



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE
MÉXICO

CENTRO UNIVERSITARIO UAEM ECATEPEC

**Desarrollo de una Impresora Flexográfica
Automatizada con Microcontroladores**

T E S I S

QUE PARA OBTENER EL GRADO DE:
Maestra en Ciencias de la Computación

PRESENTA:
Ing. Verónica Garnica Acosta

TUTOR ACADÉMICO:

DRA. TERESA IVONNE CONTRERAS TROYA

TUTORES ADJUNTOS:

DR. ISMAEL DÍAZ RANGEL

DR. FARID GARCÍA LAMONT

ECATEPEC DE MORELOS, ESTADO DE MÉXICO A 8 DICIEMBRE 2016

Índice general

Índice de figuras	IX
Índice de cuadros	XIII
1. Introducción	1
Definición del problema	3
Justificación	4
Pregunta de Investigación	5
Objetivo general	5
Objetivos particulares	5
Metodología Sistemas Suaves	6
Esquema de tesis	8
2. Marco teórico	11
2.1. Microcontrolador	11
2.2. Arduino	12
2.2.1. Programación	14
2.3. Relevador	16
2.3.1. Estructura del Relé	16
2.3.2. Características del Relé	17
2.3.3. Tipos de Relés	18
2.3.3.1. Relés Electromecánicos	18
2.3.3.2. Relés Polarizados	19
2.3.3.3. Relé de Estado Sólido	20
2.4. Bluetooth	21
2.4.1. Módulo HC-05	21

2.4.2. Módulo HC-06	22
2.5. Android	23
2.6. APP Inventor	24
3. Análisis de la situación actual	31
3.1. Etapa 1 Situación problema no estructurada	31
3.2. Etapa 2 Situación estructurada	34
3.3. Etapa 3 Definiciones básicas	37
3.3.1. CATWOE	40
4. Planeación de la Impresora	45
4.1. Etapa 4 Modelos conceptuales	45
4.1.1. Etapa 4a Modelación de sistemas formales	46
4.1.2. Etapa 4b Monitorear un sistema	54
5. Desarrollo	61
5.1. Programación de Arduino	61
5.1.1. Impresión automática	63
5.1.2. Impresión manual	64
5.2. Código fuente del microcontrolador	67
5.2.1. Declaración de bibliotecas y variables globales	67
5.2.2. Función setup	68
5.2.3. Funciones	68
5.3. Programación App Inventor	73
6. Comparación de modelos	87
6.1. Etapa 5 Comparar modelos conceptuales con realidad	87
6.2. Opiniones	87
7. Conclusiones	95
A. Diagramas de Flujo	97
A.1. Arduino	97
A.2. App Inventor	106
B. Manual de instalación	121

C. Manual de operación	125
D. Participaciones	133
Referencias	137

Índice de figuras

1.1. Metodología Sistemas Suaves de Peter Checkland (SSM)	6
2.1. Placa Arduino y Cable de conexión	13
2.2. Estructura de los Relés	18
2.3. Relé polarizado	20
2.4. Relé de Estado Sólido	20
2.5. Diagrama de Red inalámbrica vía Bluetooth	21
2.6. Módulo Bluetooth HC-05	22
2.7. Módulo Bluetooth HC-06	22
2.8. Sistema de capas de Android	24
2.9. Diagrama de funcionamiento de App Inventor	26
2.10. Sentencias en App inventor	26
2.11. Condiciones en App Inventor	27
2.12. Bucles App Inventor	28
2.13. Variables en App Inventor	28
2.14. Eventos para objetos en App Inventor	29
3.1. Situación original no estructurada.	33
3.2. Situación estructurada actual	35
3.3. Encuesta No. 1	38
3.4. Encuesta No. 2	39
3.5. Diagrama CATWOE	41
3.6. Despliegue de Casa de la Calidad (QFD)	43
3.7. Definición básica.	44
4.1. Modelo Conceptual	45

4.2.	Diagrama de desarrollo de la impresora flexográfica.	50
4.3.	Módulo 4 Relay	53
4.4.	Situación estructurada.	55
4.5.	Representación gráfica del problema	56
4.6.	Ejemplo de código Arduino	58
4.7.	Compilación del programa	58
4.8.	Programa compilado	59
4.9.	Compilando el programa	59
4.10.	Programa subido a la placa Arduino	60
5.1.	Diagrama a bloques de la impresora	62
5.2.	Diagrama a bloques de la Función Automática	63
5.3.	Diagrama a bloques de la Función Carga de tinta	64
5.4.	Diagrama a bloques de la Función de Impresión	65
5.5.	Diagrama a bloques de la Función de Finalizar Impresión	66
5.6.	Diseño de Interfaz	73
5.7.	Variables globales	73
5.8.	Programación Slider	74
5.9.	List Picker lpConectar antes	77
5.10.	List Picker lpConectar después	77
5.11.	Botones Desconectar y Automático	78
5.12.	Botón bIniciar	79
5.13.	Clock 5 y 6	80
5.14.	Clock 5 y 6	81
5.15.	Botón bCargaTinta	82
5.16.	Clock 2 y 3	83
5.17.	Botón bImprime	84
5.18.	Botón bFin	84
5.19.	Bloques totales para programación	85
5.20.	Aplicación final	85
6.1.	Opinión 1	89
6.2.	Opinión 2	91
6.3.	Opinión 3	93
6.4.	Opinión 4	94
A.1.	Diagrama de Flujo Programa Final parte 1	99

A.2.	Diagrama de Flujo Programa Final parte 2	100
A.3.	Diagrama de Flujo función Carga de Tinta parte 1	101
A.4.	Diagrama de Flujo función Carga de Tinta parte 2	102
A.5.	Diagrama de Flujo función de imprime parte 1	103
A.6.	Diagrama de Flujo función de imprime parte 2	104
A.7.	Diagrama de Flujo función de Serial flush y Fin	105
A.8.	Diagrama de Flujo listPicker lpConectar	109
A.9.	Diagrama de Flujo botón Automático	110
A.10.	Diagrama de Flujo botón Iniciar parte 1	111
A.11.	Diagrama de Flujo botón Iniciar parte 2	112
A.12.	Diagramas de flujo para timer Clock 5 y 6	113
A.13.	Diagrama de Flujo botón Desconectar	114
A.14.	Diagramas de flujo para timer Clock 2 y 3	115
A.15.	Diagrama de Flujo botón Carga de Tinta	116
A.16.	Inicialización de variables globales	117
A.17.	Diagramas de flujo para botón bCancelar y timer Clock 4	118
A.18.	Diagrama de Flujo botón Imprime	119
A.19.	Diagrama de Flujo botón Fin	120
B.1.	Instalación de impresora.apk	123
B.2.	Emparejar Bluetooth	124
C.1.	Conectando al Bluetooth Impresora Flexográfica	126
C.2.	Opción Automático opción completa acción	127
C.3.	Opción Automático opción Cancelar impresión	128
C.4.	Opción Manual Carga de tinta 1 y/o 2 veces	129
C.5.	Opción Manual Carga de tinta más de 2 veces	130
C.6.	Opción Manual Imprime y Finaliza impresiones	131
C.7.	Desconectar Bluetooth	131
D.1.	Semana Multidisciplinaria	134
D.2.	Seminario de investigación	135
D.3.	1er Coloquio de investigación	136

Índice de cuadros

1.1.	Descripción de etapas Metodología de Sistemas Suaves Checkland (SSM) . . .	7
2.1.	Tipos de Arduino	15
2.2.	Aplicaciones del Relé	16
2.3.	Relés electromecánicos	19
2.4.	Ventajas y Desventajas para usar APP Inventor	25
3.1.	Definición de agentes.	32
3.2.	Agentes de la impresora flexográfica automatizada con PLC.	36
3.3.	Descripción de los elementos del CATWOE.	40
4.1.	Modelo conceptual solución 1. Picaxe	46
4.2.	Modelo conceptual solución 2. Pic	47
4.3.	Modelo conceptual solución 3. Arduino	48
4.4.	Comparativa entre microcontroladores.	49
4.5.	Comparativa entre Bluetooth.	51
4.6.	Modelo de conectividad Bluetooth	52
4.7.	Especificaciones del 4 Relay	53
4.8.	Funciones de la impresora flexográfica programada en App Inventor.	57
6.1.	Comparación entre impresoras flexográficas	88

Capítulo 1

Introducción

El enfoque sistémico identifica los diferentes aspectos de un problema y permite que el analista pueda estudiarlo y evaluarlo con facilidad. La idea de ver un problema en el contexto de un sistema que posee componentes interconectados requiere la aplicación de técnicas específicas (cuantitativas, administrativas, económicas, tecnológicas, sociales) organizadas. En particular, los sistemas socio técnicos buscan la optimización conjunta de los sistemas técnico y social, lo que conlleva a resolver problemas reales con el uso de herramientas metodológicas y tecnológicas para el análisis, diseño, planeación y administración de la solución.

El Análisis de Sistemas trata de determinar los objetivos y límites del sistema objeto de análisis, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar sus consecuencias. Dependiendo de los objetivos del análisis se pueden tener dos vertientes:

- Análisis de un sistema ya existente para comprender, mejorar, ajustar y/o predecir su comportamiento.
- Análisis como paso previo al diseño de un nuevo sistema-producto.

Partiendo de lo anterior, el presente trabajo mejora un Entrenador Flexográfico Automatizado con PLC (Controlador Lógico Programable) [1],

utilizado actualmente en los Centros de Estudios Tecnológico Industrial y de Servicios (CETis) 119 y 165, de la Dirección General de Educación Tecnológica Industrial (DGETI).

Para desarrollar el nuevo diseño de la impresora flexográfica se tomaron en cuenta las necesidades de los usuarios, de esta forma, se cumplió con las características de calidad que conllevan a la satisfacción de éstos, al mismo tiempo se tuvo un aprendizaje en el desarrollo y mejora de la impresora y se desarrollaron habilidades aprendidas durante la maestría en Ciencias de la Computación del CU UAEM Ecatepec.

La impresora flexográfica opera con un PLC para la potencia y el control; y para su operación incorpora un panel con interruptores. La propuesta sustituye el PLC por relevadores para la gestión de potencia, un microcontrolador para la gestión de procesos y un teléfono inteligente con sistema operativo Android como interfaz de operación de la impresora.

Adicionalmente, se agregó un módulo Bluetooth HC-06 (modo maestro), una vez vinculados y haciendo uso de una aplicación creada ex profeso, se podrán realizar algunas de las siguientes funciones: Automático, Carga del rodillo de tinta, Imprime material o Finalizar impresión.

Definición del problema

Los Centros de Estudios Tecnológico Industrial y de Servicios (CETis) son bachilleratos tecnológicos que cubren la demanda de los jóvenes egresados de secundaria, así como a quienes requieren cursar una carrera técnica para incorporarse al mercado laboral, que dependen de la Dirección General de Educación Tecnológica Industrial (DGETI) a nivel nacional. El plan de estudios incluye módulos para la formación tecnológica que los prepara como técnicos del nivel medio superior en algunas de las carreras técnicas que se ofertan.

En el CETis 119 se imparte la carrera de Técnico en Producción de Prendas de Vestir y Técnico en Diseño de Modas, mientras que en el CETis 165 se imparten las carreras de Técnico en Artes Gráficas y Técnico en Sistemas de Impresión, en ambos planteles se debe usar una impresora flexográfica profesional (ésta imprime el diseño en el material a ocupar); el costo de las impresoras profesionales oscila de \$118,000.00 a \$950,000.00, con tamaños aproximados de 23m*17m*25m y con un peso de hasta 2200 kg, debido a sus altas dimensiones y costos, los planteles no pueden adquirir el equipo de impresión. Por lo cual, en el año 2014, se implementó un entrenador de impresión flexográfica automatizado con Controlador Lógico Programable (PLC), portátil y de bajo costo [1].

Sin embargo, cualquier usuario puede modificar la configuración que tiene el PLC y esto ocasionaría un daño al entrenador, entonces, se desea proteger el circuito de tal forma que no se dañe con facilidad y sea de mayor durabilidad. Por lo tanto, se propone el desarrollo de una impresora flexográfica automatizada con microcontroladores para utilizar esta impresora con fines académicos.

Justificación

Los microcontroladores a comparación de un PLC son de menor costo, se pueden implementar de diferente forma, y su reprogramación puede ser más simple. Además que algunos modelos incorporan gran diversidad de periféricos, como módulos de comunicación serial, convertidores analógicos digitales, etc.

Con base en esto se pretende modificar la impresora flexográfica que utiliza un PLC, por una aplicación de smartphone para su funcionamiento, la cual ayudará a mejorar el desarrollo académico del alumno, facilitando la operación de dicha impresora.

Se podrá implementar la impresora tanto en escuelas públicas como en microempresas, por su bajo costo económico y tamaño para una oficina.

Pregunta de Investigación

¿Es posible realizar mejoras al entrenador de impresión flexográfica implementada con PLC, utilizando para las etapas de control y potencia un microcontrolador y relevadores, y para la interfaz de operación utilizar un teléfono inteligente?.

Objetivo general

Realizar los procesos de control de una impresora flexográfica portátil mediante el uso de un microcontrolador y relevadores, en sustitución de un PLC; y diseñar e implementar una aplicación para teléfono inteligente, que funcione como interfaz de operación de la impresora.

Objetivos particulares

- Comprobar que se puede hacer uso de la Metodología de Sistemas Suaves de Checkland en ramas de la ingeniería.
- Analizar la necesidades del cliente final para el desarrollo de la impresora.
- Diseñar y programar una interfaz gráfica para la interacción hombre-máquina del sistema de impresión.
- Automatizar funciones de Automático, Manual (Carga de tinta, impresión y Finalizar) con el uso de programación de un microcontrolador.
- Optimizar el espacio de la impresora flexográfica.

Metodología de Sistemas Suaves de Checkland

La Metodología de Sistemas Suaves (SSM, por siglas en inglés) es una técnica cualitativa que se puede utilizar para aplicar los sistemas no estructurados. Es una manera de ocuparse de problemas situacionales en los que hay una actividad con un alto componente social, político y humano.

El utilizar la SSM da la estructura a las situaciones y complejidades del problema, permite la organización del problema, ayuda a buscar solución o soluciones que sean más que sólo técnicas, ayudando al componente social Figura 1.1.

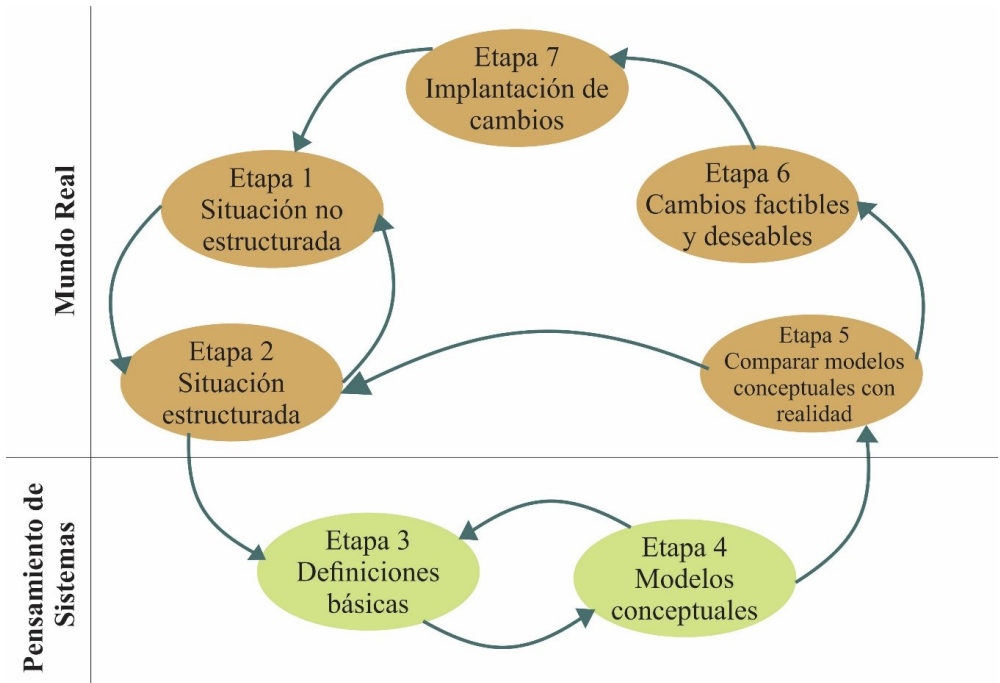


Figura 1.1: Metodología Sistemas Suaves de Peter Checkland (SSM)
Fuente: Elaboración propia basada en [2].

La SSM consta de 7 etapas como se muestra en el Cuadro 1.1.

Cuadro 1.1: Descripción de etapas Metodología de Sistemas Suaves Checkland (SSM)

Nombre de la Etapa	Descripción								
Etapa 1 Investigar el problema no estructurado	Decidir que agentes intervienen durante el proceso para la realización, es decir, especificando las herramientas que se utilizaron y quienes son los beneficiados.								
Etapa 2 Situación estructurada	Recoger y clasificar la información que provee la descripción de la situación del problema. La estructura de la organización, los factores que no cambian con facilidad. Procesos y transformaciones que se realizan dentro del sistema obteniendo la visión rica de lo recabado en la etapa 1.								
Etapa 3 Definiciones básicas.	Expresar el propósito central del sistema, las diferentes vistas de cada persona. Entra en esta etapa la realización del CATWOE, mostrando aquí lo positivo y negativo, del agente principal y cuál es su intervención durante el proceso.								
	C Clientes Receptores de la transformación de sus productos o servicios.								
	A Actores Encargados de hacer la transformación con los medios que se disponen.								
	T Transformación Es el proceso propiamente dicho mediante el cual realiza su función básica.								
	W Weltanschauung Visión rica, es la esencia o razón de ser, la necesidad que se busca satisfacer.								
	O Owner El suprasistema, el sistema que tiene poder de decisión e influencia y que puede causar que la organización exista o no.								
E Environment El medio ambiente en el que él sistema opera.									
Etapa 4 Modelos conceptuales.	<p>Ideas que se utilizaron, aplicadas con base a la metodología y llevadas a lo que percibe el cliente, el cual aportó con ideas para el sistema.</p> <table border="1"> <tr> <td>Modelación de Sistemas Formales.</td> <td>Una guía de consulta para el control del modelo conceptual ya trazado.</td> </tr> <tr> <td>Monitorear un sistema.</td> <td>La medida del funcionamiento del sistema, con los recursos y eficacia, para logro de metas satisfechas.</td> </tr> </table>	Modelación de Sistemas Formales.	Una guía de consulta para el control del modelo conceptual ya trazado.	Monitorear un sistema.	La medida del funcionamiento del sistema, con los recursos y eficacia, para logro de metas satisfechas.				
Modelación de Sistemas Formales.	Una guía de consulta para el control del modelo conceptual ya trazado.								
Monitorear un sistema.	La medida del funcionamiento del sistema, con los recursos y eficacia, para logro de metas satisfechas.								
Etapa 5 Comparar modelos conceptuales con realidad.	<p>Los modelos conceptuales construidos en la etapa 4 son comparados con la expresión de lo real al cliente, de la etapa 2; puede reiterar a la etapa 3 y 4. Definir cuál es el fin de la etapa 4 y en qué momento deberá ser comparada. Para esta comparación se siguen las siguientes opciones:</p> <table border="1"> <tr> <td>5.1</td> <td>Usar modelos conceptuales como base para cuestionamientos ordenados.</td> </tr> <tr> <td>5.2</td> <td>Comparar historia con predicción del modelo.</td> </tr> <tr> <td>5.3</td> <td>Comparación total general.</td> </tr> <tr> <td>5.4</td> <td>Recubrimiento modelo.</td> </tr> </table>	5.1	Usar modelos conceptuales como base para cuestionamientos ordenados.	5.2	Comparar historia con predicción del modelo.	5.3	Comparación total general.	5.4	Recubrimiento modelo.
5.1	Usar modelos conceptuales como base para cuestionamientos ordenados.								
5.2	Comparar historia con predicción del modelo.								
5.3	Comparación total general.								
5.4	Recubrimiento modelo.								
Etapa 6 Cambios factibles y deseables.	Resultados del sistema ejecutado, que el microcontrolador utilizado es eficaz y deseables a los cambios realizados. Si existen cambios para el mejoramiento del sistema se realizan.								
Etapa 7 Acciones para mejorar la situación-problema.	Implementar el diseño, donde los actores del sistema deberán aprobar si este soluciona la situación problemática o se realizan cambios al sistema.								

Fuente: Elaboración propia basada en [2].

Esquema de tesis

El presente trabajo se divide en siete capítulos, su desarrollo se realizó mediante la **Metodología de Sistemas Suaves de Checkland** de la siguiente forma:

Para la primera etapa se llevó a cabo la investigación del desarrollo de una impresora flexográfica automatizada con microcontroladores, en la segunda etapa se hizo un análisis exhaustivo para determinar los requerimientos necesarios a dicho proyecto, en la tercera etapa se realiza la práctica, es decir, la implementación física del circuito y la aplicación al dispositivo móvil, en la cuarta etapa se realiza la prueba y error de la implementación.

En el Capítulo 1. Se desglosa la Introducción acerca del proyecto, se realizó la definición del problema. Se tiene la pregunta de investigación, objetivo general y particulares. Se describe la Metodología a utilizar de Sistemas Suaves de Checkland y cada una de sus etapas.

En el Capítulo 2. Marco teórico se desarrollan los temas referentes a Microcontrolador, Arduino, algunos tipos de arduinos para su programación, características, estructuras y tipos de Relevadores, los Bluetooth que se puede utilizar para Arduino, Android y sentencias, condiciones, bucles, variables y eventos para programar en App Inventor.

En el Capítulo 3. Se tienen las etapas 1, 2 y 3 de la metodología de Checkland, se realiza el diagrama CATWOE y se lleva a cabo un Despliegue de la casa de la Calidad, como herramienta para saber las necesidades de los usuarios.

En el Capítulo 4. Se realizó la planeación para la impresora al llevar a cabo la Etapa 4 de la SSM. Se hace una comparación entre los diferentes elementos que pueden ser utilizados para el desarrollo y modificación de la flexográfica.

En el Capítulo 5. Se llevó a cabo el Desarrollo y la programación de todos los elementos nuevos de la impresora.

En el Capítulo 6. Se realizan la Etapa 5 de la SSM. Se comparan las impresoras, es decir, se compara la Impresora Flexográfica Automatizada con PLC con la Impresora Flexográfica implementada con microcontrolador Arduino.

En el Capítulo 7. Se utilizan las Etapas 6 y 7 de la SSM, donde se describen los elementos que le hacen falta a la impresora o bien aquellos que se hagan obsoletos. Se mencionan qué trabajos a futuro puede realizarse y la implementación de la impresora.

Capítulo 2

Marco teórico

2.1. Microcontrolador

El microcontrolador es un circuito integrado digital que puede ser usado para diversos propósitos debido a que es programable. Está compuesto por una unidad central de proceso (CPU), memorias (ROM y RAM) y líneas de entrada y salida (periféricos). Para usar un microcontrolador se debe especificar su funcionamiento por software a través de programas que indiquen las instrucciones que realizará [3].

El microcontrolador contiene las partes funcionales de una computadora:

- CPU (Central Processor Unit o Unidad de Procesamiento Central).
- Memorias volátiles Random Access Memory Memoria de Acceso Aleatorio (RAM), para datos.
- Memorias no volátiles (ROM, PROM, EPROM, EEPROM) para escribir el programa.
- Líneas de entrada y salida (I/O) para comunicarse con el mundo exterior.
- Algunos periféricos (comunicación serial, temporizador, convertidor A/D, etc).

Algunas aplicaciones que utilizan microcontroladores incluyen artefactos domésticos, sistemas de alarma, equipo médico, subsistemas de automóviles, equipo de electrónica. Los microcontroladores son comprados en “blanco” y luego son programados con un programa específico de control. Una vez programado, el microcontrolador es introducido en algún producto para hacerlo más inteligente y fácil de usar.

Un microcontrolador puede a menudo reemplazar a un gran número de partes separadas, o incluso un circuito electrónico completo. Algunas de las ventajas orientadas con el uso de microcontroladores en el diseño de productos son:

- Aumento en la confiabilidad debido al menor número de partes.
- Reducción en los niveles de existencia ya que un microcontrolador reemplaza varias partes.
- Simplificación del ensamblaje del producto y productos finales más pequeños.
- Flexibilidad y adaptabilidad del producto ya que las funciones del producto están programadas en el microcontrolador y no incorporadas en el hardware electrónico.

Los microcontroladores que se pueden utilizar son: PIC, PICAXE y la familia del microcontrolador ATmega que presenta desde 28 hasta 100 pines en la forma de Dual Inline Package (DIP), Thin Profile Plastic Quad Flat Pack (TQFP) 7mmx7mm y Micro Lead frame Package (MLF) 5mmx5mm.

2.2. Arduino

Arduino es una herramienta la cual hace que puedan sentir y controlar el mundo físico a través del ordenador personal. Es una plataforma de desarrollo de computación física de código abierto, basada en una placa con un microcontrolador y un entorno de desarrollo para crear software para la

placa.

Se puede usar Arduino para la creación de objetos interactivos, leer datos de gran variedad de interruptores y sensores, controlar multitud de tipos de luces, motores y otros actuadores físicos. El lenguaje de programación de Arduino es una implementación de Wiring, una plataforma de computación física parecida, que a su vez se basa en Processing, un entorno de programación multimedia. Los proyectos de Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en el ordenador (como Flash, Processing, MaxMSP). La placa puede ser montada por uno mismo (armarla) o bien comprarla lista para ser utilizada y con respecto al desarrollo del software es de código abierto.

El Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

En la Figura 2.1 se muestra el Arduino Uno con el microcontrolador de la familia ATmega328 y el cable de conexión para la computadora.

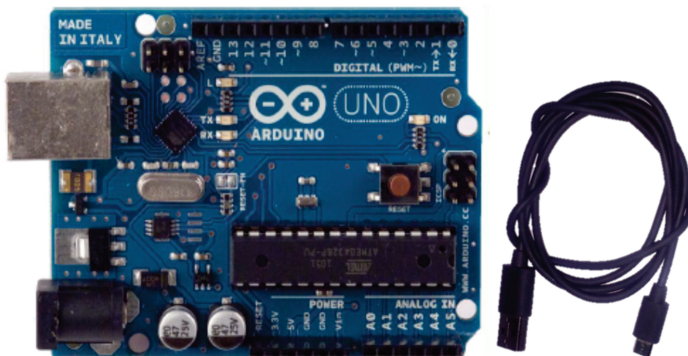


Figura 2.1: Placa Arduino y Cable de conexión
Fuente: Elaboración propia

En el Cuadro 2.1 se muestran algunos de los tipos de Arduino que existen en el mercado comercial. También se describen algunas de las características

principales para las placas de Arduino. Un hercio es la frecuencia de un suceso o fenómeno repetido una vez por segundo. Memoria RAM se guardan los datos que se están utilizando en el momento presente, su almacenamiento es considerado temporal, los datos permanecen en ella mientras la memoria tiene una fuente de alimentación. La memoria EEPROM es de sólo lectura, programable y borrable eléctricamente. La memoria FLASH es no volátil, de bajo consumo, puede grabarse y borrarse eléctricamente. Los pines digitales son la cantidad de pines de (I/O). El voltaje al que regularmente trabaja es la Tensión, mientras que dependiendo del voltaje es la corriente con la que trabaja.





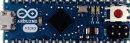
Del Microcontrolador de la familia ATmega se encuentran ATmega168 ATmega328, ATmega32u3.

En Arduino los pines analógicos pueden ser usados también como pines digitales I/O. Una interrupción es una señal que suspende la actividad normal del procesador y salta a atenderla estas pueden ser (evento hardware, previamente definido, evento programado, o Timer y una llamada por software) salta a ejecutar una función especial que se llama Interrupt Service Handler o Servicio de gestión de interrupción (ISH). La modulación de ancho de pulso (PWM), es una modulación que mantiene constante la frecuencia. La conexión de USB es un cable para cargar el programa a la placa y ejecutarlo. Al dispositivo de Arduino se le puede agregar un Bluetooth y relevadores para control de aplicación desde el dispositivo móvil y control de motores.

2.2.1. Programación

El lenguaje Arduino (basado en cableado) se implementa en C / C ++, y por lo tanto tiene algunas diferencias con el lenguaje de procesamiento, que se basa en Java. Se puede enlazar mediante Bluetooth a dispositivos móviles y tabletas, que cuenten con programación de Android.

Cuadro 2.1: Tipos de Arduino

Características	Arduino Pro Mini	Arduino Uno	Arduino Nano	Arduino Buono Uno R3	Arduino Micro
Figura					
Microcontrolador	AVR ATmega168 o 328	AVR ATmega328	AVR ATmega168 o 328	AVR ATmega328 TQFP-32	AVR ATmega32u3
Frecuencia (MHz)	16	16	16	16	16
Memoria RAM (KB)	2	2	2	2	2.5
Memoria EE-PROM (KB)	1	1	1	1	1
Memoria FLASH (KB)	16 o 32	32	16 o 32	32	32
Pines digitales I/O	14	14	14	14	20
Tensión(V) corriente (mA/A)	3.3 o 5, 40	5, 40	5, 40	3.3 o 5, 300 o 2	5, 40
Pines analógicos	6	8	8	8	12
Pines con interrupción externa	2	2	2	2	2
Pines PWM	6	6	v 6	6	7
Conexión USB	Necesita adaptador externo	Si, USB	Si, USB	Si, Nativa, MicroUSB	Si, Nativa, MicroUSB

Fuente: Elaboración propia.

