UCAB Universidad Católica
ANDRÉS BELLO

Facultad de Ingeniería

Escuela de Ingeniería Informática

BN 186341
AATG072

Caracas 07 de Diciembre del 2017
Universidad Católica Andrés Bello
Facultad de Ingeniería en informática

Dispositivo de lectura digital con salida al Sistema Alfabético Braille para personas con discapacidad visual.

Firma
Biagio Cante

Firma
Manuel Goncalves

Firma
Nombre Revisor

Firma
Luis González

UCAB Universidad Católica ANDRÉS BELLO

Facultad de Ingeniería

Escuela de Ingeniería Informática

Datos del Estudiante 1:

Apellidos, Nombres: Goncalves López, Manuel Alejandro

C.I.: 21.116.775

Teléfonos:

Celular: (0426) 8068292 Hab: (0212) 4311404

Correos electrónicos: Man281192@gmail.com

Datos del Estudiante 2:

Apellidos, Nombres: Gonzalez Lara, Luis Jose

C.I.: 18.143.153

Teléfonos:

Celular: (0414) 1057854 Hab: (0212) 4629634

Correos electrónicos: lgonzalez64@gmail.com

Datos del Tutor:

Apellidos, Nombres: Biagio Cante

C.I. : _____

Teléfonos:

Celular: _____ Hab: _____

Correos electrónicos: bcante@gmail.com

Cargo: _____

Años de Graduado:

UCAB Universidad Católica ANDRÉS BELLO

Facultad de Ingeniería

Escuela de Ingeniería Informática

FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

Dispositivo de lectura digital con salida al Sistema Alfabético Braille para personas con discapacidad visual.

Este jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: Diecinueve (19) puntos.

JURADO EXAMINADOR

Firma:	Firma:	Firma: Gloria Tarrío V.
Nombre: BIAGIO CANTE	Nombre: Ramón Ponce	Nombre: Gloria Tarrío

REALIZADO POR	**Manuel Goncalves** **Luis González**
Tutor Académico	**Biagio Cante**
Fecha	**Diciembre de 2017**

Agradecimientos

Principalmente quiero agradecer a mis padres Luis Felipe González y Sara Lara de González por todo el apoyo que me han brindado a lo largo de toda mi carrera universitaria y a la corrección de este Trabajo Especial de Grado, gracias a ellos tengo el nivel de educación que poseo hoy en día. Un reconocimiento muy especial al profesor Biagio Cante quien aceptó como tutor de este trabajo de grado por todos sus consejos, correcciones e incluso regaños cada vez que realizábamos una reunión. A Kenny Torrealba por el apoyo con los materiales del laboratorio ya que sin eso no hubiese sido posible la realización de las pruebas para el desarrollo de este trabajo especial de grado. Especialmente quiero dar una dedicación a Génesis Martínez mi fiel compañera, amiga, pilar, referencia que siempre estuvo conmigo estos últimos 3 años sin importar la situación (buena o mala). Además es oportuno recordar a todas las personas con las que trabajé a lo largo de la carrera, las múltiples amistades que se fueron formando en el camino, ya que con ellos el recorrido de este ciclo se hizo menos tortuosa la culminación de este ciclo. A todos por ser una pieza fundamental en este desarrollo.

Luis Gonzalez

Agradecimientos

Primeramente agradezco a Dios y a mi familia por haberme apoyado en mi formación como profesional y como persona, a mi padre Jose Goncalves por haberme dado la educación, que es la mejor herencia en vida que me pudo dar, a mi madre Glenda López por haber sido mi guía y pilar en los buenos momentos y en las lecciones aprendidas, al amor incondicional de mis abuelas Mireya Vera y María Goncalves que siempre celebraron cada triunfo en este grato caminar del aprendizaje y a mis abuelos que desde el cielo celebran, Regner López y Ferdinando Goncalves, quienes son motivo de inspiración y ejemplos de superación.

Agradezco a mis tías María Fernanda Goncalves, Rosa Goncalves, Cecilia López, a mis tíos Regner López, Leonardo López, Arturo López, a mi padrino Joaquin Pereira, a mis primos y hermanos que fueron los comentarios más cercanos y certeros en los momentos oportunos y la mano amiga en los momentos de fuerza, a Daniel Goncalves, Luis Fernando Goncalves, José Martinez, Diego Goncalves y Valeria Goncalves; Agradezco también a todos los compañeros de trabajo y amigos que han facilitado mi camino al aprendizaje siendo pacientes antes mis errores e impulsores de mis triunfos, brindándome oportunidades únicas para mi vida profesional, a Daniel Prato por ser un gran amigo y colega, ,a Loretty Delgado quien brindó su apoyo incondicional en gran parte de este camino apostando a mi crecimiento continuo como persona y como profesional, a Daniel Ayala por brindarme la oportunidad de aprender bajo su tutela ,a Dayriana Ruiz por ser guía e impulso consecuente a mi superación y a Daniel Farrera por ser un hermano adoptivo con el que siempre conté en todo momento.

Un agradecimiento muy especial a la Universidad Católica Andrés Bello por su esfuerzo continuo de construir mejores profesionales y a cada profesor que bajo su pasión y entrega pudieron nutrir mis conocimientos y me hicieron saber que la vida es un aprendizaje continuo y que la mejor forma de aprender es enseñando.

Manuel Goncalves López

Índice de Contenido

Agradecimientos iv
Agradecimientos v
Sinopsis xii
Introducción 1
Capítulo I 2
Planteamiento del problema 2
Objetivos 4
Objetivo general 4
Objetivos específicos 4
Alcance 5
Diseñar e implementar un entorno de contenido para la carga y manejo de archivos de texto plano 5
Diseñar e implementar un módulo de reconocimiento de posicionamiento y transición inteligente del cursor entre caracteres alfanuméricos presentados en torno a la lectura 5
Construir un dispositivo de salida que permita representar textos alfanuméricos en el sistema alfabético Braille 6
Investigar y analizar los distintos tipos de microcontroladores a utilizar en el dispositivo con salida al sistema alfabético Braille para el transporte y conexión de la información contenida en el computador 7
Crear un dispositivo traductor que permita la transformación de las señales digitales a código Braille 7
Diseñar e implementar un módulo de reconocimiento de frontera de contenido 8
Diseñar e implementar un módulo de notificación 8
Limitaciones 9

Justificación 10
Estimación del costo del dispositivo 11
Dispositivos de lectura Braille 11
Dispositivo a desarrollar 14
Solenoides varios 14
Microcontroladores 15
Precio más bajo 18
Precio más alto 18
Capítulo II 19
Marco referencial 19
El Alfabeto Braille 19
Interfaz gráfica de usuario 20
Arduino 20
Solenoide 21
Transistor 22
Resistencia 23
GIT 24
Comandos básicos 25
Branch 25
Github 25
Pruebas unitarias 25
Java 26
Arquitectura MVC. 27
Modelo 27
Vista 27
Controlador 27

Capítulo III 29

Marco Metodológico 29

Diagrama de Gantt 32

Capítulo IV 34

Desarrollo 34

Diseñar e implementar un entorno de contenido para la carga y manejo de archivos de texto plano 35

Diseñar e implementar un módulo de reconocimiento de posicionamiento y transición inteligente del cursor entre caracteres alfanuméricos presentados en torno a la lectura 36

Construir un dispositivo de salida que permita representar textos alfanuméricos en el sistema alfabético Braille. 37

Investigar y analizar los distintos tipos de microcontroladores a utilizar en el dispositivo con salida al sistema alfabético Braille para el transporte y conexión de la información contenida en el computador 39

Crear un dispositivo traductor que permita la transformación de las señales digitales a código Braille. 41

Diseñar e implementar un módulo de reconocimiento de frontera de contenido. 42

Diseñar e implementar un módulo de notificación. 43

Capítulo V 44

Resultados 44

Diseñar e implementar un entorno de contenido para la carga y manejo de archivos de texto plano 44

Diseñar e implementar un módulo de reconocimiento de posicionamiento y transición inteligente del cursor entre caracteres alfanuméricos presentados en torno a la lectura 46

Construir un dispositivo de salida que permita representar textos alfanuméricos en el sistema alfabético Braille. 47

Investigar y analizar los distintos tipos de microcontroladores a utilizar en el dispositivo con salida al sistema alfabético Braille para el transporte y conexión de la información contenida en el computador. 48

Crear un dispositivo traductor que permita la transformación de las señales digitales a código Braille. 48

Diseñar e implementar un módulo de reconocimiento de frontera de contenido. 49

Diseñar e implementar un módulo de notificación. 49

Software de lectura y conversión Braille 50

Dispositivo de lectura Braille 50

Manuales de uso 51

Manuales de programación 51

Diagramas de clases 51

Capítulo VI 52

Conclusiones y recomendaciones 52

Conclusiones 52

Recomendaciones 53

Anexos 59

Referencias Bibliográficas 54

Índice de Figuras

Figura 1 Teclado Braille 3
Figura 2 Línea Braille 3
Figura 3 Representación de caracteres 6
Figura 4 Polímero Electroactivo 6
Figura 5 Conversión de carácter. 7
Figura 6 Alfabeto Braille 20
Figura 7 Arduino UNO 21
Figura 8 Campo magnético en solenoide 22
Figura 9 Transistor TIP 122 23
Figura 10 Código de colores de los resistores 24
Figura 11 Modelo vista controlador 28
Figura 12 Ciclo del TDD 31
Figura 13 Interfaz para el manejo de archivos 35
Figura 14 Patrón Composite 36
Figura 15 Circuito de los pulsadores. 37
Figura 16 Circuito de representación Braille..... 38
Figura 17 Carga de documento (DragAndDrop)..... 44
Figura 18 Configuración UCABraille..... 45
Figura 19 Posición Inteligente del cursor..... 46
Figura 20 Pulsadores 47
Figura 21 Límite de frontera inicial.. 49

Índice de Tablas

Tabla 1 Comparación de líneas Braille. 11
Tabla 2 Evaluación de solenoides.. 14
Tabla 3 Evaluación de microcontroladores. 15
Tabla 4 Selección del microcontrolador 40
Tabla 5 Materiales en divisa externa. 50
Tabla 6 Materiales en divisa interna. 50

Sinopsis

El Braille es un alfabeto diseñado para personas con discapacidad visual, el cual representa letras por medio de combinaciones de puntos en relieve, dicho alfabeto fue creado por el Francés Louis Braille.

