO III</b>	
DESARROLLO. ....	31
3.1 Monitoreo y Regulación.....	32
3.2 Fuente de alimentación.....	33
3.2.1 El transformador.....	34
3.2.2 Rectificador.....	34
3.2.3 Filtrado.....	34
3.2.4 Regulación.....	34
3.2 Circuito Eléctrico.....	35
3.3 Programación en Android.....	36
3.4.1 Monitoreo.....	42
3.4.2 Regulación.....	43

# Residencia profesional ingeniería electrónica

---

3.4.3 Control de retroiluminación LCD .....	43
3.4.4 Ayuda.....	44
3.5 Programación de Android y Comunicación.....	45
<b>CAPITULO IV</b>	
<b>RESULTADOS. ....</b>	<b>48</b>
4.1 Conexiones Físicas del sistema.....	49
4.2 Pruebas y Resultados.....	50
4.3 Conclusión.....	51
4.4 Referencia.....	52
4.5 Anexos.....	53
4.5.1 Interfaz App Inventor.....	53
4.5.2 Código Arduino.....	54



# **CAPITULO I**

## **Generalidades**

## 1.1 Resumen

El proyecto propuesto consiste en el monitoreo y regulación soportado en android de la temperatura de una incubadora de aves, e implementado en la plataforma arduino, para una comunicación inalámbrica, de tal manera que exista una comunicación entre el arduino y un dispositivo móvil para realizar un monitoreo y regulación en tiempo real de la temperatura.

En este reporte, se describe el procedimiento de la elaboración de un sistema de monitoreo y regulación de la temperatura de una incubadora de aves soportado en android, también puede funcionar en cualquier requerimiento de control de temperatura, desde un cuarto frío hasta un horno casero o industrial. Conociendo el valor de la temperatura a la que se encuentra nuestra incubadora, es necesaria alguna acción de regulación que opere en el área de nuestra incubadora para así poder manejar la variable temperatura hasta un punto de referencia requerido.

Dentro de las diferentes maneras que se pueden monitorear un proceso hay una por medio de un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles, como teléfonos inteligentes o tabletas, llamado android.

En el sistema operativo android crearemos una aplicación en App Inventor. App inventor es una aplicación para crear aplicaciones en android. Con la aplicación realizada en el sistema operativo proporcionaremos la información necesaria sobre la variable, y se obtendrá el monitoreo y regulación de la temperatura en tiempo real.

Todo este intercambio de información se debe de hacer por medio de un sistema que envíe y reciba información o datos. La comunicación inalámbrica bluetooth es una herramienta que nos permite enviar y recibir, en nuestro caso monitorear y regular en tiempo real el estado de temperatura en nuestra incubadora.

La forma de hacer el monitoreo se da por medio de nuestra aplicación en android ya que cuenta con una vinculación que nos permite visualizar los cambios de temperatura, de este modo nuestra incubadora podrá ser monitoreada y regulada en tiempo real desde cualquier punto siempre y cuando nuestros dispositivos de comunicación estén al alcance de una vinculación apropiada.

## 1.2 Palabras claves

Android, Bluetooth, Arduino y App inventor.

## 1.3 Introducción

Hay lugares donde es necesario mantener una temperatura constante o en donde es necesario el poder variar la temperatura de cierto recinto hasta un punto determinado. Esta temperatura a su vez debe estar siendo monitoreada continuamente.

Para el diseño del sistema de monitoreo y regulación se realizó el análisis y estudio de diferentes tecnologías de comunicación para realizar el enlace inalámbrico para la temperatura de incubación y el dispositivo móvil, como es el WIFI, Bluetooth, IrDA, Zigbee; de esta forma se optó por la utilización de la tecnología Bluetooth, debido a que es un estándar de comunicación que la mayoría de los dispositivos móviles con el sistema operativo android como son Celulares, Smartphone, Tablet o PC disponen de forma integrada en la actualidad, por lo que ya no es necesario adaptarle un accesorio más al equipo a utilizar.

Con ello se plantea el diseño del sistema de monitoreo y regulación de temperatura de como muestra la Figura 1. Utilizando el sistema operativo android para crear una aplicación en nuestro dispositivo móvil en este caso nuestra comunicación inalámbrica para enviar y recibir datos será con un módulo Bluetooth para el correcto control de temperatura de nuestra incubadora.

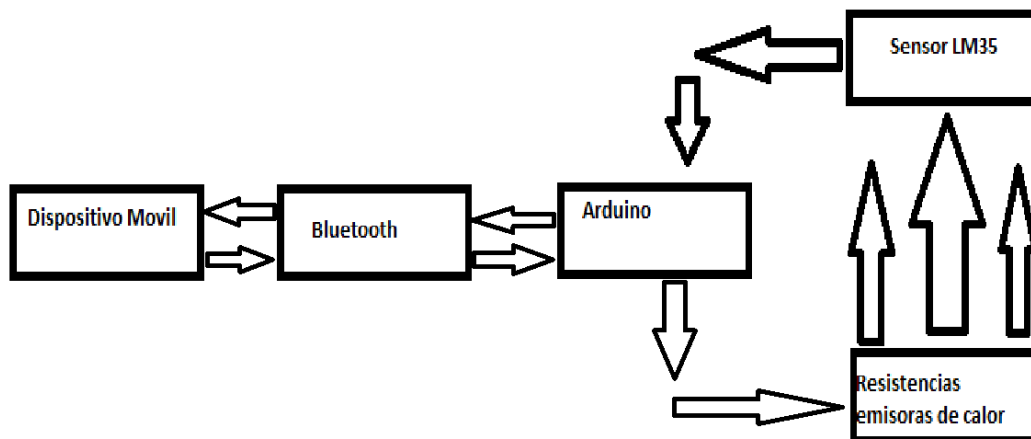


Figura 1.- Bloques del Sistema de monitoreo y regulación



El sistema de control de transmisión de datos Bluetooth, está conectado al arduino que es una placa con un microcontrolador, en el cual se ejecutan las tareas de lógica y control del sistema mediante la recepción de comandos de control provenientes del dispositivo móvil con la aplicación creada en android, los cuales son recibidos por el módulo Bluetooth RN-41 través de un puerto hacia el microcontrolador, mediante los cuales nuestras resistencias se mantendrán con la temperatura establecida o bien la temperatura que sea necesaria haciendo esta regulación en tiempo real desde nuestro dispositivo móvil.

Para el monitoreo instalaremos sensores de temperaturas dentro de la incubadora, la temperatura que existe en nuestra incubadora será enviada con el mismo sistema de transmisión de datos bluetooth a un dispositivo móvil haciendo así un perfecto monitoreo de la temperatura que existe dentro de nuestro sistema de incubación

## **1.4 Justificación**

La temperatura es una de la variables más importante de un proceso de incubación ya que en la gran mayoría esta requiere de un monitoreo y regulación de dicha variable,

Lo que se propone como proyecto es un sistema de monitoreo y regulación que permite supervisar el proceso de la temperatura y hacer cambios de regulación en tiempo real con una aplicación en el sistema operativo android que nos permite usar los recursos de comunicación de los dispositivos móviles, lo cual garantiza su funcionalidad, portabilidad y comodidad en el uso.

En este trabajo estudiaremos y analizaremos esta importante variable de una incubadora ya que si aprendemos a controlar esta variable sabremos controlar a las demás como son, oxigenación, ventilación, humedad y rotación.

## **1.5 Objetivos**

### **1.5.1 Objetivo general**

Diseñar e implementar un sistema soportado en android para monitorear y regular en tiempo real la temperatura de una incubadora de aves.

### **1.5.2 Objetivo específico**

Estudiar, analizar y elegir el tipo de aplicación que será la adecuada para implementar el sistema de monitoreo y regulación, basado en el sistema operativo android, app inventor, bluetooth, arduino y sensor de temperatura.

## **CAPITULO II**

### **Marco teórico**

## 2.1 importancia de la temperatura para la incubación

La temperatura en las fases iniciales de la incubación es tan importante como en la parte final.

La planta de incubación es hoy día una parte indiscutible de la cadena de producción avícola. La vitalidad o la denominada "calidad" de los pollitos de un día se aceptan como parámetro crítico para aprovechar todo el potencial genético de los pollos. Los pollitos de un día son un eslabón crucial entre la granja de reproductoras y la granja de pollos.

A escala global, podemos observar alteraciones significativas en el manejo de la producción de pollitos de un día. Desde la década del 60 del siglo pasado, el número de plantas pequeñas de incubación ha ido disminuyendo gradualmente. Aunque aún existan plantas de incubación pequeñas, en áreas rurales, las grandes plantas de incubación industriales para la producción de 200.000-600.000 pollitos al día se han vuelto cada vez más comunes. Observamos el aumento del tamaño de las plantas de incubación, el manejo y la tecnología de incubación han cambiado de acuerdo a la escala de la planta y el sistema de incubación más adecuado para la incubación a gran escala carga múltiple o carga única.

La temperatura de incubación es el principal punto. Se conoce bien el riesgo de sobrecalentamiento del embrión de líneas de alta rentabilidad en una incubadora de cargas múltiples, así como en una de carga única. Además de la alta temperatura durante las fases finales de incubación, encontramos usualmente bajas temperaturas en los primeros días de incubación, especialmente en incubadoras de cargas múltiples. Generalmente, se desconoce el hecho de que bajas temperaturas también afectan los resultados en la planta de incubación.

Nos enfocaremos en la temperatura de incubación como principal factor ambiental de la incubación, la importancia y la afirmación de que la temperatura en las fases iniciales de la incubación es tan importante como en la parte final.

La meta básica de la incubación es la obtención de eclosión y viabilidad/calidad de pollo ideales para cada línea genética específica. Las condiciones de incubación, especialmente la temperatura, deben satisfacer las necesidades del embrión para alcanzar esta meta. Esta afirmación es verdadera y seguirá siendo verdadera para la práctica actual de la incubación. Los aspectos básicos de la incubación no solamente influye en la eclosión, sino que pequeñas variaciones de la temperatura ideal también afectan al desempeño después de la eclosión.

Existe una ventana de temperatura sensible crítica para cada una de las diferentes fases del desarrollo embrionario. Por ejemplo, se observó baja eficiencia

alimenticia en aves expuestas a baja temperatura durante los primeros días de la incubación. Por otro lado, la eficiencia alimenticia fue mejor cuando la temperatura de la incubadora se aumentó por 6 horas al día durante los 10 primeros o los 10 últimos días de incubación.

En las últimas décadas, el número de artículos científicos sobre la temperatura de incubación ha aumentado. Estos artículos enfocan a menudo la influencia de la temperatura de incubación durante la última fase de incubación sobre el desempeño tras la eclosión de los pollos.

El énfasis en las fases finales de la incubación se debe al riesgo hoy ampliamente reconocido de sobrecalentar al embrión en plantas de incubación de cargas múltiples y a la idea de que el embrión se desarrolla y crece mejor en incubadoras de carga única. Sin embargo, podemos preguntarnos si la temperatura de incubación en fases anteriores no es tan importante como en las fases finales, especialmente con relación a la uniformidad en la eclosión. Esto nos lleva a preguntarnos si las incubadoras de cargas múltiples pueden controlar de manera ideal la temperatura del embrión en las fases iniciales de la incubación.