2.3. Relevador

Un relé es un interruptor accionado por un electroimán, permiten abrir o cerrar circuitos eléctricos. Un electroimán está formado por una barra de hierro dulce, llamada núcleo, rodeada por una bobina de hilo de cobre. Al pasar corriente eléctrica por la bobina, el núcleo de hierro se magnetiza por efecto del campo magnético producido por la bobina, convirtiéndose en un imán tanto más potente cuanto mayor sea la intensidad de la corriente y el número de vueltas de la bobina. Al abrir de nuevo el interruptor y dejar de pasar corriente por la bobina, desaparece el campo magnético y el núcleo deja de ser un imán. Se utiliza en aplicaciones como se muestra en el cuadro 2.2.

Cuadro 2.2: Aplicaciones del Relé

Aplicación	Descripción
Automatismo	El componen eléctrico que activa los motores de una puerta automática, las luces de un semáforo, un ascensor, un secador de manos y multitud de otros sistemas automáticos.
Control de motores eléctricos industriales	Utilizados para poner en marcha, para y cambiar el sentido de giro inclusive la velocidad de los motores que mueven máquinas industriales diversas.
Computadoras (Ordenadores)	Los primeros ordenadores funcionaban con relés como base para realizar cálculos matemáticos. Posteriormente fueron sustituidos por válvulas de vacío y más tarde por transistores miniaturizados en un chip de silicio.

Fuente:Elaboración propia.

2.3.1. Estructura del Relé

Un relé está constituido por los siguientes bloques:

- Circuito de entrada, control o excitación
- Cicuito de acoplamiento

- Circuito de salida, carga o maniobra, constituido por: circuito excitador (dispositivo conmutador de frecuencia, protecciones).

2.3.2. Características del Relé

- El aislamiento entre los terminales de entrada y de salida
- Adaptación sencilla a la fuente de control
- Posibilidad de soportar sobrecargas, tanto en el circuito de entrada como en el de salida
- Las dos posiciones de trabajo en los bornes de salida de un relé se caracterizan por: Estado abierto (alta impedancia) y estado cerrado (baja impedancia)

Para los relés de estado sólido se pueden añadir:

- Gran número de conmutaciones y larga vida útil
- Conexión en el paso de tensión por cero, desconexión en el paso de intensidad por cero
- Ausencia de ruido mecánico de conmutación
- Escasa potencia de mando, compatible con TTL y MOS
- Insensibilidad a las sacudidas y a los golpes.

2.3.3. Tipos de Relés

Un relé es un sistema mediante el cual se puede controlar una potencia mayor con un consumo en potencia reducido algunos tipos son: electromecánicos, polarizados y de estado sólido, como se puede observar en la Figura 2.2.

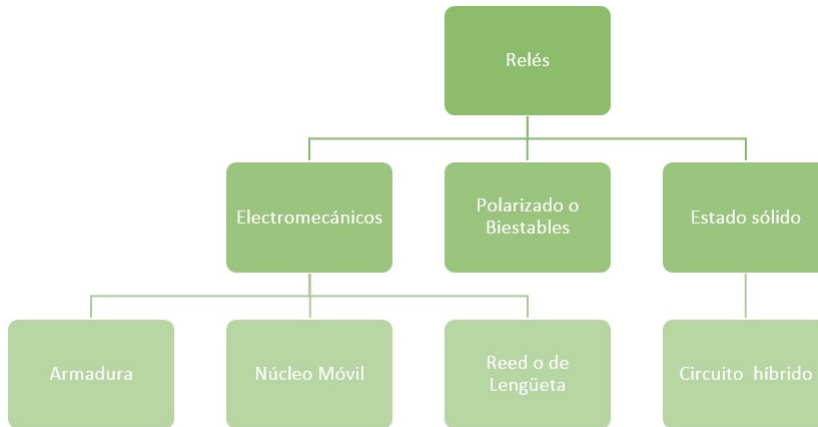


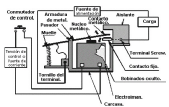

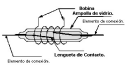
Figura 2.2: Estructura de los Relés

Fuente: Elaboración propia.

2.3.3.1. Relés Electromecánicos

Los relés electromecánicos están formados por una bobina y unos contactos, los cuales pueden conmutar corriente continua o bien corriente alterna. Esta clasificación cuenta con tres tipos: Armadura, Núcleo Móvil y Reed o Lengüeta. Los cuales se pueden observar en el Cuadro 2.3.

Cuadro 2.3: Relés electromecánicos

Tipos de Relés Electro-mecánicos	Descripción	Imagen
Armadura		El electroimán hace vascular la armadura al ser excitada cerrando los contactos dependiendo si es N.O. o N.C (Normalmente abierto o Normalmente cerrado).
Núcleo Móvil		Estos tienen un émbolo en lugar de la armadura del tipo anterior. Se utiliza un solenoide para cerrar sus contactos, debido a su mayor fuerza atractiva (por ello es útil para manejar altas corrientes).
Reed o de Lengüeta		Formados por una ampolla de vidrio, en cuyo interior están situados los contactos (pueden ser múltiples) montados sobre delgadas láminas metálicas. Dichos contactos se cierran por medio de la excitación de una bobina, que se encuentra alrededor de dicha ampolla.

Fuente:Elaboración propia.

2.3.3.2. Relés Polarizados

Llevan una pequeña armadura, solidaria a un imán permanente. El extremo inferior puede girar dentro de los polos de un electroimán y el otro lleva una cabeza de contacto. Si se excita al electroimán, se mueve la armadura y cierra los contactos. Si la polaridad es la opuesta girará en sentido contrario, abriendo los contactos o cerrando otro circuito o varios, como la Figura 2.3.

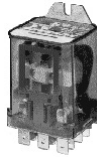


Figura 2.3: Relé polarizado

2.3.3.3. Relé de Estado Sólido

Relés de estado sólido (Solid State Relay) es un circuito electrónico que contiene en su interior un circuito disparado por nivel, acoplado a un interruptor semiconductor, un transistor o un tristor. Por SSR se entenderá un producto construido y comprobado en una fábrica, no un dispositivo formado por componentes independientes que se han montado sobre una placa de circuito impreso, como la Figura 2.4.

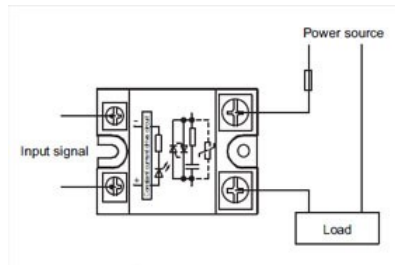


Figura 2.4: Relé de Estado Sólido

2.4. Bluetooth

Las redes inalámbricas de área personal WPAN (Wireless Personal Area Network) son redes que comúnmente cubren distancias del orden de los 10 metros como máximo, normalmente utilizadas para conectar varios dispositivos portátiles personales sin la necesidad de utilizar cables.

La tecnología inalámbrica bluetooth es un estándar global abierto para enlaces de radio, que ofrece conexiones inalámbricas económicas para transmisiones de voz y datos entre computadoras portátiles, dispositivos de mano, teléfonos celulares y varios aparatos más mediante un enlace de radiofrecuencia; así como acceso a otros recursos en la red. La Figura 2.5 muestra una red inalámbrica

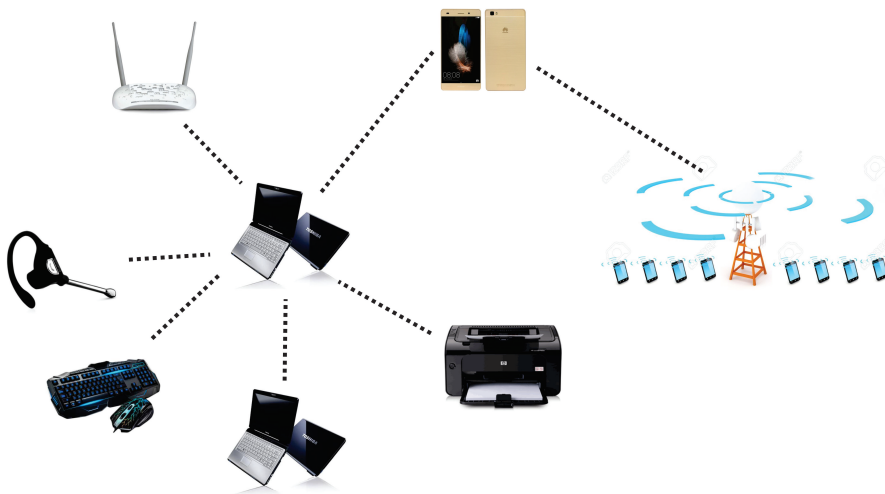


Figura 2.5: Diagrama de Red inalámbrica vía Bluetooth
Fuente: Elaboración propia.

2.4.1. Módulo HC-05

El módulo Bluetooth HC-05 Módulo maestro-esclavo, que quiere decir que además de recibir conexiones desde una PC o Tablet, también es capaz

de generar conexiones hacia otros dispositivos bluetooth; permite conectar dos módulos de bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos, como se puede observar en la Figura 2.6.

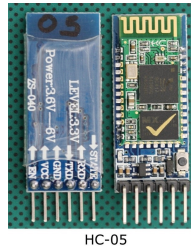


Figura 2.6: Módulo Bluetooth HC-05
Fuente: Elaboración propia.

2.4.2. Módulo HC-06

El módulo Bluetooth HC-06 acepta un set básico de comandos, que permite configuraciones, con menos opciones para su configuración, económico para conectar dispositivos con microcontroladores a la PC, tablets, relojes inteligentes, etc, como se puede observar en la Figura 2.7.

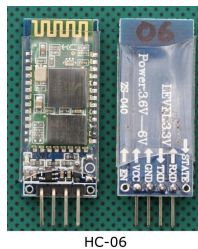


Figura 2.7: Módulo Bluetooth HC-06
Fuente: Elaboración propia.

2.5. Android

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, Symbian y Blackberry OS. Lo que lo hace diferente es que está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en lenguaje Java y permite programar aplicaciones en una variación de Java llamada Dalvik.

Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hace que una de las cosas más importantes del sistema operativo sea la cantidad de aplicaciones disponibles. Una de las mejores características de dicho sistema operativo es que es completamente libre, lo que da seguridad a los usuarios, ya que algo que es abierto permite detectar fallos más rápidamente; y también a los fabricantes, pues pueden adaptar mejor el sistema operativo a los terminales ver Figura 2.8.

Se distinguen claramente cada una de las capas: la que forma parte del propio Kernel de Linux, donde Android puede acceder a diferentes controladores, las librerías creadas para el desarrollo de aplicaciones Android, la siguiente capa que organiza los diferentes administradores de recursos, y por último, la capa de las aplicaciones a las que tiene acceso.

Google App Inventor es una plataforma 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.

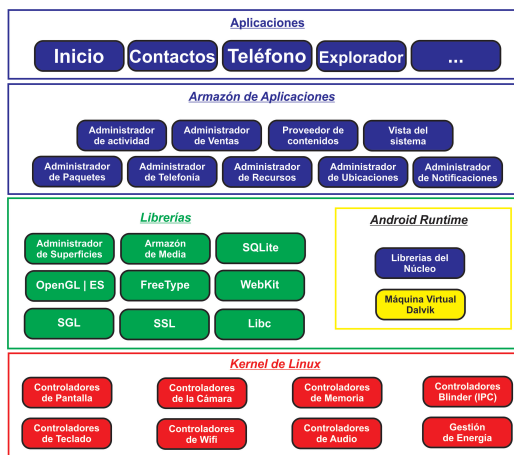


Figura 2.8: Sistema de capas de Android basado en ¿Qué es Andriod? [4].

2.6. APP Inventor

Es una herramienta de código abierto que pretende realizar la programación y la creación de aplicaciones accesibles a una amplia gama de audiencias. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones de App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

App Inventor se puede tener la primera aplicación en funcionamiento en una hora o menos, y se pueden programar aplicaciones más complejas en mucho menos tiempo que con los lenguajes más tradicionales, basados en texto.

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, ver la Figura 2.9.

En el Cuadro 2.4 se puede ver las ventajas y desventajas que tiene el usar la plataforma de APP Inventor, qué actividades permite y cuáles no.

Cuadro 2.4: Ventajas y Desventajas para usar APP Inventor

Ventajas	Desventajas
No es necesario instalar un IDE.	No se pueden subir al Android Market.
Son necesarios mínimos conocimientos de programación	No permite diferentes actividades en una aplicación.
Desarrollo rápido de aplicaciones con bajos niveles de error.	Tamaño elevado de APK.
Almacenamiento en la nube.	No permite aplicaciones complejas, aunque si completas.

Fuente: Elaboración propia.

Con App Inventor se programa usando bloques de programación. Estos bloques están hechos con elementos comunes a la mayoría de los lenguajes de programación existentes. Se colocan bloques para construir bucles, condiciones, variables, entre otros; que permiten pensar lógicamente y solucionar los problemas de forma metódica, sin perder el tiempo en encontrar el punto y coma o los dos puntos que están donde no deben y producen errores de compilación o ejecución.

Algunos elementos de programación en App Inventor que se utilizan en forma de bloques como:

- **Sentencias:** Al programar se necesita enviarle al ordenador una orden para que haga algo. En App Inventor existen numerosos bloques que son sentencias de programación. Se distinguen rápidamente ya que expresan una acción a realizar con un verbo en imperativo. Los más habituales son `call`, `set` y `def`. Además, los bloques que son sentencias sólo están disponibles en dos colores: azul para modificar variables o propiedades de un objeto y naranja para llamar a funciones,

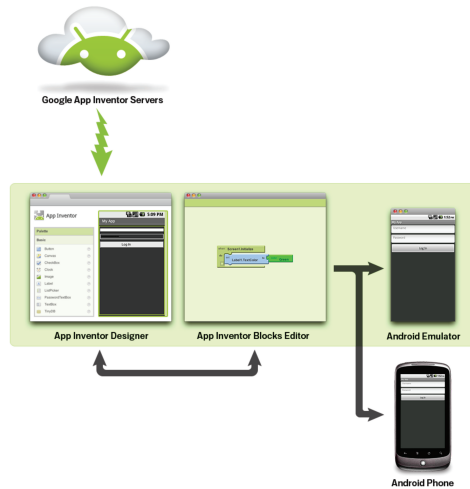


Figura 2.9: Diagrama de funcionamiento de App Inventor basado en [5].

como se puede observar en la Figura 2.10 .

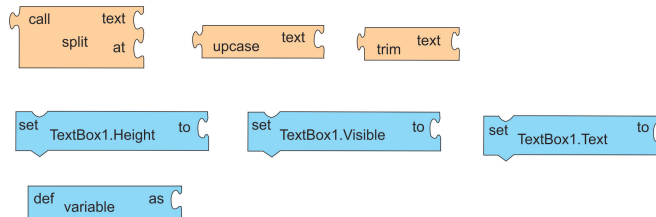


Figura 2.10: Sentencias en App Inventor basado en [5].

En la Figura 2.10 se ven algunos ejemplos de las sentencias para dividir un texto en partes, convertirlo a mayúsculas, cambiar la altura, texto o visibilidad de un recuadro o definir una variable.

- **Condiciones:** Con frecuencia se necesita realizar distintas acciones en función de que ocurra o no algo. En App Inventor para condicionar

el programa se cuenta con las estructuras if-then, if-then-else, while y choose ver Figura 2.11 .

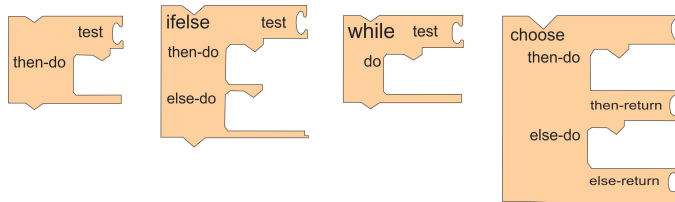


Figura 2.11: Condiciones en App Inventor basado en [5].

En todos los bloques se puede observar cómo en la parte superior hay un espacio para encajar el test o pregunta que pide al programa, para decidir por dónde continúa su flujo. En la Figura 2.11 Condiciones en App inventor se pueden observar que las tres primeras estructuras son habituales en la mayoría de los lenguajes, la última, choose, es prácticamente igual a if-then-else con la diferencia de que permite devolver una variable en función de que se cumpla o no el test.

- Bucles:** Un bucle provoca la ejecución repetida de varias sentencias. En App Inventor se dispone de los bucles while, for range y foreach, ver Figura 2.12 . El primero ejecutará el bloque de sentencias que se encajen en el apartado 'do' mientras se cumpla la condición que refleje el test, for range, es el típico bucle for de otros lenguajes de programación y, como tal, tiene las opciones de elegir el nombre de la variable que itera, su comienzo, fin y el tamaño de los saltos de iteración. Finalmente, foreach es un iterador sobre los elementos de una lista.
- Variables:** Tiene la posibilidad de las herramientas para definir y cambiar variables. El tipo puede ser numérico o un texto y “se declara” al asignarle un valor por primera vez, ver Figura 2.13.

En la Figura 2.13 se muestra cómo se declara una variable llamada

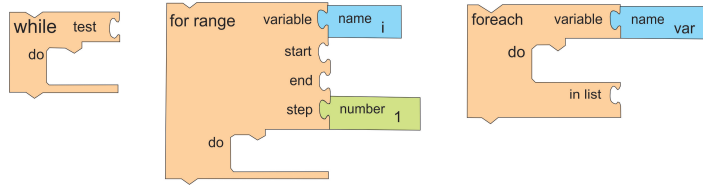


Figura 2.12: Bucles App Inventor basado en [5].

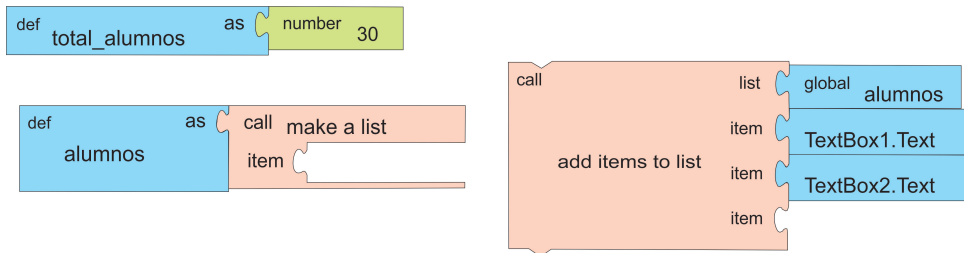


Figura 2.13: Variables en App Inventor basado en [5].

“total_alumnos” y se le da un valor numérico de 30. Además se observan dos bloques. Su lenguaje cuenta con una estructura de datos llamada list (lista) que es similar a los Arrays de otros lenguajes de programación. Se muestra incluso el bloque con el que se define la lista “alumnos” (se define vacía al no añadirle ningún item) y también se ve el bloque en el que se le añaden dos elementos a la lista de alumnos. En el ejemplo en la imagen se le añade el texto javascript:void(0); que contienen los cuadros de texto “TextBox1” y “TextBox2”.

- **Eventos:** Aquí es donde los programas son capaces de responder a la interacción del usuario con la interfaz, toman la importancia los bloques, para ejecutar las acciones en respuesta a los eventos que se produzcan en el teléfono o Tablet. El nombre de los distintos eventos depende del objeto; un botón avisa cuando se ha hecho click (significa pulsar aquí) sobre él, mientras que el acelerómetro avisa cuando se ha agitado el móvil, la mensajería cuando se ha recibido un mensaje, la cámara de fotos cuando se ha hecho una foto, etc.

En la Figura 2.14 se puede observar algunos de estos eventos mencionados.

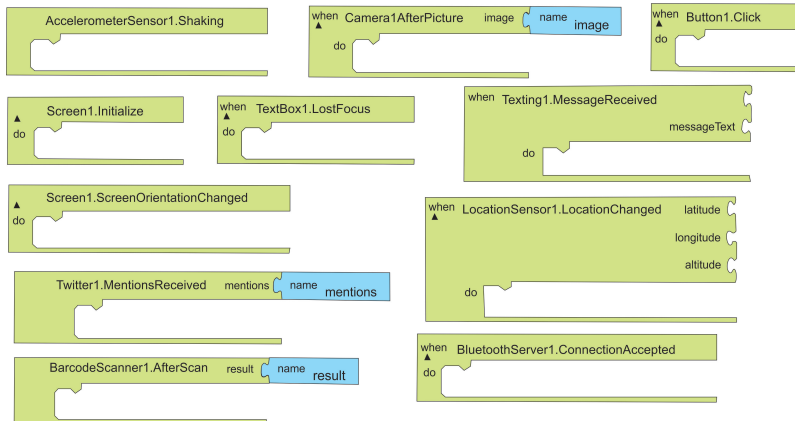


Figura 2.14: Eventos para objetos en App Inventor basado en [5].

Capítulo 3

Análisis de la situación actual








3.1. Etapa 1 Situación problema no estructurada

Dentro de esta etapa se definen los agentes actores dentro de la situación que se quiere mejorar y los cuales intervienen en el entorno de la impresora flexográfica, se especifican las herramientas que se utilizan y los usuarios que son beneficiados por la misma. En el Cuadro 3.1 se describen los agentes que intervienen en esta situación.

En las carreras de Artes Gráficas y Offset de la DGETI CETis 165 y 119, según el plan de estudios, se debe impartir la teoría y la práctica en varias materias, un caso particular es la práctica en el uso de una impresora flexográfica. Debido a los altos costos que dichas impresoras tienen, éstas no son adquiridas en las instituciones educativas de gobierno, por ello los profesores imparten sólo la teoría de la asignatura y los alumnos carecen de esta competencia y la destreza para utilizar una impresora flexográfica profesional, de aquí que realizan su servicio social o prácticas profesionales en la industria sin el conocimiento necesario para poder llevarlo a cabo con la eficacia y eficiencia necesarias.

En la Figura 3.1, se muestra la situación no estructurada del problema y

Cuadro 3.1: Definición de agentes.

# Agente	Agente	Icono de agente	Descripción
1	Profesor		El profesor imparte y da instrucciones de como usar la impresora.
2	Alumno		El alumno utiliza la impresora con responsabilidad y compromiso.
3	Asignatura		El alumno utiliza la impresora con responsabilidad y compromiso.
4	Plan de estudios		El plan de estudios esta enfocado, que al termino de la carrera se obtenga el conocimiento esperado.
5	Institución pública DGETI CETis		La institución DGETI CETis utilizará la impresora con fines académicos para sus estudiantes.
6	Industria		La industria tendrá gente de servicio y prácticas, que estarán capacitados para uso de sus impresoras profesionales.
7	Impresora flexográfica profesional		Es de alto costo que solo pueden adquirir empresas de renombre.

Fuente: Elaboración propia.

las relaciones entre los actores o elementos, tales como la industria, plan de estudios, asignatura, flexografía profesional, institución, usuarios (profesores y alumnos).

Después de recabar información se tienen las siguientes relaciones. Relación 1: El profesor imparte la asignatura de Flexografía; Relación 2: El profesor no logra cumplir con el plan de estudios; Relación 3: El profesor no culmina la parte práctica por ausencia de una impresora flexográfica; Relación 4: El plan de estudios en la institución DGETI CETis; Relación 5: La asignatura es impartida en la institución DGETI CETis; Relación 6: La asignatura se imparte al alumno; Relación 7: El alumno asiste a las asignatura de Flexografía; Relación 8: El alumno desconoce el uso en la práctica de una impresora flexográfica; Relación 9: La institución DGETI CETis no puede adquirir por recursos económicos una impresora flexográfica profesional; Relación 10: Los alumnos terminan conociendo el funcionamiento de una impresora flexográfica, hasta que realizan su servicio social y prácticas profesionales.

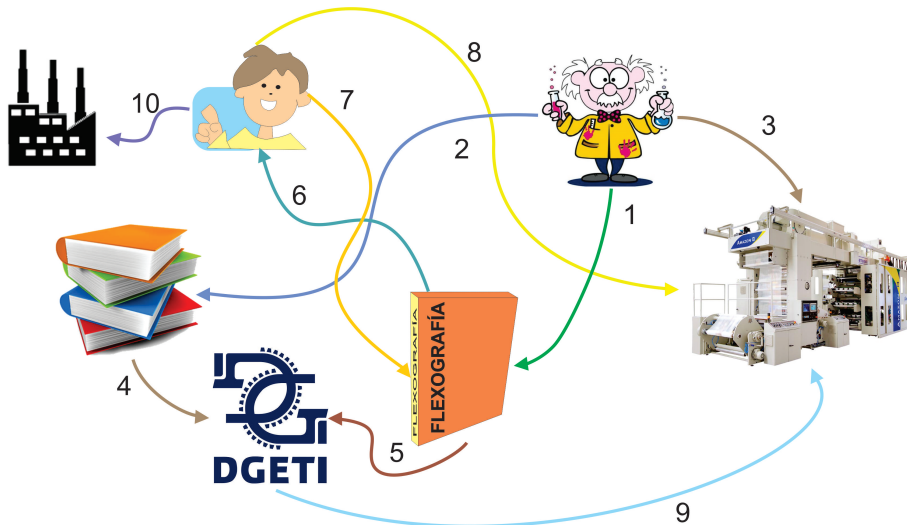


Figura 3.1: Situación original no estructurada.

3.2. Etapa 2 Situación estructurada

Paula Feldman, Directora de inteligencia de negocios de The Association for Packaging and Processing Technologies (PMMI) indica: “El 70 % de las empresas que se entrevistaron están haciendo algún tipo de cambio a proceso de etiquetado, marcado o codificación. Y lo que requieren de sus proveedores y fabricantes de equipos originales (OEMs por sus siglas en inglés) es que les ayuden a implementar los cambios a la perfección”. A lo que, en respuesta a este empeño creciente, algunos de los proveedores revelan nuevas tecnologías para realizar este cambio que se busca [6].

Una vez que se hizo la investigación acerca del porqué no se adquiría alguna impresora flexográfica profesional para usos académicos, se determinó que su gran tamaño, costo y magnitud no eran prácticos para el uso que es requerida.

En un proyecto anterior [1] se desarrolló una impresora flexográfica automatizada con PLC, con ciertas características como portabilidad, PLC, cliché, diseño, impresión a un color, bajo costo y peso. En el Cuadro 3.2, se anotan los nuevos agentes para esta impresora.

En la Figura 3.2, se muestra la situación actual en la que se encuentra la impresora flexográfica portátil programada con PLC dentro de la institución, para efectos de no repetir, aquí solo se mencionan las nuevas relaciones del sistema. Relación 2: El profesor cumple con el plan de estudios; Relación 3: El profesor enseña el uso de la impresora; Relación 4: La asignatura de Flexografía es impartida en la institución DGETI CETis; Relación 5: La asignatura se imparte al alumno; Relación 6: El alumno asiste a la asignatura de Flexografía; Relación 7: El alumno practica en el Entrenador de impresión flexográfica automatizado con PLC, Relación 8: El alumno tiene conocimientos previos para practicar en la industria, mediante servicio social y prácticas profesionales; Relación 9: El plan de estudios que cuenta la DGETI CETis; Relación 10: La institución DGETI CETis cuenta con un entrenador de impresión; Relación 11: El entrenador imprime a una tinta; Relación 12: La impresora es portátil; Relación 13: En los rodillos se coloca la matriz (diseño); Relation 14: El entrenador es programado con PLC;

Relación 15: La impresora tiene 10 kg de peso; Relación 16: El entrenador cuenta con un precio accesible.

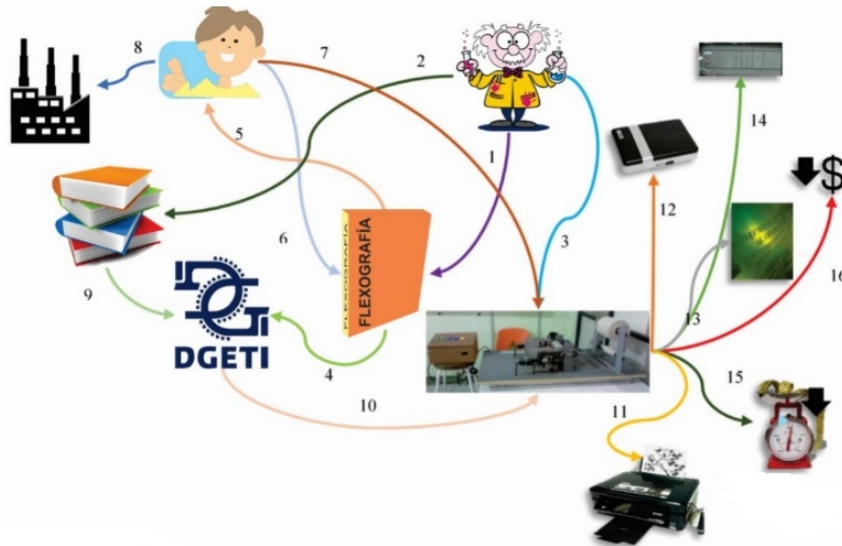
















Figura 3.2: Situación estructurada actual impresora portátil programada con PLC.

Después del uso de dicha flexográfica en los CETIS 119 y 165, se encuestó a 30 personas ver la Figura 3.3 y se encontró que ésta era susceptible a cambios. La encuesta dió como resultados los beneficios de la impresora, de igual forma mostró el orden de importancia de las características, que ésta tiene hasta el momento, además se toman en cuenta las que consideraron los usuarios y sugerencias de que elementos se quitan y/o agregan.

Cuadro 3.2: Agentes de la impresora flexográfica automatizada con PLC.