El objetivo de este trabajo de grado es ofrecer una alternativa a las personas que padecen de discapacidad visual para realizar lectura de documentos digitales. Uno de los objetivos es brindar mayor integración a las personas que padecen este tipo de discapacidad con el mundo de la tecnología y que tengan la oportunidad de poder leer textos, libros o artículos de una manera cómoda y sencilla. Para el desarrollo de este sistema se utilizó la metodología de desarrollo Orientado a pruebas o por sus siglas en inglés TDD (Test Driven Development). Ésta es una metodología de desarrollo ágil con una fortaleza fundamental, y es que unifica el proceso de prueba de una aplicación con el proceso de desarrollo, lo que hace que el producto final sea un programa robusto y a prueba de fallos, a la vez que ahorra costos. En el desarrollo del dispositivo de lectura se emplearon solenoides, este componente electrónico es un conjunto de espiras enrolladas consecutivamente a una varilla de metal, que al paso de la corriente genera un campo magnético, transformando la energía eléctrica en energía mecánica, lo que permite que cualquier metal interno a su espiral pase de un extremo al otro. Para el desarrollo del circuito del teclado se optó por la integración de una placa Arduino DUE trabajando en conjunto con solenoides para representar la salida del texto en Braille. El Arduino es una plataforma de desarrollo con un microcontrolador central creado por Atmel, posee puertos de entrada y salida (I/O), puertos analógicos y puertos USB, esto permite que la placa se conecte a un computador y sea programable por un lenguaje de alto nivel como java o C/C++, por lo cual se podrá acceder a cada una de estas entradas/salidas mediante la programación en el lenguaje seleccionado para suministrar o recibir corriente que servirá de activadores de funcionalidades para la solución.

Introducción

En la actualidad, existen aproximadamente 285 millones de personas que presentan discapacidad visual según la Organización Mundial de la Salud (OMS), 39 millones de ellas son totalmente ciegas, 246 millones presentan baja visión, adicionalmente el 82% de estas personas con discapacidad visual son mayores a 50 años. Actualmente la tendencia en el mercado es la era digital, en el que se pretende usar la menor cantidad de papel posible para crear un planeta más verde, esto representa un problema para las personas con discapacidad visual debido a su condición. Para solventar este problema y evitar la exclusión a estas personas se crearon dispositivos denominados teclados Braille, que se utilizan para la lectura de documentos digitalizados, conectados a un computador, pero la construcción de este tipo de dispositivos es muy costosa. La mayoría de las personas con discapacidad visual de la población mundial son de escasos recursos y no tienen la manera económica de adquirir estos teclados, ya que en el mercado actual un teclado con línea Braille para la lectura de documentos digitales está valorado en más de $1.500, como bien se sabe las dificultades económicas del país y la adquisición de divisas son muy complicadas, por lo cual se planteó la idea de construir un teclado Braille a un costo más económico, para que la tecnología sea accesible a las personas más necesitadas, por ésta razón el trabajo especial de grado tiene como objetivo brindar una propuesta de acceso a la tecnología para las personas con discapacidad visual a un costo más accesible y que cumpla con la mayoría de los estándares que requieran las personas con esta discapacidad visual para poder realizar la lectura de un documento digital.

Capítulo I

Planteamiento del problema

El acceso a la información digital se ha incrementado al paso del tiempo siendo cada vez más necesario el poder contar con las herramientas suficientes para poder hacer uso de este servicio.

La documentación digital exigida por las instituciones educativas se encuentra a disposición de cualquier persona con acceso a internet y hace posible su consulta en cualquier momento que se desee. Esto es así para casi toda las personas que tengan acceso a internet y a medios digitales, sin embargo hay un grupo grande de personas, según la OMS más de 285 millones de personas en el mundo padecen de discapacidad visual [1], las cuales se ven en la dificultad de acceder a servicios de internet y medios digitales, y en consecuencia a todos sus beneficios, ya que para el uso de dicho servicio se requiere de un monitor en el cual se despliega la información que a través del sentido de la vista se recibe.

En la actualidad existen ciertos dispositivos que permiten a las personas con discapacidad visual poder navegar en un computador. Está el teclado Braille digital [2] que corresponde a un teclado con 6 u 8 botones que mediante la combinación de sus pulsaciones en conjunto se puede lograr digitar una letra o un número según sea el caso.

Entre estos dispositivos también se encuentra uno más avanzado el cual es la línea Braille [3] que tiene una forma rectangular en la mayoría de sus presentaciones y contiene diversos puntos Braille en una fila sucesiva horizontal, mediante los cuales se transmite la información recibida desde el computador, realizando levantamiento de los puntos Braille para causar el relieve y la combinación necesaria para la representación al lenguaje Braille, algunos además cuentan con un teclado Braille incorporado y otros botones

con los cuales se podrá retroceder una palabra, saltar a otra palabra, entre otros.

Figura 2 Teclado braille [2]

Figura 1 Línea braille [4].

Estos tipos de dispositivos cubren las necesidades de las personas no videntes para poder acceder a la información contenida en un computador, pero son sumamente costosos. Un equipo de estos puede llegar a costar alrededor de mil quinientos dólares ($1.500) [5] dependiendo del modelo y los costos de importación asociados al país en que se solicite.

Muchas de estas personas no están en la mejor situación económica como para poder costear este tipo de productos o se encuentran en países donde le es difícil conseguir el cambio de moneda local a dólar o simplemente importar productos.

Pensando en un mundo incluyente donde las oportunidades para cualquier persona sean equivalentes sin importar el color, etnia, género o discapacidad, se pretende desarrollar un dispositivo de salida en el cual se emitirá la información transcrita del alfabeto común al alfabeto Braille desde el computador a una tarjeta microcontroladora y un software de lectura con el cual se podrá cargar y transformar la información de documentos de texto a lenguaje de caracteres Braille.

Objetivos

Objetivo general

Desarrollar un dispositivo para la lectura de texto plano que permita a personas con discapacidad visual, poder interactuar con un computador utilizando el sistema alfabético Braille.

Objetivos específicos

1. Diseñar e implementar un entorno de contenido para la carga y manejo de archivos de texto plano.
2. Diseñar e implementar un módulo de reconocimiento de posicionamiento y transición inteligente del cursor entre caracteres alfanuméricos presentados en torno a la lectura.
3. Construir un dispositivo de salida que permita representar textos alfanuméricos en el sistema alfabético Braille.
4. Investigar y analizar los distintos tipos de microcontroladores a utilizar en el dispositivo con salida al sistema alfabético Braille para el transporte y conexión de la información contenida en el computador.
5. Crear un dispositivo traductor que permita la transformación de las señales digitales a código Braille.
6. Diseñar e implementar un módulo de reconocimiento de frontera de contenido.
7. Diseñar e implementar un módulo de notificación.

Alcance

En este trabajo de grado se pretende desarrollar un dispositivo de lectura que emule el comportamiento habitual de la lectura Braille mediante un software de carga de archivos de texto plano en el computador que será procesado y emitido por un dispositivo de salida en su forma del alfabeto Braille.

1. Diseñar e implementar un entorno de contenido para la carga y manejo de archivos de texto plano

Este módulo corresponde al entorno de carga de documentos para el procesamiento de lectura y transcripción de caracteres en el cual se podrá importar documentos de texto plano a un software de lectura que administrará el contenido será procesado, transmitido y finalmente leído a través del dispositivo de lectura en forma de caracteres Braille.

2. Diseñar e implementar un módulo de reconocimiento de posicionamiento y transición inteligente del cursor entre caracteres alfanuméricos presentados en torno a la lectura

En el módulo de posicionamiento, se estudiará e implementará la detección de caracteres planos que estén cargados en el entorno de lectura, para luego ser procesados y emitidos a la tarjeta microcontroladora la cual se encargará de realizar la representación en puntos Braille; Se estudiará la posibilidad de identificar si el cursor está sobre la presencia de un enlace o imagen haciendo uso de las librerías que permiten acceder tanto a funcionalidades como a información del entorno del sistema operativo. La librería OS de Python [7] proporciona acceso a las funcionalidades del sistema operativo UNIX o Windows, con lo cual se realizará un análisis de los métodos que esta librería ofrece para identificar los que permitan la recolección de texto plano del entorno.

El módulo de posicionamiento y transición inteligente del contenido permitirá la detección de cadenas de caracteres de texto plano la cual podrá

ser simultánea ya que se podrá identificar una cadena de caracteres (palabra completa) por lo cual se estudiará e implementará un método de transición de caracteres para recorrer la palabra, de manera automática o manual según el usuario lo disponga mediante el uso de algoritmos de lectura y procesamiento que agilicen la recolección del texto contenido de manera horizontal y el procesamiento y emisión entre tiempos estimados que permitan una lectura fluida acorde al sistema de lectura Braille.