### **2.1.1 Temperatura de la incubadora y del embrión**

Hasta el final del siglo XX, la mayoría de los artículos científicos y consultores de plantas de incubación utilizaban la temperatura de la incubadora cuando se referían a los efectos de la temperatura sobre el desarrollo embrionario y la eclosión. La baja calidad de los pollitos que a menudo se observa en las plantas de incubación se atribuye a altas temperaturas. Temperaturas inadecuadas en incubadoras de cargas múltiples también pueden explicar la reducción de la eclosión observada en los EUA.

Actualmente, las investigaciones sobre incubación consideran la temperatura del embrión, y no la temperatura de la incubadora como el factor más crítico para la eclosión y la calidad del pollito.

La temperatura de la cáscara del huevo es un equilibrio entre el calor metabólico producido por el embrión y el intercambio de calor con el ambiente inmediato del huevo. En las incubadoras modernas con ventilación forzada, la transferencia hacia los huevos ocurre por convección: el aire caliente o frío fluye sobre los huevos, calentándolos o enfriándolos.

El enfoque "temperatura del embrión" tiene gran impacto en la rutina de la planta de incubación. La temperatura del aire en la incubadora determina la temperatura

del embrión / cáscara del huevo. Para calentarlos, el aire caliente debe fluir sobre los huevos.

Cuando el sistema de enfriamiento está en operación, la temperatura del aire que fluye sobre los huevos es más baja que la temperatura de la cáscara, lo que reduce su temperatura. El aumento y la reducción de la temperatura de la cáscara dependen, principalmente, de diferencias de temperatura y de la velocidad del aire sobre los huevos. Se ha demostrado que la humedad relativa es menos importante, por lo menos en el caso del enfriamiento de los huevos en las últimas fases de la incubación.

En la incubadora, el punto seleccionado de la temperatura es un factor constante y la temperatura del aire próximo a los huevos varía de acuerdo al calor metabólico producido por los huevos.

La temperatura de la cáscara en incubadoras es regulada alternando huevos "viejos" (huevos que producen calor) con huevos nuevos, que demandan calor.

El crecimiento embrionario y tras la eclosión en temperaturas bajas de la cáscara del huevo (36.7°C) y control (37.8°C) durante la primera semana de incubación

### **2.1.2 Temperatura ideal de la cáscara**

Hasta hace 5 a 10 años, nunca pensaba en la temperatura de la cáscara porque estaba satisfecho con la eclosión y con la calidad de los pollitos. Sin embargo, hoy en día, se trabaja con la misma incubadora y se tiene que enfrentar reducción de la eclosión, peor calidad de los pollitos y pollitos de segunda clase con signos de sobrecalentamiento del embrión, como ombligos abiertos y jarretes rojos. Si se aplica un perfil inadecuado de temperatura (demasiado alto)

La producción de calor de los huevos de líneas genéticas modernas de alto rendimiento aumentó en comparación con la producción de calor de las líneas tradicionales. Consecuentemente, para mantener la temperatura ideal del embrión entre la cáscara del huevo, se tiene que retirar más calor de los huevos que contienen embriones de líneas modernas. En algunas incubadoras, la forma tradicional de enfriar a los huevos colocando huevos fríos al lado de los calientes no es eficaz para regular la temperatura de la cáscara en niveles ideales.

En algunas incubadoras, se aumentó la capacidad de enfriamiento, para evitar el sobrecalentamiento reduciendo el punto de la temperatura. El tiempo total necesario para calentar los huevos se determina principalmente por la temperatura inicial de los huevos y por el volumen de aire alrededor de los huevos: bandejas

completamente llenas necesitan más tiempo para calentarse que aquellas con espacios vacíos. Además, huevos más grandes necesitan más tiempo para pasar de la temperatura de almacenaje a la de la incubadora que huevos más chicos.

Se ha demostrado que la velocidad del aire sobre el huevo y las diferencias de temperatura entre el aire y la cáscara son los factores físicos más importantes en la transferencia de calor.

En las incubadoras, la capacidad de calentamiento es suficiente para llevar la temperatura de la cáscara a una temperatura ideal de incubación de 37,8°C en 6-12 horas, dependiendo si los huevos fueron precalentados o no. Mientras que en otras incubadoras, el aire se calienta a través del calentador de la incubadora y del calor metabólico producido por los embriones.

Aunque se sabe que es más rápido calentar los huevos hasta la temperatura de incubación, generalmente la temperatura de la cáscara no alcanza la temperatura ideal de incubación.

### **2.1.3 Temperatura del embrión**

Después de calentar los huevos, el perfil de temperatura de carga única mantiene la temperatura de la cáscara a un nivel constante de 37,8°C durante los primeros 12-14 días y después se permite un aumento gradual de hasta 38.3- 38.6°C en el día de la transferencia.

En las incubadoras, vemos un perfil muy diferente: inferior a 37,8°C durante los cuatro primeros días de incubación y demasiado elevada (hasta 39.5- 40.0°C) en los últimos días antes de la transferencia.

Concluimos que en las plantas de incubación de pollos actuales, la incubación tradicional ha dejado de ser económicamente eficiente, porque las temperaturas del embrión son demasiado bajas o demasiado altas ya que las plantas no cuentan con las herramientas tecnológicas necesarias para controlar la temperatura de la cáscara. En general, las condiciones por debajo de las ideales de las incubadoras resultan en menor eclosión y en pollitos de menor calidad.

## 2.1.4 Necesidades de temperatura del embrión

Como explicamos antes, los puntos de temperatura aplicados históricamente no necesariamente sostienen el desarrollo ideal de embriones de diferentes líneas genéticas. A menudo observamos temperaturas demasiado bajas durante la fase de diferenciación del desarrollo. La duda se refiere a las consecuencias para el desarrollo embrionario y para la calidad de los pollitos. En otros artículos anteriores, se definieron tres fases naturales de desarrollo, las temperaturas ideales para el embrión en la fase inicial del desarrollo: la fase de diferenciación.

## 2.1.5 Eventos de la diferenciación

En las aves, la fertilización y las divisiones iniciales de clivaje del ovocito ocurren en el oviducto, antes que se formen el albumen y la cáscara. Mientras la yema en la glándula de la cáscara gira por 20 horas a una tasa de 10-12 revoluciones por hora, las células embrionarias se someten a las primeras especificaciones que determinarán su futuro destino y función.

En el huevo fresco, el blastodermo contiene 20.000- 30.000 células con el destino especificado: algunas formarán las estructuras de la cabeza y otras las estructuras de la cola. Además, el eje dorso-ventral se ha formado: el lado próximo al albumen forma las estructuras dorsales. El lado ventral está en contacto íntimo con la yema. Todos los tejidos embrionarios y extra-embriónicos surgen de la zona pelúcida del blastodermo. Los tejidos extra-embriónicos ofrecen al embrión un ambiente constante y protector (amnio y alantoide) y los nutrientes del albumen y de la yema (membranas del saco vitelino y del saco del albumen). En la fase posterior (crecimiento), la membrana corioalantoide se desarrolla para quitar el dióxido de carbono metabólico y para suministrar oxígeno suficiente al embrión. Los tejidos extra-embriónicos se diferencian junto con el embrión y la diferenciación normal del desarrollo extra-embriónico es esencial para el desarrollo del embrión.

Cuando la temperatura del huevo aumenta hasta la temperatura de incubación de 37.5 - 37.8°C, la diferenciación continúa. El resultado de la diferenciación puede acompañarse fácilmente, porque el blastodermo se transforma de un disco plano en un pequeño embrión, con las estructuras prematuras de la cabeza, corazón y de la cola claramente visibles después de 60 horas de incubación.

En esta fase de diferenciación, las células dentro de la zona pelúcida cambian de posición debido a movimientos drásticos por influencia de temperaturas más altas. La migración de células forma parte del proceso de gastrulación, durante el que se



## Residencia profesional ingeniería electrónica

---

diferencian tres capas germinales el endoderma, el mesoderma y el ectoderma. Además, como resultado de migración masiva de grupos diferentes de células, el blastoderma pasa por una transición, de una estructura bidimensional a una tridimensional. El proceso de gastrulación, con su intensiva migración de células, es altamente dependiente de la temperatura.

Muchos artículos sobre la embriología de gallinas ofrecen información acerca de la influencia de la temperatura sobre los eventos de diferenciación que ocurren al principio del desarrollo. Sin embargo, es necesario destacar que casi todos los experimentos se refieren a la temperatura de la incubadora y no a la temperatura del embrión. Por lo tanto, todas las temperaturas mencionadas abajo se refieren a la temperatura de la incubadora, excepto cuando indicado. Sin embargo, como la mayoría de los experimentos citados es de pequeña escala, la temperatura del embrión está dentro de  $0.1^{\circ}\text{C}$  de la temperatura de la incubadora.

Se observaron anomalías embrionarias después de la exposición precoz tanto a temperaturas bajas como altas. Los primeros eventos de la diferenciación ocurren en el oviducto a una temperatura corporal de  $41^{\circ}\text{C}$ . Cuando se produce la puesta del huevo, el embrión está en la etapa de desarrollo denominada pregástrula. Su apariencia es la de una rosquita, con el área opaca como si fuera una rosquita y la zona pelúcida al centro. Entre los huevos de la misma parvada, se observó variación en el tamaño del blastoderma.

Se concluyó que, en base al tamaño del embrión, el blastoderma de parvadas de reproductoras más viejas se desarrolló más que los embriones de parvadas de reproductoras más jóvenes. Aún es necesario investigar si esta variación se debe a diferencias en la temperatura corporal de reproductoras más jóvenes o más viejas. La diferenciación embrionaria sincronizada continúa en la incubadora cuando la temperatura del embrión alcanza la temperatura ideal de incubación de  $37.5 - 37.8^{\circ}\text{C}$ . En temperaturas inferiores a los  $35^{\circ}\text{C}$ , la tasa de desarrollo disminuye y la eclosión se retrasa en aproximadamente tres días, pero las diferentes estructuras morfológicas se desarrollan en sincronía.

Aunque no haya datos específicos disponibles sobre temperaturas entre  $27$  y  $35^{\circ}\text{C}$ , suponemos desarrollo heterocrónico de los grupos de células embrionarias. Por debajo del cero fisiológico de  $25-27^{\circ}\text{C}$ , el desarrollo de las diferentes estructuras no está sincronizado: algunas estructuras dejan de desarrollarse, mientras otras siguen desarrollándose, pero a una tasa más lenta.

El desarrollo embrionario prácticamente cesa cuando la temperatura es igual o menor que  $14^{\circ}\text{C}$ , aunque en embriones almacenados a esta temperatura, se observa un aumento del número de células muertas.