# Agente	Agente	Icono de Agente	Descripción
1	Impresora flexográfica automatizada con PLC		Impresora a la que se realizan mejoras pedidas por el cliente.
2	Controlador lógico programable		Dispositivo programado con código para impresora.
3	Carcasa		Plataforma que lleva montado cada elemento de la impresora.
4	Botones		Plataforma que lleva montado cada elemento de la impresora.
5	Rodillos		Caja porta PLC para las funciones de Encendido/Apagado, Entintado, Velocidades desde la mínima hasta la máxima, Reversa y Jogueo.
6	Relevadores		Rodillos para portar la matriz, tomar tinta, dosificar tinta y sujetador de material.
7	Step 7 Micro/Win		Software donde se realizó el programa para el funcionamiento del PLC.
8	Fuente de poder		Fuente de alimentación para el funcionamiento de la impresora.
9	Motores		Motor Rebobinador, entintador e impresión de matriz.

10	Portátil		El usuario puede moverla con facilidad acomodarla en otro lugar de uso.
11	Impresión a un color(tinta)		Posibilidad de practicar con un color, es decir, solo se tiene un rodillo con una tina de entintado para un color.
12	Cliché		Diseño colocado en rodillo porta cliché.
13	Peso bajo		Peso de 10 kg. contando con la fuente de poder y el PLC.
14	Costo accesible		El precio considerado fue de \$13,000.00 MX.

Fuente: Elaboración propia.

3.3. Etapa 3 Definiciones básicas

Aquí se da a conocer el propósito central del sistema, tomando en cuenta la perspectiva de las personas. Se elaboran definiciones de lo que idealmente se necesita, según los diferentes elementos involucrados en el sistema. La construcción de estas definiciones se fundamenta en seis factores que deben aparecer explícitos en todas ellas agrupados como CATWOE (Cuadro 1.1 en el que se describe las etapas SSM).

A partir del proyecto que se realizó en Ingeniería en Computación Entrenador de impresión flexográfica automatizada con PLC, puede obtenerse

Encuesta No. 1

1. El tiempo que utilizaste el entrenador de impresión flexográfica automatizado con PLC

3 meses

menos de 3 meses

nunca

2. ¿Qué opinión tienes del entrenador de impresión flexográfica?

3. ¿Qué porcentaje de avance crees que pueden obtener los alumnos con el uso del entrenador?

3. La dimensión para la impresión es de 15cm x 15 cm ¿Esta dimensión es suficiente para la práctica de impresión? Sí No

¿Por qué? _____

4. De las características que tiene el entrenador con PLC, anota del 1-10 el orden de importancia

- Es portátil para desplazarse
- La instalación es sencilla de hacer
- Mantenimiento sencillo y práctico
- Imprime materiales como plástico, bolsas de papel chicas, servilletas
- Imprime a un color
- Cuenta con manuales de operación, instalación y mantenimiento
- Pesa 10kg
- Tiene bajo costo
- Programado en PLC (Controlador Lógico Programable)
- Su desempeño es similar al de una impresora profesional
- Dimensión de 1m x 57cm

5. ¿Qué le quitarías o agregarías?

Figura 3.3: Encuesta No. 1
Características y funcionalidad de la impresora anterior.
Fuente: Elaboración propia.

una mejora de ésta. Para ello, se aplicó una segunda encuesta a 30 usuarios para saber las necesidades del cliente para mejorar la impresora que ya está en uso. La encuesta No.2 como se puede observar en la Figura 3.4, en la que se muestra los cambios propuestos para dicha impresora.

Esta encuesta mostró como resultados los elementos importantes de la impresora, que agregar una aplicación para celular es indispensable y añadir el bluetooth para tener un ambiente más amigable con el usuario.

ENCUESTA 2

Si se realiza una **aplicación** que **enlace** el celular con el entrenador

1. ¿Qué características te gustaría que tuviera dicha aplicación?

2. De las opciones siguientes indica del 1 al 10 cuáles son las más importantes para el nuevo entrenador
 - Bluetooth
 - Programado con Arduino
 - Instalación sencilla
 - Agregar ruedas para desplazamiento
 - Dimensión 80cm x 60cm
 - Aplicación del sistema de impresión flexográfica para el celular
 - Dimensión de 20 cm menos que la de antes
 - Manuales de operación, instalación y mantenimiento en la aplicación para el celular
 - Impresora Flexográfica Automatizada con microcontroladores
 - Desarrollo de la aplicación en App inventor
3. ¿Te gustaría que hubiera en la aplicación información sobre la historia de impresión flexográfica? Sí No
¿Por qué?
4. ¿De qué forma crees que impactaría el uso del bluetooth como tecnología inalámbrica para la facilidad del uso del entrenador?

Figura 3.4: Encuesta No. 2
Nuevas características para la impresora.
Fuente: Elaboración propia.

3.3.1. CATWOE

Se utiliza principalmente con el fin de analizar las sentencias de la definición raíz. Se puede, observar en el Cuadro 3.3 el CATWOE con las respectivas características de cada uno de los elementos que actúan en el sistema solución del problema.

Cuadro 3.3: Descripción de los elementos del CATWOE.

Inicial	Concepto	Descripción.
C	Cliente	Experto (profesor de impresión flexográfica) a la que se están prestando los servicios para la realización de su sistema. Alumnos
A	Actor	Analista, Diseñador de la impresora flexográfica y Programador.
T	Proceso de la transformación	Se muestra el proceso que se realiza para transformar las ideas del cliente sobre el diseño y funcionamiento del proyecto (entradas cualitativas), de esta forma se obtienen las salidas en las que se observa el proyecto en funcionamiento con los requerimientos que pide el cliente salidas cuantitativas).
W	Weltanschauung	Se refiere al punto de vista desde la perspectiva del cliente, la visión positiva o negativa de los diferentes roles para hacer que el proceso de transformación sea significativo en su contexto.
O	Dueño o Propietario	El cliente con el aporte de ideas respecto al proyecto, puede ayudar con su opinión para el proyecto, si está siguiéndose sus requerimientos o bien se sale de lo solicitado.
E	Apremios ambientales	Institución, CU UAEM Ecatepec, con el apoyo económico de CONACYT para el desarrollo y producción del proyecto.

Fuente: Elaboración propia.

Se puede apreciar en la Figura 3.5 el diagrama el análisis CATWOE de la tabla anterior; en éste se muestran las visiones positivas y negativas de los agentes en cuestión, así como la transformación. Se describe cada uno de los conceptos que contiene el CATWOE. Se muestran las visiones positivas y negativas de los agentes en cuestión así como la transformación.

En esta etapa se incluye una herramienta para mejorar el producto lla-

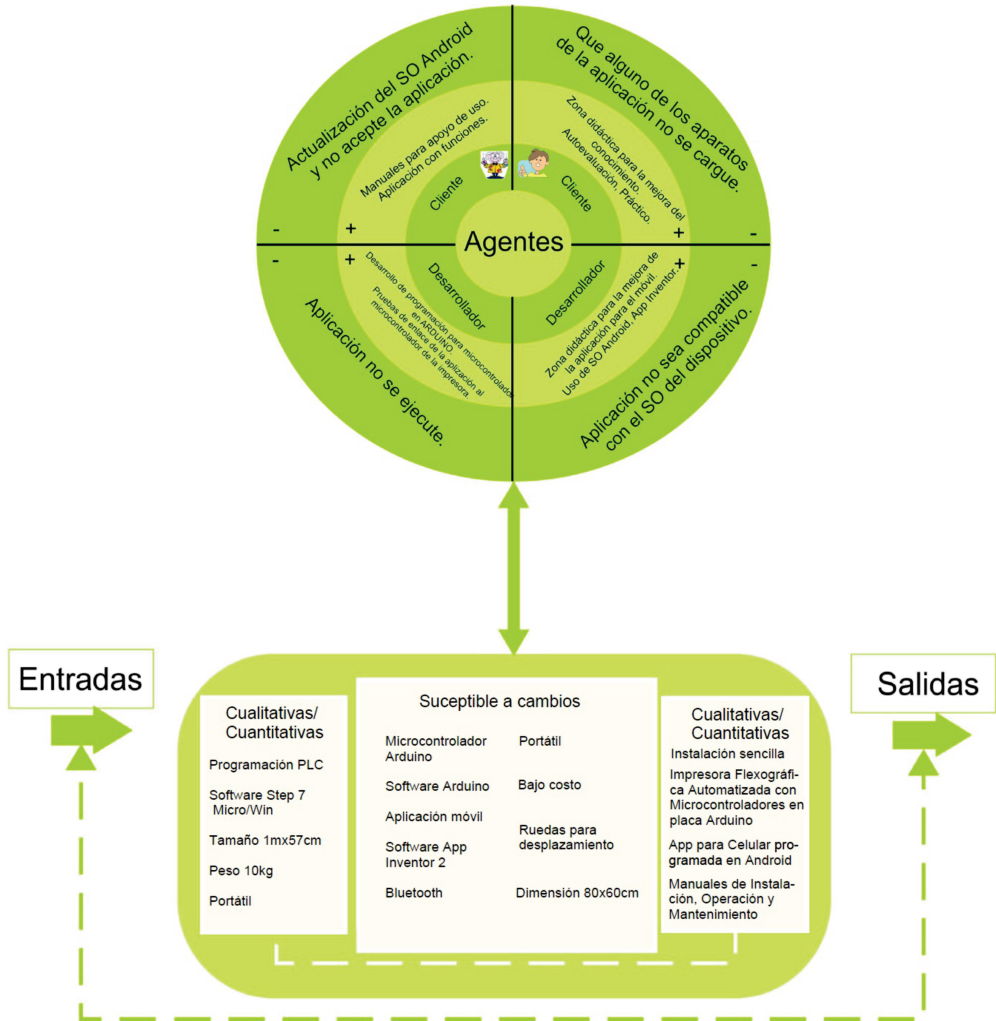


Figura 3.5: Diagrama CATWOE
Elaboración propia basado en paso 7 de [7].

mada Despliegue de la función de calidad (Quality Function Deployment) QFD.

El QFD es una de las claves para lograr la mejora continua, en el que se asegura que los deseos y necesidades de los clientes sean traducidas en características técnicas, así mismo que el cliente se involucre en el proceso de desarrollo del producto. En la Figura 3.6 se muestra la casa de la calidad con respecto a las necesidades del cliente y lo que se ofrece, con los resultados obtenidos de la encuesta No.2.

Después de utilizar el QFD se tienen los elementos necesarios para mejorar la impresora flexográfica programada con PLC. Como puede verse los valores más altos del ranking son 1) Impresora Flexográfica Automatizada con Microcontroladores, 2) Aplicación del sistema de impresión flexográfica para el celular y 3) Instalación sencilla. La Figura 3.6 muestra los elementos que se toman a cuenta en la mejora (parte superior horizontal sección B del QFD).

Después de realizar el CATWOE y el QFD se tienen los siguientes elementos ver Figura 3.7 que intervienen en el proceso de desarrollo de dicha impresora, especificando cada una de las herramientas que se necesita, se muestra el diagrama de las definiciones básicas.

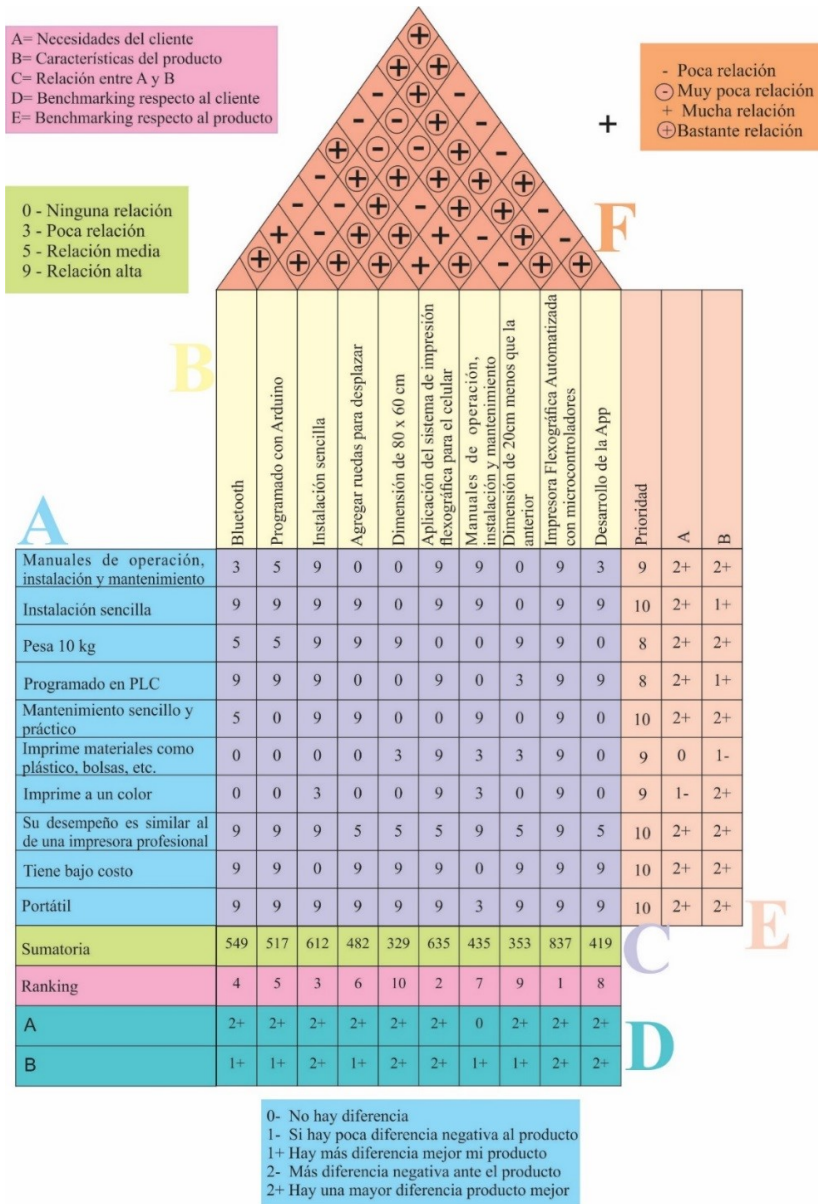


Figura 3.6: Despliegue de Casa de la Calidad (QFD) basado en [8]

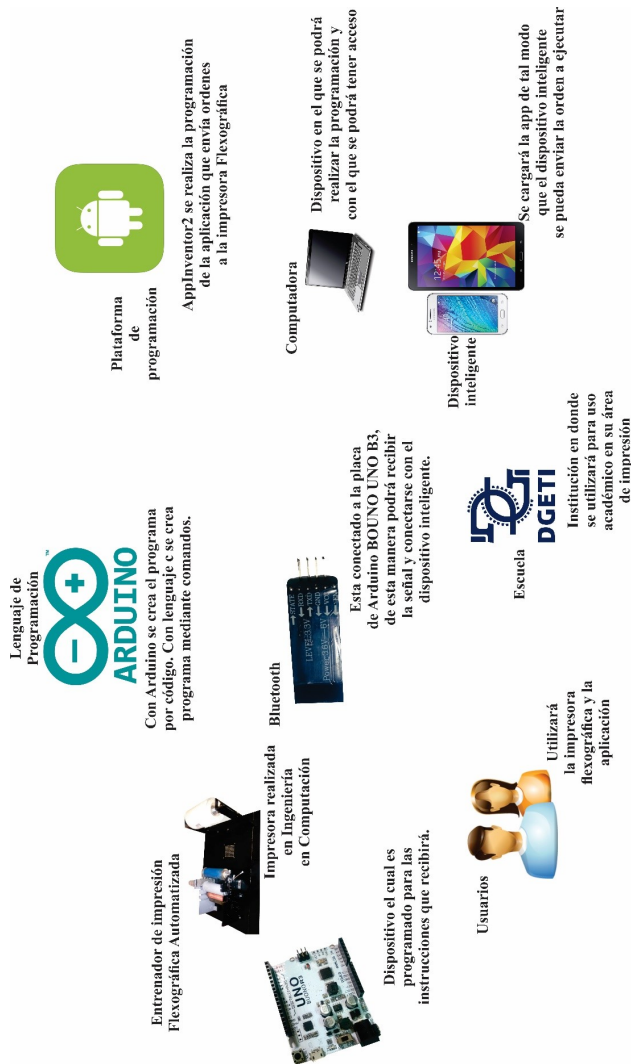


Figura 3.7: Definición básica.
Fuente: Elaboración propia

Capítulo 4

Planeación de la Impresora

4.1. Etapa 4 Modelos conceptuales

En la Figura 4.1 muestra una visión sistémica de solución al problema, en la que como entorno están los usuarios y la institución DGETI CETis 119 y 165 (clientes), es decir, institución y usuarios, y en el sistema solución la forma en la que se desarrolla la impresora flexográfica.

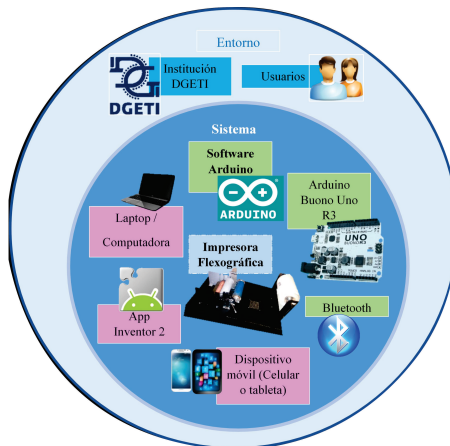


Figura 4.1: Modelo Conceptual
Fuente: Elaboración propia

4.1.1. Etapa 4a Modelación de sistemas formales

Consiste en el uso de un modelo general de sistema de la actividad humana que se puede usar para verificar que el modelo conceptual no sea fundamentalmente deficiente, es decir, sirve como una guía de consulta para el modelo conceptual que se trazó.

En el Cuadro 4.1, el modelo conceptual propuesta solución 1 Picaxe, muestra los microcontroladores de la familia Picaxe 08M2, 14M2 y 18M2, con las características más importantes, esto para elegir el adecuado para la programación de la impresora.

Cuadro 4.1: Modelo conceptual solución 1. Picaxe

Características	Picaxe 08M2	Picaxe 14M2	Picaxe 18M2
Pines	8	14	18
I/O	6	12	16
ADC	3	7	10
Data Memory	256	256	256
MHz	32	32	32
Diseñar placa	Si	Si	Si

Fuente: Elaboración propia.

Se puede observar que el Picaxe 18M2 tiene 18 pines con 16 entradas y salidas, las cuales sirven para colocar todas las conexiones, que se deseen realizar para la impresora.

En el Cuadro 4.2, el modelo conceptual propuesta solución 2 Pic, muestra las características de los microcontroladores de la familia Pic 16F873, 16F874 y 16F876.

El modelo conceptual solución 2 Pic, compara entre los microcontroladores de la familia Pic, donde la opción elegida para el desarrollo es el Pic16F873 por conjunto de instrucciones, memoria, entradas y salidas, esta es la opción de microcontrolador a emplear en el desarrollo de la impresora.

Cuadro 4.2: Modelo conceptual solución 2. Pic

Características	Pic16F873	Pic16F874	Pic16F876
Instruction set	35	35	35
I/O	A,B,C	A,B,C,D,E	A,B,C
Timer	3	3	3
Data Memory	192	192	368
Flash Program Memory	4K	4K	4K
MHz	20	20	20
Diseñar placa	Si	Si	Si

Fuente: Elaboración propia.

Como último el Cuadro 4.3, el modelo conceptual propuesta solución 3, donde se muestra los tres microcontroladores de la familia ATmega.

En el modelo conceptual solución 3 Arduino, se hace la comparativa con el Arduino UNO R3 y Arduino pro mini, el Arduino Buono UNO R3, donde el indicado para la realización en la programación de la impresora es el Arduino bueno UNO R3 por el microcontrolador ATmega328 TQFP-32, Micro USB, los dos tipos de voltaje y corriente que maneja, entradas y salidas, cuenta con un cristal de cuarzo de alta calidad es ideal como oscilador externo para el microcontrolador.

Por lo que la opción adecuada es del modelo conceptual solución 3, ya que la placa ya viene implementada, cargar el programa es fácil por medio del micro usb, el voltaje que se recibe es de 3.3V o bien 5V.

El Uno se puede programar con el software de Arduino (IDE). Seleccione "Arduino/Genuino Uno" desde el menú Herramientas Junta (de acuerdo con el micro controlador en su tablero).

Los ATmega328 en el Uno viene pre programado con un cargador de arranque que le permite cargar nuevo código a ella sin el uso de un programador de hardware externo. Se comunica utilizando el protocolo original STK500.

También puede pasar por alto el gestor de arranque y programar el microcontrolador a través de la (programación serial en circuito) ICSP encabezado utilizando Arduino ISP o similar.

Cuadro 4.3: Modelo conceptual solución 3. Arduino

Características	Arduino UNO R3	Arduino Buono Uno R3	Arduino pro mini
USB	Type B Female	Micro USB	Necesita adaptador externo
Pins I/O	14/14	14/14	14/14
Operating Voltage	5	3.3 o 5	3.3 o 5
3.3V Current	50mA	300mA	40mA
5V Current	500mA	2A	2A
Input Range (V)	7-12	6-23	6-23
MCU	ATmega 328	ATmega328 TQFP-32	AVR ATmega168 o 328
Clock (MHz)	16	16	16
Flash Memory (KB)	32	32	16 o 32
Diseñar placa	Está hecha	Está hecha	Está hecha
Others	6 (salidas PWM)	Group terminal LPF for AVCC, LITE version available, Cristal de cuarzo	EEPROM de 512 bytes

Fuente: Elaboración propia.

Obsérvese el cuadro 4.4, la comparación entre los microcontroladores que se tomaron en cuenta y el que se usó para el desarrollo.

Se tomaron las características más importantes para la toma de decisión entre los microcontroladores Picaxe 18M2, Pic16F873 y Arduino Buono Uno R3. En el que el Arduino Buono Uno R3 con ATmega328 TQFP-32,

es el indicado para la impresora.

Cuadro 4.4: Comparativa entre microcontroladores.

Características	Picaxe	Pic	Arduino Buono Uno R3
Microcontrolador	18M2	16F873	Atmega328 TQFP-32
Pines	18	-	-
I/O	16	A,B,C	-
Data Memory Kb	256	192	-
MHz	32	20	16
Instruction set	-	35	-
Timer	-	3	16
Flash Program Memory (KB)	-	4	32
ADC	10	-	-
USB	-	-	Micro USB
Operating Voltage	-	-	3.3 o 5
3.3V Current	-	-	300mA
5V Current	-	-	2A
Diseñar placa	Si	Si	Esta hecha
Input Range (V)	-	-	6-23
Others	-	-	Ground terminal LPF for AVCC LI-TE version available

Fuente: Elaboración propia.

La Figura 4.2 muestra el modelo conceptual para el desarrollo de la impresora flexográfica programada con el microcontrolador ATmega328 TQFP-32 Arduino Buono Uno R3.

Respecto al Bluetooth se hizo la comparación entre los módulos HC-05 y HC-06, los resultados se encuentran en el cuadro 4.5.

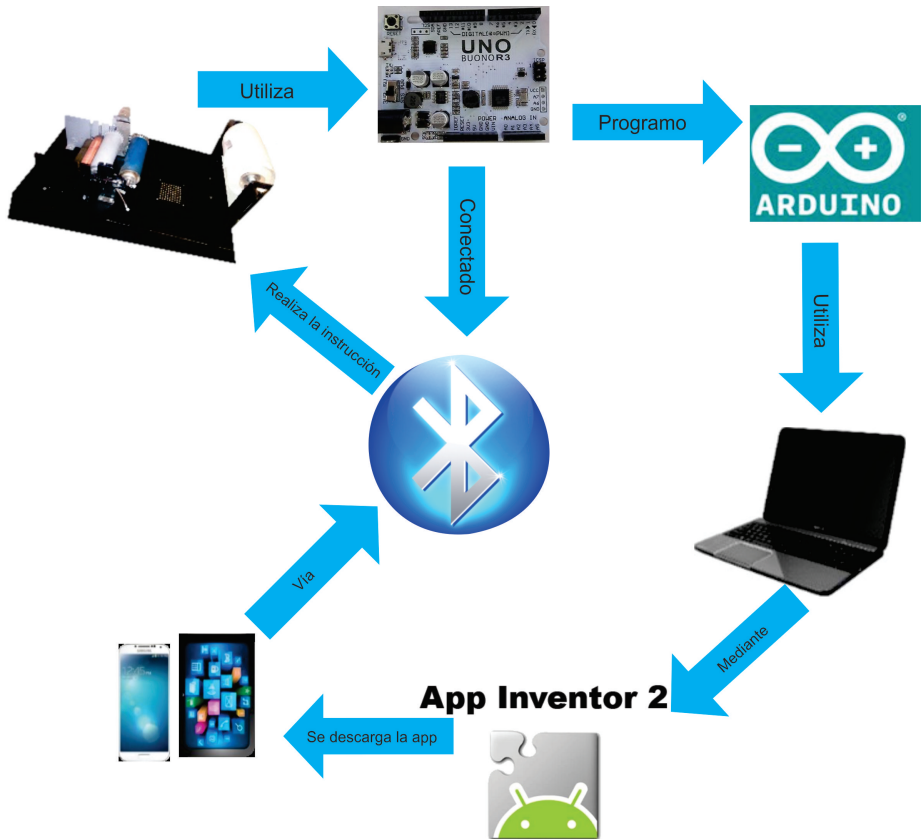
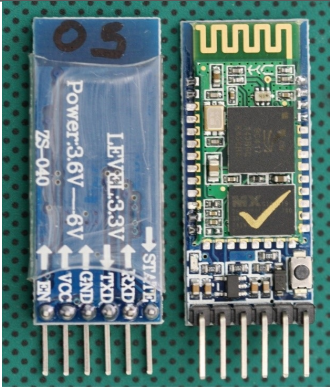
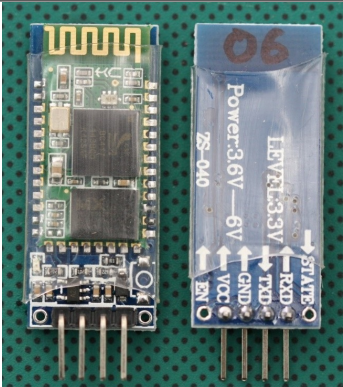


Figura 4.2: Diagrama de desarrollo de la impresora flexográfica.
Fuente: Elaboración propia.

Cuadro 4.5: Comparativa entre Bluetooth.

Bluetooth HC-05	Bluetooth HC-06
Son idénticos (vista)	Son idénticos (vista)
Realiza comunicaciones inalámbricamente a 10 mts máx SSP (Serial Port Protocol)	Realiza comunicaciones inalámbricamente a 10 mts máx SSP (Serial Port Protocol)
Bajo consumo 3.3 - 6V	Bajo consumo 3.3 - 6V
Tamaño 12.7mmx27mm	Tamaño 12.7mmx27mm
Modo maestro-esclavo	Modo esclavo
6 pines	4 pines
Vcc, GND, TX, RX, Key, Reset y Status	Vcc, GND, TX y RX
 <p style="text-align: center;">HC-05</p>	 <p style="text-align: center;">HC-06</p>

Fuente: Elaboración propia.

Como puede observarse en el cuadro 4.6 que se muestran las características principales del Bluetooth modelo HC-06 y las versiones del mismo.

Se decidió tomar el Bluetooth HC06, ya que para el proyecto sólo se requiere que envíe y reciba el dato elegido. La diferencia entre estos dispositivos es: Sí no es adquirido el HC-06 como maestro y se adquiere un esclavo, éste no visualiza ni empareja ningún dispositivo bluetooth.

En cuanto al relevador utilizado es el Ywrobot 4 relay, este módulo de relevador puede ser controlado directamente por una amplia gama de microcontroladores tales como Arduino, AVR, PIC, ARM y MSP430. Los 4 relés son incluidos en el módulo, con los puertos NC (Normalmente Cerrado) y CO (Normalmente Abierto), cuenta con 4 leds para mostrar el estado de los relés. Sus especificaciones se muestran en el Cuadro 4.7.

Cuadro 4.6: Modelo de conectividad Bluetooth

Bluetooth HC-06	V1.04 esclavo	V1.2 esclavo	V1.3 esclavo	V1.02 maestro	V1.2 maestro	V1.04 maestro-esclavo	RS232 esclavo
The power to prevent reverse connect	Si	No	Si	No	No	Si	No
Enable	No	Si	Si	No	No	No	No
Led indicator	Si	Si	Si	Si	Si	Si	Si
Status output	Reservado	Si	Si	Si	Si	Si	No
Interface level	3.3V	3.3V	3.3 – 5V	3.3V	3.3V	3.3V	3.3V/RS232
Input voltage	3.6 – 6V	3.6 – 6V	3.6 – 6V	3.6 – 6V	3.6 – 6V	3.6 – 6V	3.6 – 6V
Operating mode	Modo esclavo	Modo esclavo	Modo esclavo	Modo principal	Modo principal	Modo Principal y esclavo	Modo esclavo

Fuente: Elaboración propia.

En la Figura 4.3 se muestra la placa con los 4 relés, que cuenta con Vcc, GND, INT1, INT2, INT3, INT4, un led de encendido y 4 ledes respectivos a cada entrada, cada relé cuenta con su NC y NO, incluso cuenta con 4 orificios para colocar tornillos y ajustarlo a la base.

Cuadro 4.7: Especificaciones del 4 Relay

Especificaciones	Tipo
Tipo de módulo:	Control
Tamaño de la placa:	8x4.8x2cm
Nivel de operación:	5V Digital
Peso:	70g
Versión:	1
Fuente de alimentación:	5V externa

Fuente: Elaboración propia.