3. Construir un dispositivo de salida que permita representar textos alfanuméricos en el sistema alfabético Braille

El módulo de representación corresponde a la emisión de la información recibida del módulo de transcripción mediante el uso de micro solenoides que estarán conectados a la tarjeta microcontroladora para emular el comportamiento de representación Braille realizando levantamientos de los pines correspondientes al carácter en representación similar a lo que se muestra en la figura 3. Este módulo podrá ser suplantado posteriormente, en una mejora del trabajo de grado por polímeros electroactivos [9] ya que este material posee la capacidad de alterar su estado físico mediante la conducción de energía.

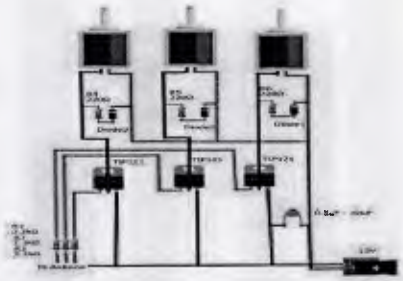


Figura 3 Representación de caracteres [8].

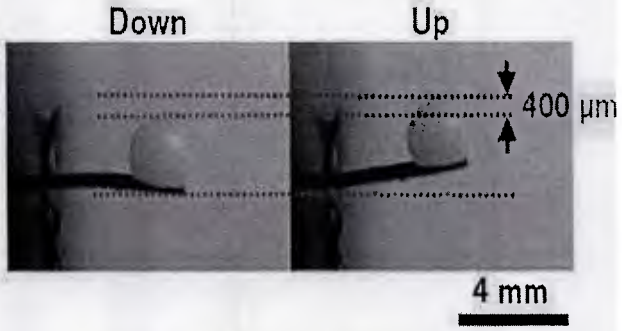


Figura 4 Polímero Electroactivo [9].

4. Investigar y analizar los distintos tipos de microcontroladores a utilizar en el dispositivo con salida al sistema alfabético Braille para el transporte y conexión de la información contenida en el computador

Se realizará el estudio de los diferentes tipos de microcontroladores a implementar para la conexión, procesamiento e intercambio de información entre el módulo de salida al sistema alfabético Braille y el entorno de lectura contenido en el computador, será seleccionado el componente que mejor se adapte y brinde mayores beneficios a los requerimientos para el desarrollo de la solución.

5. Crear un dispositivo traductor que permita la transformación de las señales digitales a código Braille

El módulo de transcripción permitirá la conversión de caracteres de texto plano a su formato en el alfabeto Braille. Mediante la recolección de caracteres planos en el módulo anterior se podrá realizar la conversión a un arreglo de longitud seis, el cual contendrá la representación en Braille correspondiente en bits (1 ó 0), para luego ser transmitido a una tarjeta Microcontroladora mediante una conexión USB y se estudiará el uso de la biblioteca Pyserial [6], así como también otras bibliotecas para la interconexión entre la tarjeta microcontroladora y el computador.

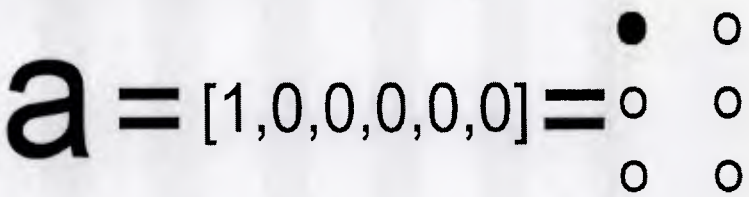


Figura 5 Conversión de carácter [Fuente propia].

6. Diseñar e implementar un módulo de reconocimiento de frontera de contenido

Este módulo corresponde a la identificación de terminación de contenido en el entorno a desarrollar, con el cual se podrá identificar si se ha llegado al borde horizontal o vertical del contenido o si bien hemos llegado al final de un archivo o inicio del mismo.

7. Diseñar e implementar un módulo de notificación

En el módulo de notificaciones se analizará la forma en que se comunicará el contenido especial o notificaciones de la aplicación para que el usuario pueda ser informado de notificaciones como el comienzo de la lectura o finalización de la misma, posteriormente a dicho análisis se procederá a desarrollar.

Limitaciones

- Los módulos de manejo de contenido en el computador serán desarrollados en Python o Java.
- El controlador de la interacción entre los módulos es una placa microcontroladora.
- La solución será desarrollada para el funcionamiento bajo el sistema operativo Windows debido a la librería implementada (PANAMAHITEK_ARDUINO) en la comunicación serial con el Arduino solo detecta puertos COM de Windows.
- El dispositivo de lectura solo recibirá información a través de los documentos cargados (Word, txt y PDF) en el entorno de lectura Braille.
- La solución recibirá información del contenido de texto plano cargado en el entorno de lectura Braille.
- La representación de puntos Braille será en su formato de 6 pines con un máximo de 64 caracteres a representar.
- La licencia de Microsoft Word tiene que estar activa al momento de la carga del documento ya que la librería valida que el software no sea activado ilícitamente.

Justificación

Actualmente no existe un software independiente que permita detectar el texto en pantalla basándose en la posición del cursor y pueda transmitirlo a una unidad de control que sea capaz de realizar una salida con representación en el alfabeto Braille.

Esta solución pretende emular el comportamiento de lectura habitual del Braille de manera automática, donde el usuario podrá recibir los caracteres en un tiempo estimado entre cada uno, para lograr una lectura fluida y privada para el usuario del dispositivo.

Con esta solución el usuario podrá tener una experiencia similar computarizada donde podrá leer de manera típica y privada los caracteres presentados en el entorno de lectura a desarrollar, con la simplicidad de inicializar el proceso mediante la carga de la información en el entorno de lectura que será interpretada mientras simultáneamente se van representando en su formato Braille, permitiendo la interacción del usuario al contenido de documentos de texto reflejados en el entorno de lectura de una manera práctica y sencilla.

El entorno de lectura permitirá la detección de caracteres de texto plano mediante el posicionamiento del cursor, para luego ser transformando en su formato Braille que será enviado a la placa microcontroladora.

La placa microcontroladora es el receptor de la información que emite el entorno de lectura desde el computador y se encargará de dar salida en los pines correspondientes para realizar la representación de la información en su forma Braille.

Para la representación de la información se utilizarán seis solenoides que cumplen la función de levantamiento del punto Braille que representará la salida correspondiente al carácter en relieve.

Estimación del costo del dispositivo

Dispositivos de lectura Braille

En la siguiente tabla de comparaciones se pretende aproximar el costo de producción para la solución propuesta en contraparte con algunos dispositivos ya existentes en el mercado que cumplen funciones similares. Para la estimación del costo se utilizarán dos precios referenciales del mercado, siendo uno el más económico encontrado y el otro el más costoso encontrado en los diferentes tiendas de internet que cumplan las mismas funcionalidades básicas requeridas para la comparación como lo es la emisión de caracteres en puntos Braille.

Tabla 1 Comparación de líneas Braille [Fuente propia].

Imagen	Dispositivo	Características	Costo
	Brailliant Bi 32 [10]	32 celdas Braille USB 2.0 y Bluetooth V2.1 Teclado Braille incorporado Tamaño 260 x 87 x 18 mm Peso: 524 g	$ 2,595
	Braille sense U2 [11]	32 celdas Braille Bluetooth incorporado 3 puertos usb Telado incorporado Procesador de 1 Ghz, Memoria flash 32 Gb 17 horas de batería continua Tamaño: 250x157x20mm	$4,795

		Peso:924 g	
	BrailleNote Apex Bt 32 [12]	32 celdas Braille Tres USB 2.0 GPS receptor via bloototh Teclado Braille incorporado Conexión para micrófono y audífonos. Tamaño: 19 x 244 x 143 mm Peso: 812 g	$5,495
	SyncBraille [13]	20 celdas Braille USB 2.0 Sensor de voz Par de botones en cada lado Tamaño:187 x 85 x 20 mm Peso:280 g	$1,995
	Baum Supervario 32 [14]	32 celdas Braille Cursor integrado Batería de hasta 25 hrs usando Bluetooth 3 navigation keys left and 3 navigation keys right of Braille display Tamaño: 260 x 20 x 76 mm (appr.	$2,995

		10.2 x 0.8 x 3 in.) Peso 480 g	
	Baum Supervario 40 [15]	40 celdas Braille Cursos integrado Batería de hasta 25 hrs usando Bluetooth 3 navigation keys left and 3 navigation keys right of Braille display Tamaño: 310 x 20 x 76 mm (appr. 12.2 x 0.8 x 3 in.) Peso: 560 g	$3,495

Dispositivo a desarrollar

Solenoides varios

Para la estimación del precio de los solenoides se tomaron como referentes dos modelos que por su diminuto tamaño y características pueden ser aplicados a la solución siendo uno el más económico encontrado y la opción más costosa encontrada, se requieren 16 unidades en total para conformar las dos celdas Braille de 8 pines cada una que realizará la representación.

Tabla 2 Evaluación de solenoides. [Fuente propia].

Imagen	Dispositivo	Características	Costo unitario	Cantidad	Total
	DC Solenoid Electromagnet[16]	• Modelo ZYE1-0530 • Material Metal • Voltaje DC 12V • Tipo empujar, halar • Amperaje 1ª • Oscilación 10mm • Fuerza 600g • Tamaño 30x15x14mm(LxWxT) • Longitud del cable 17cm • Peso 32g	$ 3.23	16	$ 51.68
	DIY DC Electromagnet Magnet Solenoid[17]	• Modelo FJ-0630B • Material Metal • Voltaje DC 6V • Tipo empujar, halar • Amperaje 0.8V • Oscilación 10mm • Fuerza 6N	$ 10.54	16	$168.64

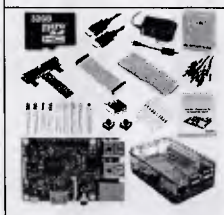
		• Tamaño 30x16x20mm (LxWxT) • Peso 45g			

Microcontroladores

Para la estimación del costo del microcontrolador a utilizar se realizó una búsqueda sobre los tres tipos de microcontroladores a evaluar con sus diferentes modelos, tomando en cuenta el modelo factible más económico y el más costoso de cada uno, como se presenta a continuación:

Tabla 3 Evaluación de microcontroladores. [Fuente propia].