## 2.1.6 Temperatura durante la primera semana

- El calentamiento de los huevos durante la incubación se produce mediante el intercambio de calor entre el aire y los huevos.
- La temperatura de trabajo en las incubadoras se enmarca entre 37 y 38°C. Los embriones mueren a menos de 35 y a más de 40°C.
- El nivel de temperatura óptimo a aplicar depende de:
  1. El tipo de incubadora
  2. La calidad y el tamaño de los huevos
  3. La edad de los embriones,
  4. Además de la especie de que se trate.
- En todos los casos, es necesario disminuir el nivel de temperatura durante los últimos días (2 a 3) de incubación; es decir, la temperatura se diferencia de acuerdo a las etapas de incubación

Si buscamos datos sobre la influencia de la temperatura de la incubadora sobre la etapa inicial de desarrollo, casi seguramente volveremos al libro "Patogénesis del Embrión Aviario" de Romanoff y Romanoff (1972). Estos investigadores incomparables demostraron que pequeños desvíos de la temperatura ideal, definida por Romanoff como 37.5°C, tienen efectos significativos sobre el crecimiento del embrión y de las membranas extra-embriónicas.

A 96 horas de incubación (día 4), el crecimiento relativo de las 31 estructuras necesita de 5-10% más tiempo para desarrollarse cuando el embrión es incubado a 37°C. Consecuentemente, un embrión incubado a 37°C necesita 100.8 horas ( $0.05 \cdot 96 + 96$ ) para alcanzar la misma etapa embrionaria que un embrión incubado a 37.5°C.

Además, la mortalidad embrionaria aumenta en temperaturas de incubación más bajas.

En discusiones más recientes, se ha dado más atención a la temperatura del embrión como factor importante para la calidad de los pollitos. Aunque la mayoría de las investigaciones se enfocan en el sobrecalentamiento del embrión en las últimas fases del desarrollo, Lourens y colaboradores (2005) también estudiaron el efecto de la baja temperatura (36.7°C) de la cáscara del huevo sobre el desarrollo de embriones. La temperatura de la cáscara se usó como referencia de la temperatura interna del embrión porque ésta sólo puede determinarse a través de métodos invasivos.

A partir de estos experimentos, Lourens y colaboradores concluyeron que una temperatura de la cáscara de 36.7°C en los primeros 7 días de incubación redujo

significativamente la masa corporal sin la yema de los embriones en los días 7, 14 y 21,5. También demostraron que, para huevos de reproductoras más viejas, el efecto de la baja temperatura de la cáscara en la primera semana de incubación permaneció hasta el desarrollo tras eclosión. Embriones de los lotes de reproductoras más viejas incubados en la menor temperatura de la cáscara tuvieron crecimiento significativamente menor durante la primera semana.

La temperatura constituye un factor fundamental en la incubación, ya que el calentamiento de los huevos se produce debido al intercambio de calor entre el aire y los huevos.

Durante la incubación, el nivel de temperatura óptimo de la incubadora se enmarca entre 37 y 38°C. Aunque al llegar a los últimos días (2 o 3) de incubación es necesario reducir la temperatura.

### 2.2 App Inventor

Es una aplicación de Google Labs para crear aplicaciones de software para el sistema operativo Android . De forma visual y a partir de un conjunto de herramientas básicas, el usuario puede ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones fruto de App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con Google App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y el Android Market, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente

Para crear una aplicación con app inventor hay que realizar algunos pasos como:

- El diseño de la aplicación, en la que se seleccionan los componentes para su aplicación.
- El editor de bloques, donde irás escogiendo los bloques que te sean necesarios según la aplicación que tenga que deseases hacer.

## 2.2.1 Historia

La aplicación se puso a disposición del público el 12 de julio de 2010 y está dirigida a personas que no están familiarizadas con la programación informática. En la creación de App Inventor, Google se basó en investigaciones previas significativas en informática educativa. Fue creada a mediados del 2009 por el profesor Harold Abelson del MIT. Antes de salir al mercado se ha probado en diferentes centros educativos y la han utilizado desde niños de 12 años hasta licenciados universitarios sin nociones de programación

## 2.2.2 Características

El editor de bloques de la aplicación utiliza la librería Open Blocks de Java para crear un lenguaje visual a partir de bloques. Estas librerías están distribuidas por Massachusetts Institute of Technology (MIT) bajo su licencia libre (MIT License). El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza Kawa como lenguaje de programación, distribuido como parte del sistema operativo GNU de la Free Software Foundation

## 2.3 Bluetooth

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. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como PDA, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

### 2.3.1 Nombre y logo

El nombre procede del rey danés y noruego Harald Blåtand, cuya traducción al inglés es Harald Bluetooth, conocido por unificar las tribus noruegas, suecas y danesas y por convertirlos al cristianismo. La idea de este nombre fue propuesto

por Jim Kardach que desarrolló un sistema que permitiría a los teléfonos móviles comunicarse con los ordenadores y unificar la comunicación de los sistemas digitales.



**Figura 2.- logotipo Bluetooth**

El logo de Bluetooth son las runas de las iniciales del nombre y el apellido. † La (Hagall) y la B (Berkana).

### **2.3.2 Usos y aplicaciones**

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo.

Los dispositivos que incorporan este protocolo pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión es suficiente. Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de transmisión, siendo totalmente compatibles los dispositivos de una caja de ordenador como se muestra en la fig.

**Figura 3.- Potencia máxima permitida y alcance**

Clase	(nW)	(dBm)	(aproximado)
Clase 1	100 mW	20 dBm	-30 m
Clase 2	2.5 mW	4 dBm	-10
Clase 3	1 Mw	0 dBm	-1

En la mayoría de los casos, la cobertura efectiva de un dispositivo de clase 2 se extiende cuando se conecta a un transceptor de clase 1. Esto es así gracias a la mayor sensibilidad y potencia de transmisión del dispositivo de clase 1, es decir, la mayor potencia de transmisión del dispositivo de clase 1 permite que la señal llegue con energía suficiente hasta el de clase 2. Por otra parte la mayor sensibilidad del dispositivo de clase 1 permite recibir la señal del otro pese a ser más débil.

Los dispositivos con Bluetooth también pueden clasificarse según su ancho de banda

**Figura 4.- Clasificación por su ancho de banda**

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s
Versión 4.0	24 Mbit/s

### 2.3.3 Aplicacion

El 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. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar los cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la información personal, como PDA, Teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

### **2.3.4 Especificaciones**

La utilidad Bluetooth fue desarrollada como un reemplazo del cable en 1994 por Jaap Haartsen y Mattisson Sven, que estaban trabajando para Ericsson en Lund, Suecia. La utilidad se basa en la tecnología de saltos de frecuencia de amplio espectro.

Las prestaciones fueron publicadas por el Bluetooth Special Interest Group (SIG). El SIG las anunció formalmente el 20 de mayo de 1998. Hoy cuenta con una membresía de más de 20,000 empresas en todo el mundo. Fue creado por Ericsson, IBM, Intel, Toshiba y Nokia, y posteriormente se sumaron muchas otras compañías. Todas las versiones de los estándares de Bluetooth están diseñadas para la compatibilidad hacia abajo, que permite que el último estándar cubra todas las versiones anteriores.

#### **3.3.4.1 Bluetooth v1.0 y v1.0b**

Las versiones 1.0 y 1.0b han tenido muchos problemas, y los fabricantes tenían dificultades para hacer sus productos interoperables. Las versiones 1.0 y 1.0b incluyen en hardware de forma obligatoria la dirección del dispositivo Bluetooth (BD\_ADDR) en la transmisión (el anonimato se hace imposible a nivel de protocolo), lo que fue un gran revés para algunos servicios previstos para su uso en entornos Bluetooth.

#### **3.3.4.2 Bluetooth v1.1 (2002)**

- Ratificado como estándar IEEE 802.15.1-2002
- Se corrigieron muchos errores en las especificaciones 1.0b.
- Añadido soporte para canales no cifrados.
- Indicador de señal recibida (RSSI).

### 2.3.4.3 Bluetooth v1.2 (2003)

Esta versión es compatible con USB 1.1 y las principales mejoras son las siguientes:

- Una conexión más rápida y Discovery (detección de otros dispositivos bluetooth).
- Salto de frecuencia adaptable de espectro ampliado (AFH), que mejora la resistencia a las interferencias de radio frecuencia, evitando el uso de las frecuencias de lleno en la secuencia de saltos.
- Mayor velocidad de transmisión en la práctica, de hasta 721 kbit/s, que en v1.1.
- Conexiones Sincrónicas extendidas (ESCO), que mejoran la calidad de la voz de los enlaces de audio al permitir la retransmisión de paquetes corruptos, y, opcionalmente, puede aumentar la latencia de audio para proporcionar un mejor soporte para la transferencia de datos simultánea.
- Host Controller Interface (HCI) el apoyo a tres hilos UART.
- Ratificado como estándar IEEE 802.15.1-2005
- Introdujo el control de flujo y los modos de retransmisión de L2CAP.

### 2.3.4.4 Bluetooth v2.0 + EDR (2004)

Esta versión de la especificación principal Bluetooth fue lanzado en 2004 y es compatible con la versión anterior 1.2. La principal diferencia es la introducción de una velocidad de datos mejorada (EDR "Enhanced Data Rate" "mayor velocidad de transmisión de datos") para acelerar la transferencia de datos. La tasa nominal de EDR es de 3 Mbit / s, aunque la tasa de transferencia de datos práctica es de 2,1 Mbit / s. EDR utiliza una combinación de GFSK y Phase Shift Keying modulación (PSK) con dos variantes,  $\pi/4$ -DQPSK y 8DPSK. EDR puede proporcionar un menor consumo de energía a través de un ciclo de trabajo reducido.

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. Por lo menos un dispositivo de estados comerciales "sin EDR Bluetooth v2.0" en su ficha técnica.



### 2.3.4.5 Bluetooth v2.1 + EDR (2007)

Bluetooth Core Version especificación 2.1 + EDR es totalmente compatible con 1.2, y fue adoptada por el Bluetooth SIG (Bluetooth Special Interest Group) el 26 de julio de 2007.

La función de titular de la 2.1 es Secure Simple Pairing (SSP): se mejora la experiencia de emparejamiento de dispositivos Bluetooth, mientras que el aumento del uso y la fuerza de seguridad. Vea la sección de enlace de abajo para más detalles.

2.1 permite a otras mejoras, incluida la "respuesta amplia investigación" (EIR), que proporciona más información durante el procedimiento de investigación para permitir un mejor filtrado de los dispositivos antes de la conexión, y oler subrating, lo que reduce el consumo de energía en modo de bajo consumo.

### 2.3.4.6 Bluetooth v3.0 + HS (2009)

La versión 3.0 + HS de la especificación principal Bluetooth fue aprobado por el Bluetooth SIG el 21 de abril de 2009. Bluetooth 3.0 + HS soporta velocidades de transferencia de datos teórica de hasta 24 Mbit / entre sí, aunque no a través del enlace Bluetooth propiamente dicho. La conexión Bluetooth nativa se utiliza para la negociación y el establecimiento mientras que el tráfico de datos de alta velocidad se realiza mediante un enlace 802.11. Su principal novedad es AMP (Alternate MAC / PHY), la adición de 802,11 como transporte de alta velocidad. Estaban inicialmente previstas dos tecnologías para incorporar en AMP: 802.11 y UWB, pero finalmente UWB no se encuentra en la especificación.

La incorporación de la transmisión a alta velocidad no es obligatoria en la especificación y por lo tanto, los dispositivos marcados con "+ HS" incorporan el enlace 802.11 de alta velocidad de transferencia de datos.