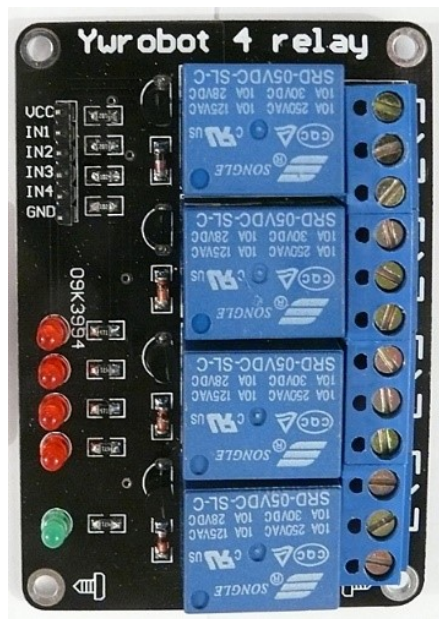


Figura 4.3: Módulo 4 Relay

Fuente: Elaboración propia.

4.1.2. Etapa 4b Monitorear un sistema

Consiste en transformar el modelo obtenido en alguna otra forma de pensamiento sistémico que, dadas las particularidades del problema, pueda ser conveniente, en el que la medida del funcionamiento desde la eficacia de su trabajo, eficiencia del trabajo terminó con los recursos y las metas satisfechas.

Dentro de esta etapa se definen los agentes actores dentro de la situación y los cuales intervienen en el entorno de la impresora flexográfica automatizada con microcontroladores, aquí se especifican las herramientas que se utilizan.

Se establecen los agentes necesarios para la mejora de la impresora flexográfica automatizada con PLC. En esta etapa se recoge y clasifica la información que provee la descripción de la situación del problema, los procesos y transformaciones que se realizan dentro del sistema.

En la Figura 4.4, se muestra la situación estructurada para la impresora flexográfica. La Relación 1: La impresora es adquirida por la institución DGETI CETis; Relación 2: El profesor imparte su asignatura con prácticas en la impresora; Relación 3: El alumno realiza prácticas con la impresora flexográfica; Relación 4: El alumno puede realizar sus prácticas profesionales con conocimientos previos en la industria; Relación 5: La impresora es programada en el software de ARDUINO; Relación 6: Una vez creado el programa, se carga para ejecución en la placa de ARDUINO BUONO UNO R3; Relación 7: Los motores son conectados en el relevador; Relación 8: El bluetooth es conectado en la placa de ARDUINO para espera de enlace; Relación 9: La aplicación es programada en App inventor; Relación 10: La aplicación es disponible para celulares Android; Relación 11: La impresora se activa y es puesta en ejecución; Relación 12: Cuenta con movilidad para su traslado.

La impresora flexográfica contiene un programa que es cargado al dispositivo, es decir, al circuito que se colocará en el Arduino Buono Uno R3, está conectado con un bluetooth, para la comunicación (envío y recepción

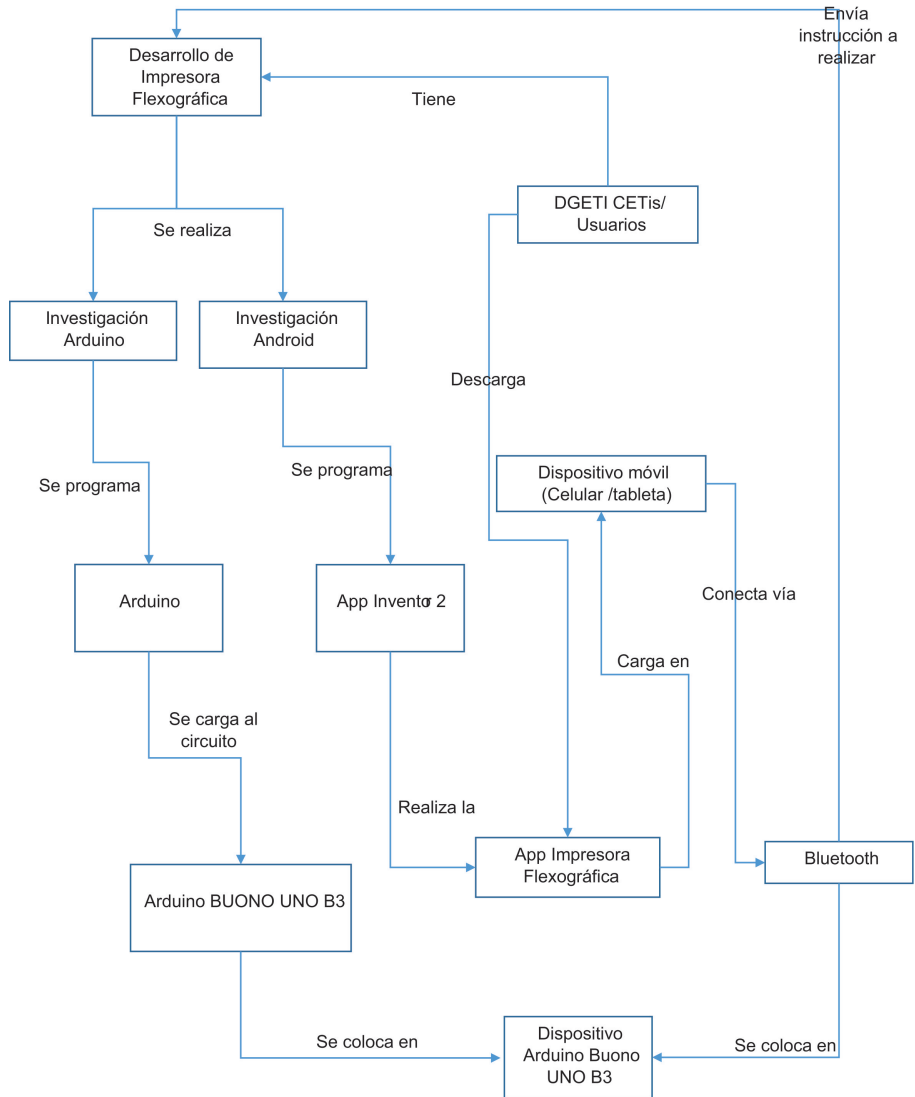


Figura 4.5: Representación gráfica del problema
Fuente: Elaboración propia.

Cuadro 4.8: Funciones de la impresora flexográfica programada en App Inventor.

Funciones	Descripción
Conectar	Busca los Bluetooth con los que ha estado emparejado.
Automático	Activa el Slider para colocar la cantidad del proceso de impresión.
Iniciar	Realiza carga de tinta y comienza el proceso de impresión respecto a la cantidad seleccionada en el Slider.
Cancelar	Cancela el proceso de impresión de Automático.
Carga de tinta	Función forma Manual, activa los actuadores y rodillo entintador, el cual pasa la tinta al rodillo dosificador, para así alimentar el rodillo de impresión. Lo realiza en 1739 milisegundos por tres veces. Muestra un aviso en dado caso que se haya realizado carga de tinta más de dos veces
Imprime	Función manual, Imprime n cantidad de impresiones.
Finalizar	Función manual, Finaliza el proceso de impresión.
Desconectar	Desconecta el Bluetooth.

Fuente: Elaboración propia.

La Figura 4.7 se puede observar cómo se compila el programa, antes de que se ejecute (se cargue a la placa de Arduino).

En la Figura 4.8 se observa que el programa ha sido compilado correctamente.

Ejecutar o cargar el programa del Software Arduino a la placa, se conecta vía cable USB-micro, de esta manera se carga el código ya programado como muestra la Figura 4.9.

Una vez que termina de cargar el programa envía un mensaje como el que se muestra en la Figura 4.10.

```
void setup() {  
  pinMode(13,OUTPUT);  
  // put your setup code here, to run once:  
}  
void loop() {  
  digitalWrite(13,HIGH);  
  // put your main code here, to run repeatedly:  
  delay(500);  
  digitalWrite(13,LOW);  
  delay(150000);  
}
```

Figura 4.6: Ejemplo de código Arduino
Fuente: Elaboración propia.

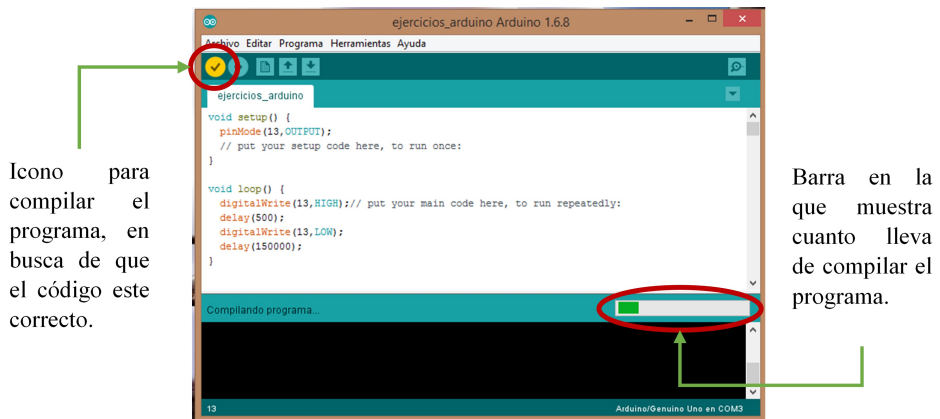


Figura 4.7: Compilación del programa
Fuente: Elaboración propia.

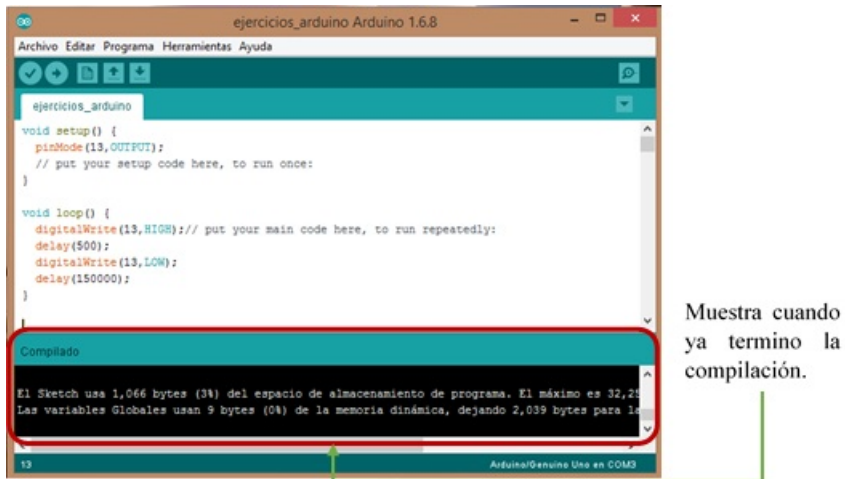


Figura 4.8: Programa compilado
Fuente: Elaboración propia.

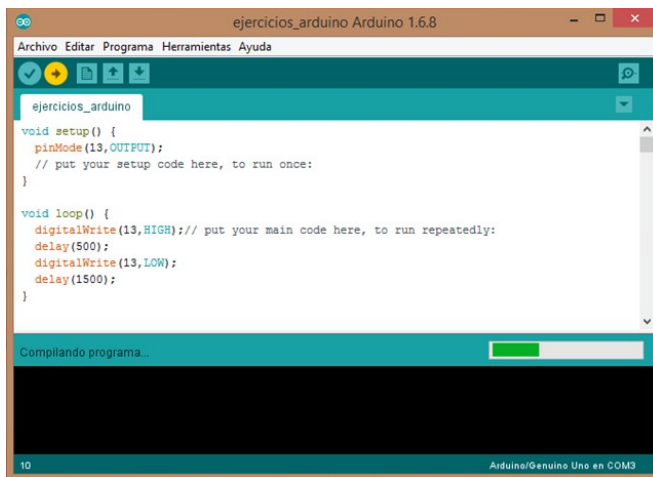
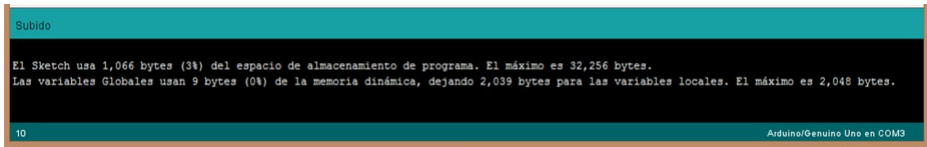


Figura 4.9: Compilando el programa
Fuente: Elaboración propia.



```
Subido
El Sketch usa 1,066 bytes (3%) del espacio de almacenamiento de programa. El máximo es 32,256 bytes.
Las variables Globales usan 9 bytes (0%) de la memoria dinámica, dejando 2,039 bytes para las variables locales. El máximo es 2,048 bytes.
10 Arduino/Genuino Uno en COM3
```

Figura 4.10: Programa subido a la placa Arduino
Fuente: Elaboración propia.

Capítulo 5

Desarrollo

5.1. Programación de Arduino

Este trabajo está orientado al mejoramiento de una impresora flexográfica que opera con un PLC para la potencia y el control; y para su operación, incorpora un panel con interruptores. La propuesta sustituye el PLC por relevadores para la gestión de potencia, un microcontrolador para la gestión de procesos y un teléfono inteligente con sistema operativo Android como interfaz de operación de la impresora. El diagrama a bloques de la Figura 5.1 ilustra los elementos del sistema.

El microcontrolador (Arduino) envía y recibe datos de forma bidireccional, mientras que el bluetooth de Arduino recibe el dato que emite el celular. Con respecto a los relevadores, uno recibe el dato de Arduino para activar al solenoide (actuador), otro recibe el dato del Arduino para activar al Motor del rodillo de carga y el tercero sirve para activar al Motor (rodillo de impresión).

El microcontrolador recibe y envía señales hacia el teléfono mediante el protocolo de comunicación inalámbrica IEEE 802.15.1 WPAN (por sus siglas en inglés Wireless Personal Area Network) bluetooth basada en el estándar Homerf que conecta a los teléfonos celulares entre sí, siempre y cuando estén en un radio cercano (frecuencia de 2.4 GHz). Mediante el programa creado para la operación de la impresora, el usuario puede, me-

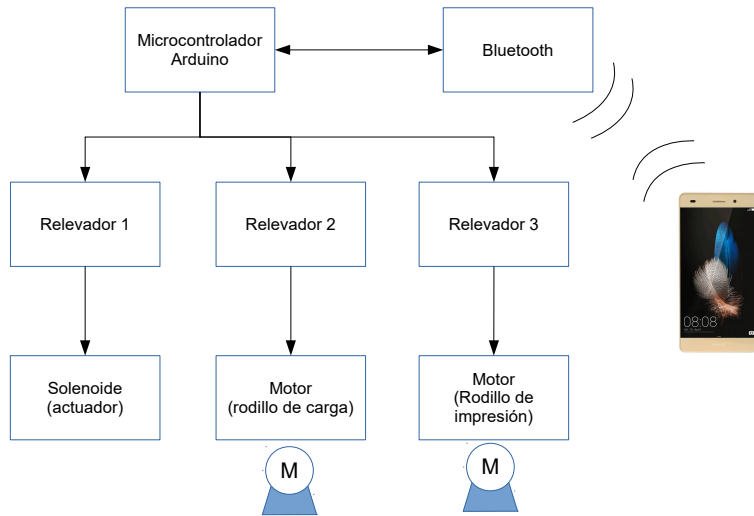


Figura 5.1: Diagrama a bloques de la impresora
Fuente: Elaboración propia.

diante la interfaz, solicitar las acciones que se han incorporado, las cuales son:

- Impresión automática.
- Impresión manual.
- Carga de tinta.
- Fin de impresión.

En la sección 5.3 de este capítulo se hablará a mayor detalle del programa del teléfono.

De acuerdo a la petición realizada por el usuario, el microcontrolador realiza las acciones pertinentes, las cuales se indican a continuación:

5.1.1. Impresión automática

Cuando el usuario solicita una impresión automática, también debe indicar cuántas impresiones requiere, siendo el mínimo una y el máximo nueve. A continuación se muestra Figura 5.2 el diagrama a bloques de las acciones que realiza el microcontrolador cuando el usuario pide una impresión automática.

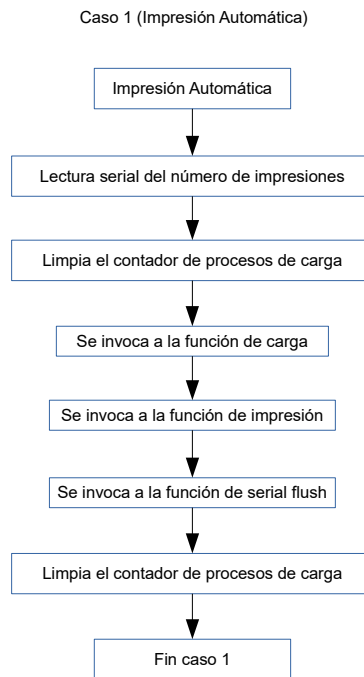


Figura 5.2: Diagrama a bloques de la Función Automática
Fuente: Elaboración propia.

En el caso '1', realiza la función de Automático, y se vuelve a realizar una lectura que indicará la cantidad de impresiones que solicitó el usuario, el contador de cargas se establece en cero, se llama a la funciones carga e imprime, y Serialflush vacía el buffer de datos recibidos, por último se

establece en cero el contador de cargas nuevamente.

```

{
  case '1':    //caso 1 Automático
    cantImpre = Serial.parseInt();
    //valor que recibe el número de vueltas de rodillo
    contCargas=0;
    //valor del contador de cargas lo devuelve a 0
    carga(); //manda llamar la función de carga
    imprime(); //manda llamar la función de imprime
    Serialflush(); //vacía el buffer
    contCargas=0;
    //valor del contador de cargas lo devuelve a 0
    break;

```

En el Apéndice A se muestran los diagramas de flujo que modelan todo el funcionamiento del sistema.

5.1.2. Impresión manual

- Carga de Tinta:

Cuando el usuario solicita realizar carga de tinta. A continuación se muestra 5.3 el diagrama a bloques de las acciones que realiza el microcontrolador cuando el usuario pide carga de tinta.

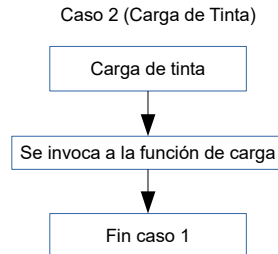


Figura 5.3: Diagrama a bloques de la Función Carga de tinta
Fuente: Elaboración propia.

Los siguientes casos se utilizan en modos de operación manual; el caso '2' llama a la función de carga.

```
case '2': //caso 2 Carga de tinta
    carga(); //manda llamar la función de carga
    break;
```

- Impresión:

Cuando el usuario solicita realizar impresión manual. A continuación se muestra Figura 5.4 el diagrama a bloques de las acciones que realiza el microcontrolador cuando el usuario pide impresión manual.

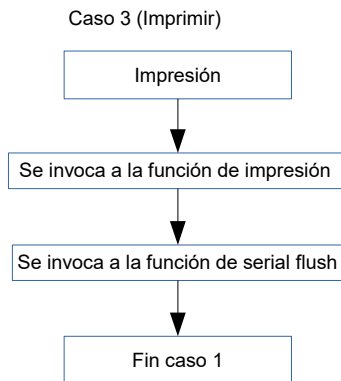


Figura 5.4: Diagrama a bloques de la Función de Impresión
Fuente: Elaboración propia.

El caso '3' llama a la función imprime (está inicia una impresión que se realizará de manera constante), además se invoca a la función Serialflush para vaciar el buffer de datos.

```
case '3': //caso 3 impresión
    imprime(); //manda llamar la función de imprime
    Serialflush(); //vacía el buffer
    break;
```

- Finalizar impresión:

Cuando el usuario solicita realizar finalizar impresión. A continuación se muestra Figura 5.5 el diagrama a bloques de las acciones que realiza el microcontrolador cuando el usuario pide finalizar la impresión manual.

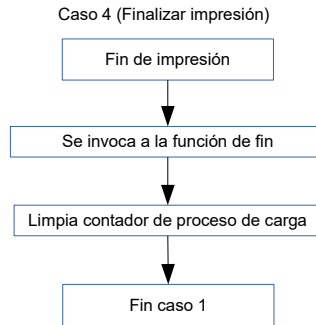


Figura 5.5: Diagrama a bloques de la Función de Finalizar Impresión
Fuente: Elaboración propia.

El caso '4' llama a la función fin, como su nombre lo dice cancela la impresión manual.

```

    case '4': //caso 4 finalizar impresión
        fin(); //manda llamar la función de fin de impresión
        contCargas=0;
        //valor del contador de cargas lo devuelve a 0
        break;
    }
}
}

```

5.2. Código fuente del microcontrolador

En esta sección se muestra y describen las líneas de código de todo el programa que realiza el microcontrolador para el funcionamiento del sistema.

5.2.1. Declaración de bibliotecas y variables globales

La biblioteca `<SoftwareSerial.h>` sirve para realizar transmisiones de datos serialmente; además permite que el buffer de datos de recepción pueda ser vaciado mediante la función “flush”.

La variable “contCargas” se utiliza para llevar un conteo de la cantidad de veces consecutivas que se ha solicitado cargar de tinta al rodillo de impresión, de ser así se enviará al usuario una alerta de que está pidiendo más de dos cargas sin haber realizado impresiones, lo cual no es una situación que se pueda considerar necesaria.

```
#include <SoftwareSerial.h> //librería para usar serial flush
int contCargas=0; //inicializa contador de Cargas en 0
int actuadorCarga = 3; //asignación del nombre al pin 3
int rodilloImpresionCarga = 4;
//asignación del nombre al pin 4
int rodilloImpresion = 5; //asignación del nombre al pin 5
int i = 0; //inicializa i en 0 para switch
int cantImpre = 0; //inicializa cantidad de impresiones en 0
int t= 0; //inicializa t en 0
```

Se declaran los nombres de los pines que servirán para su comunicación con los relevadores y el actuador, donde: al pin 3 del arduino se le asigna el nombre de “actuadorCarga” este se conectará físicamente al relevador del actuador. Al pin 4 del Arduino se le asigna el nombre “rodilloImpresiónCarga” este se conectará físicamente al relevador de carga de tinta. Al pin 5 se le asigna el nombre de “rodilloImpresión” este se conectará físicamente al relevador de impresión. Se declaran las variables globales “cantImpre” la cual recibe la cantidad de impresiones. Se declara las variables globales

i (para estructura de control switch) y *t* (el tiempo en dar una vuelta de rodillo).

5.2.2. Función setup

Esta es una función obligatoria en el IDE de Arduino, es la primera función que se ejecuta y se suele utilizar para configurar los periféricos y puertos que se emplearán durante la ejecución del programa principal y demás funciones. Esta función sólo se ejecuta una vez.

```
void setup()
{
  pinMode(actuadorCarga, OUTPUT);
  //pin actuador de carga activado como salida
  pinMode(rodilloImpresionCarga, OUTPUT);
  //pin rodillo de impresión de carga activado como salida
  pinMode(rodilloImpresion, OUTPUT);
  //pin rodillo de impresión activado como salida
  Serial.begin(9600); //transmisión serial a 9600 bps
}
```

5.2.3. Funciones

El bloque que se muestra a continuación es el cuerpo del programa (void loop), el cual es una función obligatoria de arduino y que se ejecuta ciclicamente de manera indefinida. Se observa un if que está revisando si han llegado datos por el puerto serial (este puerto recibe información del módulo bluetooth), si hay dato disponible se ejecuta el if, si no hay dato no se realizará ninguna acción. De ser válida la condición del if, se procede a leer el dato recibido y se guarda en la variable *i*, la cual es evaluada mediante una estructura switch-case.

```
void loop()
{
  if (Serial.available() > 0) //función para conteo del switch
  {
    i = Serial.read(); //i para monitor serial
    switch (i)          //comienza función de switch
```

Se realiza la función Serial flush para vaciar el buffer de datos, esto es necesario para evitar que se hagan colas de impresión.

```
void Serialflush() //función de Serial flush
{
    while(Serial.available(>0)
    //lee lo que llega al puerto serial
    {
        char i = Serial.read(); //obtiene el caracter
    }
}
```

La siguiente función se utiliza en modo Automático; en el caso '1', se realiza la cuenta para las cargas de rodillo de tinta realizadas, por lo que si ésta excede las dos cargas en su cuenta, a la que sigue, envía un mensaje de aviso, para enseguida revisar que el rodillo de impresión esté desactivado, lo mantiene durante 500 milisegundos, después activa los actuadores y rodillo de carga de tinta. Se declarará trc=1739 (milisegundos) lo que equivale al tiempo que tarda en completar un giro el rodillo de carga, éste se multiplica por 3 veces para que el rodillo tenga una carga completa, lo cual tarda un tiempo total de 5217 milisegundos en completarse, termina y desactiva actuador y rodillo de carga, después incrementa el contador de cargas. Pero si la función carga fue invocada para la impresión manual no se incluye el aviso e incrementa el contador.

```
void carga() //carga para automático
{
    if(contCargas>=2) //si el contCargas es mayor o igual a 2
    { //manda el siguiente mensaje
        //Serial.println("Ya cargaste tinta más de una vez");
        digitalWrite(rodilloImpresion, 0);
        //revisa si el rodillo de impresión está apagado
        delay(500); //espera 500 milisegundos
        digitalWrite(actuadorCarga, 1);
        //activa el actuador de carga
        delay(500); //espera 500 milisegundos
        digitalWrite(rodilloImpresionCarga, 1);
    }
}
```

```

        //activa rodillo de carga
        int trc=1739; //tiempo de 1 giro del rodillo de carga
        int tc=3*trc; //tiempo para completar carga de tinta
        delay(tc); //espera el tiempo 5217 milisegundos
        digitalWrite(rodilloImpresionCarga, 0);
        // desactiva rodillo de carga
        delay(500); //espera 500 milisegundos
        digitalWrite(actuadorCarga, 0);
        //descactiva actuadorCarga
        delay(500); //espera 500 milisegundos
        contCargas++; //incrementa el contador de cargas
    }
else
    {
        digitalWrite(rodilloImpresion, 0);
        //revisa si el rodillo de impresión está apagado
        delay(500); //espera 500 milisegundos
        digitalWrite(actuadorCarga, 1);
        //activa el actuador de carga
        delay(500); //espera 500 milisegundos
        digitalWrite(rodilloImpresionCarga, 1);
        //activa rodillo de carga
        int trc=1739; //tiempo de 1 giro del rodillo de carga
        int tc=3*trc; //tiempo para completar carga de tinta
        delay(tc); //espera el tiempo 5217 milisegundos
        digitalWrite(rodilloImpresionCarga, 0);
        // desactiva rodillo de carga
        delay(500); //espera 500 milisegundos
        digitalWrite(actuadorCarga, 0);
        //descactiva actuadorCarga
        delay(500); //espera 500 milisegundos
        contCargas++; //incrementa el contador de cargas
    }
}

```

La función de imprime, cuando es invocada por el modo Automático, activa el rodillo de impresión, se reinicia el contador de cargas, se declara

el tiempo que tarda en dar un giro de rodillo de impresión (1447 milisegundos), se realiza una multiplicación de dicho tiempo por la cantidad de impresiones solicitadas por el usuario, el resultado se divide en lapsos de 500 milisegundos, esto es para monitorear cada medio segundo la solicitud de cancelación, por lo que se coloca la estructura de control While, ya que se utiliza para ejecutarse mientras que reciba algún dato por bluetooth, dado el caso sea '4' entonces desactiva el rodillo de impresión, espera 500 milisegundos y termina para volverse a utilizar. Si fue de forma Manual y se recibe un '3' activa el rodillo de impresión sin límite hasta que sea cancelado.

```
void imprime() //inicia función imprime
{
    digitalWrite(rodilloImpresion, 1);
    //activa rodillo de impresión
    contCargas=0; //valor del contador de cargas lo devuelve a 0
    if (i == '1') //si se encuentra en '1' Automático
    {
        int t = 1447;
        //duración de una vuelta en milisegundos del rodillo
        de impresión
        int total = t * cantImpre;
        //total de tiempo en realizar las vueltas de impresión
        int numtotal = total/500;
        //lo que se obtiene del total lo divide en secciones de
        500 milisegundos
        int ciclos=0;
        while(ciclos<numtotal+1)
        //se ejecuta mientras reciba dato del bluetooth
        {
            if (Serial.available() > 0)
            //espera que llegue un dato al puerto serial
            {
                int r = Serial.read(); //se obtiene el carácter
                if (r=='4') //si el dato obtenido es '4'
                {
                    digitalWrite(rodilloImpresion,0);
```

```

        //desactiva el rodillo de impresión
        delay(500);
        break;
    }
}
delay(500); //espera 500 milisegundos
ciclos++;
}
digitalWrite(rodilloImpresion, 0);
//desactiva rodillo de impresión
delay(500);
}
else
{ // si solo llego el '3'
    digitalWrite(rodilloImpresion, 1);
    //activa rodillo de impresión
    delay(500); //espera 500 milisegundos
}
}

```

La última función creada es para finalizar o cancelar el funcionamiento del rodillo de impresión.

```

void fin() //función fin de impresión
{
    digitalWrite(rodilloImpresion, 0);
    //desactiva rodillo de impresión
    delay(500); //espera 500 milisegundos
}

```

5.3. Programación App Inventor

En esta sección se muestra la interfaz de la aplicación en Android. En la Figura 5.6 se ilustra cada elemento con la definición de nombre.

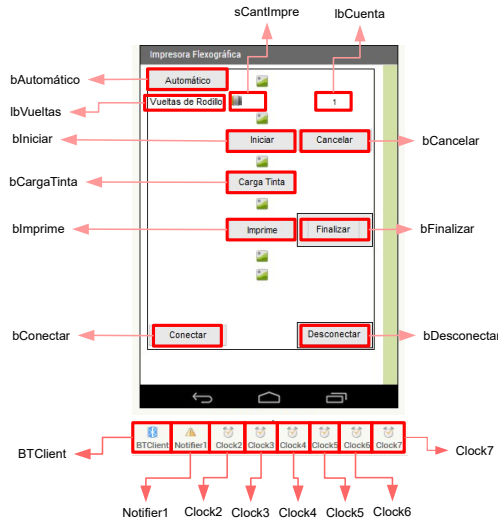
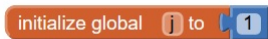
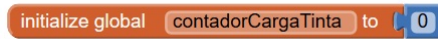


Figura 5.6: Diseño de Interfaz
Fuente: Elaboración propia

En la Figura 5.7 se muestra la programación en bloques para la inicialización de las variables globales en Figura 5.7(a) $j=1$ y en Figura 5.7(b) `contadorCargaTinta=0`.