Imagen	Dispositivo	Características	Costo unitario	Cantidad	Total
	Arduino Leonardo [18]	• Modelo ATmega32u4 • Peso 20g • Velocidad del reloj 16 MHz • Memoria Flash 32K • DC 5V • SRAM 2.5K • Pin Digital 20	$ 21.99	1	$ 21.99
	Arduino UNO R3 [19]	• Modelo A000066 • Peso 25g • Velocidad del reloj 16 MHz • Memoria Flash 32K • DC 5V • Pin Digital 14 • SRAM 2KB	$ 21.27	1	$ 21.27
	Raspberry Pi 3 [20]	• Procesador 1.2 GHz x64 con 4 núcleos • 802.11n Wireless LAN	$ 39.20	1	$ 39.20

		• Bluetooth 4.0 • 4 puertos USB • 1 puerto HDMI • Conector de audio y video 3.5mm • Ranura para tarjetas Micro SD • Tamaño 4.8x2.99x1.34 inches. • DC 5V, 2.5A			
	Raspberry Pi 2 [21]	• Modelo RPi2 B • Procesador Quad-Core 900 MHz 1 GB RAM • CanaKit WiFi Adapter / Dongle (RT5370 chipset) • CanaKit 2.5A Micro USB • Ranura para tarjetas Micro SD • Bluetooth 4.0 • 4 puertos USB	$ 84.99	1	$ 84.99
	PIC16F877A-I/P DIP-40 Microchip Microcontroller IC [22]	• Modelo PIC16F877A-I/P DIP-40 • Procesador 8 Bits 20 MHz • Memoria de capacidad programable 14KB • Ram 368x8 • Voltaje 4V~5.5V • Rango de temperatura	$ 9.28	1	$ 9.28

		para trabajar -40°C ~ 85°C • I/O 33			
	AT32UC3A3256-ALUR IC MCU 256KB FLASH 144LQFP 32UC3 AT32UC3 [23]	Modelo AT32UC3A3256-ALUR • Procesador 32 Bits 66 MHz • Rango de temperatura para trabajar -40°C ~ 85°C • Memoria de capacidad programable 256k • Voltaje 1.65V~1.95V • Ram 128kx8 • I/O 110	$ 19.53	1	$ 19.53

Como se logra apreciar, el costo de los solenoides, los microcontroladores y los PIC suman un total de $119.35 aproximadamente para la versión más económica y $351.28 para la versión más costosa, agregándole en ambas partes un estimado de gastos varios asumidos por $100 para la cobertura de cables, placas y materiales para el diseño del contenedor del dispositivo, con lo cual se logra apreciar que la propuesta a desarrollar es más económica que las alternativas existentes en el mercado.

Precio más bajo

Dispositivo	Costo
DC Solenoid Electromagnet	$ 51.68
PIC16F877A-I/P DIP-40 Microchip Microcontroller IC	$ 9.28
Gastos varios	$ 100
Total	$ 160.96

Precio más alto

Dispositivo	Costo
Raspberry Pi 2	$ 84.99
DIY DC Electromagnet Magnet Solenoid	$ 168.64
Gastos varios	$ 100
Total	$353.63

Capítulo II

Marco referencial

El Alfabeto Braille

El Braille es un sistema diseñado para personas con discapacidad visual creado por el francés Louis Braille a mediados del siglo XIX, el cual representa letras por medio de una combinación de puntos; este sistema no es un idioma sino un alfabeto, la distribución de los puntos se basa en dos columnas y tres filas para un total de 64 combinaciones, para identificar las letras mayúsculas y los números se utilizan diferenciadores especiales antes de la misma, indicando la aproximación de uno de estos caracteres. Al principio el Braille fue creado con la finalidad de utilizarlo como un código para tácticas y estrategias militares como una manera de comunicación silenciosa entre las tropas, con la introducción de los sistemas informáticos el Braille se amplía a ocho (8) puntos para que de esta manera se pueda representar el carácter ASCII en una celda. Se necesita el sentido del tacto para detectar los puntos sobre el relieve y realizar la interpretación de los caracteres, y se realiza de izquierda a derecha en forma horizontal, existen distintas formas de leer Braille, una de ella es realizada con un dedo que por lo general es el dedo índice, con esta técnica suele ser difícil el cambio de línea y se producen alteraciones de ritmo lo que ocasionando que la velocidad de lectura no sea la adecuada aumentando la fatiga, otra forma de leer es ayudando el dedo lector con los otros dedos bien sea de la misma mano o de la otra para reforzar la información obtenida, esto reduce la fatiga y se requiere menor atención ya que mientras una mano "lee", la otra mano puede ubicar la siguiente línea o la siguiente letra. El alfabeto Braille actualmente es implementado en múltiples sitios, se puede encontrar en billetes canadienses para ser fácilmente identificados por gente con problemas en la vista, por ejemplo, en Paraguay el banco central desde el año 2009 puso en circulación un billete de 2000 guaraníes con el sistema de Braille para personas ciegas, en Inglaterra es de carácter obligatorio que las medicinas tengan el nombre en Braille.

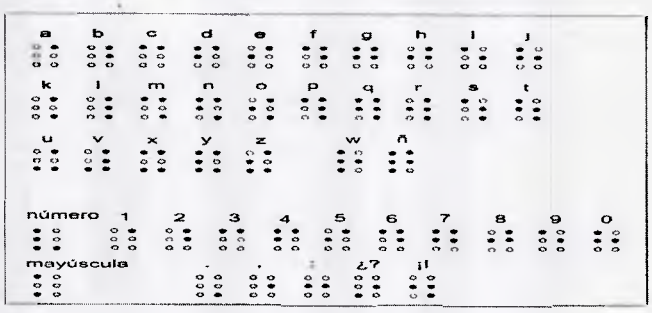


Figura 6 Alfabeto Braille [24].

Interfaz gráfica de usuario

Conocida también como GUI (Graphical User Interface), es la interfaz que los usuarios utilizan para comunicarse con el sistema operativo del computador bien sea bajo un ambiente en Windows o en un ambiente Unix, la intención de introducir GUI a los sistemas operativos fue con la finalidad de facilitar el trabajo al usuario final puesto que ya no tendrían que aprender múltiples comandos para la ejecución de los procesos en el sistema operativo manejado. Para las personas ciegas es necesario desarrollar soluciones no solo a un precio razonable, sino que adicionalmente puedan ser de utilidad a la persona ciega a una velocidad razonable y esto se puede lograr con un nivel de información adicional, de allí la importancia de que las personas ciegas puedan acceder a los mismo sistemas y aplicaciones informáticas que las personas no ciegas, para integrarse en un entorno profesional. Las soluciones más viables para los usuarios ciegos, es que incluyan una mezcla de salida de audio y táctil como el método Braille.

Arduino

Es una plataforma de código abierto que se basa en facilitar el uso de componentes electrónicos y la programación, posee una herramienta de programación para escribir código e insertarlo en la tarjeta controladora. El hardware es una placa de circuito impreso tal como se demuestra en la figura 7, con un microcontrolador por lo general es el Atmel AVR posee puertos digitales y analógicos de I/O (entrada y salida por sus siglas en inglés),

adicionalmente posee un puerto USB de donde puede alimentarse la placa y establecer la comunicación con una computadora. El software es un entorno de desarrollo (IDE por sus siglas en inglés) se basa en el entorno Processing y en el lenguaje de programación se basa en Wiring, además posee es de gran portabilidad ya que puede ser ejecutado en múltiples sistemas operativos como son Linux, Windows y Mac, es de hardware económico si se compara con otros microcontroladores desde 8 bits en adelante, proporciona una ambiente de desarrollo simple, ya que puede ser utilizado desde principiantes, profesores, hasta usuarios avanzados.

Figura 7 Arduino UNO [25].

Solenoide

Es un conjunto de espiras arrolladas consecutivamente creando un campo magnético intenso en su interior y débil en su exterior como se muestra en la figura 8 capaz de transformar la energía eléctrica entrante en energía mecánica, la ventaja del solenoide consiste en que, el campo magnético generado dentro de solenoide es más uniforme cuanto más larga sea, su principal desventaja es que no puede producir un campo magnético elevado. Los solenoides pueden ser encontrados principalmente en los seguros de las puertas de un carro sirven para abrir o cerrar la(s) puerta(s) de un carro al momento de accionar la alarma, también es utilizado para permitir o no el paso de corriente eléctrica y de alto amperaje, que es conducido por un cable de alto de grueso calibre.

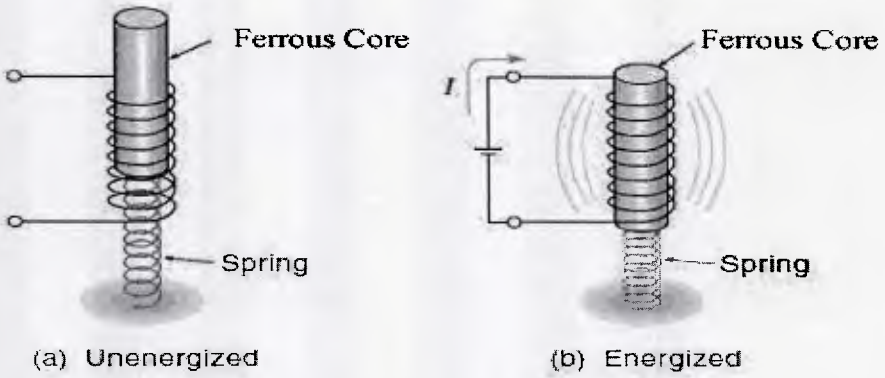


Figura 8 Campo magnético en solenoide [26].