### 2.3.4.7 Bluetooth v4.0 (2010)

El SIG de Bluetooth ha completado la especificación del Núcleo de Bluetooth en su versión 4.0, que incluye Bluetooth clásico, Bluetooth de alta velocidad y protocolos Bluetooth de bajo consumo. Bluetooth de alta velocidad se basa en Wi-Fi, y Bluetooth clásico consta de protocolos Bluetooth preexistentes. Esta versión ha sido adoptada el 30 de junio de 2010. Bluetooth de baja energía (*Bluetooth Low Energy* o BLE) es un subconjunto de Bluetooth v4.0 con una pila de protocolo

completamente nueva para desarrollar rápidamente enlaces sencillos. Como alternativa a los protocolos estándar de Bluetooth que se introdujeron en Bluetooth v1.0 a v4.0 está dirigido a aplicaciones de muy baja potencia alimentados con una pila botón. Diseños de chips permiten dos tipos de implementación, de modo dual, de modo único y versiones anteriores mejoradas.

- En implementaciones de modo único sólo se incluye la pila de protocolo de baja energía. CSR, Nordic Semiconductor y Texas Instruments han dado a conocer solo las soluciones Bluetooth modo de baja energía.
- En implementaciones de modo dual, la funcionalidad de Bluetooth de bajo consumo está integrada en un controlador Bluetooth clásico existente. En la actualidad (2011-03) los siguientes fabricantes de semiconductores han anunciado la disponibilidad de chips que cumplen esta norma: Atheros, CSR, Broadcom y Texas Instruments. La arquitectura resultante comparte la radio y funcionalidades del Bluetooth clásico, resultando en un incremento de coste negligible comparado con el Bluetooth clásico.

El 12 de junio de 2007, Nokia y Bluetooth SIG anunciaron que Wibree formará parte de la especificación Bluetooth, como una tecnología de muy bajo consumo Bluetooth.

El 17 de diciembre de 2009, el Bluetooth SIG adoptó la tecnología Bluetooth de bajo consumo como el rasgo distintivo de la versión 4.0. Los nombres provisionales Wibree y Bluetooth ULP (*Ultra Low Power*) fueron abandonados y el nombre BLE se utilizó durante un tiempo. A finales de 2011, los nuevos logotipos "*Smart Bluetooth Ready*" para los anfitriones y "*Smart Bluetooth*" para los sensores se presentó como la cara pública general de BLE.

### 2.3.5 Información técnica

La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s (1 Mbit/s de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores).

Opera en la frecuencia de radio de 2,4 a 2,48 GHz con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos por segundo. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz; esto permite dar seguridad y robustez.

La potencia de salida para transmitir a una distancia máxima de 10 metros es de 0 dBm (1 mW), mientras que la versión de largo alcance transmite entre 20 y 30 dBm (entre 100 mW y 1 W).

Para lograr alcanzar el objetivo de bajo consumo y bajo costo se ideó una solución que se puede implementar en un solo chip utilizando circuitos CMOS. De esta manera, se logró crear una solución de 9x9 mm y que consume aproximadamente 97% menos energía que un teléfono celular común. El protocolo de banda base (canales simples por línea) combina conmutación de circuitos y paquetes. Para asegurar que los paquetes no lleguen fuera de orden, los slots pueden ser reservados por paquetes síncronos, empleando un salto diferente de señal para cada paquete.

La conmutación de circuitos puede ser asíncrona o síncrona. Cada canal permite soportar tres canales de datos síncronos (voz) o un canal de datos síncrono y otro asíncrono.

Cada canal de voz puede soportar una tasa de transferencia de 64 kbit/s en cada sentido, la cual es suficiente para la transmisión de voz. Un canal asíncrono puede transmitir como mucho 721 kbit/s en una dirección y 56 kbit/s en la dirección opuesta. Sin embargo, una conexión síncrona puede soportar 432,6 kbit/s en ambas direcciones si el enlace es simétrico.

### 2.3.6 Arquitectura hardware

El hardware que compone el dispositivo Bluetooth está compuesto por dos partes:

- **un dispositivo de radio**, encargado de modular y transmitir la señal.
- **un controlador digital**, compuesto por una CPU, un procesador de señales digitales (DSP - Digital Signal Processor) llamado Link Controller (o controlador de Enlace) y de las interfaces con el dispositivo anfitrión. El LC o Link Controller se encarga del procesamiento de la banda base y del manejo de los protocolos ARQ y FEC de la capa física; además, se encarga de las funciones de transferencia tanto asíncrona como síncrona, la codificación de audio y el cifrado de datos. La CPU del dispositivo se encarga de las instrucciones relacionadas con Bluetooth en el dispositivo anfitrión, para así simplificar su operación. Para ello, sobre la CPU corre un software denominado Link Manager cuya función es la de comunicarse con otros dispositivos por medio del protocolo LMP.

## 2.3.7 Dispositivo de radio Bluetooth genérico

Entre las tareas realizadas por el LC y el Link Manager, destacan las siguientes:

- Envío y Recepción de Datos.
- Paginación y Peticiones.
- Establecimiento de conexiones.
- Autenticación.
- Negociación y establecimiento de tipos de enlace.
- Establecimiento del tipo de cuerpo de cada paquete.
- Establecer el dispositivo en modo sniff o hold: El primero, sniff, significa olfatear, pero en castellano y en informática se traduce por escuchar (el medio): en este caso es la frecuencia o frecuencias en la que está funcionando el dispositivo. Así, cualquier paquete de datos enviado en esa frecuencia será "leído" por el dispositivo, aunque no vaya dirigido a él. Leerá todos los datos que se envíen en esa frecuencia por cualquier otro dispositivo Bluetooth, es lo que se denomina rastreo de paquetes. Una técnica parecida pero a nivel de frecuencias es la que se utiliza para detectar redes wi-fi, generalmente para encontrar redes abiertas (sin contraseña), al escanear todas las frecuencias se obtiene información de cada frecuencia o canal de las redes wi-fi disponibles. Hold por su parte significa mantener, retener; esto quiere decir que el dispositivo se mantendrá en esa frecuencia aunque no emita ni reciba nada, manteniendo esa frecuencia siempre disponible aunque otros dispositivos la utilicen.

### 2.3.7.1 Usos de Bluetooth

Bluetooth se utiliza principalmente en un gran número de productos tales como teléfonos, impresoras, módems y auriculares. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda. Su uso más común está integrado en teléfonos y PDA, bien por medio de unos auriculares Bluetooth o en transferencia de ficheros. además se puede realizar y confeccionar enlaces o vincular distintos dispositivos entre sí.

Bluetooth simplifica el descubrimiento y configuración de los dispositivos, ya que estos pueden indicar a otros los servicios que ofrecen, lo que permite establecer la conexión de forma rápida (sólo la conexión, no la velocidad de transmisión)

## 2.3.7.2 Bluetooth contra Wi-Fi.

Bluetooth y Wi-Fi cubren necesidades distintas en los entornos domésticos actuales: desde la creación de redes y las labores de impresión a la transferencia de ficheros entre PDA y ordenadores personales. Ambas tecnologías operan en las bandas de frecuencia no reguladas (banda ISM).

## 2.4 Sistema Operativo Android

Android (figura 5) es un sistema operativo basado en Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas, inicialmente desarrollado por Android, Inc. Google respaldó económicamente y más tarde compró esta empresa en 2005. Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.



**Figura 5.- Logotipo android**

Los componentes principales del sistema operativo de Android (cada sección se describe en detalle):

### 2.4.1 Aplicaciones

Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

### 2.4.2 Marco de trabajo de aplicaciones

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo **permite que los componentes sean** reemplazados por el usuario.

### 2.4.3 Bibliotecas

Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

### 2.4.4 Runtime de Android

Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para

memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

### **2.4.5 Núcleo Linux**

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

### **2.4.6 Aspecto histórico de Android**

Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43,6% en el tercer trimestre. A nivel mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (iOS de Apple, Inc.).

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se ha llegado ya a 1.000.000 de aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java.

## **CAPITULO III**

### **Desarrollo**



## 3.1 Monitoreo y regulación

Para el diseño del sistema de monitoreo y regulación se realizó el análisis y estudio de diferentes tecnologías de comunicación para realizar el enlace inalámbrico la temperatura de incubación y el dispositivo móvil, como es el WIFI, Bluetooth, IrDA, Zigbee; de esta forma se optó por la utilización de la tecnología Bluetooth, debido a que es un estándar de comunicación que la mayoría de los dispositivos móviles con el sistema operativo android como son Celulares, Smartphone, Tablet o PC disponen de forma integrada en la actualidad, por lo que ya no es necesario adaptarle un accesorio más al equipo a utilizar.

Con ello se plantea el diseño del sistema de monitoreo y regulación de temperatura utilizando el sistema operativo android para crear una aplicación en nuestro dispositivo móvil en esta caso nuestra comunicación inalámbrica para enviar y recibir datos será con un módulo Bluetooth para el correcto control de temperatura de nuestra incubadora.

El sistema de control de transmisión de datos Bluetooth, está conectado al arduino que es una placa con un microcontrolador, en el cual se ejecutan las tareas de lógica y control del sistema mediante la recepción de comandos de control provenientes del dispositivo móvil con la aplicación creada en android, los cuales son recibidos por el módulo Bluetooth RN-41 través de un puerto hacia el microcontrolador, mediante los cuales nuestras resistencias se mantendrán con la temperatura establecida o bien la temperatura que sea necesaria haciendo esta regulación en tiempo real desde nuestro dispositivo móvil.

Para el monitoreo instalaremos sensores de temperaturas dentro de la incubadora, la temperatura que existe en nuestra incubadora será enviada con el mismo sistema de transmisión de datos bluetooth a un dispositivo móvil haciendo así un perfecto monitoreo de la temperatura que existe dentro de nuestro sistema de incubación.

## 3.2 Fuente de alimentación

Esta etapa se encarga de suministrar energía a los diferentes módulos de control del sistema de monitoreo y regulación de la incubadora. Esta etapa inicia con un voltaje de AC, luego se filtra a un nivel CD y por último, se regula para obtener el voltaje CD fijo deseado. La regulación se obtiene mediante un CI regulador de voltaje que toma un voltaje de CD y proporciona un voltaje CD inferior que permanece constante, aunque el voltaje de entrada varíe a la carga conectada a la salida de voltaje de CD cambie. En la figura 6 se muestra un diagrama a bloques que contiene las partes de una alimentación típica y el voltaje en diversos puntos de la fuente.

La fuente de alimentación está constituida por las siguientes etapas.