(a) Variable `j`



(b) Variable contador carga de tinta

Figura 5.7: Variables globales
Fuente: Elaboración propia

Se manda llamar el `.PositionChanged` de `sCargaTinta` (Slider para definir cantidad de impresiones), se redondea su valor con `round` get `thumbPosition` para obtener número entero del 1-9 ver la Figura 5.8.



Figura 5.8: Programación Slider
Fuente: Elaboración propia

La Figura 5.9 muestra la programación del elemento `listPicker` incorporado a la interfaz (antes de que el usuario lo invoque); lo que hace esta función es cargar una lista con todos los dispositivos bluetooth con los que el teléfono se haya vinculado en alguna ocasión.

La Figura 5.10 muestra el después de ser seleccionado el elemento `listPicker` (botón Conectar), que al ser pulsado por el usuario muestra en la pantalla del celular el listado de los dispositivos bluetooth para que seleccione con cuál desea emparejarse, aquí se espera elegir el correspondiente al bluetooth de sistema de impresión. Después la interfaz habilita los botones con los que el usuario podrá interactuar.

La Figura 5.11 Cuando se presiona el botón `bDesconectar` (Figura 5.11(a)), algunos elementos se deshabilitarán, otros se habilitarán, y otros se volverán invisibles al usuario.

La programación para el botón de Automático mostrará algunos elementos habilitados, deshabilitados, con y sin visibilidad Figura 5.11(b).

El botón Iniciar Figura 5.12 se programó, para que al ser pulsado muestre una barra de progreso con el mensaje y cantidad en el que se colocó el Slider con un join y título del mensaje Carga. Algunos elementos se ha-

bilitan, deshabilitan y muestran con y sin visibilidad 5.13(a) y al Clock6 5.13(b) el cual realiza la operación de la posición del Slider multiplicada por el tiempo que tarda en dar una vuelta el rodillo de impresión, a ese resultado se le suma 7800 milisegundos y el resultado obtenido al finalizar la carga e impresión volverá habilitar los botones que contiene Clock6. Revisará que el dato recibido sea 1, redondee el Slider y de un salto de línea.

La Figura 5.13 muestra la programación para Clock5, el intervalo que tiene el temporizador es de 7500 milisegundos por lo que después de ese tiempo quita el mensaje en pantalla de proceso de la carga de tinta que realiza, deshabilita botones bAutomático, bIniciar, bCargaTinta e bImprime, para bFin (habilita visibilidad y deshabilita botón) y por último al botón bCancelar (habilita botón y visibilidad) ver Figura 5.13(a). El Clock6, una vez que obtiene el resultado de la operación que se realiza de redondeo de la posición del Slider (sCantImpre), multiplicado por 1447 milisegundos. Entonces el Clock6, habilita y muestra los botones: bAutomático, bCargaTinta, bImprime; y hace visibles pero inhabilitados: bCancelar y bFin; y por último se hacen invisibles: lbVueltas, sCantImpre, lbCuenta y bFin ver Figura 5.13(b)..

La Figura 5.14 muestra el botón bCancelar, al darle click coteja la conexión del bluetooth, muestra una barra de progreso con el mensaje de la cancelación del proceso, el redondeo de la posición del Slider sCantImpre y el título. Se habilita el Clock4. Revisa que el dato enviado al bluetooth sea el 4 ver Figura 5.14(a).

El Clock4 estará deshabilitado hasta que es mandado llamar por bCancelar, llama al Notifier1 para que la barra de progreso pare, los elementos que deja sin visibilidad (lbVueltas, sCantImpre, lbCuenta) en pantalla. Los elementos habilitados (bAutomático, bCancelar, bCargaTinta y bImprime) en pantalla. Deshabilita bFin y el botón bCancelar lo habilita y deshabilita su visibilidad, ver Figura 5.14(b).

Dentro de la programación del botón bCargaTinta (forma Manual) al dar click incrementa el contador a 1, envía el dato 2 al bluetooth deshabilita los botones bAutomático, bImprime y bCargaTinta. Al pasar el contador

de dos veces (0 y 1) muestra un mensaje en pantalla, realiza la carga correspondiente, muestra la barra de progreso de la carga de tinta, habilita el Clock 2 y Clock 3.

Si se da click al botón `bCargaTinta` muestra una barra de progreso, habilita `Clock2` y `Clock3`, deshabilita `bAutomático`, `bImprime` y `bCargaTinta`. El valor que envía por el bluetooth es 2. Al concluir el `If` de bluetooth, incrementa a `contadorCargaTinta + 1` ver Figura 5.15.

En la Figura 5.16, el `Clock2` estará deshabilitado hasta que lo llame `bCargaTinta`, deshabilita a los botones `bAutomático`, `bIniciar`, `bCargaTinta`, `bImprime` y `bFin`, llama a `Notifier1`, quita la barra de progreso 5.16(a). El `Clock3` estará deshabilitado hasta que lo invoque `bCargaTinta`, mientras que los botones `bAutomático`, `bCargaTinta` e `Imprime` son habilitados, los botones `bIniciar` y `bFin` son deshabilitados, muestra la pantalla principal de la aplicación 5.16(b).

El botón `bImprime` revisa si esta conectado el bluetooth, para deshabilitar los botones `bAutomático`, `bCargaTinta`, `bImprime` y habilitar el botón `bFin`. El dato que envía al bluetooth es 3, mientras que el `contadorCargaTinta` lo reinicia a 0, ver Figura 5.17.

El botón de `bFin`, al darle click, revisa si está conectado el bluetooth, llama al `Notifier1` mostrando una barra de progreso de que se realiza cancelación, Habilita el `Clock4`, y por último envía el dato 4 por el bluetooth ver Figura 5.18.

En siguiente Figura 5.19 se observa el total de los bloques utilizados y las variables globales declaradas.

El usuario proporcionó el fondo para la aplicación, por lo que el diseño final queda como puede verse en la Figura 5.20.

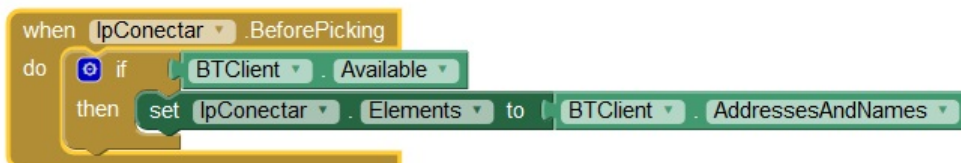


Figura 5.9: List Picker IpConectar antes
Fuente: Elaboración propia

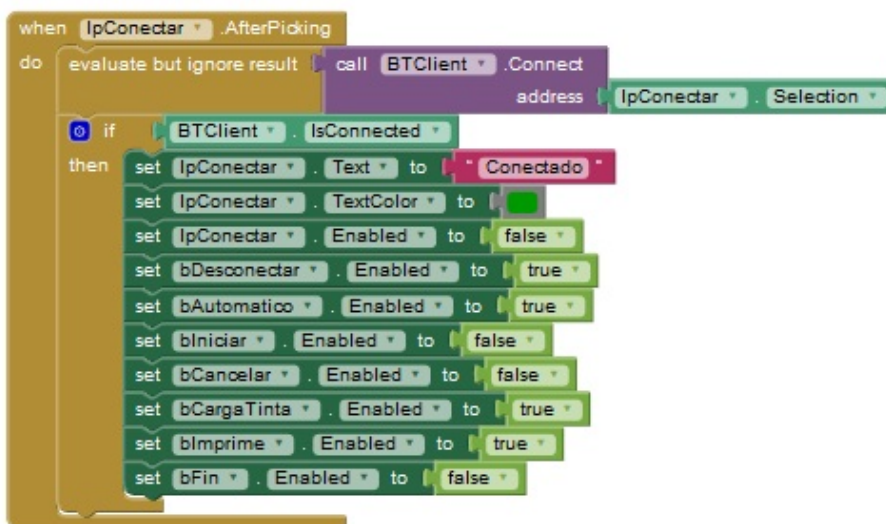


Figura 5.10: List Picker IpConectar después
Fuente: Elaboración propia

```

when bDesconectar .Click
do
  call BTClient .Disconnect
  set IpConectar .Text to "Conectar"
  set IpConectar .TextColor to 
  set IpConectar .Enabled to true
  set bDesconectar .Enabled to false
  set bAutomatico .Enabled to false
  set lbVueltas .Visible to false
  set lbCuenta .Visible to false
  set sCantImpre .Visible to false
  set bIniciar .Enabled to false
  set bCancelar .Enabled to false
  set bCargaTinta .Enabled to false
  set blprime .Enabled to false
  set bFin .Enabled to false

```

(a) bDesconectar

```

when bAutomatico .Click
do
  if BTClient .IsConnected
  then
    set lbVueltas .Visible to true
    set lbCuenta .Visible to true
    set sCantImpre .Visible to true
    set sCantImpre .ThumbEnabled to true
    set bIniciar .Enabled to true
    set bIniciar .Visible to true
    set bCancelar .Enabled to true
    set bCancelar .Enabled to true
    set bAutomatico .Enabled to false
    set bCargaTinta .Enabled to false
    set blprime .Enabled to false
    set bFin .Enabled to false
    set bFin .Visible to false

```

(b) bAutomático

Figura 5.11: Botones Desconectar y Automático
Fuente: Elaboración propia

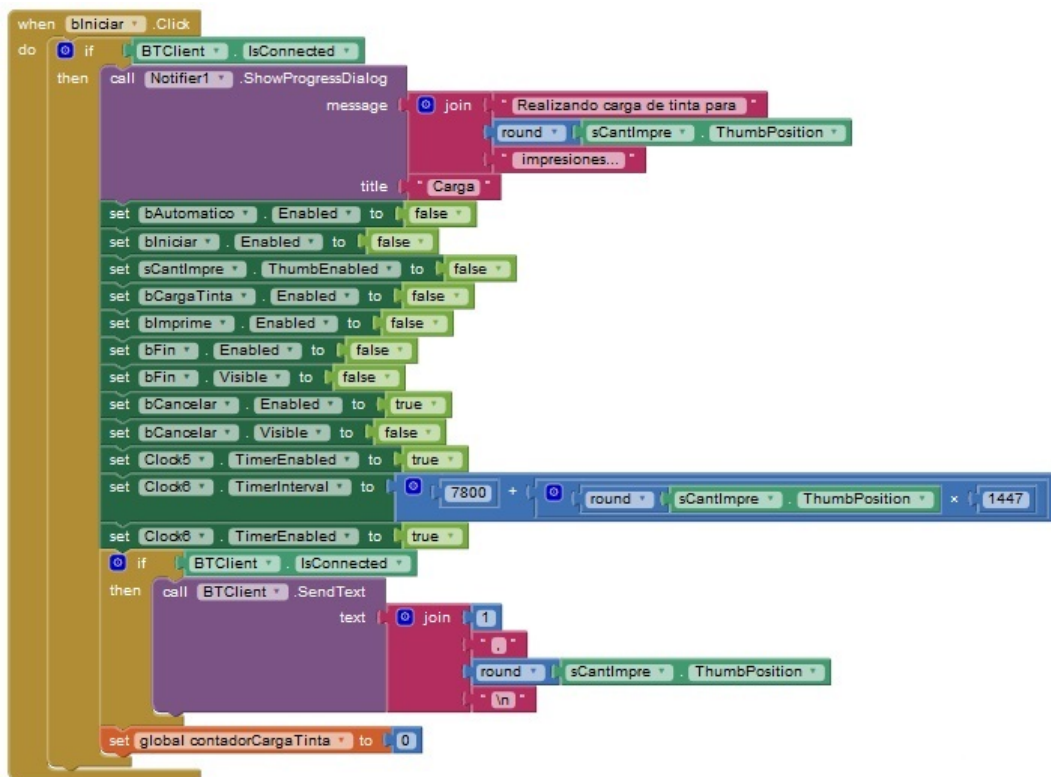


Figura 5.12: Botón bIniciar
Fuente: Elaboración propia

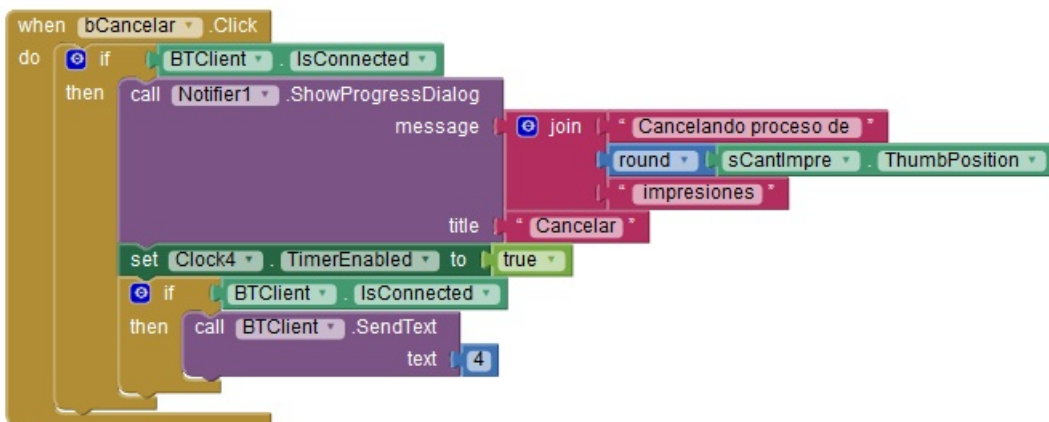
```
when Clock5.Timer
do
  set Clock5.TimerEnabled to false
  call Notifier1.DismissProgressDialog
  set bAutomatico.Enabled to false
  set bIniciar.Enabled to false
  set bCargaTinta.Enabled to false
  set bImprime.Enabled to false
  set bCancelar.Enabled to true
  set bCancelar.Visible to true
  set bFin.Visible to true
  set bFin.Enabled to false
```

(a) Clock5

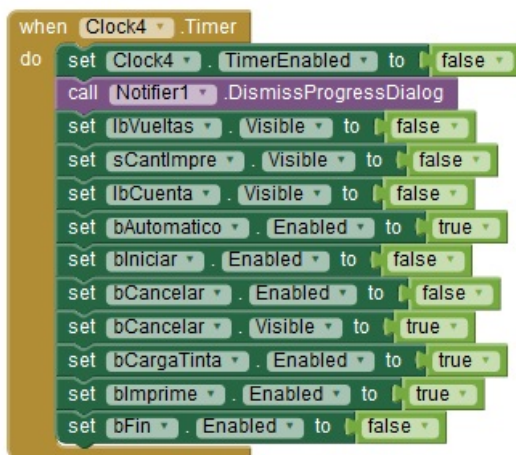
```
when Clock6.Timer
do
  set Clock6.TimerEnabled to false
  set bAutomatico.Enabled to true
  set bIniciar.Enabled to false
  set bCancelar.Visible to true
  set bCancelar.Enabled to false
  set lbVueltas.Visible to false
  set sCantImpre.Visible to false
  set lbCuenta.Visible to false
  set bCargaTinta.Enabled to true
  set bImprime.Enabled to true
  set bFin.Enabled to false
  set bFin.Visible to true
```

(b) Clock6

Figura 5.13: Clock 5 y 6
Fuente: Elaboración propia



(a) Botón bCancelar



(b) Clock4

Figura 5.14: Clock 5 y 6
Fuente: Elaboración propia

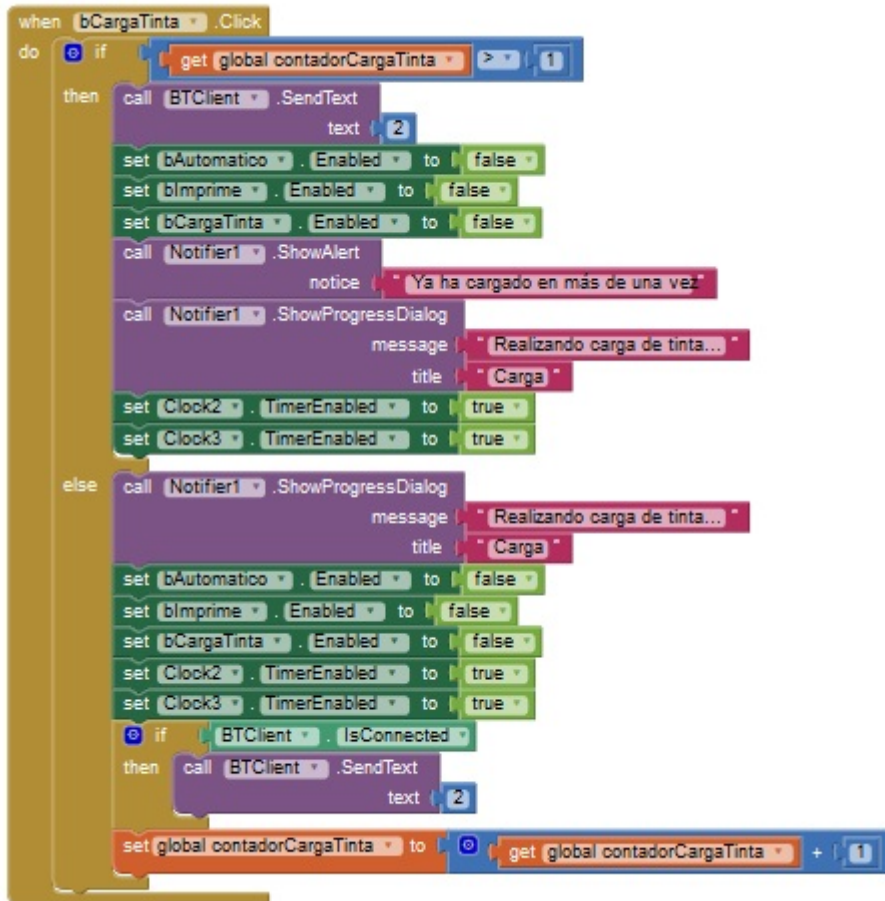


Figura 5.15: Botón bCargaTinta
Fuente: Elaboración propia



```
when Clock2 .Timer
do
  set Clock2 . TimerEnabled to false
  set bAutomatico . Enabled to false
  set bIniciar . Enabled to false
  set bCargaTinta . Enabled to false
  set bImprime . Enabled to false
  set bFin . Enabled to false
  call Notifier1 .DismissProgressDialog
```

(a) Clock2



```
when Clock3 .Timer
do
  set Clock3 . TimerEnabled to false
  set bAutomatico . Enabled to true
  set bIniciar . Enabled to false
  set bCargaTinta . Enabled to true
  set bImprime . Enabled to true
  set bFin . Enabled to false
```

(b) Clock3

Figura 5.16: Clock 2 y 3
Fuente: Elaboración propia

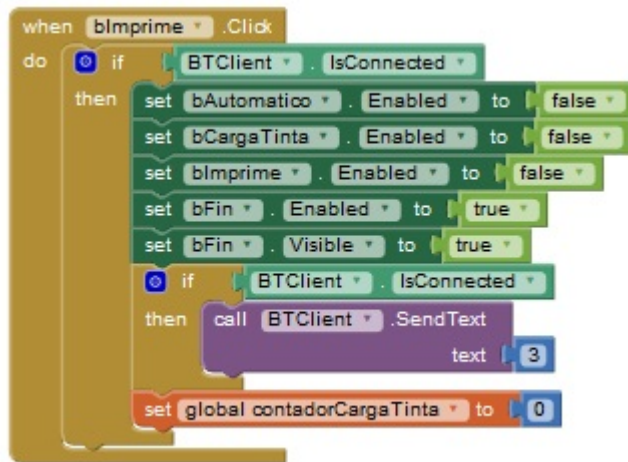


Figura 5.17: Botón bImprime
Fuente: Elaboración propia

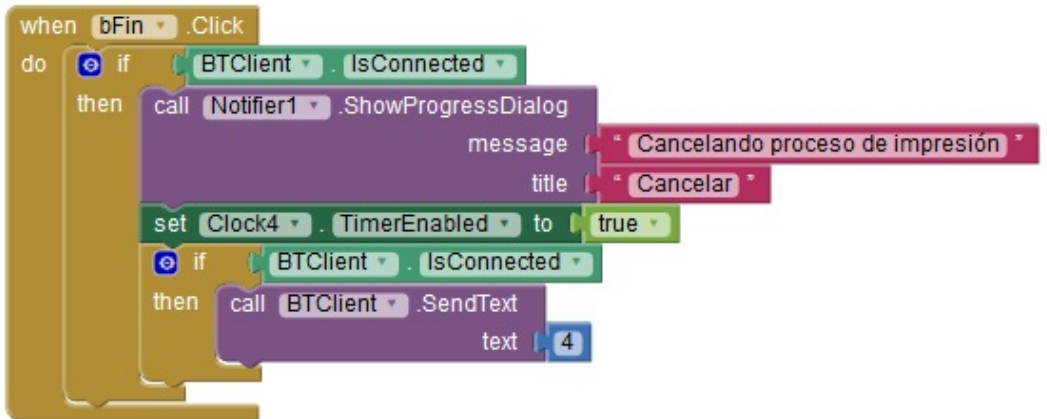


Figura 5.18: Botón bFin
Fuente: Elaboración propia

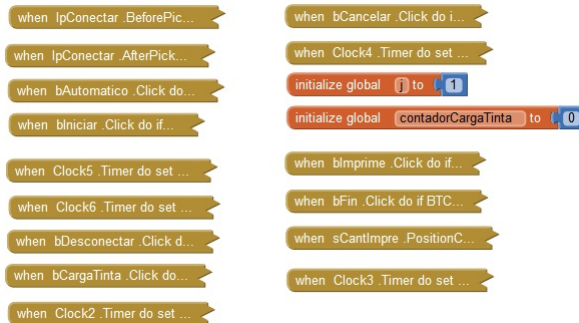


Figura 5.19: Bloques totales para programación
Fuente: Elaboración propia



Figura 5.20: Aplicación final
Fuente: Elaboración propia

Capítulo 6

Comparación de modelos

6.1. Etapa 5 Comparar modelos conceptuales con realidad

Se puede observar en el cuadro 6.1, la comparación que se realizó con la impresora flexográfica profesional, la impresora portátil programada con PLC (realizada como trabajo de titulación de la Carrera de Ingeniería) y la impresora flexográfica portátil programada con microcontroladores), como puede observarse se consideraron desde la dimensión hasta el software de programación para cada una.




6.2. Opiniones

Se realizaron las pruebas de uso con expertos a través de las opiniones de la aplicación y funcionamiento de la impresora flexográfica automatizada con microcontroladores.

- Opinión 1:

Es evidente que la Tecnología de la Información y Comunicación (TIC's) ha avanzado. Me gusta la aplicación del celular mediante

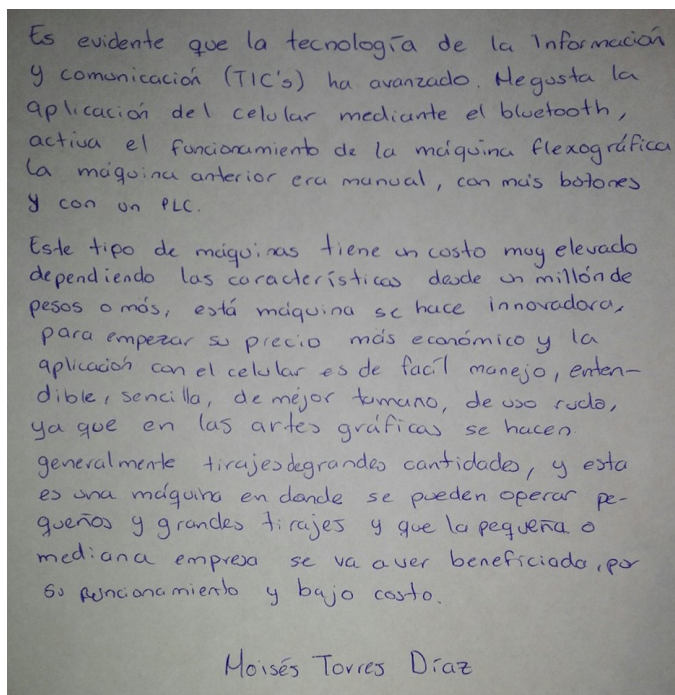
Cuadro 6.1: Comparación entre impresoras flexográficas

Características	Impresora Flexográfica Profesional	Impresora Flexográfica Automatizada con PLC	Impresora Flexográfica automatizada con microcontroladores
Figura			
Ancho máx. del pliego (cm)	100	18	18
Peso total (Kg)	5000	10	7
Dimensiones (cm)	510x206x295	97x51x40	80x60x36
Sistema de registro (matriz)	Si	Si	Si
Rodillos anilox	Si	Si	Si
Motores	Si	Si	Si
Microcontrolador	No especificado	Controlador Lógico Programable (PLC)	Arduino Buono UNO R3 ATmega328 TQFP-32
Mando de Control	Si	Si	Si
Relevadores	No	2	3
Portátil	No	Si	Si
Cantidad de tintas de impresión	4, 6 y 8	1	1
Cliché	Si	Si	Si
Costo (\$)	1,385,500	30,000	25,000
Aplicación para control (móvil)	No	No	Si
Bluetooth	No	No	Si
Movilidad (ruedas)	No	No	Si
Software de programación	No especificado	Step7 MicroWin	Arduino y App Inventor (para Android)

Fuente: Elaboración propia

el bluetooth, activa el funcionamiento de la máquina flexográfica. La máquina anterior era manual, con más botones, con un PLC.

Este tipo de máquinas tiene un costo muy elevado dependiendo las características desde un millón de pesos o más, esta máquina se hace innovadora para empezar su precio más económico y la aplicación con el celular es de fácil manejo, entendible, sencilla, de mejor tamaño, de uso rudo, ya que en las artes gráficas se hacen generalmente tirajes de grandes cantidades, y ésta es una máquina en donde se pueden operar pequeños y grandes tirajes y que la pequeña o mediana empresa se va a ver beneficiada, por su funcionamiento y bajo costo ver Figura 6.1.



Es evidente que la tecnología de la Información y comunicación (TIC's) ha avanzado. Me gusta la aplicación del celular mediante el bluetooth, activa el funcionamiento de la máquina flexográfica. La máquina anterior era manual, con más botones y con un PLC.

Este tipo de máquinas tiene un costo muy elevado dependiendo las características desde un millón de pesos o más, esta máquina se hace innovadora, para empezar su precio más económico y la aplicación con el celular es de fácil manejo, entendible, sencilla, de mejor tamaño, de uso rudo, ya que en las artes gráficas se hacen generalmente tirajes de grandes cantidades, y esta es una máquina en donde se pueden operar pequeños y grandes tirajes y que la pequeña o mediana empresa se va a ver beneficiada, por su funcionamiento y bajo costo.

Moisés Torres Díaz

Figura 6.1: Opinión 1
Fuente: Elaboración propia

- Opinión 2:

Soy usuario de una máquina flexográfica, éstas máquinas son muy complicadas, considero que, cuando me mostraron la primer máquina era totalmente manual, posteriormente le pusieron un PLC (Controlador Lógico Programable) y una adaptación hacía la computadora.

Ahora es más sofisticado, me llama la atención un paquete de una tablilla que utiliza y que sustituyó al Controlador Lógico Programable, en este caso están aplicando un arduino que es una programación más amigable, más fácil, menos robusta; me gusta la interacción con un celular que es una aplicación de encendido, paro, arranque, cierto número de impresiones, carga de tinta, cosas que no se podían hacer con el PLC. Simplemente al inicio era sólo girar la manivela y comenzaba a imprimir.

Creo que muchos no conocen el beneficio de esta máquina, ya que en el mercado empresarial son muy caras tienen un costo desde el medio hasta un millón, incluso están a la venta en dólares dependiendo las características, por eso esta máquina se me hace muy eficaz e innovadora por la planeación y programación. Cualquier persona aunque no conozca de artes gráficas con un manual de operación lo pueden operar ver Figura 6.2.

- Opinión 3:

Teniendo en cuenta los avances que se producen con respecto a las Tecnologías de la Información y Comunicación éste proyecto ofrece un cambio al momento de realizar trabajos gráficos, ya que en un momento son de acceso exclusivo y ciertamente no permitían un manejo en general para los usuarios por su exhuberante tamaño, costo y complicado manejo.

Este tipo de máquina resuelve en gran medida estas desventajas, ya que es práctica, económica, de mejor tamaño, y con la integración de

SOY USUARIO DE UNA MAQUINA FLEXOGRAFICA, ESTAS MAQUINAS SON MUY COMPLICADAS, CONSIDERO QUE CUANDO ME MOSTRARON LA PRIMER MAQUINA ERA TOTALMENTE MANUAL, POSTERIORMENTE LE PUSIERON UN PLC (CONTROLADOR LOGICO PROGRAMABLE) Y UNA ADAPTACION HACIA LA COMPUTADORA.

AHORA ES MAS SOFISTICADO, ME LLAMA LA ATENCION UN PAQUETE DE UNA TABULLA QUE UTILIZA Y QUE SUSTITUYO AL CONTROLADOR LOGICO PROGRAMABLE, EN ESTE CASO ESTAN APLICANDO UN ARDUINO QUE ES UNA PROGRAMACION MAS AMIGABLE, MAS FACIL, MENOS ROBUSTA, ME GUSTA LA INTERACCION CON UN CELULAR QUE ES UNA APLICACION DE ENCENDIDO, PERO, AUNQUE, CIERTO NUMERO DE IMPRESIONES, CARGA DE TINTA, COSAS QUE NO SE PODIA HACER CON EL PLC. SIMPLEMENTE AL INICIO ERA SOLO GIRAR LA MANIVELA Y COMENZABA A IMPRIMIR.