Transistor

El transistor es un dispositivo electrónico semiconductor que emite una señal en respuesta a una señal de entrada, este puede cumplir diversos roles dentro de un circuito como son: amplificador, oscilador, conmutador o rectificador, actualmente se puede encontrar en diversos aparatos eléctricos tales como: radio, televisores, computadores, celulares, etc. El transistor consta de tres partes dopadas cada una de ellas contaminadas con materiales específicos en cantidades específicas formando dos uniones bipolares: el emisor que emite los portadores (partículas libres portadora de cargas eléctricas), el colector lo recibe y los recolecta de allí su nombre de recolector y la base que está intercalada, esta se encarga de modular el paso de los portadores.

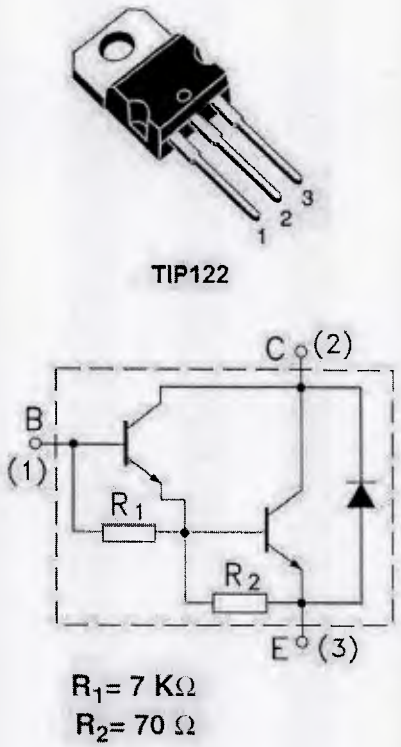


Figura 9 Transistor TIP122 [27].

Resistencia

Es un dispositivo electrónico diseñado para oponer el flujo de electrones al moverse dentro del conductor en dos puntos determinados del circuito eléctrico, a diferencias de otros componentes electrónicos las resistencias no tienen polaridad definida (se pueden colocar de cualquier forma). La característica principal de la resistencia son los 3 valores: resistencia eléctrica disipación máxima y la precisión o tolerancia, estos valores son denotados con un conjunto de rayas de colores sobre el cuerpo de la resistencia, se lee de derecha a izquierda dejando la raya de tolerancia (normalmente plateada o dorada) a la derecha, se lee de izquierda a derecha, 1° cifra significativa, 2° cifra significativa, 3° multiplicador, 4° tolerancia.

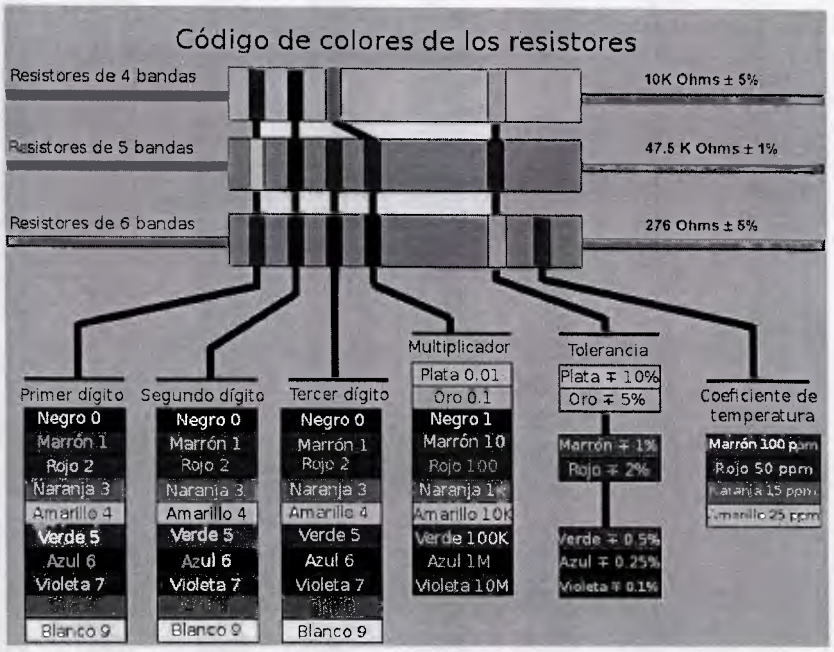


Figura 10 Código de colores de los resistores [28].

GIT

Es una herramienta que permite la integración y control de las versiones de un software [36], brindando los beneficios de registrar un histórico de modificaciones realizadas a los archivos que se encuentren en la carpeta reconocida por esta tecnología. GIT es implementado en ambientes de desarrollo distribuidos, debido a que facilita la integración entre las modificaciones que cada integrante pueda realizar sobre un documento, quitando la labor de integración que podría conllevar la misma y permitiendo el desarrollo no lineal, brindando la oportunidad que cada integrante pueda realizar sus tareas en paralelo. Entre las características principales de GIT se tiene:

- Un desarrollo no lineal, lo que garantiza rapidez en la gestión de ramas y mezclado en las versiones
- Una gestión distribuida, GIT le da a cada uno una copia local y los cambios se propagan entre cada uno de los repositorios locales
- La información pueden publicarse vía HTTP, FTP, TCP/IP o SSH.
- Todas las versiones previas a un cambio determinado implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio

Comandos básicos

1. **Commit**: Es la acción que se realiza para guardar los cambios de manera local.
2. **Pull**: Ésta acción permite obtener la última versión subida al servidor.
3. **Push**: Con la opción del Push se logra subir los cambios guardados de manera local al servidor.
4. **Merge**: El Merge permite unificar dos ramas (Branch) distintos en una única rama como resultado final.

Branch

El branch o rama de desarrollo es un espacio de trabajo que permite llevar la secuencia de cambios de manera secuencial entre archivos. Como buena práctica se suele dividir los branch por módulos o estaciones de trabajo, con la finalidad de separar el desarrollo en distintos ambientes e integrándolos en una fase final en una rama superior, la cual comúnmente suele ser la rama Master.

Github

GitHub es una plataforma de desarrollo para alojar proyectos usando el sistema de control de versiones Git, este cliente proporciona un ambiente predispuesto al desarrollo, sin tener que contar con un servidor propio e instalaciones y configuraciones de una plataforma.

Pruebas unitarias

Las pruebas unitarias tienen como finalidad revisar el correcto funcionamiento de un módulo o unidad en particular para prever distintos tipos de variantes situacionales para el mismo, lo que garantiza la cobertura de eventualidades que puedan suscitar en un futuro a dicho módulo o unidad, esto provee mejoras en el desarrollo ya que cada prueba realizada garantiza, que desde momento se cumplirá su función y si llega a suscitar un cambio que altere el funcionamiento de la unidad, la prueba unitaria arrojará un valor inesperado

que será visto como una falla a corregir. Entre las características principales de una prueba unitaria se tiene:

- Sean automatizables no deben requerir una intervención manual.
- Sean completas, deben cubrir la mayor cantidad de código posible.
- Sean repetibles o reutilizables, no deben ser creadas para sean utilizadas una sola vez
- Independientes, la ejecución de una prueba no debe afectar la ejecución de otra.
- Documentadas, al igual que un código debe ser documentada.

Java

Java es un lenguaje orientado a Objetos desarrollado por James Gosling de Sun MicroSystem en el año 1996, años después fue comprado por Oracle el cual es actualmente el propietario de dicho lenguaje, la mayor parte de su sintaxis deriva de C y C++ pero tiene menos utilidades de bajo nivel que cualquiera de ellos, la principal intención de este lenguaje de programación es permitir a los desarrolladores de aplicaciones escribir el programa una sola vez para luego ser ejecutado en cualquier sistema operativo, lo que quiere decir es que el código ejecutado en un plataforma no tiene que ser recompilado para correr en otra. Java a partir del año 2012 es uno de los lenguajes de programación más populares y cuenta con más de 8 millones de aplicaciones a nivel mundial, lo que lo hace un lenguaje de programación competitivo en el mercado laboral. Las aplicaciones de Java pueden ser ejecutadas en cualquier Java Virtual Machine (JVM) sin importar el sistema operativo o arquitectura del computador.

Arquitectura MVC

La arquitectura MVC (Modelo – Vista – Controlador) es una arquitectura de tres capas que permite separar las funcionalidades de una aplicación, facilitando las futuras mejoras sobre cambios en cualquiera de sus capas y brindando mayor entendimiento a los desarrolladores al momento de interpretar y mantener un código brindando los beneficios de modularidad del código, obligando a separarlo por capas, de esta manera se evita el llamado "código espageti"[29]. Ésta arquitectura es empleada indirectamente por la tecnología de JavaFX.

Las capas en que se divide la arquitectura MVC se describen a continuación:

Modelo

Es la capa en la cual se trabaja con datos, por lo tanto, esta capa contiene métodos o funciones para acceder a la información, por ejemplo, las estructuras de los objetos a usar en la aplicación, conexión con base de datos, carga de archivos físicos, entre otras.

Vista

Esta capa contiene toda la parte de la visualización de la interfaz de usuario, es decir solamente el código que permite mostrar la salida por pantalla de la información y entrada de datos que permite al usuario interactuar con el software.