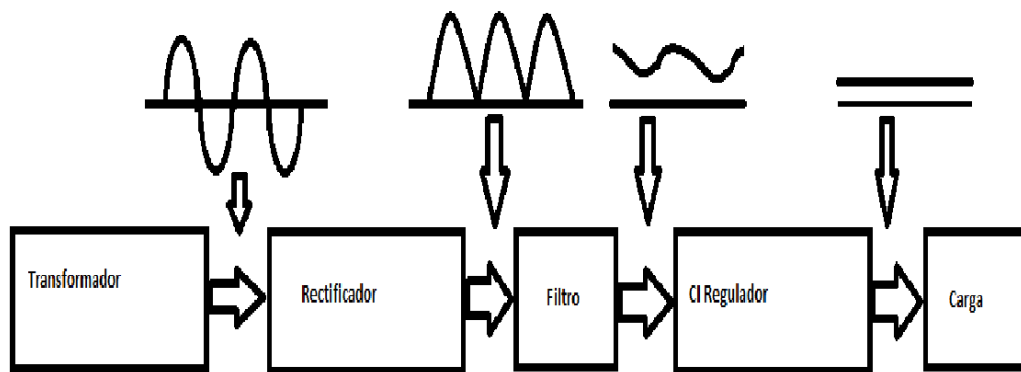


Figura 6.- diagrama a bloques de una fuente de alimentación

El voltaje de AC de entrada es de 120 V y se conecta a un transformador que disminuye ese voltaje al nivel de la salida de CD deseado 15V.

## 3.2.1 El transformador

Se tomó en cuenta que el transformador para una alimentación establecida es un transformador separador esto quiere decir que dispone por seguridad de dos devanados separados galvánicamente (eléctricamente). A la tensión que se desea que la fuente de alimentación suministre, se le añade alrededor de 3 V a 4 V por las caídas de tensión producidas al rectificar y regular en tensión e intensidad.

## 3.2.2 Rectificador

Esta etapa se implementó mediante un puente de diodos, el cual proporciona un voltaje rectificado, debido a que el voltaje de salida que se obtiene del puente rectificador es una señal de cd pulsante.

## 3.2.3 Filtrado

Esta etapa se encarga de filtrar la señal pulsante obtenida del puente de diodos, para obtener una señal de CD, por lo general esta señal aun contiene algo de rizo o variación de CA.

Para determinar el valor del condensador electrolítico que se aplicó a la salida del puente rectificador de doble onda, la regla que se suele aplicar, suele estar sobre los 2000 $\mu$ f por Ampere de salida y la tencion del orden del valor superior estándar al requerido, ósea, según esto para una fuente de 1.5a /15 V, el condensador electrolítico debe ser 3000 $\mu$ f /35V.

## 3.2.4 Regulación

Esta etapa se encarga de fijar el voltaje de salida y es realizada mediante los CI reguladores de voltaje de la serie 78H05 o 79H12, los cuales proporcionan voltajes fijos de 12V, 5V.

12V se utiliza para alimentar el arduino que se encarga de controlar la temperatura y los sensores de temperatura (LM35)

120 V se utiliza para alimentar las cargas de CA (resistencias emisoras de calor).

## 3.3 Circuito Electrónico

Dentro de nuestro sistema de regulación de temperatura, fue muy importante trabajar en 2 etapas del circuito físico: censar y controlar la temperatura. Para ello, se trabajó con 2 sensores de temperatura LM35, los cuales nos entregan una lectura analógica. Dicha señal, leída con el Arduino, fue condicionada con un amplificador operacional LM358 configurado como seguidor de línea, para así mejorar la lectura del Arduino (figura 7).

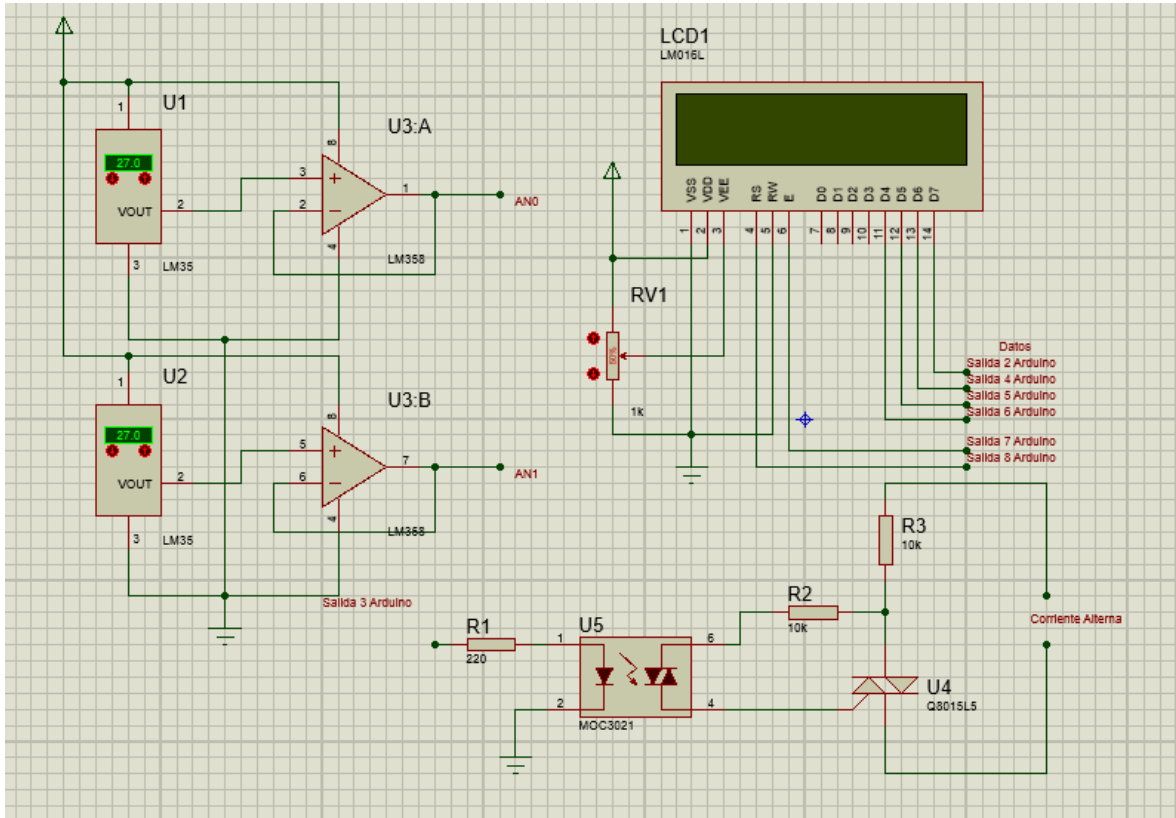


Figura 7.- diagrama electrónico

La siguiente etapa, fue la de la regulación de la temperatura, para ello fue necesario crear una etapa de potencia, la cual controla el voltaje que alimenta a las resistencias caloríficas. Para ello tenemos una etapa en la cual disponemos de un optoacoplador MOC3010, y un triac MAC15A. Esta etapa consiste en, hacer un control del voltaje de las resistencias por modulación por ancho pulso.

El control PWM (modulación por ancho de pulso) es necesario para poder hacer un control pleno de la temperatura, pues fue necesario tener controlado en que momento la resistencia calorífica debía incrementar, disminuir, o mantener su temperatura estable.

Se incorporó también una pantalla lcd, para poder mostrar en el Arduino la temperatura o cambios que estén ocurriendo en el sistema.

### 3.4 Programación en Android

La programación de la Aplicación en Android, fue una parte primordial de este trabajo de residencia, en ella se logró implementar, un programa que dentro de la plataforma Android, lograra comunicarse por Bluetooth con un dispositivo físico (Arduino) y así mismo, se lograra el monitoreo y control de la temperatura. Después de un análisis a distintos software y plataformas para la programación de Android, se optó por trabajar bajo APP INVENTOR.

Esta plataforma, nos ofreció la accesibilidad de tener como interfaz una página web, desde la cual se trabajó sin tener a necesidad de hacer la instalación de algún software dentro de nuestro equipo de cómputo. Uno de los requerimientos básicos para trabajar en APP INVENTOR, fue el uso del navegador GOOGLE CHROME, pues sin ello los creadores mencionan que puede haber problemas de compilación de código en el momento de diseñar (anexo 3.6.1).

El diseño de nuestro código, tiene 3 etapas fundamentales:

- Acceso
- Monitoreo y Control
- Ayuda

Para ellos, fue necesario previamente, hacer lectura del uso del interfaz, diagramas de bloque y conceptos básicos, como se puede observar en el punto 5 de la bibliografía.

- \* Acceso

En la primera ventana (Screen1), se vio la necesidad de crear un acceso por contraseña, pues se tuvo el cuidado de crear una aplicación, la cual sea de uso de un usuario asignado, por lo cual si otro usuario tuviera la aplicación y quisiera hacer modificaciones desde otro dispositivo móvil, tendría que conocer la contraseña de la acceso, la dirección mac del Bluetooth, y la contraseña de emparejamiento del Bluetooth.



**Figura.- 8 ventana de bienvenida**

Para ello, podemos observar la figura 8, en la cual se observa nuestra ventana de bienvenida, que consta de los elementos: Textbox, Button, y Label. El código podemos observarlo en la figura 9, donde podemos observar como existe una condición para el acceso a la aplicación, la cual compara el texto escrito en el Textbox, con la contraseña guardada en memoria, de ser errónea, arrojará un mensaje de error, un tono predeterminado y una vibración del teléfono.

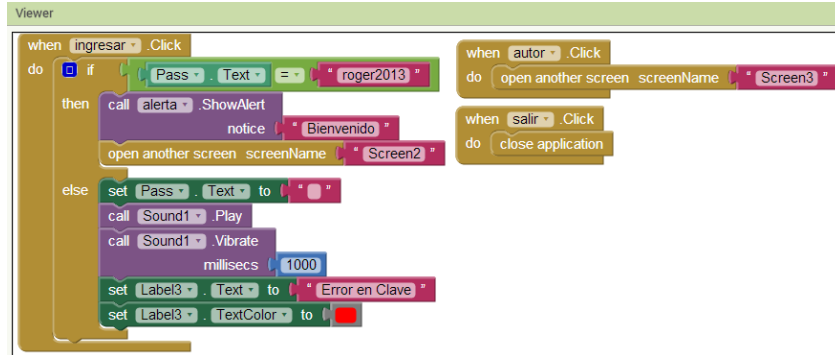


Figura 9.- código de Textbox, Button, y Label

De ser la contraseña establecida, la cual el diseñador puede cambiar en cualquier momento, el móvil emitirá una vibración y posteriormente se dirigirá al Screen2, en el cual se encuentra nuestra ventana de Monitoreo y Control.

## \* Monitoreo y Regulación

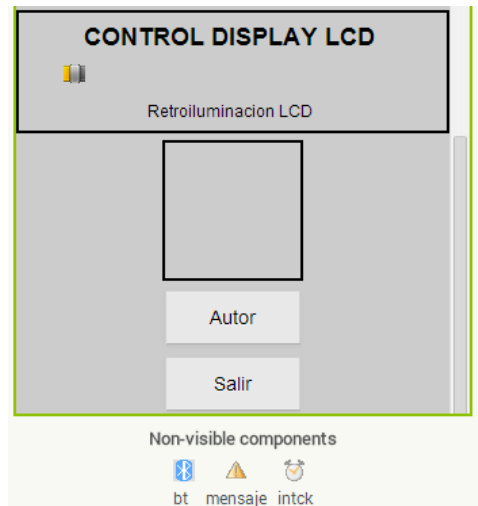
En nuestra ventana de monitoreo y regulación, como podemos ver en la figura 10, tenemos los siguientes elementos: Button, Label, y Slider. Con dichos elementos, logramos crear un screen en el cual tenemos 3 secciones principales: Monitoreo, Control, e Iluminación LED.