CREO QUE MUCHOS NO CONOCEN EL BENEFICIO DE ESTA MAQUINA, YA QUE EN EL MERCADO EMPRESARIAL SON MUY CARAS, TIENEN UN COSTO DESDE EL MEDIO HASTA UN MILLON, INCLUSO ESTAN A LA VENTA EN DOLARES DEPENDIENDO LAS CARACTERISTICAS, POR ESO ESTA MAQUINA SE ME HACE MUY EFICAZ E INNOVADORA POR LA PLANEACION Y PROGRAMACION, CUALQUIER PERSONA AUNQUE NO CONOZCA DE ARTES GRAFICAS CON UN MANUAL DE OPERACION LO PUEDEN OPERAR.

FELIPE GARNICA BARRIOS

Figura 6.2: Opinión 2
Fuente: Elaboración propia

uno de los artefactos más utilizados por los usuarios en la actualidad que es el teléfono celular hace que la máquina conlleve una implementación de tecnologías que están a la vanguardia como es el uso del bluetooth, placa Arduino, aplicaciones móviles que este último se emplea como control remoto que transmite las instrucciones de forma eficaz y con un rango de distancia más aceptable, es sencillo y práctico de usar. Está a la mano del usuario en general, con esto y más hace que cualquier persona pueda acceder a esta herramienta y pueda a la vez aprender de las artes gráficas ver Figura 6.3.

Opinión 4:

La impresora flexográfica es una herramienta de gran apoyo ya que sus dimensiones son menores a las convencionales. El haber implementado una app para el manejo de las funciones de dicha impresora la hace más sencilla de utilizar.

El contar con la opción de realizar el proceso automáticamente solo indicando el número de impresiones a realizar, permite visualizar como hace el proceso y solo esperar a que esten las impresiones solicitadas o bien tener la opción de hacer el proceso manual encargarte de cargar tinta y después mandar a imprimir.

Considero que los cambios fueron novedosos al incorporar una app ya que nos encontramos en una etapa de constantes cambios ver Figura 6.4.

Teniendo en cuenta los avances que se producen con respecto a los tecnológicos de la información y comunicación este proyecto que ofrece un cambio al momento de realizar trabajos gráficos ya que en un momento fueron de acceso exclusivo y espontáneamente no permitían un manejo en general para los usuarios por su exuberante tamaño, costo y su complicado manejo.

Este tipo de máquina resuelve en gran medida estos desventajas, ya que es práctica, económica, de mejor tamaño y con la integración de uno de los artefactos más utilizados por los usuarios en la actualidad que es el teléfono celular hace que la máquina conlleve una implementación de tecnologías que están a la vanguardia como es el uso del bluetooth, placa Arduino, aplicaciones móviles que este último se emplea como control remoto que transmite las instrucciones de forma eficaz y con un rango de distancia más que aceptable, y que es sencillo y práctico de usar y que está a la mano del usuario. En general, con esto y más hacen que cualquier persona pueda acceder a esta herramienta y pueda a la vez aprender de los artes gráficos.

Luis Enrique Maldonado Navarro

Figura 6.3: Opinión 3
Fuente: Elaboración propia

La impresora flexográfica es una herramienta de gran apoyo ya que sus dimensiones son menores a las convencionales. El haber implementado una app para el manejo de las funciones de dicha impresora la hace más sencilla de utilizar.

El contar con la opción de realizar el proceso automáticamente solo indicando el número de impresiones a realizar, permite visualizar como hace el proceso y solo esperar a que estén las impresiones solicitadas, o bien tener la opción de hacer el proceso manual encargarte de cargar tinta y después mandar a imprimir.

Considero que los cambios fueron novedosos al incorporar una app ya que nos encontramos en una etapa de constantes cambios.

Saúl Rubio Rosas.

Figura 6.4: Opinión 4
Fuente: Elaboración propia

Capítulo 7

Conclusiones

Etapa 6 Cambios factibles y deseables

Durante la realización del desarrollo de la impresora flexográfica automatizada con microcontroladores, se realizaron varios cambios, ya que el PLC (Controlador Lógico Programable) era de tamaño amplio para la impresora, adicionalmente analizando las encuestas realizadas fue evidente la necesidad de incorporar un elemento para facilitar desplazar la impresora de un lugar a otro, por lo que se modificó el diseño para incorporar cuatro ruedas, lo cual facilita en gran medida la movilidad de la impresora.

Otro cambio estructural, consistió en reemplazar el material de la base de la impresora, que al ser de madera comprimida, tuvo problemas de deformación principalmente provocados por humedad y el mismo peso de la impresora. La nueva propuesta consistió en el uso de madera y formáica, proporcionando así mayor resistencia y solidéz.

Con respecto a la etapa de control y potencia, se pudo comprobar funcionalmente que para este sistema, el uso de microcontroladores y relevadores, pueden reemplazar por completo al PLC que tenía la impresora.

Con respecto a la operación, la sustitución de un panel físico por una aplicación programada para el sistema operativo Android, ha sido satisfactoria, ya que es posible operar a la impresora con cualquier teléfono o tableta en

el que se pueda instalar la aplicación.

Por lo que cabe mencionar que los cambios incorporados derivaron en una impresora, de menor peso, menor tamaño, mejor movilidad, menor costo y que es más fácil y conveniente su operación por parte de los estudiantes. Por lo tanto se concluye que los objetivos planteados fueron cumplidos de manera satisfactoria.

Etapa 7 Implantación de cambios

Implementar el diseño, donde los usuarios del sistema deberán verificar si el sistema soluciona la situación problemática o se realizan cambios al mismo.

Durante las pruebas, los usuarios que utilizarán la impresora flexográfica de ambos planteles educativos para optimizar sus clases y unidades de aprendizaje, opinaron que es práctica para sus necesidades, ya que los alumnos pueden aprender de mejor forma, optimizar sus conocimientos tanto teóricos como prácticos, ya que es lo solicitado dentro del ámbito empresarial.

A su vez, los usuarios aprobaron el sistema tanto de forma física como lógica, al obtener la aplicación del celular ya que la mayoría de sus alumnos cuenta con dispositivos móviles con Sistema Android, la aplicación es amigable con el usuario y cuenta con los manuales de operación y mantenimiento dentro de la aplicación.

De igual forma los usuarios indicaron que el uso de la placa Arduino implica el consumo de menores recursos económicos y físicos; y que la decisión de implementarla tiene como consecuencia su posible adopción en proyectos de características similares.

Apéndice A

Diagramas de Flujo

A.1. Arduino

Se desarrollaron los diagramas de flujo para la programación de Arduino, en el que la Figura A.1 se observa la configuración de transmisión serial, configuración de los pines 3 para actuador de carga, 4 a rodillo de impresión de carga y 5 para rodillo de impresión, se declaran las variables globales, lee los datos que recibe el puerto serial. Si llega el dato '1' asigna la cantidad de impresiones.

En la Figura A.2 inicializa el valor de la cantidad de impresiones a cero, manda llamar la función de carga, imprime y Serialflush; se establece el contador de cargas a cero y regresa a revisar si llegó dato. Si el dato que llegó fue '2' manda llamar la función de carga. Si el dato que llegó fue '3' manda llamar la función de imprime y Serialflush. Si el dato que llegó fue '4' manda llamar la función de fin y establece el contador de cargas a cero. Al final cada uno revisa si llegó dato.

La Figura A.3 se observa el diagrama de flujo de la función de carga de tinta, en el que si el contador de carga es mayor o igual a 2 entonces, pone en cero el rodillo de impresión, espera 500 milisegundos, actuador de carga se prende, espera 500 milisegundos, rodillo de impresión de carga se prende, espera 500 milisegundos, se declara tiempo de rodillo de carga $trc=1739$, tiempo de carga $tc=3*trc$, espera tc milisegundos, actuador de

carga se apaga y continúa en la parte 2. Si el contador de carga no es igual o mayor de 2, entonces pone en cero el rodillo de impresión, espera 500 milisegundos, actuador de carga se prende, espera 500 milisegundos, rodillo de impresión de carga se prende, espera 500 milisegundos, se declara tiempo de rodillo de carga $trc=1739$, tiempo de carga $tc=3*trc$ continúa en la parte 1.

La Figura A.4 la continuación de la parte 2 espera 500 milisegundos e incrementa el contador de cargas y finaliza. La parte continuación de la parte 1 espera el tiempo de carga en milisegundos, rodillo de impresión de carga se apaga, espera 500 milisegundos, actuador de carga lo apaga, espera 500 milisegundos e incrementa el contador de cargas y finaliza su función.

La figura A.5 muestra el rodillo de impresión se pone en 1, el contador de cargas en cero, Si i es igual a '1', entonces se declara $t=1447$, se declara total igual al tiempo de t por la cantidad de impresiones seleccionada, se declara ciclos igual a 0. Comienza si el ciclo es menor que el $numtotal + 1$, entonces revisa si llegó algún dato y continúa en la parte 1. Si no continúa en la parte 2. Si el dato recibido no es '1', entonces pone al rodillo de impresión prendido, espera 500 milisegundos, continúa en la parte 4. Si no llegó dato pasa a la parte 2. La parte 3 regresa a revisar el ciclo.

La continuación de la parte 1 se declara r para recibir dato serial. Si el dato que recibe es '4', entonces el rodillo de impresión se apaga, espera 1000 milisegundos, incrementa a ciclos, rodillo de impresión lo apaga, espera 500 milisegundos y regresa a la parte 3, que revisa si llegó dato. La continuación de la parte 2 espera 500 milisegundos, incrementa a ciclos, apaga rodillo de impresión, espera 500 milisegundos y termina su función ver Figura A.6

La Figura A.7 para la función de serial flush revisa si hay dato en el buffer serial, si lo hay limpia el dato que llegó. Si no regresa a revisar si hay dato serial. En la función de fin desactiva el rodillo de impresión, espera 500 milisegundos y termina su función.

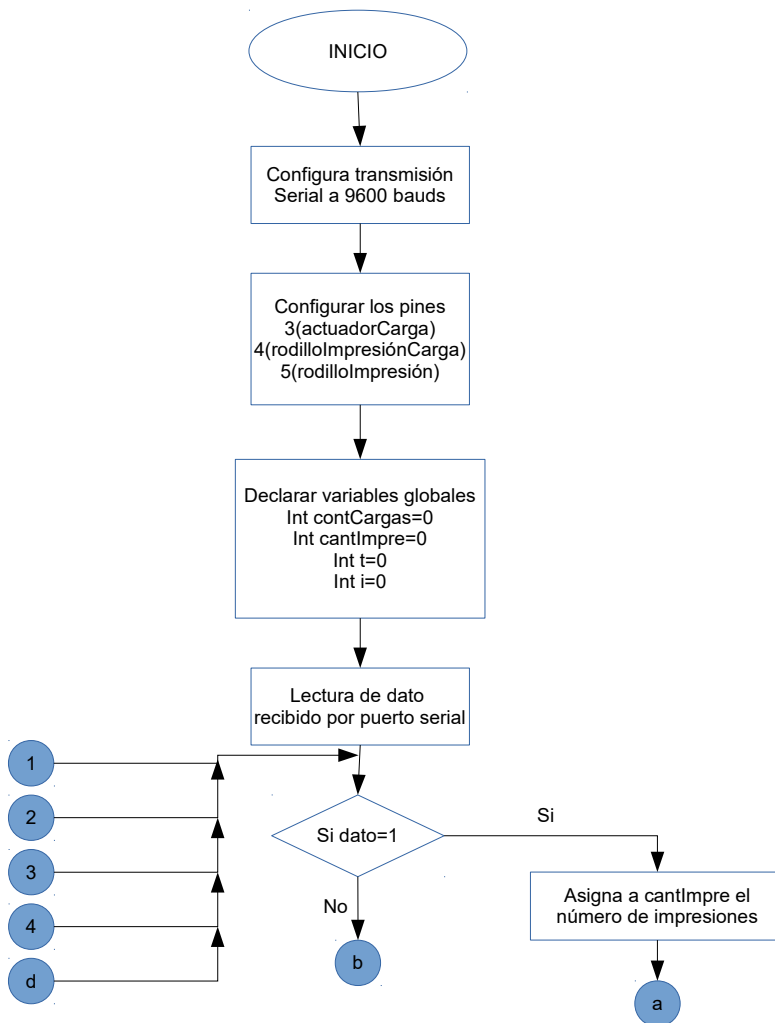


Figura A.1: Diagrama de Flujo Programa Final parte 1
Fuente: Elaboración propia.

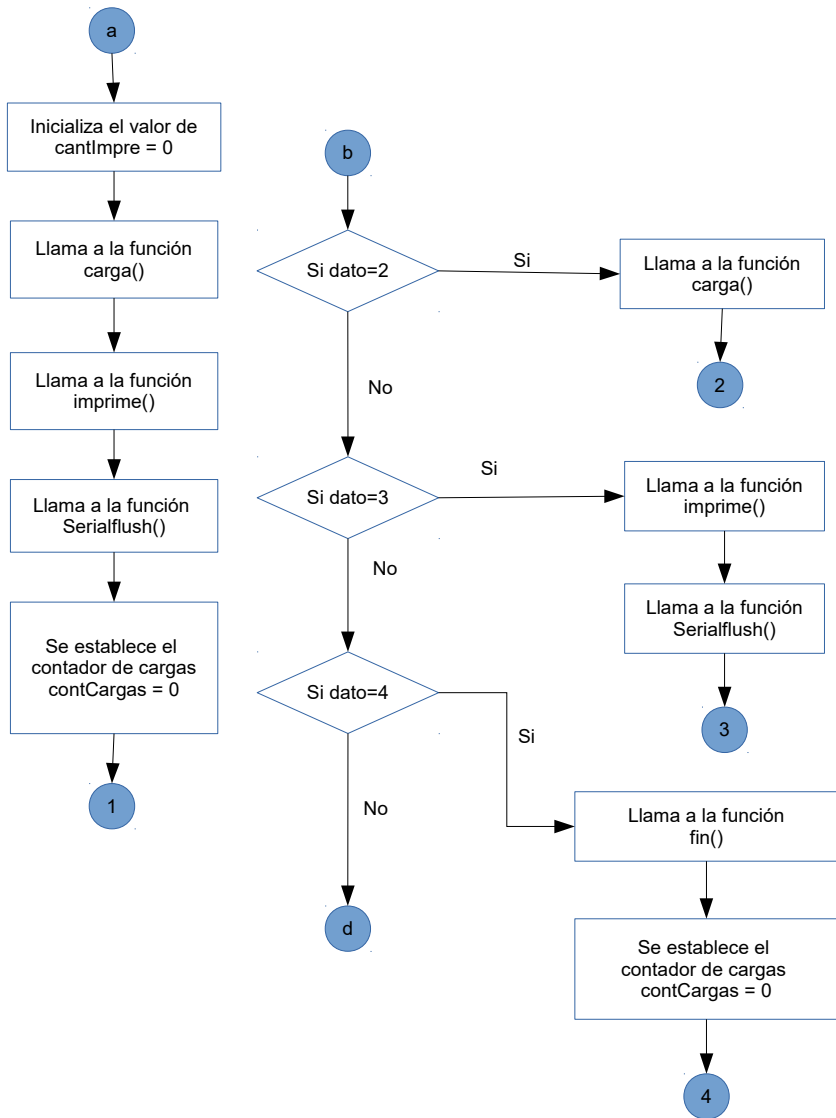


Figura A.2: Diagrama de Flujo Programa Final parte 2

Fuente: Elaboración propia.

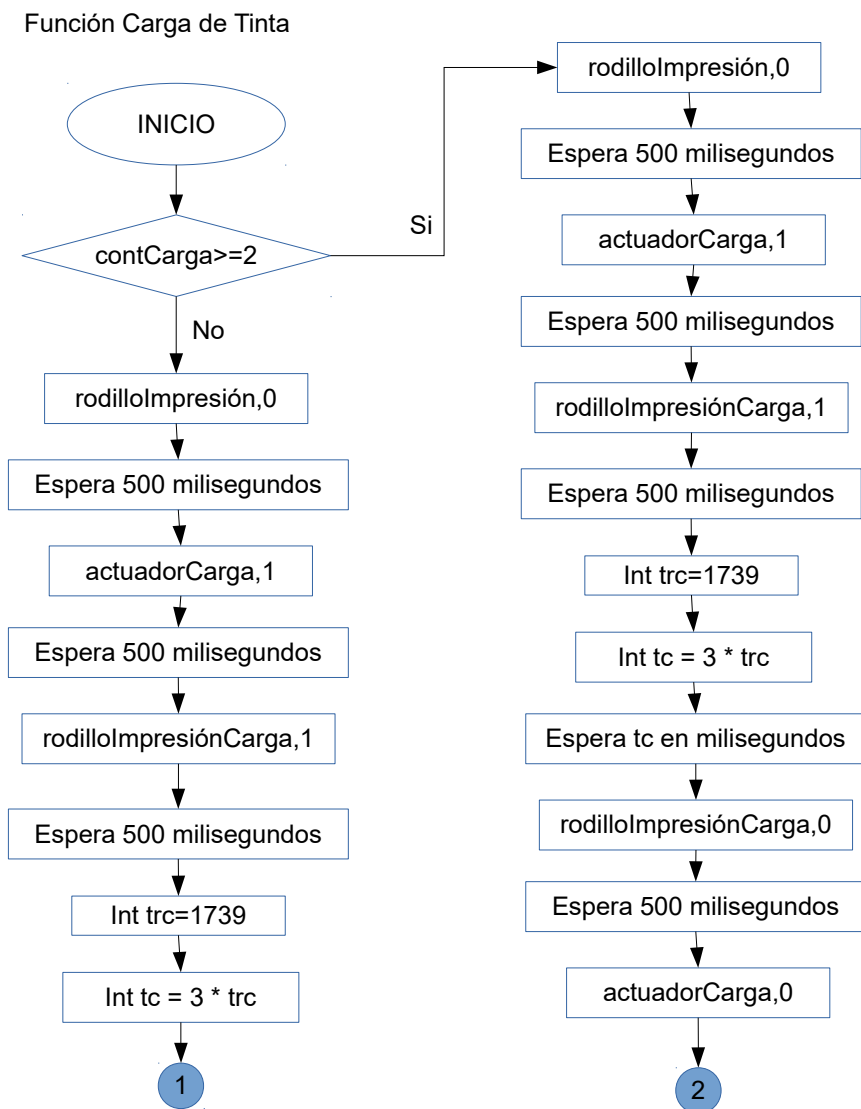


Figura A.3: Diagrama de Flujo función Carga de Tinta parte 1

Fuente: Elaboración propia.

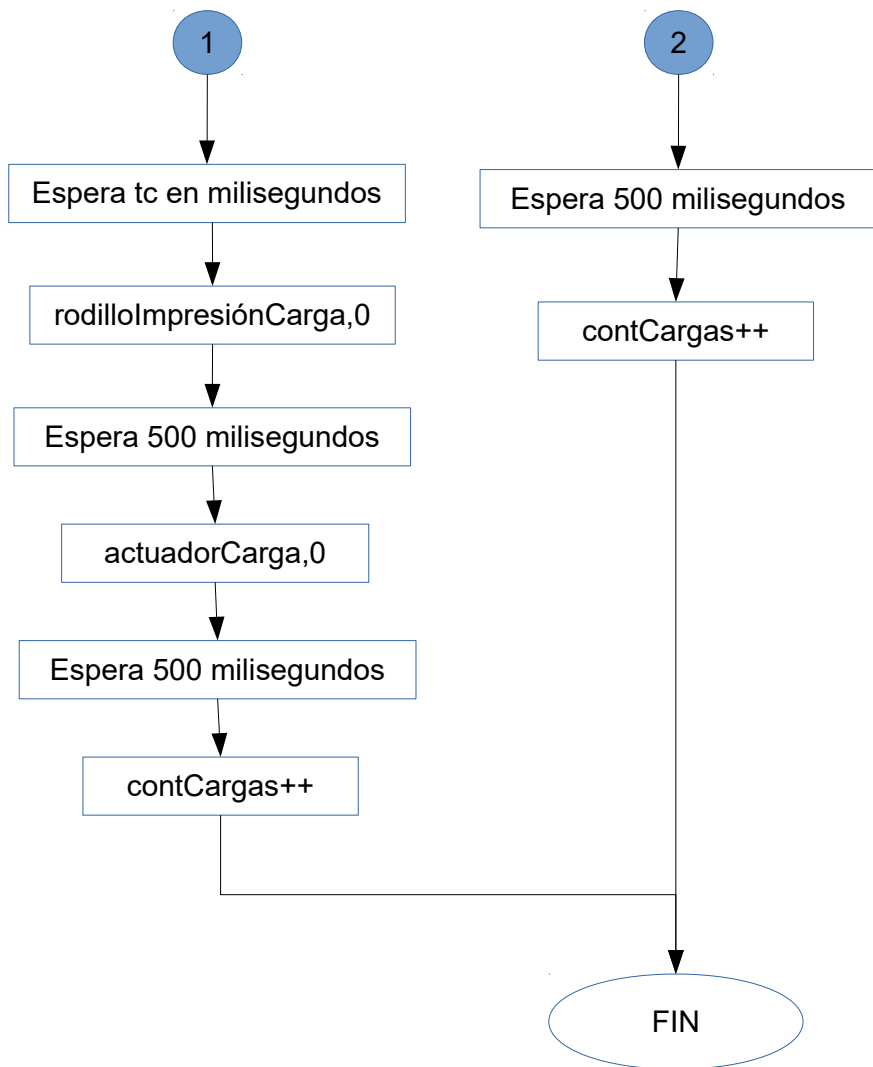


Figura A.4: Diagrama de Flujo función Carga de Tinta parte 2
Fuente: Elaboración propia.

Función de imprimir

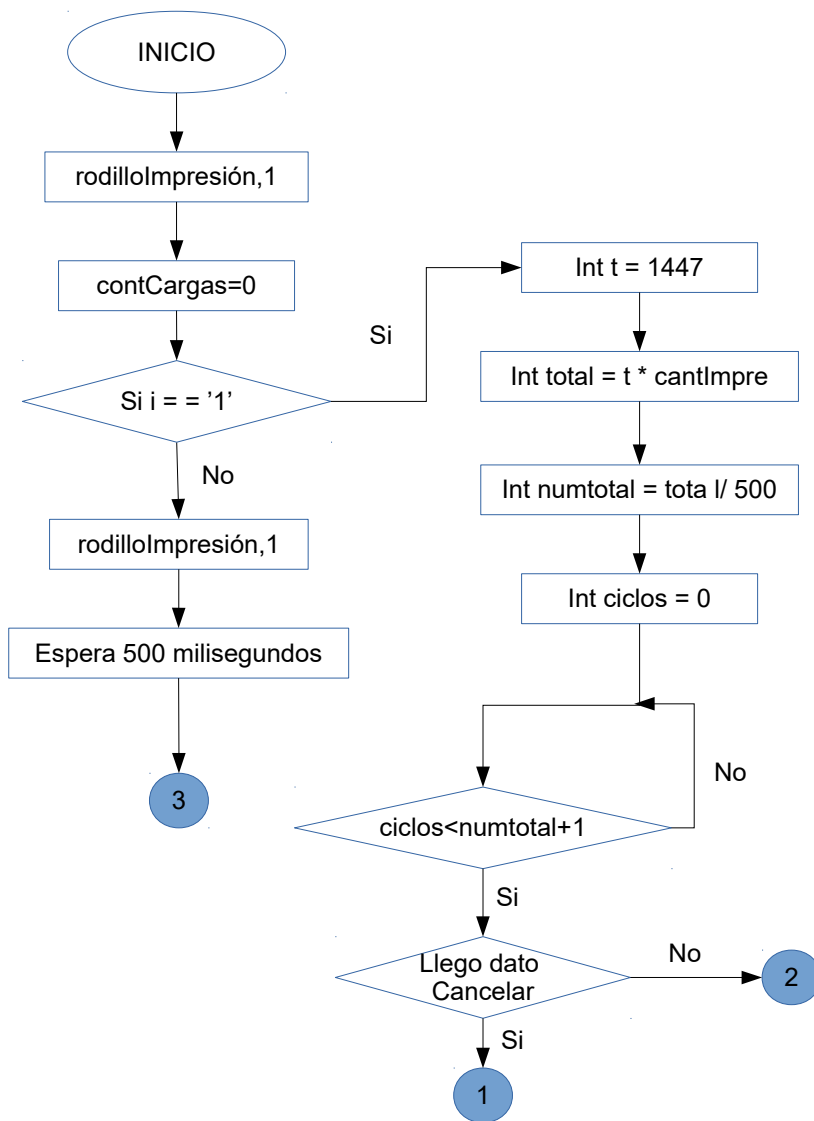


Figura A.5: Diagrama de Flujo función de imprimir parte 1
 Fuente: Elaboración propia.

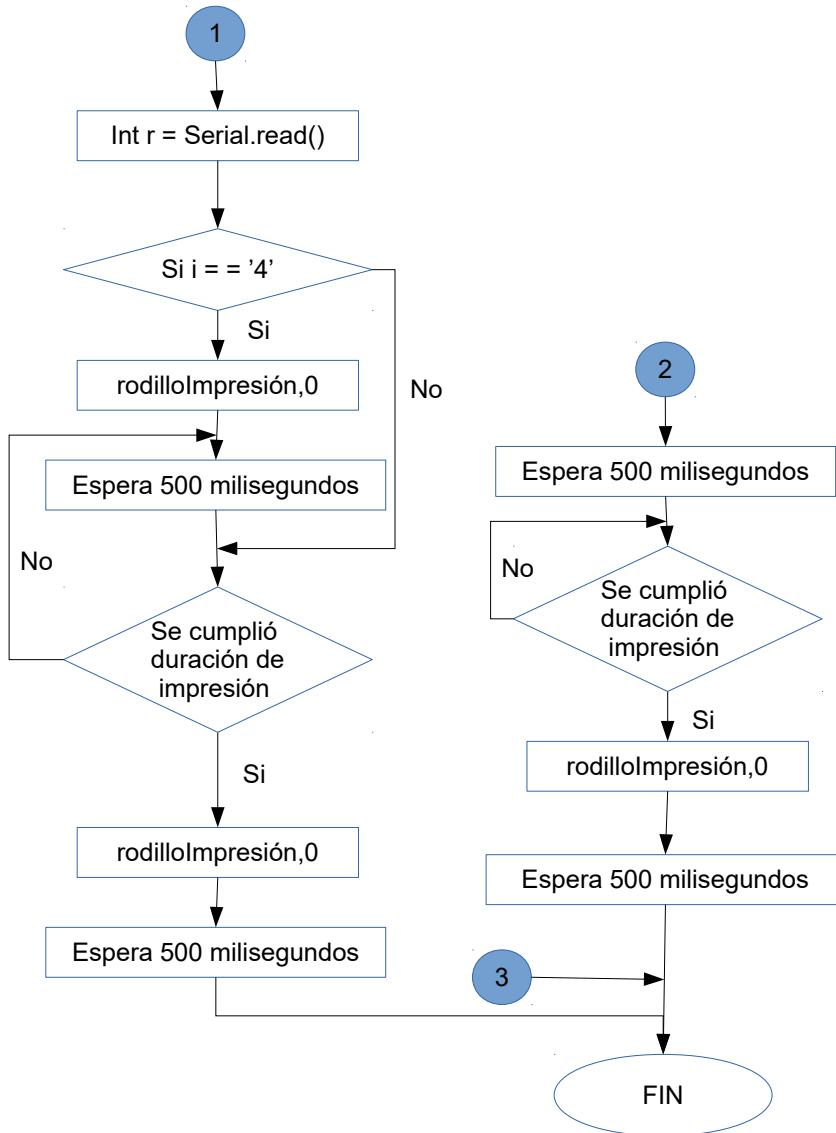


Figura A.6: Diagrama de Flujo función de imprime parte 2
Fuente: Elaboración propia.

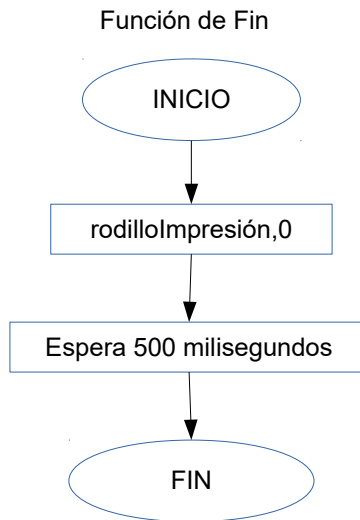
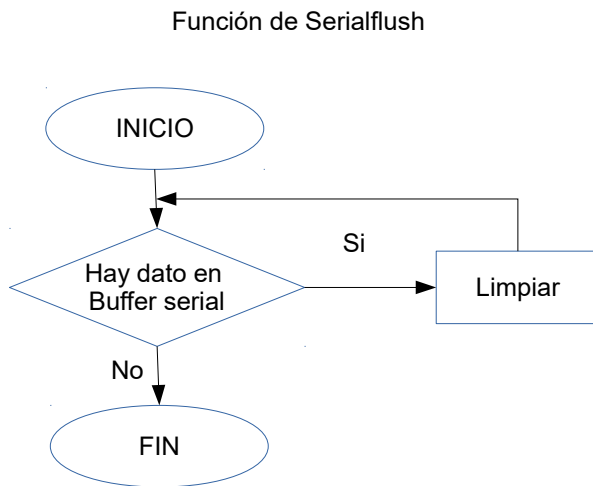


Figura A.7: Diagrama de Flujo función de Serial flush y Fin
Fuente: Elaboración propia.

A.2. App Inventor

Los siguientes diagramas de flujo se realizaron para el sistema de la aplicación en App Inventor.