Controlador

Es la capa que realiza el enlace entre la vista y los modelos, responde a los mecanismos que puedan surgir para implementar las necesidades de la aplicación, sin embargo, esta capa no manipula ni almacena directamente datos, solo cumple con la finalidad de obtener los datos, interpretarlos y generar una respuesta a la capa de vista para que pueda dar respuesta a una acción del usuario.

Figura 11 Modelo vista controlador [30]

Capítulo III

Marco Metodológico

Para el desarrollo de este sistema se utilizará la metodología de desarrollo Orientado a Pruebas. Esta es una metodología de desarrollo ágil con una fortaleza fundamental, y es que unifica el proceso de prueba de una aplicación con el proceso de desarrollo, lo que hace que el producto final sea un programa robusto y a prueba de fallos, a la vez que ahorra costos.

Dicha metodología requiere una organización muy meticulosa y un sistema de desarrollo muy estructurado, pero a cambio obtienen productos finales probados en un tiempo muy razonable.

Se comienza diseñando y escribiendo las pruebas unitarias en base a requisitos del software y a la arquitectura del proyecto, seguido se codifica el sistema de tal manera que el código implementado cumpla las pruebas diseñadas previamente, de esta forma se logra un código funcional y seguro; después se usa la refactorización para hacer el código más eficiente, así al final se vuelve a ejecutar la prueba verificando que todo esté acorde a lo planteado. Se parte de los requisitos, los cuales deben ser traducidos a pruebas de tal forma que cuando todas superen los requisitos y todos sus derivados, se podrá asegurar que la aplicación funciona correctamente.

La metodología cumple el siguiente ciclo:

1. **Elegir requerimiento**

   Se escoge el requerimiento más simple con el cual se comenzará a trabajar.

2. **Escribir prueba del requerimiento**

   El requerimiento seleccionado se convierte en una prueba en la cual se representará las posibles salidas y comportamiento del mismo.

3. **Verificar prueba**

   Inicialmente la prueba debe fallar debido a que el código no está implementado, si la prueba es exitosa se deberá revisar ya que no se realizó de forma correcta.

4. **Realizar código**

   En esta etapa se procede a escribir el código correspondiente al requerimiento el cual deberá cumplir con la prueba anteriormente realizada.

5. **Ejecutar prueba**

   Se verifica la prueba correspondiente al requerimiento la cual deberá ser exitosa lo cual implica que el código escrito en la etapa anterior es correcto y cumple con lo establecido de esta manera se garantiza que el código se comportara de la manera que se ha previsto en la prueba, de no ser así se repite el paso anterior hasta conseguir el éxito en la prueba.

6. **Refactorizar**

   Una vez culminado los pasos anteriores se procede a optimizar el código mejorando varios aspectos como la legibilidad, eficiencia y disminuyendo la complejidad.

7. **Volvemos al paso 1**

   Se repite el ciclo nuevamente hasta cubrir todos los requerimientos y así concluir la aplicación.

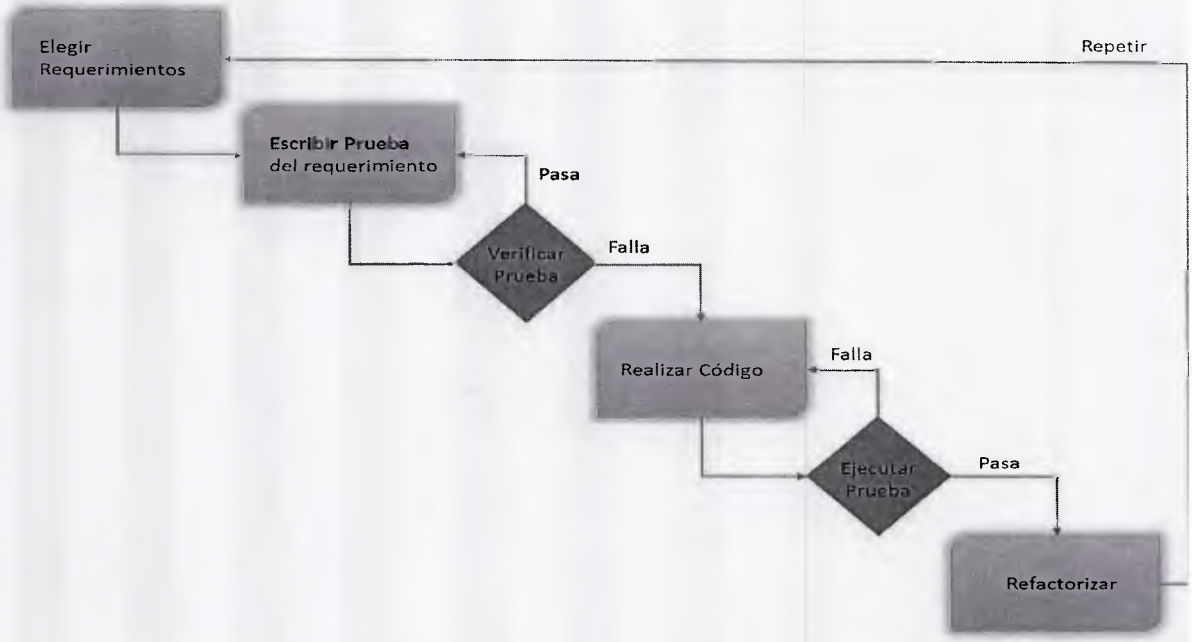


Figura 12 Ciclo del TDD [Fuente propia].

Se utilizará el desarrollo orientado a pruebas (TDD) ya que optimiza el proceso del desarrollo acotando el tiempo de prueba ya que estas se cubren antes de codificar, lo que garantiza un comportamiento previsto en el software además de un aumento de la calidad del software debido a que se realiza la refactorización del código de forma inmediata.

Al realizar la prueba antes que el código es necesario escribir el mínimo de funcionalidad necesaria, lo cual evita el diseño reiterativo; por otra parte, las pruebas realizadas sirven de documentación del sistema a desarrollar. Esto deja un producto final completo y probado reduciendo considerablemente la cantidad de tiempo invertido en desarrollo de la solución.

Diagrama de Gantt

Actividad	Fecha inicio	Fecha de fin
Documentación y reunión con el centro de servicios para personas con discapacidad visual	13/06/2016	13/06/2016
Investigación de los diferentes tipos de microcontroladores	14/06/2016	17/06/2016
Análisis comparativo y selección del microcontrolador a usar	18/06/2016	20/06/2016
Módulo de contenido		
Diseño y desarrollo de la interfaz del entorno de lectura Braille	21/06/2016	22/06/2016
Desarrollo del módulo de carga de documentos en el entorno de lectura Braille	23/06/2016	28/06/2016
Desarrollar y realizar pruebas de interacción entre el microcontrolador y el entorno de lectura	28/06/2016	01/07/2016
Módulo de posicionamiento		
Diseñar y desarrollar un módulo de posicionamiento inteligente del foco de lectura	04/07/2016	11/07/2016
Desarrollar la acotación de captura de contenido de forma horizontal en el desplazamiento entre caracteres	12/07/2016	17/07/2016
Desarrollar el desplazamiento del foco de lectura a la parte inicial y final de un párrafo	18/07/2016	22/07/2016
Módulo de representación		
Diseño de circuito electrónico para el módulo de representación Braille	25/07/2016	27/07/2016
Desarrollar algoritmos de captura y emisión de datos para la representación Braille	27/07/2016	29/07/2016
Realizar pruebas para el reconocimiento de instrucciones en el microcontrolador que serán emitidas por los solenoides	29/07/2016	31/07/2016
Módulo de transcripción		
Desarrollar módulo de transcripción de caracteres alfanuméricos a caracteres Braille	01/08/2016	05/08/2016
Desarrollar el módulo de frontera		
Desarrollar reconocimiento de inicio de párrafo en un texto	08/08/2016	11/08/2016

Desarrollar reconocimiento de finalización de párrafo de un texto	08/08/2016	11/08/2016
Desarrollar reconocimiento de finalización de contenido	12/08/2016	16/08/2016
Desarrollar reconocimiento de comienzo de contenido	12/08/2016	16/08/2016
Módulo de notificaciones		
Diseñar y analizar las notificaciones que se desarrollaran	17/08/2016	20/08/2016
Desarrollo de las notificaciones para la notificación del contenido especial	22/08/2016	26/08/2016
Módulo de integración		
Realizar pruebas integrales para la conexión con el microcontrolador y el entorno de lectura Braille	29/08/2016	02/09/2016
Realizar pruebas integrales para la conexión entre el microcontrolador y el módulo de representación Braille	19/09/2016	23/09/2016
Realizar pruebas de conexión general	26/09/2016	28/09/2016
Realizar pruebas de funcionamiento con personas invidentes	03/10/2016	14/10/2016
Realizar correcciones sobre resultados de pruebas unitarias	17/10/2016	21/10/2016

Capítulo IV

Desarrollo

Debido a que el desarrollo fue realizado por dos integrantes y motivado a la ejecución de procesos en paralelo que permita la agilización en el cumplimiento de las actividades, se hizo uso de la tecnología de GIT, alojando el desarrollo en los servidores de GitHub, bajo la licencia gratuita que tiene como condicional permitir la visualización e integración de terceras personas al desarrollo, por consiguiente, el proyecto es OpenSource.

Para el flujo de trabajo se emplearon tres (3) ramas que permiten el desarrollo en paralelo de actividades independientes y la integración de las mismas de manera sencilla, las cuales están compuestas en Backend, Frontend y como rama principal el Master:

Backend: En dicha rama, se encuentra la lógica del desarrollo y las estructuras de los objetos que serán manipulados a través de las aplicaciones para permitir el funcionamiento de la aplicación desktop como el código para la programación del Arduino.

Frontend: La rama del frontend, recopila toda la parte estética de la aplicación, que tiene relación directa con el usuario, como lo son las imágenes, interfaces gráficas, diagramas y pruebas unitarias.