Figura 10.- ventana de monitoreo y regulación

Previo a esto, creamos los botones de Conectar y desconectar, los cuales fueron necesarios para poder hacer la conexión entre el dispositivo móvil y Arduino.

Dentro del screen2, observamos como los botones están por cada sección, como en la de monitoreo, tenemos el de iniciar y detener, y en la de regulación, existen botones para incrementar o disminuir la temperatura que se va controlar. Se agregó en el diseño del interfaz, un control de la retroiluminación del LED de la LCD, como se observa en la figura 11, en la cual, se colocó el slider que nos ayudara a controlar la retroiluminación LED.



**Figura 11.- ventana de control LCD**

En la figura 12, podemos observar nuestro código para el screen2, en el cual tenemos dividido nuestro código de bloques debido a la complejidad de nuestras condiciones. En primer lugar, tenemos nuestra declaración de variables, en las cuales se observan 5 de ellas, una correspondiente a la dirección Mac de nuestro Bluetooth, 2 variables para nuestro control y monitoreo de temperatura, una para el control de la retroiluminación del LED, y una cadena, que nos servirá para enviar toda la información en una sola trama de información.



```
initialize global mac to " 20:13:06:14:31:19 "
initialize global d1 to 0
initialize global d2 to 30
initialize global d3 to 128
initialize global cad to " BT: "

when Screen2.Initialize
do
  set desconectar.Enabled to false
  set intck.TimerEnabled to false
  call mensaje.ShowAlert
  notice Verif...
```

Figura 12.- Código para la ventana de control LCD

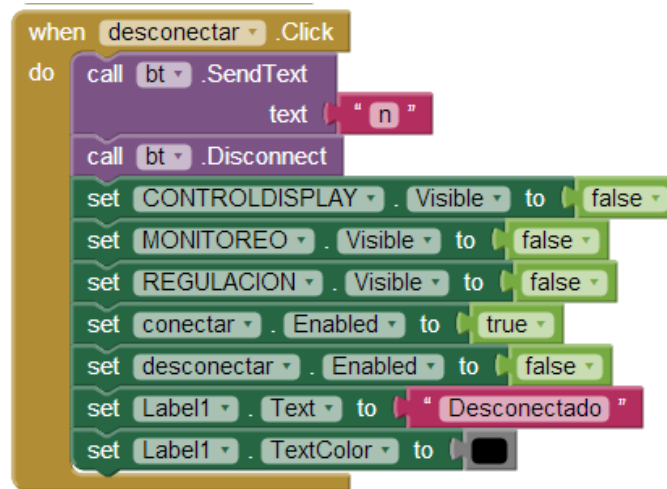
Debajo (figura 13) observamos nuestras condiciones y sentencias para los botones Conectar y Desconectar, en las cuales, hacemos la conexión de Android con Arduino, preguntando si existe una conexión al dispositivo con la dirección MAC requerida, de ser así, el interfaz habilita todos los controles y etiquetas, y envía un carácter 'y', a Arduino, para que Arduino lo interprete y sepa que existe un dispositivo conectado.

```
when conectar.Click
do
  if
    call bt.Connect
    address get global mac
  then
    set Label1.Text to " Conectado "
    set Label1.FontSize to 18
    set Label1.TextColor to blue
    set MONITOREO.Visible to true
    set REGULACION.Visible to true
    set CONTROLDISPLAY.Visible to true
    set conectar.Enabled to false
    set desconectar.Enabled to true
    set detener.Enabled to false
    call bt.SendText
    text " y "
  else
    set Label1.Text to " Error "
    set Label1.TextColor to red
```

Figura 13.- condiciones y sentencias para el botón conectar.

De no ser así, se arroja un mensaje de error y no se habilita ningún control, indicando que posiblemente el dispositivo no esté disponible o se encuentre fuera de rango de alcance del bluetooth.

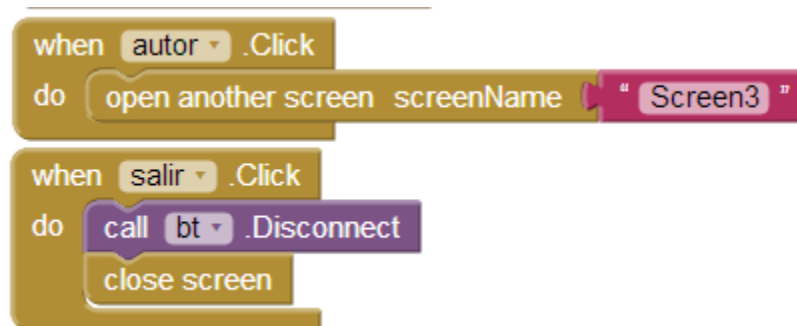
Posteriormente, en la figura 14, tenemos el botón desconectar, que cumple las siguientes sentencias, que se observan en la figura siguiente:



**Figura 14.-sentencias para el botón desconectar**

Cuando se da click en desconectar, primero enviamos un carácter 'n', para que Arduino sepa que ha sido desconectado el dispositivo móvil y pueda continuar con su regulación sin seguir recibiendo datos. así mismo, apagamos todos los elementos del screen2 para que no sean visibles y no puedan ser manipulados.

Los botones de 'autor' y 'salir', cumplen con la función de abrir la ventana del screen3, en el caso de 'autor', y el de salir en cerrar el puerto bluetooth, y cerrar el screen2, para posteriormente regresar a la ventana de inicio.



**Figura 15.- código para las ventanas de Autor y Salir**

## 3.4.1 Monitoreo

En la sección de monitoreo, tenemos 2 botones, que son los de inicio y detener. Con ellos, estamos controlando, tanto la llegada de datos de Arduino, como el momento de decirle a Arduino que deje de enviarnos esa información, dicha información corresponde a las lecturas de los sensores analógicos LM35. Nuestro código se observa en la figura 16.

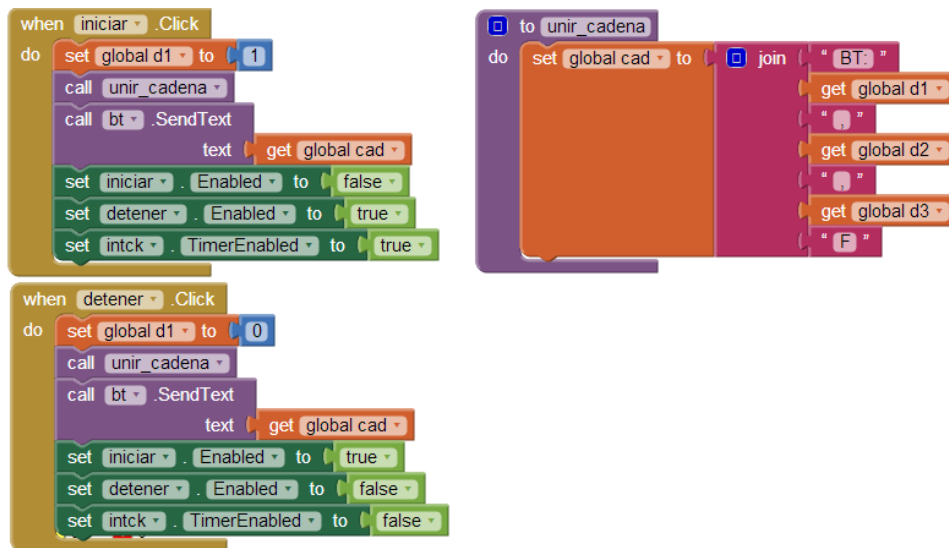


Figura 16.-código de lectura de los sensores analógicos lm35

De lado derecho de la figura 16, observamos la subrutina unir\_cadena, en la cual estamos uniendo arreglos string, que conformaran la cadena que será enviada a Arduino. Se observa que contiene las 3 variables con las que estamos trabajando, una para monitorear la temperatura, una para controlar la temperatura y la otra para el control de la retroiluminación LCD. En la figura 17 se observa el código del timer1, el cual marca la pauta de la lectura de los datos que están llegando de Arduino, y los imprime en pantalla.

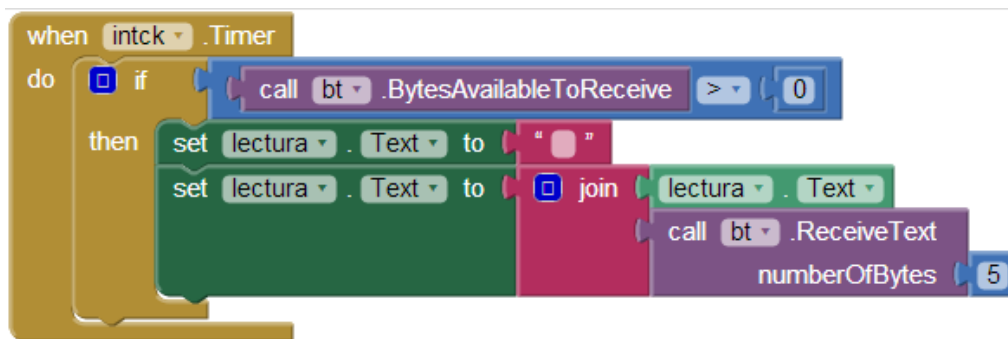


Figura 17.- código del timer1

## 3.4.2 Regulación

En la sección de regulación (figura 18), tenemos presentes 5 botones y un Label. Aquí como podemos observar, 4 botones están siendo usados para hacer el cambio del valor de la temperatura que se desea tener en la incubadora, y un botón que es el de 'Establecer' nos servirá para enviar la variable modificada por el puerto Bluetooth y llegue a Arduino, donde posteriormente se interpretara la cadena para que separe cada dato y haga el control que se está enviando.

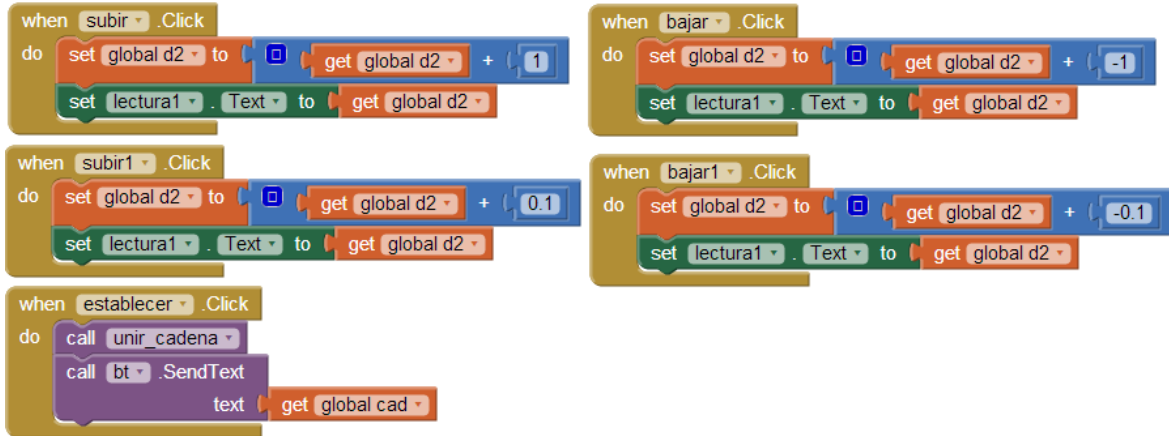


Figura 18.- Regulación

## 3.4.2 Control Retroiluminación LCD

Por último, tenemos el control LCD, en el cual, estamos modificando la retroiluminación LED de la pantalla LCD que tenemos en nuestra incubadora. en ella se observa una única sentencia, en la cual se toma el valor del slider una vez que se ha modificado, y se envía en conjunto con la cadena de caracteres al puerto Bluetooth (figura 19).