El diagrama que se muestra en la Figura A.8 explica el proceso que realiza el listPicker lpConectar, al ser pulsado, revisa si el bluetooth esta conectado, si lo está, muestra en pantalla la lista de dispositivos con los que ha estado emparejado el celular, se conecta al bluetooth Impresora Flexográfica, vuelve a revisar si esta conectado el bluetooth, si lo esta despliega en el listPicker el texto de 'Conectado', cambia el color de ese texto a verde, inhabilita el listPicker al igual que los botones de bIniciar, bCancelar, bFin. Después habilita los botones bDesconectar, bAutomático, bCargaTinta y bImprime.

En la Figura A.9, al pulsar el botón de Automático revisa si el bluetooth está conectado, si es así habilita la visibilidad de los label lbVueltas y lbCuenta, el Slider sCantImpre y los botones bIniciar y bCancelar; mientras que deshabilita los botones de bAutomático, bCargaTinta, bImprime y bFin (también su visibilidad).

Para el botón de iniciar revisa la conexión del bluetooth, si se encuentra conectado manda llamar el notificador, éste muestra mensaje en lo que realiza la carga de tinta y la cantidad seleccionada del Slider; deshabilita los botones bAutomático, bIniciar, bCargaTinta, bImprime, bFin y Slider sCantImpre; inhabilita visibilidad en botones bFin y bCancelar, habilita el timer Clock5 y Clock6, en éste timer realiza la siguiente operación $7800 + (\text{posición Slider} * 1447)$ el resultado se muestra al habilitar el Clock6 ver la Figura A.10. Posteriormente revisa la conexión del bluetooth, si hay conexión envía un '1', la posición del Slider y salto de línea, establece el contador de cargas a cero ver Figura A.11.

En la Figura A.12 se muestran los diagramas de flujo de los timer Clock 5A.12(a) se muestran las instrucciones a seguir del Clock5, el cual deshabilita los siguientes elementos: timer, notificador, bAutomático, bIniciar, bCargaTinta, bImprime y bFin; mientras que habilita visibilidad de los

botones bCancelar y habilita botón bFin. En la Figura A.12(b) las instrucciones que se realizan para el Clock6 son: deshabilita los siguientes elementos Clock6, bAutomático, bCargaTinta, bImprime, bCancelar y bIniciar, habilita visibilidad de lbVueltas, lbCuenta y Slider sCantImpre.

Las instrucciones que se utilizaron para el botón desconectar fueron las siguientes: desconecta el bluetooth, habilita el listPicker de conectar, cambia el color de texto del listPicker y lo habilita; deshabilita los botones bDesconectar, bAutomático, bIniciar, bCancelar, bCargaTinta, bImprime y bFin; por último habilita visibilidad de lbVueltas, lbCuenta y Slider sCantimpre ver Figura A.13.

En la Figura A.14 se muestra el diagrama de flujo para las instrucciones que realiza el Clock2 son las siguientes: deshabilita Clock2, bAutomático, bIniciar, bCargaTinta, bImprime, bFin y notificador Figura A.14(a). El timer Clock3 deshabilita los botones bIniciar y bFin; habilita bAutomático, bCargaTinta y bImprime ver Figura A.14(b).

El botón para rodillo de carga las instrucciones que realiza son las siguientes: el contador de cargas es mayor que 1 (cuenta el 0 y 1, lo que hace un total de 2 veces), entonces la opción envía un '2' por el bluetooth, habilita bAutomático, bCargaTinta, bImprime, Clock2 y Clock3; muestra alerta con el notificador y barra de progreso. De lo contrario muestra el notificador, deshabilita bAutomático, bCargaTinta y bImprime; habilita los timer Clock2 y Clock3, revisa la conexión del bluetooth, si hay conexión envía un 2 por el bluetooth e incrementa el contador de cargas y finaliza ver Figura A.15.

En la Figura A.16 Para el Slider muestra en el label de lbCuenta el número que fue seleccionado y se redondea ver Figure A.16(a). En las Figuras A.16(b) y A.16(c) se inicializan las variables globales j y contador de cargas.

La Figura A.17 se muestra el botón de Cancelar A.17(a), revisa la conexión del bluetooth, si la hay entonces muestra la barra de progreso del notificador, habilita el timer Clock4; y vuelve a revisar la conexión del bluetooth, si hay conexión envía un 4 por el bluetooth. Las instrucciones de la

Figura A.17(b) deshabilita el Clock4, habilita visibilidad de Slider sCantImprime, lbVueltas, lbCuenta y bCancelar; habilita bAutomático, bCargaTinta, bImprime; y deshabilita bIniciar, bCancelar y bFin.

Las instrucciones del botón de imprimir son las siguientes: revisa la conexión del bluetooth, si hay conexión deshabilita bAutomático, bCargaTinta y bImprime; habilita visibilidad y botón bFin; revisa la conexión del bluetooth, si la hay envía un 3 por el bluetooth y establece el contador de cargas a cero ver Figura A.18.

Las instrucciones del botón bFin revisa la conexión del bluetooth, si hay conexión muestra la barra de progreso (notificador), habilita el Clock4, revisa conexión del bluetooth, hay conexión envía un 3 por el bluetooth ver Figura A.19.

ListPicker lpConectar

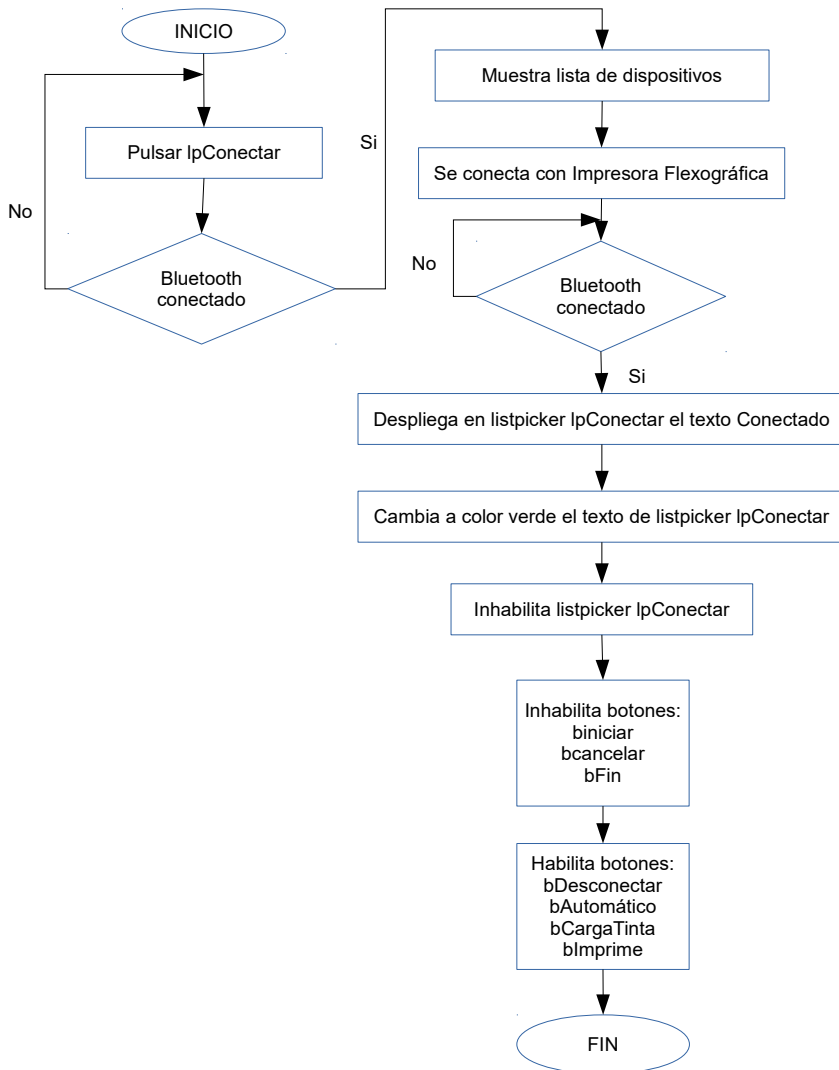


Figura A.8: Diagrama de Flujo listPicker lpConectar
Fuente: Elaboración propia.

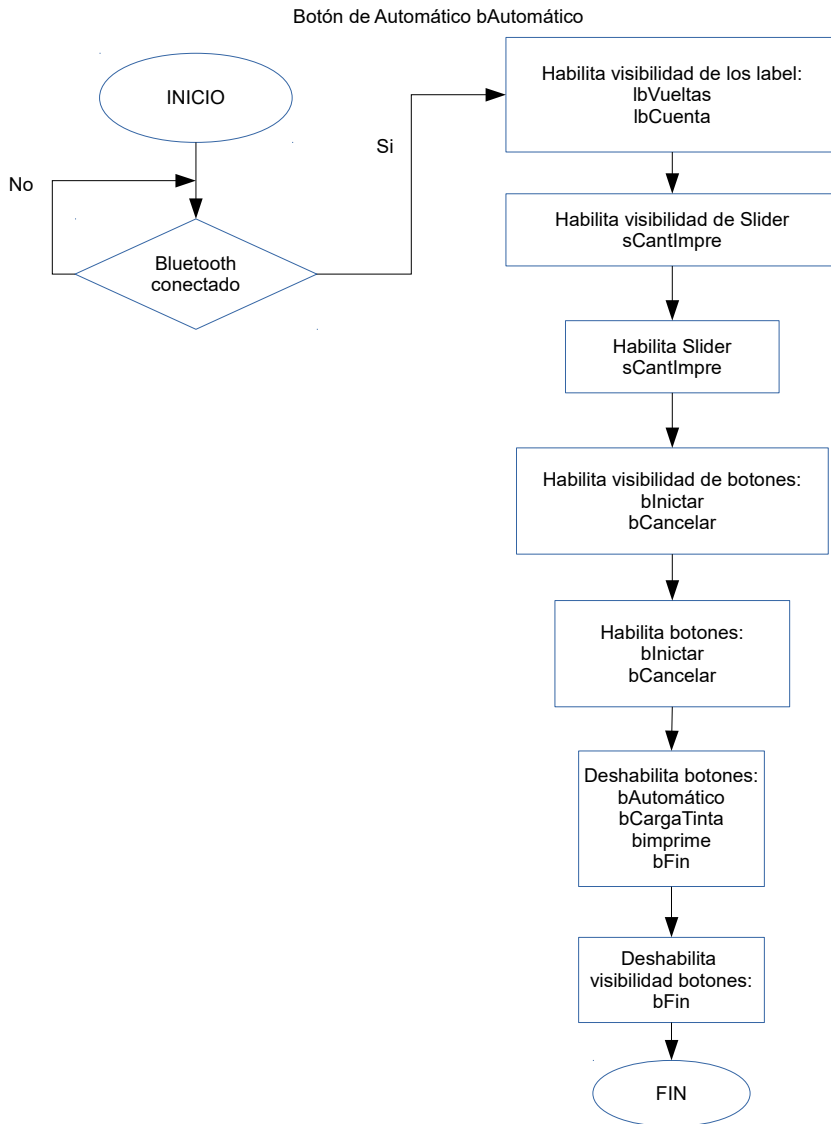


Figura A.9: Diagrama de Flujo botón Automático

Fuente: Elaboración propia.

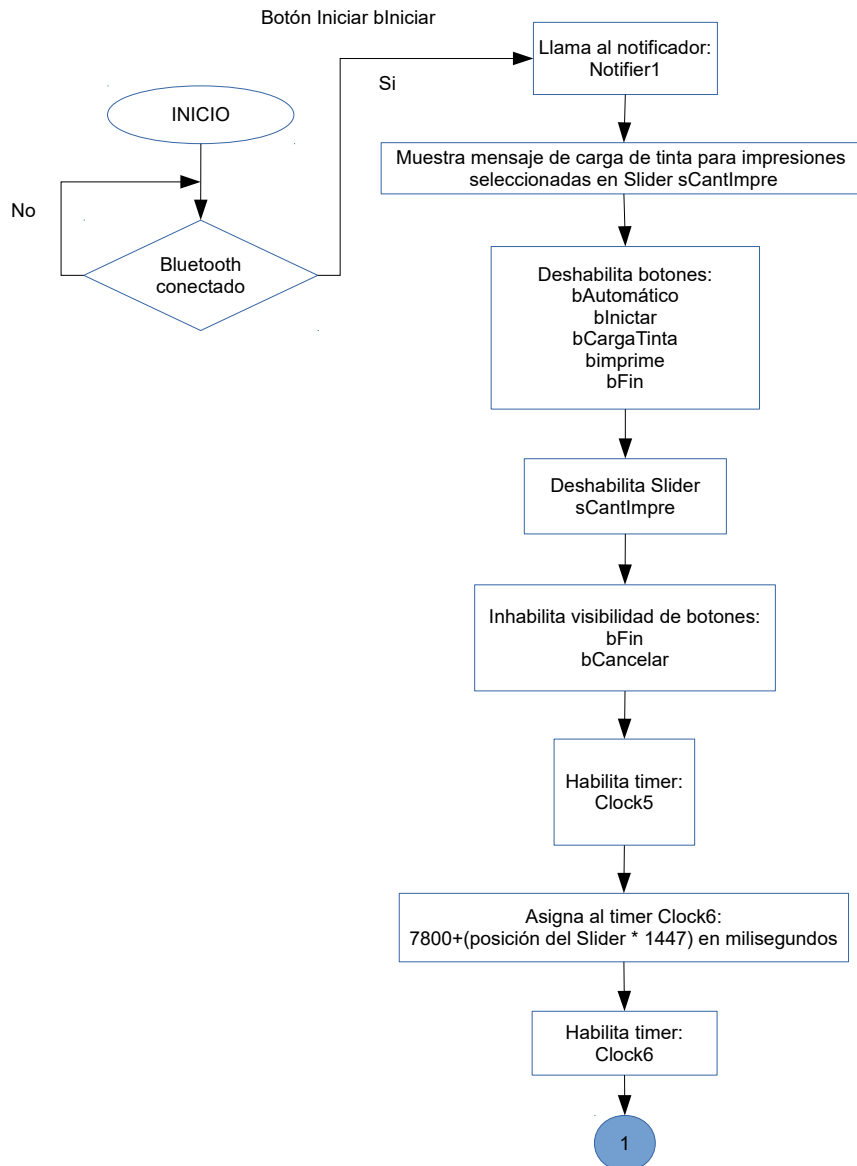


Figura A.10: Diagrama de Flujo botón Iniciar parte 1

Fuente: Elaboración propia.

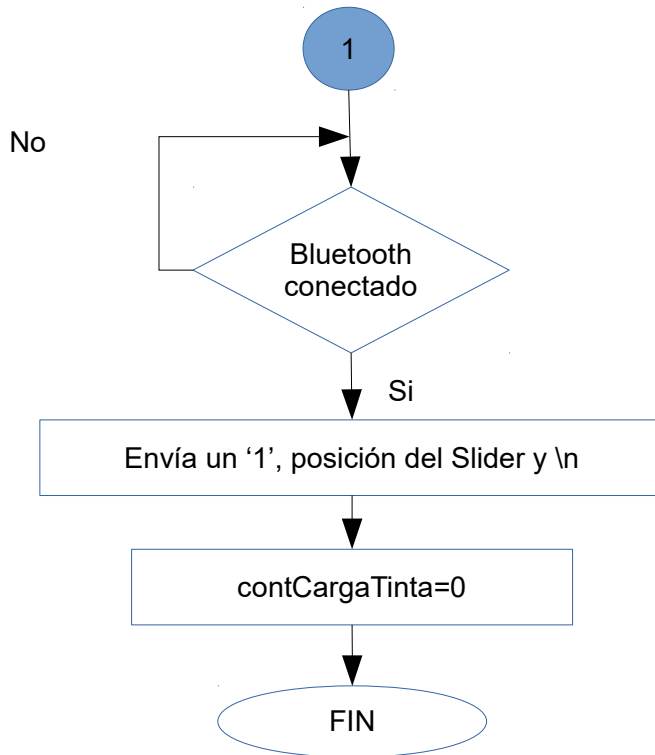


Figura A.11: Diagrama de Flujo botón Iniciar parte 2
Fuente: Elaboración propia.

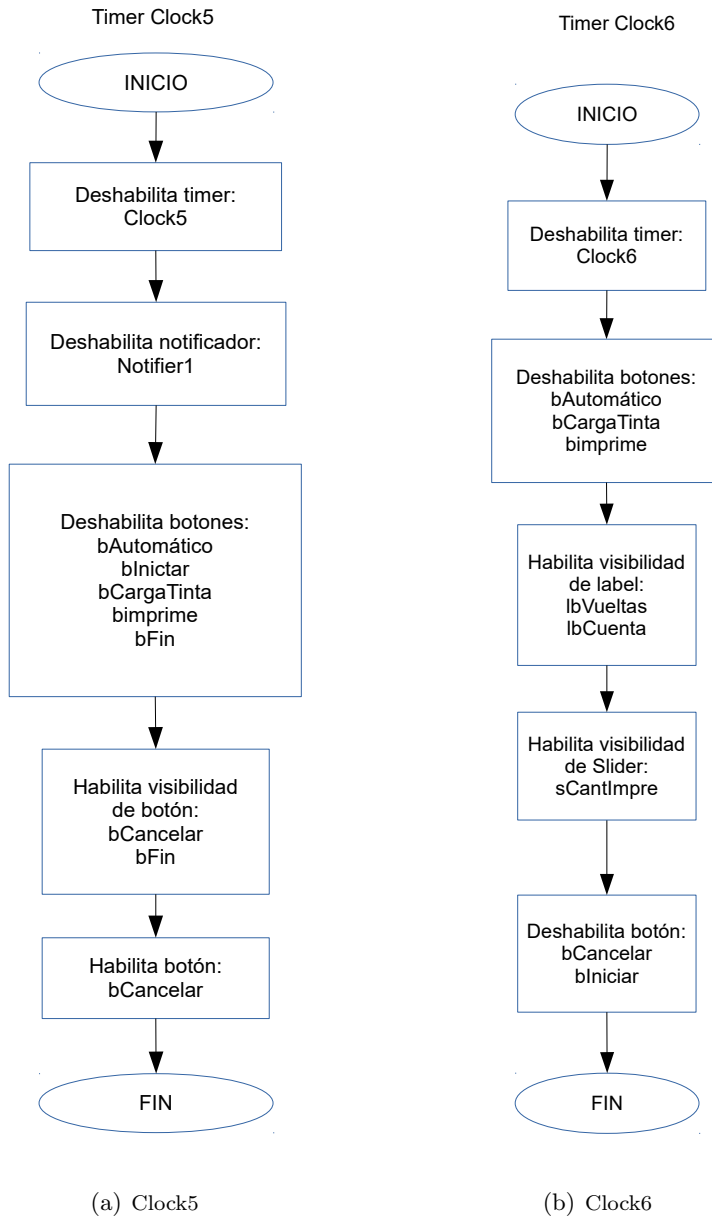


Figura A.12: Diagramas de flujo para timer Clock 5 y 6
Fuente: Elaboración propia

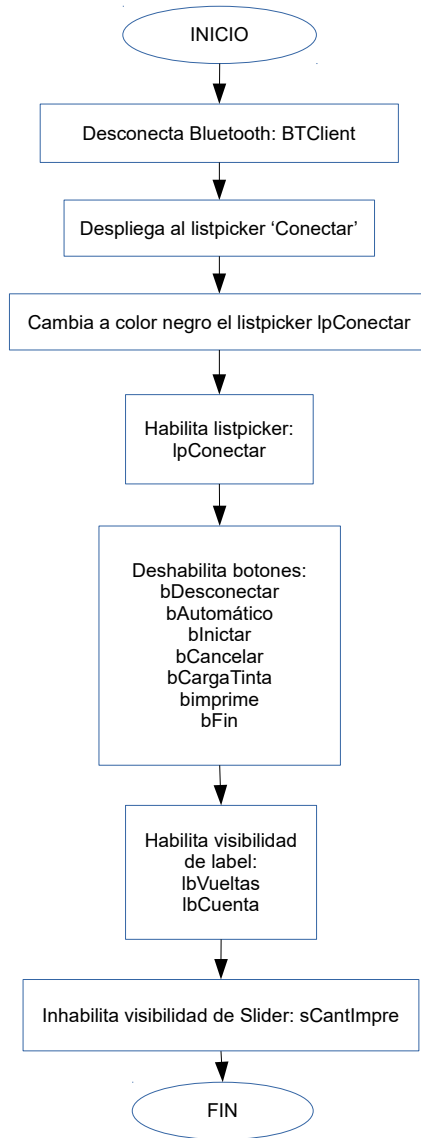


Figura A.13: Diagrama de Flujo botón Desconectar
Fuente: Elaboración propia.

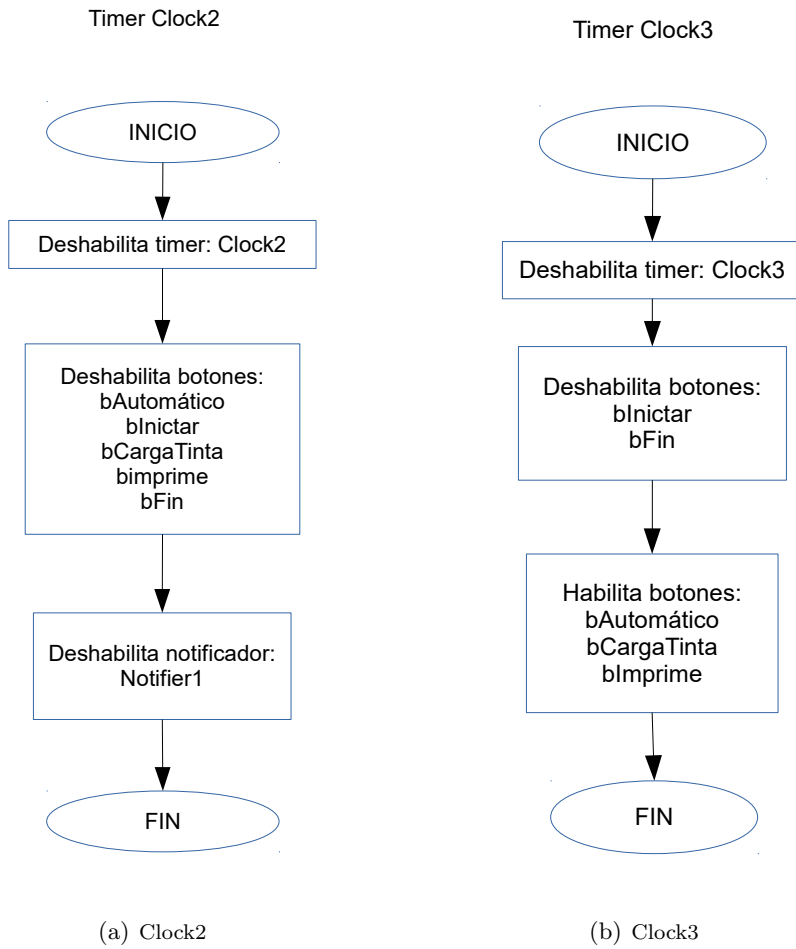


Figura A.14: Diagramas de flujo para timer Clock 2 y 3
Fuente: Elaboración propia

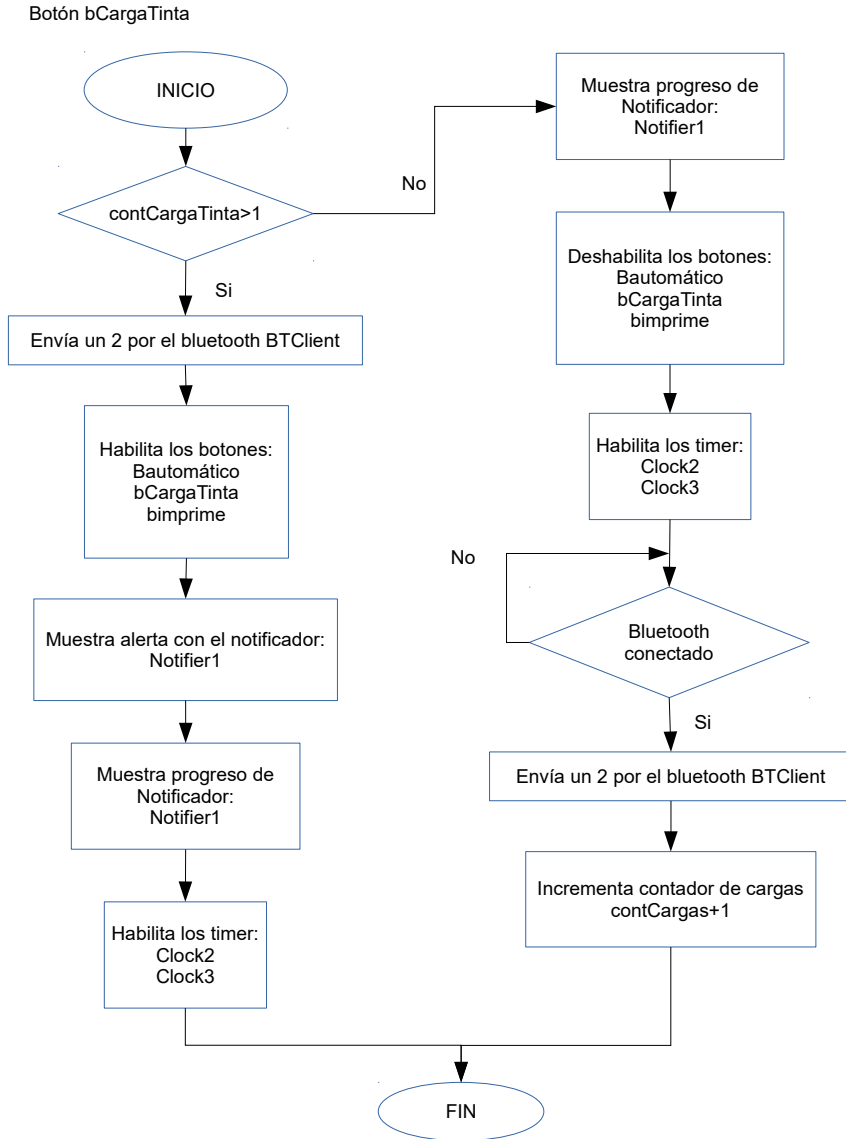
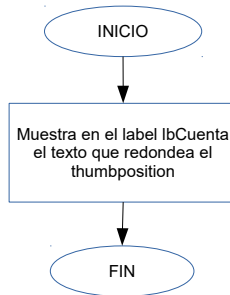


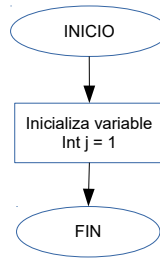
Figura A.15: Diagrama de Flujo botón Carga de Tinta
Fuente: Elaboración propia.