Master: Ésta es la rama más importante ya que integra el frontend y el backend en un único flujo de trabajo que da como resultado el producto final funcional.

1. Diseñar e implementar un entorno de contenido para la carga y manejo de archivos de texto plano

Cumpliendo con el flujo de la metodología se realizaron prototipos de diseños para la interfaz de carga para el manejo de archivos de texto plano utilizando la herramienta de Balsamiq Mockups [31], lo que permite realizar mejoras y correcciones antes de implementar la interfaz en código, posterior a ello se desarrolló la interfaz gráfica utilizando la tecnología de JavaFx, el cual cuenta con una arquitectura de trabajo dinámica e interfaces agradables que permiten la agilización en la construcción de dicha interfaz. Para la carga de documentos de texto plano se realizó las estructuras de los métodos que serán invocados para la carga de los mismos, a los cuales se le realizaron pruebas unitarias para definir los parámetros correspondientes a dichos métodos y los resultados que se iban a obtener mediante dichas llamadas; Luego de ello se realizó el código, dando como resultado métodos genéricos para la carga de archivos de texto plano, que fueron llamados a través del menú de carga en la barra superior y en el proceso de "Drag and drop" que se activa al arrastrar un documento a la parte central de la interfaz.

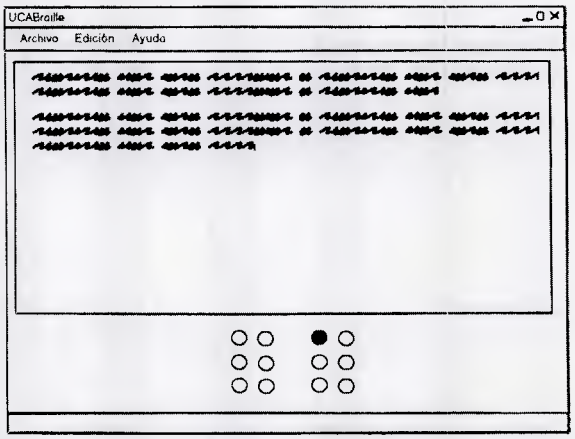


Figura 13 Interfaz para el manejo de archivos [Fuente propia]

2. Diseñar e implementar un módulo de reconocimiento de posicionamiento y transición inteligente del cursor entre caracteres alfanuméricos presentados en torno a la lectura

Para la implementación del módulo de reconocimiento y posicionamiento del cursor se definieron los métodos que realizan la acción de adelantar o retroceder en la lectura de los caracteres, los cuales fueron probados a través de pruebas unitarias definiendo la llamada a esas funciones y la respuesta esperada, como resultado se obtuvo una interfaz la cual posteriormente fue implementada en el patrón de diseño composite, esta permite relacionar objetos de manera recursiva haciendo uso de las similitudes entre objetos con cumplimiento a una jerarquía, por consiguiente, se pueden generar objetos más complejos a través de objetos simples.

En el desarrollo de UCABraille, fue preciso identificar que un texto se compone de párrafos, los párrafos de palabras y las palabras de letras, con lo cual es posible implementar el uso del patrón de diseño composite mediante la jerarquía previamente mencionada.

Dicho patrón permite de manera sencilla mantener niveles de ubicación que nos facilita la transición entre caracteres, ya que si deseamos acceder a la siguiente palabra solo es cuestión de obtener el carácter en foco, buscar su ancestro (Palabra) y solicitar la siguiente en la lista mediante el método de .la interfaz resultante de las pruebas unitarias.

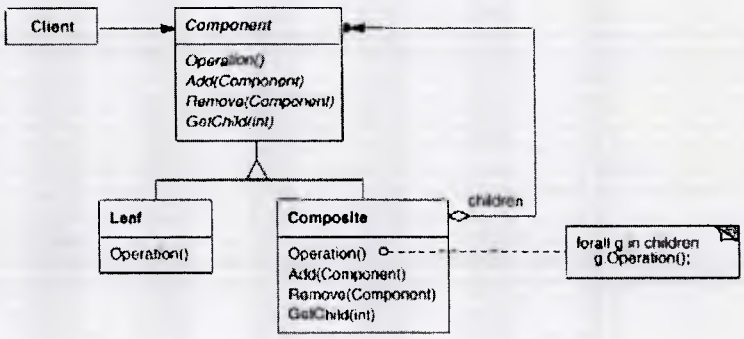


Figura 14 Patrón Composite [32].

Para realizar el desplazamiento manual de caracteres primeramente se diseñó un circuito en Circuit Market [33] que permitió determinar el comportamiento esperado para los pulsadores, posteriormente se realizó el desarrollo con dos pulsadores normalmente abiertos los cuales permiten que, al realizar una pulsación sobre ellos, se mande una señal al pin de entrada correspondiente para la acción y ésta pueda ser interpretada en el computador, para luego generar la acción correspondiente la cual es devuelta al Arduino en forma de arreglo de bits para su representación Braille.

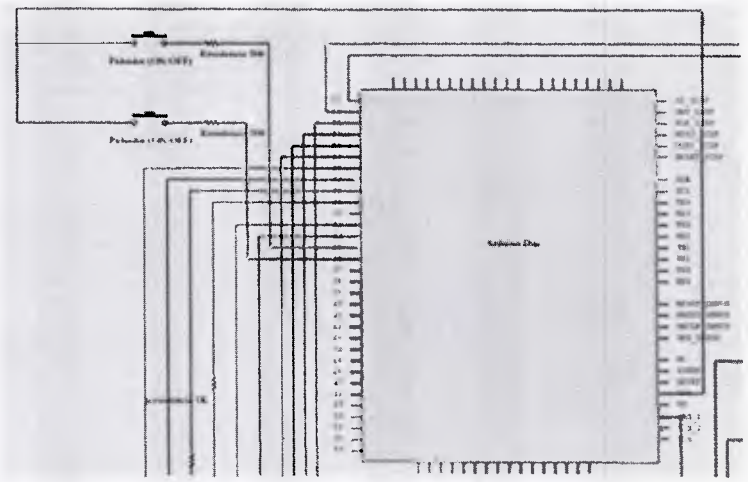


Figura 15 Circuito de los pulsadores [Fuente propia].

3. Construir un dispositivo de salida que permita representar textos alfanuméricos en el sistema alfabético Braille.

Para la representación en el alfabeto Braille se realizó el diseño de un circuito que permitiera el paso de corriente mediante una pulsación emitida por el arduino como activador de la representación, para ello se empleó el uso de transistores modelo Tip122 que permiten realizar la conmutación de corriente de forma digital, siendo este transistor de tipo NPN (Negativo - Positivo - Negativo). En el diseño del circuito se conectó al colector la salida negativa del componente representativo al pin Braille, que es alimentado a través de una corriente continua de una fuente; La base va conectado a los pin del Arduino a través de una resistencia de 500 Ohm que evita el rebote de señal al

dispositivo y por último el emisor es conectado a la tierra común para completar el circuito. El transistor trabaja como conmutador lógico, lo cual garantiza que mediante una salida del Arduino, el transistor permita que la señal negativa del componente representativo se dirija a tierra y sea activado.

Para validar que el comportamiento de los transistores era el deseado, se realizaron pruebas con diodos LED que permitieron detectar de manera visual el paso de corriente mediante las salidas del Arduino emitida a la base del transistor, utilizando un total de 12 diodos LED en paralelo, siendo el consumo de cada diodo de 1,6V, 5 miliamperios (mA) con lo cual el consumo máximo del circuito es de 1,6V, 60 miliamperios (mA).

Para la representación en el alfabeto Braille se decidió usar solenoides como salida ya que dicho componente permite generar un relieve mediante su barra central al recibir una señal eléctrica.

Se utilizó un total de 12 solenoides que sustituyeron a los diodos LED empleados en las pruebas de transistor, cada uno cuenta con medidas de 3 cm de alto y 1.2 cm de ancho, y un consumo de 5V, 1.75 Amperios con lo cual el consumo máximo del circuito es de 5V, 21 Amperios, al estar los 12 solenoides trabajando en conjunto para representar todos los puntos Braille posibles.

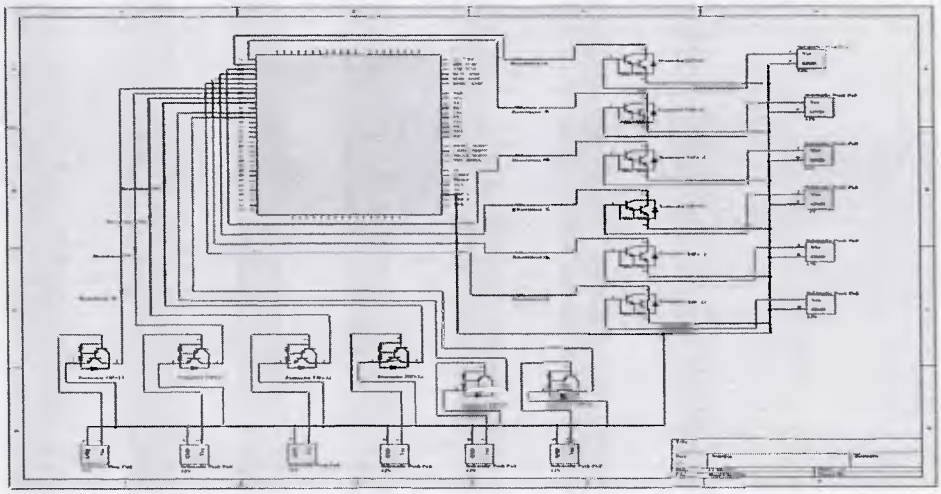


Figura 16 Circuito de representación braille [Fuente propia].