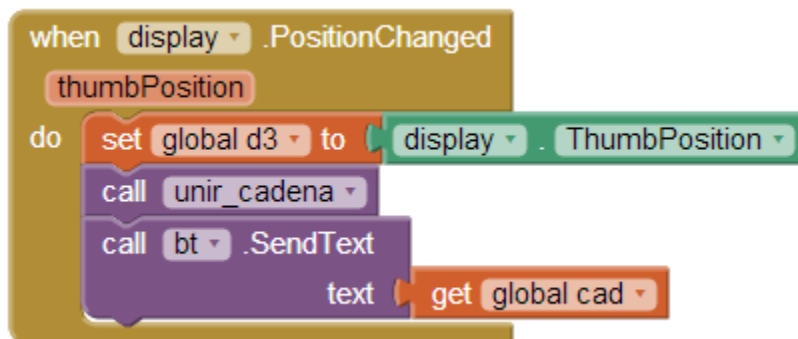
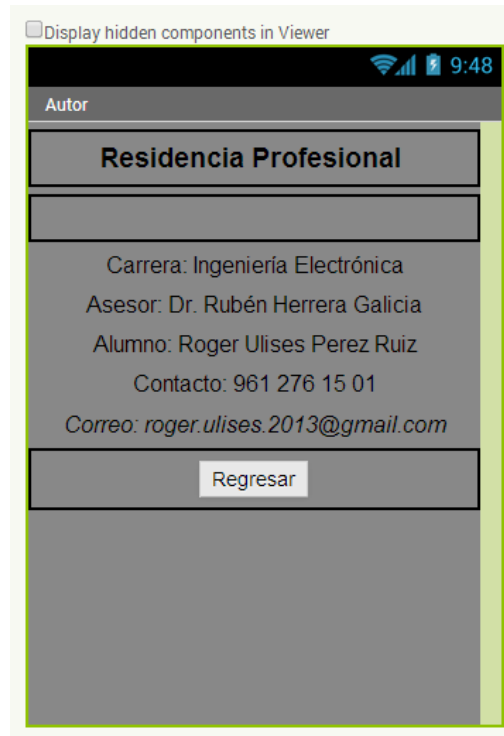


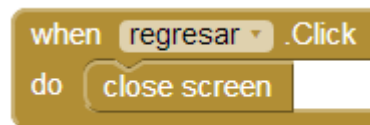
Figura 19.- Control Retroalimentación LCD

## 3.4.4 Ayuda

En el screen3, podemos observar la ventana de ayuda figura 20 (a), la cual solo contiene elementos Label y un único Button, el cual nos hara regresar a la ventana anterior que se haya tenido abierta. Como podemos observar en la figura 20 (b) se encuentra su sentencia, la cual cierra al screen para poder regresar al anterior.



**Figura 20(a).- Ventana de ayuda**



**Figura 20 (b).- sentencia de la ventana de ayuda**

## 3.5 Programación de Arduino y Comunicación

Una vez realizado el programa en Android, se realizó el código en Arduino, en el cual se tomó en cuenta todas las entradas y salidas, puertos de comunicación, librerías, y variables que necesitamos para lograr hacer el monitoreo y control de la temperatura de la incubadora.

En la figura 21, podemos observar el interfaz, en el cual, comenzamos por declarar las librerías que necesitamos para poder controlar nuestro display LCD, y la librería SoftwareSerial, para controlar el dispositivo Bluetooth. Posteriormente se crean los objetos correspondientes a estas librerías, y se configuran dependiendo de en donde conectamos cada componente.



```
bluetooth $
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
LiquidCrystal lcd(8,7,6,5,4,2);
SoftwareSerial bt(11,12);

char dato_bt,dato_serial,cad[25];
float lm35_1,lm35_2,lm35=1,setpoint=20;
int x,retro=128,aux1=0,aux2=0,aux3=0,aux4=0;
byte m0=0,m1=0,cont=0;

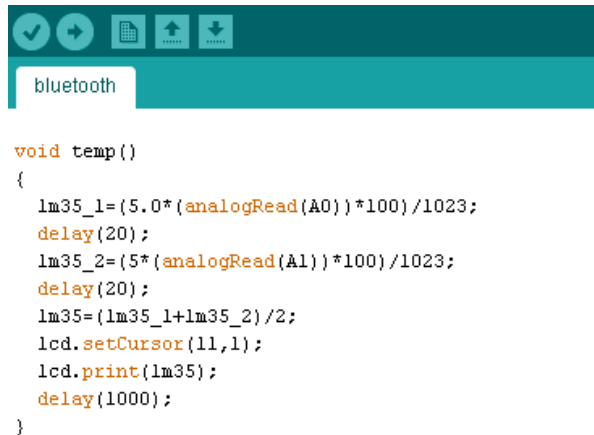
void setup()
{
  Serial.begin(9600);
  bt.begin(9600);
  lcd.begin(16,2);
  lcd.clear();
  //SALIDAS PWM
  pinMode(3,OUTPUT);
  analogWrite(3,128);
  pinMode(9,OUTPUT); //resistencia calorifica 1
  analogWrite(9,0);
  pinMode(10,OUTPUT); //resistencia calorifica 2
  analogWrite(10,0);
  delay(500);
  //MENSAJE DE BIENVENIDA
  welcome();
}
```

Figura 21.-Programación de Arduino y Comunicación

Posterior a ello, se declararon las variables que nos sirvieron para recibir los datos por el puerto bluetooth y serial, para guardar la cadena de caracteres que provenía del dispositivo móvil, variables para guardar los valores que entregaron los sensores analógicos LM35, y variables auxiliares que nos servirían para complementar lecturas, ciclos, y el uso de banderas.

Una vez declaradas las librerías y variables en el encabezado, se configuro el Arduino dentro del void setup(), donde asignamos la velocidad de transmisión de los puertos serial y bluetooth, se configuro la lcd, se asignaron puertos de salida para el control pwm de las resistencias, y se agregó una subrutina que contiene un mensaje de bienvenida. Podemos ver el código ampliamente en el anexo

Una vez realizada dicha parte del código, procedimos a programar la parte más importante de nuestra residencia, el monitoreo y control de la temperatura. Para ello, comenzamos con crear una subrutina en la cual hiciéramos solamente la lectura de los sensores analógicos, se hiciera la conversión de la lectura, y se imprimirá el promedio de esa lectura en la pantalla LCD. Como observamos en la figura 22, ya tenemos el valor de la temperatura de la incubadora almacenada en memoria en la variable llamada lm35.



```
void temp()
{
  lm35_1=(5.0*(analogRead(A0))*100)/1023;
  delay(20);
  lm35_2=(5*(analogRead(A1))*100)/1023;
  delay(20);
  lm35=(lm35_1+lm35_2)/2;
  lcd.setCursor(11,1);
  lcd.print(lm35);
  delay(1000);
}
```

**Figura 22.- subrutina para lectura de los sensores**

Posteriormente, comenzamos con el código que se ejecutara en tiempo real en el Arduino. Desde un principio, tenemos al Arduino haciendo la lectura de los sensores en todo momento, pues es necesario que una vez llegue el usuario a modificar la temperatura a la que se va regular, se tenga conocimiento en todo momento de la temperatura pues el equipo quedara trabajando de forma automática, hasta que se vuelva a realizar otra modificación desde el dispositivo

móvil, por ello, se separó esa parte del código de la lectura de la cadena de caracteres que llegan por el puerto bluetooth.

Si existen datos disponibles en el puerto serial del bluetooth, se guardan en una variable temporal, la cual nos servirá para ir ordenando o comparando los datos que están llegando, dependiendo si pertenecen a la cadena de caracteres o solamente se está enviando una indicación del dispositivo móvil de su estado de conexión o desconexión.

El código nos muestra como al llegar la cadena de caracteres, que traen la información de la temperatura que se desea, la indicación del estado de monitoreo, y el valor de la retroiluminación LED de la pantalla LCD, dicha información es almacenada y separada, esto por medio del símbolo ',' para saber en qué momento termina y empieza el siguiente dato. En base a esto, es que se guardan los caracteres en variables distintas, para poder modificar nuestro setpoint, que nos sirvió para controlar la salida PWM que controla a la resistencia calorífica.

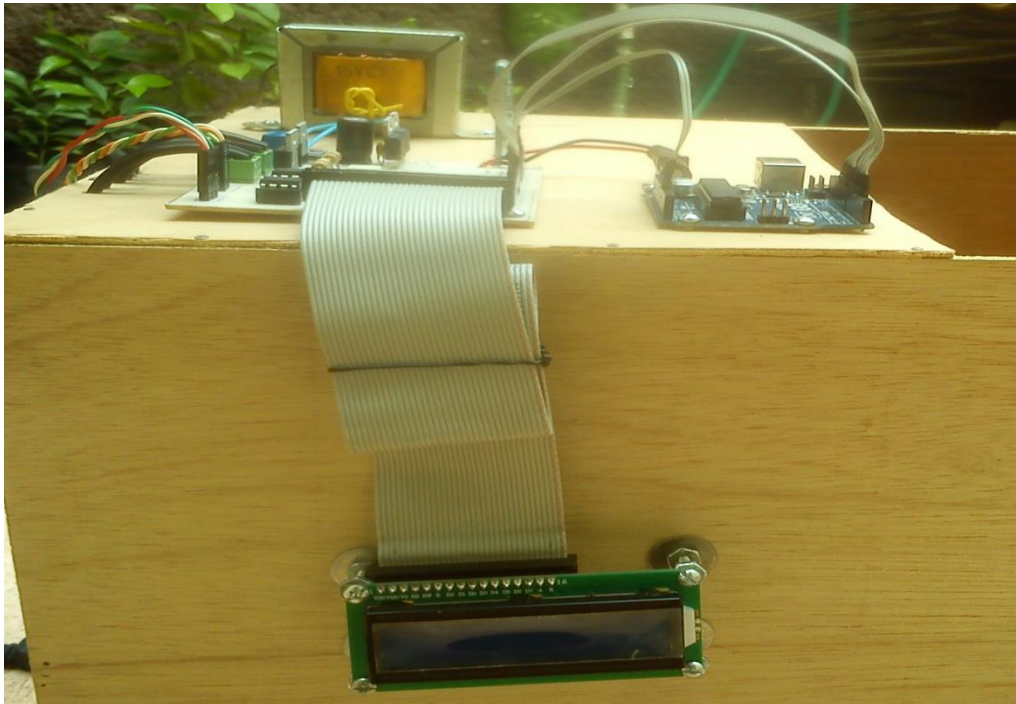


# **CAPITULO IV**

## **RESULTADOS**

### 4.1 Conexiones físicas del sistema

El sistema de monitoreo y regulación de temperatura mediante la conexión inalámbrica bluetooth fue implementado en un prototipo escala de una incubadora (figura 23) donde se realizó la conexión del arduino, LCD, sensores de temperatura así como las resistencias emisoras de calor para la incubación como lo muestra la Figura 24 .