Slider sCantImpre



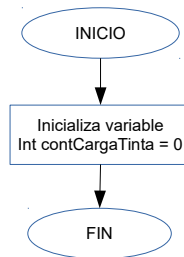
(a) sCantImpre

Variable j



(b) Inicializa j

Variable contCargaTinta



(c) Contador carga de tinta

Figura A.16: Inicialización de variables globales

Fuente: Elaboración propia

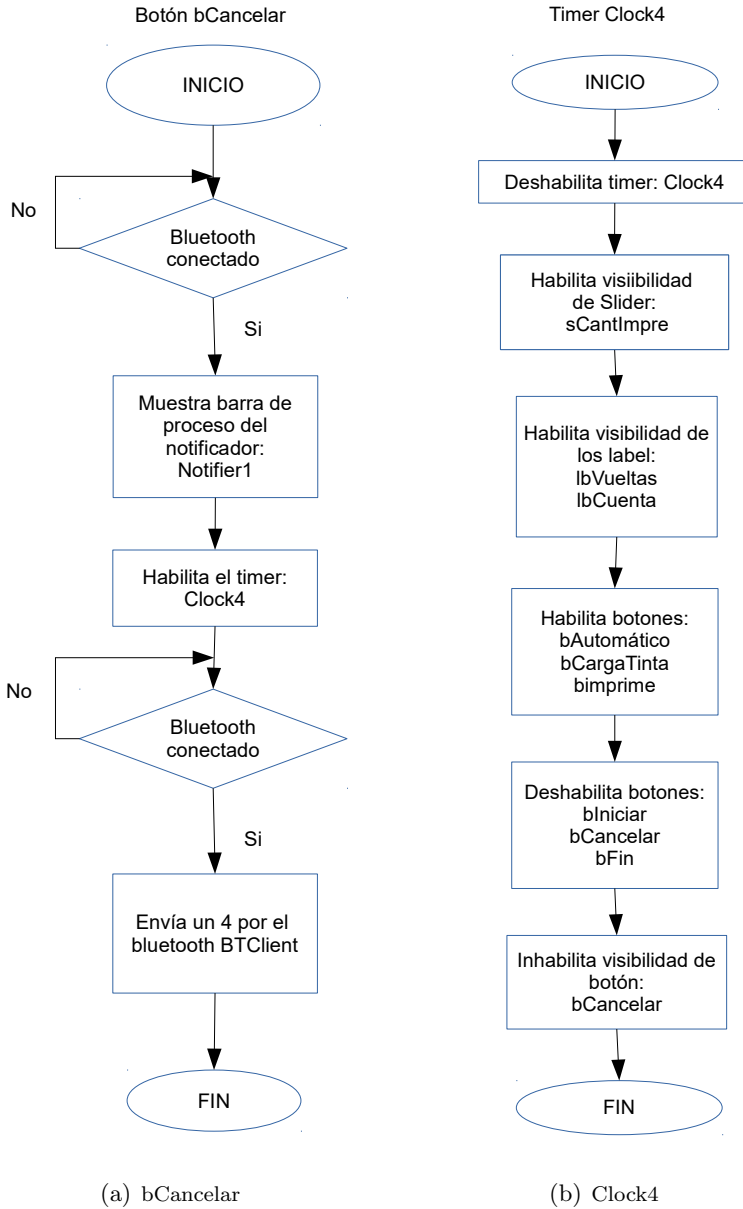


Figura A.17: Diagramas de flujo para botón bCancelar y timer Clock 4
Fuente: Elaboración propia

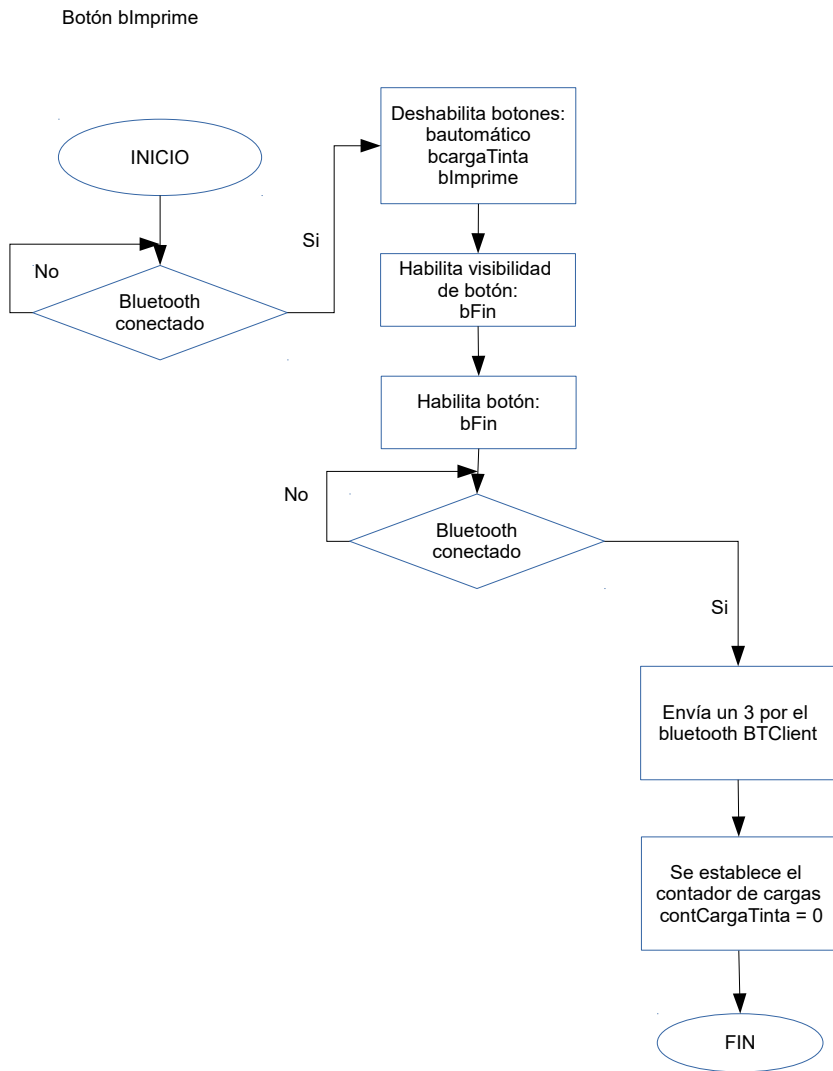
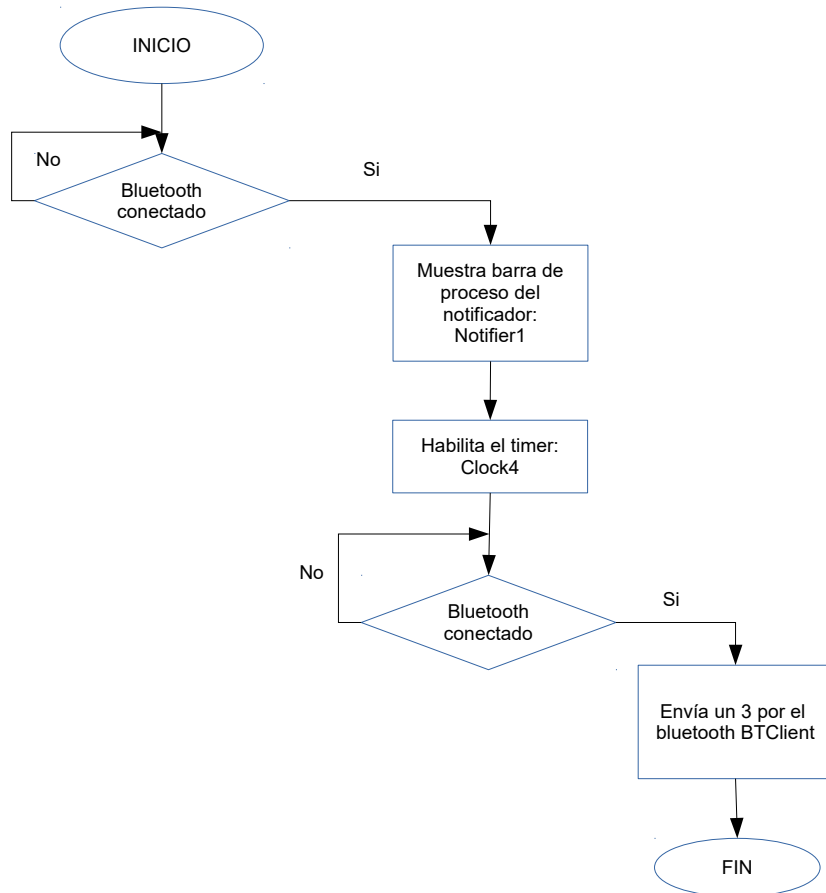


Figura A.18: Diagrama de Flujo botón Imprime

Fuente: Elaboración propia.

Botón bFin

**Figura A.19:** Diagrama de Flujo botón Fin

Fuente: Elaboración propia.

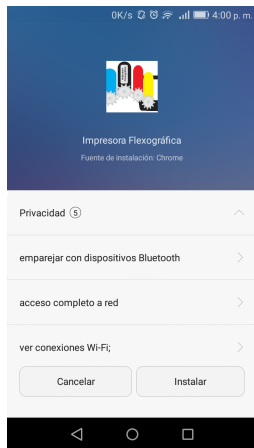
Apéndice B

Manual de instalación

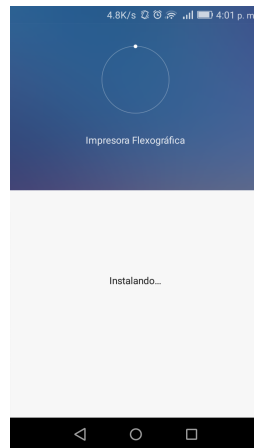
Pasos para la instalación de la impresora flexográfica automatizada con microcontroladores.

1. Conectar la fuente de poder para proporcionar energía a la placa de Arduino, actuadores, motor carga de tinta, motor porta matriz, motor rebobinador y relevadores.
2. Se coloca el material en el soporte a imprimir. Se debe verificar que el material sea colocado en el registro y que pase por el rodillo rebobinador.
3. Se descarga el apk en el siguiente link <https://mega.nz/fm/KUxV3QjR>, cabe mencionar que es necesario crear una cuenta en la página de mega en el link <https://mega.nz/register>, si ya se tiene una cuenta en mega, se procede a descargar la carpeta compartida con el archivo ImpresoraFlexo.apk.
4. Se debe configurar el celular o tablet, ya que por seguridad bloquea algunas opciones para instalar aplicaciones que no sean de Play Store. Por lo que, es necesario que dar click al icono de configuraciones del celular o tableta y seleccionar la opción de "Permitir la instalación de aplicaciones provenientes de fuentes desconocidas". Mostrará un mensaje, en el que "El usuario acepta ser el único responsable de las aplicaciones que quiera instalar, deberá dar click en el botón de Aceptar.

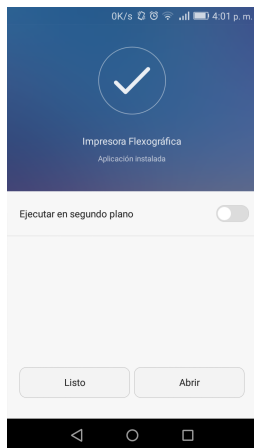
5. El cable de datos o carga, se conecta al celular y computadora, se procede a guardar el archivo .apk en el celular.
6. Buscar y abrir el archivo ImpresoraFlexo.apk y comenzar con la instalación.
7. Muestra en la Figura B.1 las siguientes pantallas, la primera B.1(a) desglosa las características de la aplicación, se da click en botón Instalar; se muestra la carga de la instalación ver Figura B.1(b). En la Figura B.1(c) muestra la instalación completa, dar click en el botón Abrir; después de dar click al botón Abrir mostrará un mensaje en pantalla como se observa en la Figura B.1(d), al que se debe dar click en el botón Cancelar para que la aplicación no se elimine y se abrirá la aplicación.
8. En la Figura B.2 se explica en cada pantalla cómo se realiza el emparejar el bluetooth. Se debe activar el bluetooth, y dar click en la opción de buscar que se encuentra en la parte inferior de lado izquierdo, aquí se busca el bluetooth con nombre Impresora flexográfica ver la Figura B.2(a). Ya seleccionado muestra la pantalla como se ve en la figura B.2(b), ya que pide una clave ya sea 0000 o 1234 como se ve en la Figura B.2(c). La siguiente pantalla muestra cómo se empareja con ese bluetooth ver Figura B.2(d) y en la siguiente pantalla muestra el bluetooth Impresora Flexográfica dentro de los dispositivos emparejados ver Figura B.2(e).



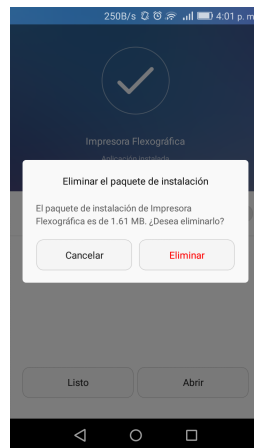
(a) Botón Instalar



(b) Instalando .apk

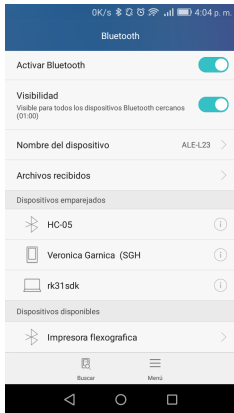


(c) Abrir



(d) Botón Cancelar

Figura B.1: Instalación de impresora.apk
Fuente: Elaboración propia



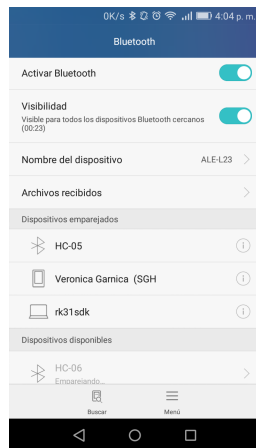
(a) Búsqueda Bluetooth



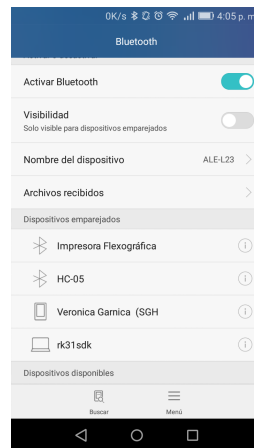
(b) Emparejando



(c) Contraseña



(d) Empareja



(e) Bluetooth Impresora Flexográfica

Figura B.2: Emparejar Bluetooth

Fuente: Elaboración propia

Apéndice C

Manual de operación

La forma de operar para la Impresora Flexográfica Automatizada con microcontroladores es la siguiente:

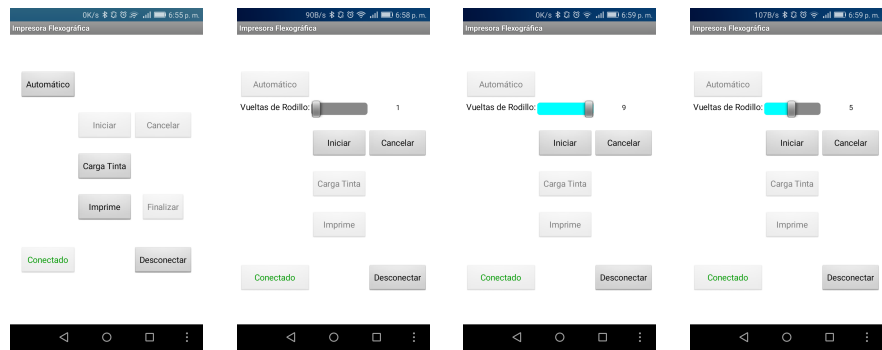
1. Verificar que la conexión de la fuente de poder se coloque dentro del enchufe de luz, la fuente cuenta con un switch, éste se debe cambiar de posición para que la impresora sea activada.
2. El bluetooth estará parpadeando. En la Figura C.1 se puede observar la pantalla principal, en la que se encuentran deshabilitados los botones Automático, Iniciar, Cancelar, Carga Tinta, Imprime, Finalizar y Desconectar. Se da click en el botón Conectar ver C.1(a). La pantalla siguiente muestra los bluetooth con los que esta emparejado ver Figura C.1(b) se selecciona el bluetooth Impresora Flexográfica, ya que se conecta muestra el Botón Conectado de color verde y Activa los botones Automático, Carga Tinta e imprime como se ve en la Figura C.1(c) y desactiva el mismo botón de Conectar.
3. Para utilizar la Función de Automático observar cada pantalla de la Figura C.2, en la que se da click al botón de Automático C.2(a), esté habilita los botones de Iniciar, Cancelar y Slider (se posiciona en una cantidad del 1-9) como se puede observar en las Figuras C.2(b) y C.2(c). En está ocasión se colocó el Slider en la posición 5, es decir, imprimirá 5 vueltas de rodillo ver Figura C.2(d), se da click en botón Iniciar esté deshabilita botón Iniciar y la pantalla siguiente mostrará



Figura C.1: Conectando al Bluetooth Impresora Flexográfica
 Fuente: Elaboración propia

el progreso de carga de tinta para la cantidad seleccionada con el Slider ver Figura C.2(e), una vez que termina de cargar muestra la siguiente pantalla deshabilita el Slider y da la opción de cancelar o terminar el proceso, se eligió terminar proceso de impresión ver Figura C.2(f), en cuanto culmina el proceso de impresión habilita los botones Automático, Carga Tinta e Imprime, mientras que el botón de Conectar lo mantiene deshabilitado ver Figura C.2(g).

- La siguiente Opción para la función de Automático ver Figura C.3 consta de seleccionar con el Slider la cantidad de vueltas de rodillo C.3(a) dar click en Iniciar, mostrará el proceso de Carga de tinta para la cantidad seleccionada ver Figura C.3(b), deshabilitará la opción del Slider y del botón Inicio mostrando sólo el botón Cancelar ver Figura C.3(c), al dar click en el botón Cancelar mostrará en la siguiente pantalla C.3(d) el progreso de la cancelación de impresiones. Al concluir habilita los botones Automático, Carga Tinta e Imprime, el botón de Conectar lo mantiene deshabilitado ver C.3(e).

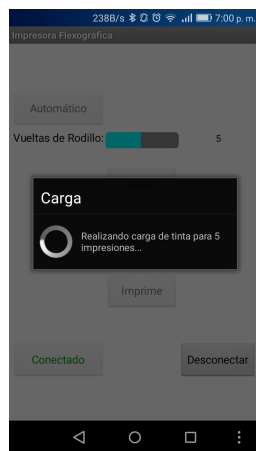


(a) Botón Automático

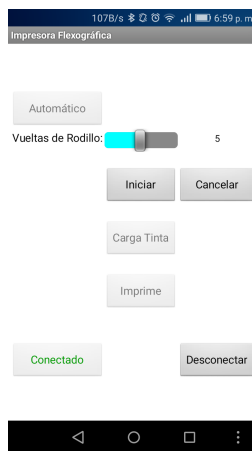
(b) Slider "1"

(c) Slider "9"

(d) Slider "5" Botón Iniciar



(e) Carga de 5 impresiones

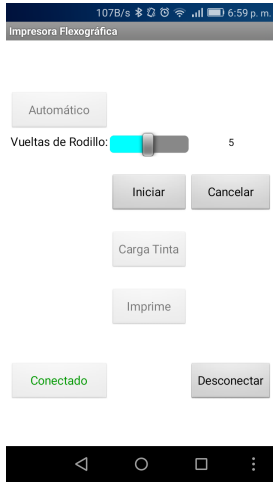


(f) Esperar que finalice impresión

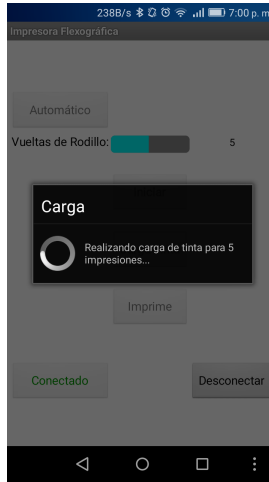


(g) Regreso a pantalla principal

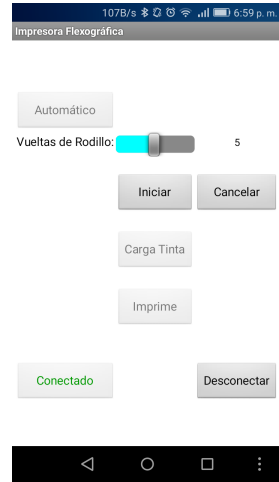
Figura C.2: Opción Automático opción completa acción
Fuente: Elaboración propia



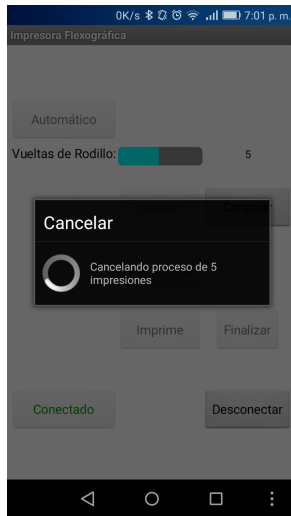
(a) Botón Iniciar "5" impresiones



(b) Carga para "5" impresiones



(c) Botón Cancelar



(d) Cancelando "5" impresiones



(e) Regreso a pantalla principal

Figura C.3: Opción Automático opción Cancelar impresión
Fuente: Elaboración propia

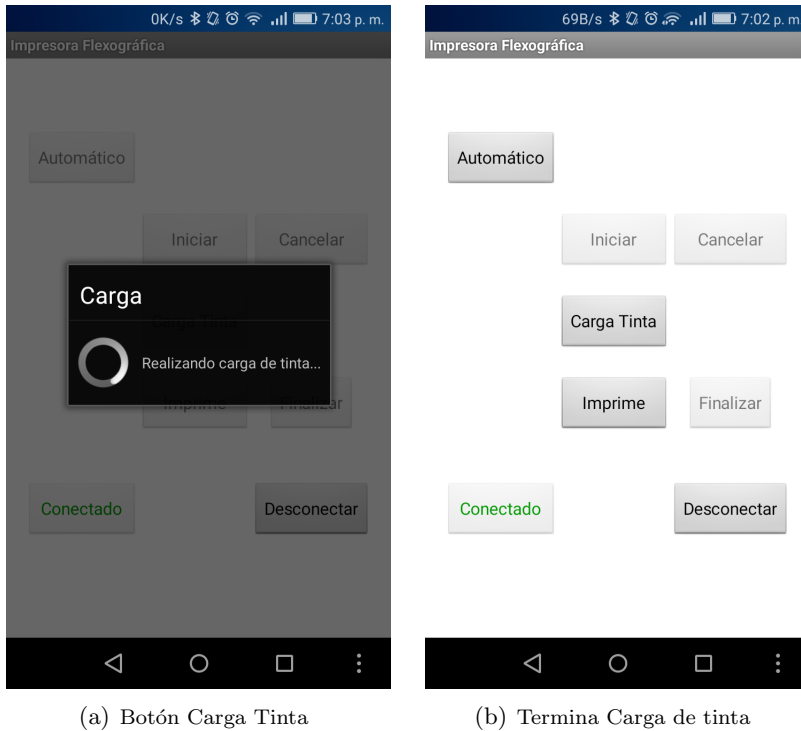
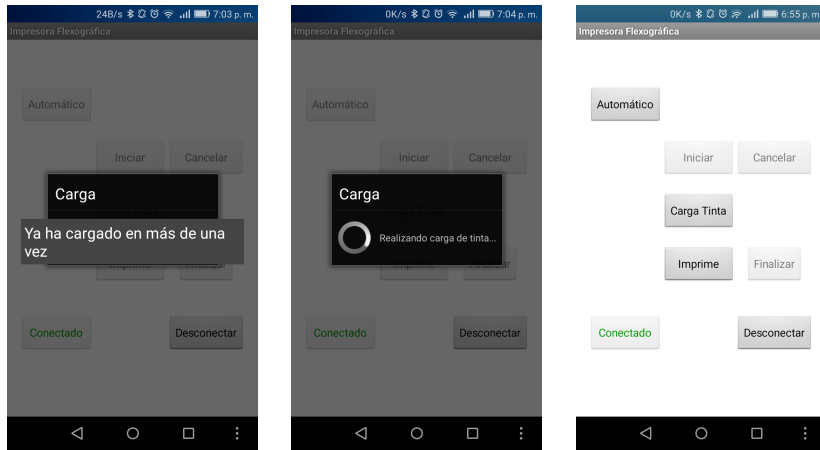


Figura C.4: Opción Manual Carga de tinta 1 y/o 2 veces
Fuente: Elaboración propia

5. En la Figura C.4 está función muestra el progreso que lleva la carga del rodillo de tinta manual, lo que equivale a 1739 por una vuelta esto multiplicado por 3 vueltas lo que da un total de 5217 milisegundos ver Figura C.4(a). En cuanto termina muestra pantalla de funciones ver Figura C.4(b).
6. En la Figura C.5 se muestra cuando el botón de carga de tinta manual ha sido presionado más de dos ocasiones, a lo que muestra un mensaje de aviso, pero realiza la carga por si no cargo totalmente C.5(a), muestra la carga que realiza C.5(b) al finalizar habilita los botones de Automático, Carga Tinta e Imprime C.5(c).



(a) Carga de tinta más de 2 veces

(b) Carga de Tinta

(c) Regreso a pantalla principal

Figura C.5: Opción Manual Carga de tinta más de 2 veces
Fuente: Elaboración propia

7. En la Figura C.6, se da click al botón de Imprime ver Figura C.6(a), lo que deshabilita el botón de Imprimir y habilita el de Finalizar, ver Figura C.6(b) a esta opción se dará click en el botón Finalizar hasta que se desee ver C.6(c) y al concluir muestra la pantalla con las funciones Automático, Carga Tinta e Imprime habilitadas ver C.6(d), mantiene el botón Conectar deshabilitado.
8. En la última Figura C.7, al dar click en el botón de Desconectar ver Figura C.7(a), muestra en la siguiente Figura C.7(b) los botones Automático, Iniciar, Cancelar, Carga Tinta, Imprime y Finalizar deshabilitados.



Figura C.6: Opción Manual Imprime y Finaliza impresiones
Fuente: Elaboración propia

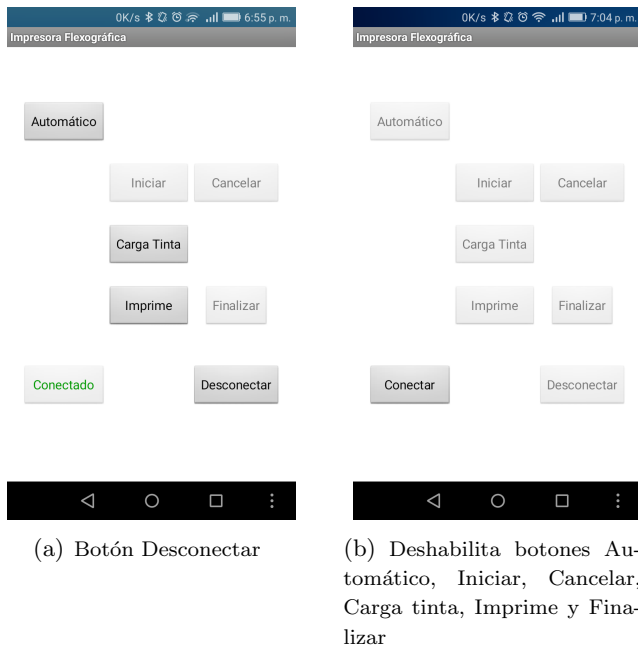


Figura C.7: Desconectar Bluetooth
Fuente: Elaboración propia

Apéndice D

Participaciones

Se participó en eventos como:

- Semana Multidisciplinaria.
- Seminario de investigación.
- El 1er Coloquio de Investigación en el Centro Universitario UAEM Ecatepec.



UAEM
Universidad Autónoma
del Estado de México

Universidad Autónoma
del Estado de México

Maestría en Ciencias de la Computación

Centro Universitario UAEM Ecatepec



DESARROLLO DE UNA IMPRESORA FLEXOGRÁFICA AUTOMATIZADA CON MICROCONTROLADORES

Objetivo:

Automatizar y controlar una impresora flexográfica, mediante microcontroladores y una interfaz desarrollada en Android.





Línea de investigación:
Inteligencia Artificial

Resultados obtenidos:

Dentro de Arduino se programarán las funciones que realizará la impresora flexográfica. Por su parte en la placa se conectarán el bluetooth y los relevadores, los cuales a su vez estarán conectados a los motores.

Adicionalmente, se cuenta con una app para Android, la cual se conectará con la impresora vía bluetooth.



Alumna participante:
Ing. Verónica Garnica Acosta
vero_-_garnica@hotmail.es





Académico responsable:
Dra. Teresa Ivonne Contreras Troya



UAEM Universidad Autónoma del Estado de México

Figura D.1: Semana Multidisciplinaria



Figura D.2: Seminario de investigación de la Maestría en Ciencias de la Computación



Figura D.3: 1er Coloquio de investigación
 Centro Universitario UAEM Ecatepec

Referencias

- [1] V. G. Acosta, *Entrenador de Impresión Flexográfica Automatizada con PLC*. Mar. 2014.
- [2] S. J. . P. Checkland, *La Metodología de los Sistemas Suaves en Acción*. 1994. Scholes J.
- [3] Z. C. . F. O. . R. S. . L. Rodríguez, “Microcontroladores,” 2007.
- [4] M. B. . . B. . J. C. . L. C. . M. González, “Introducción a android,” Apr. 2013.
- [5] J. L. Rederjo, “Uso de appinventor en la asignatura de tecnologías e la información y la comunicación,” Feb. 2013.
- [6] P. Feldman, “La transformación de la tecnología flexográfica,” Feb. 2014.
- [7] C. I. M. H. . T. I. C. T. . R. A. Corona, “Metodología con enfoque de sistemas complejos en áreas económico - financieras,” *Temas de Ciencia y Tecnología*, vol. 19, pp. 21 – 23, Apr. 2015.
- [8] E. S. O. E. . C. J. C. R. . O. G. D. Velasco, “Despliegue de función de calidad (qfd) apoyado mediante técnicas difusas: Caso prótesis mioeléctrica de mano,” *Ingeniería e Investigación*, pp. 4–14, 2005.
- [9] A. Zenon, “La microelectrónica en experimentos de física,” *Fides et Ratio - Revista de DifusiÃ cultural y cientÃfica de la Universidad La Salle en Bolivia*, vol. 5, pp. 88 – 93, 04 2012.

- [10] A. A. Borrego, “Vinculación universidad – empresa y su contribución al desarrollo regional*,” *Ra Ximhai*, vol. 5, pp. 407–414, Sept. 2009.
- [11] Óscar Torrente, *ARDUINO Curso práctico de formación*. 1 ed., Feb. 2013.
- [12] P. A. . B. P. . R. A. . B. Jaramillo, “Desarrollo de aplicación con sensores de temperatura usando una versión del lenguaje java llamada javelin adecuada para el uso en microcontroladores que admiten tecnología.,” Oct. 2010.
- [13] S. B. . L. J. I. . M. J. Agüero, “Sistemas expertos: fundamentos, metodologías y aplicaciones,” *Ciencia y Tecnología*, vol. 13, pp. 349–364, 2013. ISSN 1850-0870.
- [14] H. G. . C. Castillo, “El modelo de la triple hélice como un medio para la vinculación entre la universidad y empresa,” *revista nacional de administración*, vol. 1, pp. 85–94, Jan. 2010.
- [15] A. E. Chávez, “Inteligencia artificial,” Oct. 2012. Docente Facultad de ciencias y Sistemas - UNI.
- [16] P. Checkland, *Pensamiento de sistemas, práctica de sistemas*. primera edición ed., 1993.
- [17] A. B. . F. G. Barrios, *Automatización del proceso de envasado en líquidos mediante un PLC*. Tecnológico de Estudios Superiores de Ecatepec, Ecatepec de Morelos, Estado de México., 1996.
- [18] A. G. Hernández, “Agentes inteligentes,” 2013.
- [19] M. Constantino, “Introducción a la inteligencia artificial,” 2012. Sitio Web Universitas Nebrissensis S.A. Nebrija Universidad.
- [20] A. M. I. Badia., *Autómatas Programables*. No. 3 in 84-267-0672-X, 1992.
- [21] N. M. Hernández, *Tarjeta de desarrollo para microcontroladores PIC*, vol. 1. Instituto Politécnico Nacional, Nov. 2009.
- [22] C. A. R. Castillo, “Historia de los plc s,” Sept. 2009.

- [23] C. S. Torres, “Inteligencia artificial,” Dec. 2010. Lenguaje Natural.
- [24] J. A. M. Velasco, “Inteligencia artificial y conciencia,” 2012. Ponencia.
- [25] E. Vildósola, “Actuadores,” 2014.