**Figura.- 23 Prototipo a Escala**



## Interiores

- Casa Habitación u Oficina
- Obstáculos y objetos con poca densidad
- Potencia máxima de transmisión
- Módulo Bluetooth con antena unitaria
- Condiciones climatológicas optimas para comunicaciones RF

## Exteriores

- Lugares con línea de vista
- Sin obstáculos y objetos
- Potencia máxima de transmisión
- Módulo Bluetooth con antena unitaria
- Condiciones climatológicas optimas para comunicaciones RF

Ambiente	Distancia
Interiores	7m.
Exteriores	10m.

**Figura 25.- Distancias Efectivas**

## 4.3 Conclusiones

El desarrollo de este sistema de control, propició la incorporación y estudio de tecnologías inalámbricas actuales como es el caso del Bluetooth, así como el desarrollo de software aplicado a plataformas Windows Mobile, logrando el control de la temperatura en incubadoras, así también este sistema puede ser implementado en un proceso industrial ya que la gran mayoría de esta requieren el monitoreo y regulación de temperatura procesos como, hornos, calderas de vapor, saunas, invernaderos pasteurizados etc.

Una mejora en características del sistema podría estar dada con la implementación de módulos Bluetooth de mayor potencia para así alcanzar mayores distancias de operación efectiva.

Finalmente el desarrollo de este proyecto está orientado y fortalecido mediante los conocimientos adquiridos en el estudio profesional en el ámbito de Programación, Electrónica Digital y Sistemas de Comunicación; con la finalidad de desarrollar Sistemas Embebidos en la solución de problemas de la sociedad e incursionar en la investigación y desarrollo tecnológico, fortaleciendo nuestros conocimientos en ingeniería.

## 4.4 Referencia

### **Tecnología Bluetooth**

Serie de telecomunicaciones  
Serie McGraw-Hill de telecomunicaciones  
Autor Nathan J. Muller  
Editor McGraw-Hill, 2002  
ISBN 8448135938, 9788448135935  
N.º de páginas 386 páginas

### *Ingeniería de Control moderna*

Katsuhiko Ogata  
Pearson Educación  
2003 – 965 paginas

### *El gran libro de android*

Jesús Tomas Girones  
Actualizado a la versión 4.2  
3era Edición  
20013 - 431

### *Arduino Curs Práctico de Formación*

Oscar Torrente Artero  
Editorial RC Libros 20013

“ Guía Básica de Arduino” (Documento en Línea),  
<http://www.slideshare.net/hugoangaritaespinosa/libro-kit-básico>,

Sulbaran, Heli, “*Nace la tecnología Bluetooth*”, Diciembre 8 de 2013. (Web en línea), <http://helisulbaran.blogspot.mx/2013/12/07-de-diciembre-2000-nace-la-tecnologia.html>

Plaza, [Exequiel](#), “*La historia del Bluetooth*”, Septiembre 28 de 2011, (Web en línea), <http://www.wayerless.com/2011/09/la-historia-del-nacimiento-de-bluetooth/>,

“Arduino más allá del hardware libre”, (Web en línea),  
<http://www.somoslibres.org/modules.php?name=News&file=article&sid=5581>

*“Placa Arduino Uno”* (Web en línea),  
<http://www.menosmedia.org/spip.php?article43>,

*“Arduino”*, (Web en línea),  
<http://preciosanchez.blogspot.mx/2012/07/arduino.html>,

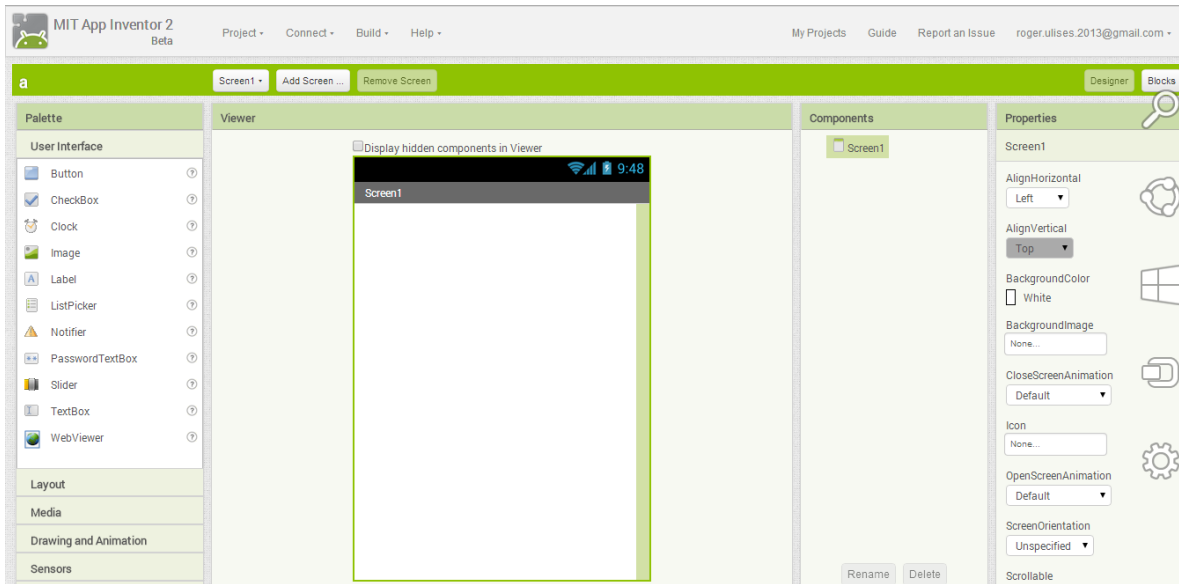
*“Plataformas de desarrollo móviles actuales”*, (Documento en línea),  
<http://es.scribd.com/doc/170157049/Plataformas-de-desarrollo-moviles-actuales-docx>,

*“Que es Android”*, (Web en línea), <http://yosoyandroid.com/diccionario-android/que-es-android-2/>

*“Android”*, (Web en línea), <http://es.wikipedia.org/wiki/Android>,

## ANEXOS

### 4.5.1 figura.- 26 Interfaz APP INVENTOR



## 4.5.2 código Arduino

Figura.- 27 Código Arduino

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
LiquidCrystal lcd(8,7,6,5,4,2);
SoftwareSerial bt(11,12);

char dato_bt,dato_serial,cad[25];
float lm35_1,lm35_2,lm35=1,setpoint=20;
int x,retro=128,aux1=0,aux2=0,aux3=0,aux4=0;
byte m0=0,m1=0,cont=0;

void setup()
{
  Serial.begin(9600);
  bt.begin(9600);
  lcd.begin(16,2);
  lcd.clear();
  //SALIDAS PWM
  pinMode(3,OUTPUT);
  analogWrite(3,128);
  pinMode(9,OUTPUT); //resistencia calorofica 1
```



```
analogWrite(9,0);

pinMode(10,OUTPUT); //resistencia calorifica 2

analogWrite(10,0);

delay(500);

//MENSAJE DE BIENVENIDA

welcome();

}

void welcome()

{

  lcd.clear();

  lcd.print("MONITOREO Y REGULACION DE LA TEMPERATURA");

  delay(500);

  for(x=0;x<=23;x++)

  {

    lcd.scrollDisplayLeft();

    delay(300);

  }

  lcd.clear();

  for(int x=0;x<=15;x++)

  {

    lcd.setCursor(x,1);

    lcd.print(".");
```

```
    delay(65);  
}  
  
lcd.clear();  
  
delay(1000);  
  
lcd.setCursor(1,0);  
  
lcd.print("BLUETOOTH V1.0");  
  
bt.print("AT");  
  
delay(1000);  
  
lcd.setCursor(7,1);  
  
lcd.print(dato_bt=bt.read());  
lcd.print(dato_bt=bt.read());  
  
delay(2000);  
  
lcd.setCursor(0,1);  
  
lcd.print("T:");  
  
lcd.setCursor(2,1);  
  
lcd.print(setpoint);  
  
lcd.setCursor(7,1);  
  
lcd.print(" TR:");  
}  
  
void temp()  
{  
  
    Im35_1=(5.0*(analogRead(A0))*100)/1023;
```

```
delay(20);

lm35_2=(5*(analogRead(A1))*100)/1023;

delay(20);

lm35=(lm35_1+lm35_2)/2;

lcd.setCursor(11,1);

lcd.print(lm35);

delay(1000);

}

void loop()

{

temp();

if(bt.available(>0) //entrada de informacion por bluetooth

{

dato_bt=bt.read();

Serial.print(dato_bt);

//Aqui inicia la cadena de caracteres

if(dato_bt=='B')

{

m0=1;

}

//Fin de cadena de caracteres

if(dato_bt=='F')
```

```
{
  if(cad[7]=='')
  {
    aux1=cad[5]-48;
    aux2=cad[6]-48;
    aux3=cad[8]-48;
    setpoint=(aux1*10)+(aux2*1)+(aux3*0.1);
  }
  else
  {
    aux1=cad[5]-48;
    aux2=cad[6]-48;
    setpoint=(aux1*10)+(aux2*1);
  }
  lcd.setCursor(2,1);
  lcd.print(setpoint);
  cont=0;
  m0=0;
}
```

//Inicio de cadena de caracteres

```
if(m0==1)
{
```

```
cad[cont]=dato_bt;

cont++;

if(cad[3]=='1')
{
    m1=1;

    lcd.setCursor(0,0);

    lcd.print("Monitoreo...");

}

else

{

    if(cad[3]=='0')

    {

        m1=0;

        lcd.setCursor(0,0);

        lcd.print("Esperando...");

    }

}

}

else

{

    if(dato_bt=='y')

    {
```

```
lcd.setCursor(0,0);  
lcd.print("Conectado... ");  
}  
if(dato_bt=='n')  
{  
  lcd.clear();  
  lcd.print("Desconectado...");  
  delay(2000);  
  welcome();  
}  
if(dato_bt=='1')  
{  
  retro=30;  
}  
if(dato_bt=='2')  
{  
  retro=60;  
}  
if(dato_bt=='3')  
{  
  retro=90;  
}  
if(dato_bt=='4')
```

```
{  
    retro=120;  
}  
  
if(dato_bt=='5')  
{  
    retro=150;  
}  
  
if(dato_bt=='6')  
{  
    retro=180;  
}  
  
if(dato_bt=='7')  
{  
    retro=210;  
}  
  
if(dato_bt=='8')  
{  
    retro=240;  
}  
  
if(dato_bt=='9')  
{  
    retro=255;  
}
```

```
}  
  analogWrite(3,retro);  
}  
  
//Envio de informacion de Arduino a Dispositivo Móvil  
  
if(m1==1)  
{  
  bt.print(lm35);  
}  
  
if(lm35<=setpoint)  
{  
  analogWrite(9,1);  
}  
  
else  
{  
  analogWrite(9,0);  
}  
}
```