4. Investigar y analizar los distintos tipos de microcontroladores a utilizar en el dispositivo con salida al sistema alfabético Braille para el transporte y conexión de la información contenida en el computador

Para la investigación de los distintos tipos de microcontroladores se tomaron en cuenta tres modelos PIC, Raspberry y Arduino cada uno de ellos con diferentes ventajas y desventajas, las cuales fueron evaluadas y como resultado se obtuvo la decisión de adquirir el microcontrolador con mayores beneficios para la aplicación de la solución.

Dispositivo	Ventajas	Desventajas
Pic	1. No es un dispositivo costoso 2. Hay mucha variedad de pics, constan de diferentes tamaños formas y capacidades que se adaptan al ambiente de desarrollo que se necesita. 3. Se tiene 100% del control del dispositivo no depende de funciones de terceros. 4. Se puede crear un entorno de desarrollo personalizado.	1. Necesita integrarse con otros dispositivos para poder funcionar. 2. Se necesita un programador ya que no posee una placa de desarrollo integrado. 3. No ejecutan muchas (millones) de instrucciones por segundo. 4. Solo se puede programar en leguaje C o ensamblador.
Arduino	1. Las placas Arduino son de código abierto pueden ser	1. No es flexible, el proyecto debe adaptarse al tamaño de la

	modificado sin violar la ley. 2. Contiene su propio IDE lo que facilita en gran parte su programación. 3. Contiene funciones ya cargadas previamente. 4. No es un dispositivo costoso. 5. Es multiplataforma, puede correr en los sistemas operativos, Linux, Windows y Mac. 6. Existe mucha documentación en internet por ser software libre.	placa.
Raspberry Pi	1. No es un dispositivo costoso. 2. Puede procesar varias tareas al mismo tiempo ya que puede ejecutar programas en segundo plano. 3. Se puede conectar de forma alámbrica e inalámbrica a internet. 4. Se puede programar bajo cualquier IDE que sea de distribución con soporte en LINUX.	1. No es flexible, por ejemplo, para la lectura de los sensores analógicos se requiere un software adicional. 2. Para escribir código en la Raspberry pi se necesita instalar el SO y utilizar librerías de código. 3. Su manipulación es mucho más compleja, por ejemplo, si no se hace un apagado adecuado y se desconecta se puede dañar la placa. 4. El voltaje máximo en los pins I/O son de 5-10 mA

Tabla 4 Selección del microcontrolador [Fuente propia].

Durante el proceso de la toma de decisiones, el elegido fue el Arduino por las numerosas ventajas que tiene a pesar de la desventaja que el proyecto debe adaptarse a la placa, el Arduino se integra con el lenguaje de programación elegido que fue Java a través de su librería RXTX a diferencia del PIC que solamente admite lenguaje ensamblador o lenguaje C, necesita un desarrollador ya que no posee una placa de desarrollo integrado, su costo en el mercado no es elevado a diferencia del Raspberry Pi que su costo es bastante elevado ya que esta placa contiene hasta sistema operativo propio, lo que hace que su manipulación sea mucho más compleja porque es necesario instalar el SO y utilizar sus librerías para que funcione, otra ventaja que influyó en la elección del Arduino fue la gran documentación que se encuentra en internet.

5. Crear un dispositivo traductor que permita la transformación de las señales digitales a código Braille.

Para el desarrollo de dicho módulo se requirió integrar procesos de interpretación y emisión de señales digitales en la placa Arduino para convertir las señales digitales entrantes en salidas representativa mediante sus pines, integrando dicho módulo con el dispositivo de salida para la representación de texto alfanuméricos en el sistema alfabético Braille; Para ello arduino cuenta con su propio lenguaje de programación y estructuras que como en su página web [34] es definida, se puede dividir en tres segmentos principales como variables, funciones y estructura de datos; Dos estructuras principales para la ejecución y comunicación con el Arduino que son setup y loop:

- Setup: es el método inicial que da valor a las configuraciones de arranque para el inicio del programa, similar al constructor de la clase en la programación orientada a objetos.

- loop: este método es llamado luego de realizar la ejecución completa de setup, su función es realizar ciclos repetitivos para realizar

funciones reiterativas que pueden estar a la escucha de alguna señal o bien sea realizando acciones de manera cíclica.

Para el desarrollo del módulo se requirió realizar la conexión con el Arduino, por consiguiente se ejecutaron pruebas con la consola que arduino dispone para emular la entrada de datos llamada "serial monitor". Una vez realizadas las pruebas y evaluando el funcionamiento del arduino se realizó la integración con el módulo para la representación de texto alfanuméricos en el sistema alfabético Braille, validando que el comportamiento y funcionamiento de la salidas emitidas desde la consola fuera el esperado.

6. Diseñar e implementar un módulo de reconocimiento de frontera de contenido.

En el desarrollo de dicho módulo se realizaron pruebas unitarias que cumplirian con la detección del límite de frontera ubicando el apuntador en el último carácter y solicitando el siguiente carácter, así mismo colocando el apuntador en el primer carácter y solicitando el carácter anterior. Del mismo modo se realizaron pruebas realizando pases y regresos de palabras y párrafos.

Para la implementación de este módulo, se hizo uso del patrón de diseño composite anteriormente desarrollado, debido a que el texto cargado se encuentra en una arquitectura jerárquica que contiene un inicio y un final, y gracias a ello es posible saber que en un punto determinado no se cuenta con más caracteres iterativos, por ello se denota que se está en un límite de frontera que para el desarrollo de la solución solo se hará notar los inicios y finalizaciones de texto ya que son de información relevante para el lector.

7. Diseñar e implementar un módulo de notificación.

Para el diseño de las notificaciones se realizaron bocetos estructurales de posibles mensajes emitidos por el software usando la herramienta balsamiq mockups, con la finalidad de detectar la información relevante que debía ser emitida al lector.

Para realizar las notificaciones fue necesario utilizar mecanismos alternos que permitiera generar un mensaje de forma directa y precisa que las personas con discapacidad visual pudieran interpretar, es por ello que se recurrió al uso del sentido auditivo, por el cual se recibirá la información mediante la reproducción de archivos de audio que realizarán las notificaciones de interés para el usuario.

El audio fue generado en un portal web [35] que permite la conversión de texto a audio, y debido a las pruebas realizadas que permitieron la identificación de la información que se iba a emitir, se realizó la conversión del texto a un archivo con formato MP3.

En dicha página se generaron las pruebas de los sonidos mediante la reproducción de los mismos para dar la bienvenida a la aplicación, notificar inicio o finalización del texto e inicio o finalización de la lectura automática.

Capítulo V

Resultados

1. Diseñar e implementar un entorno de contenido para la carga y manejo de archivos de texto plano

Se obtuvo un cuadro diálogo, en el cual se puede cargar el contenido de un documento de Word, texto plano o PDF vía "drag and drop" (arrastrándolo dentro del cuadro diálogo), este entorno extrae el contenido del archivo cargado, y lo expone en el cuadro diálogo para el comienzo de la lectura generando un sonido que indica la bienvenida al programa y que el documento se ha cargado con éxito.

La interfaz cuenta con un rectángulo central que permite la visualización del contenido para los usuarios asistidos, en la barra superior contiene un menú de tres opciones que engloban las acciones realizadas bajo la aplicación.

- Archivo: Cuenta con opciones relacionadas con el manejo de archivo como lo es:

  - Carga: Esta opción permite realizar la carga de un documento a través de un menú popup para la selección del archivo, la ruta de atajo para esta opción es "CTR + D" o que también puede ser cargado arrastrando el archivo (DragAndDrop) hasta el rectángulo de contenido como se muestra en la figura 17.

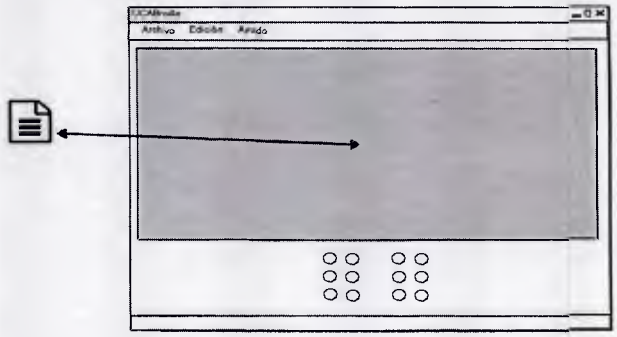


Figura 17 Carga de documento (DragAndDrop) [Fuente propia].

- Configuración: En la configuración podremos asignar los valores relacionados con la comunicación y lectura de automática del texto como la detección de la conexión por terminal serial del Arduino o la velocidad en segundos para el paso de caracteres de lectura automática.

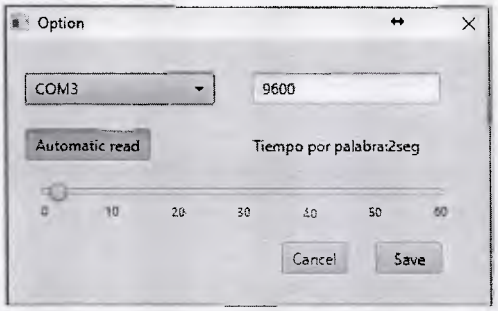


Figura 18 Configuración UCABraille [Fuente propia].

- Edición: En edición está contenido las opciones que permiten realizar alguna acción sobre el texto cargado en la aplicación como

- Anterior: Esta acción permite posicionarse en un carácter anterior al actual bien sea una letra, palabra, o párrafo los cuales cada uno cuenta con un atajo de teclado "CTR + SHIFT + L", "CTR + SHIFT + T", "CTR + SHIFT + P" respectivamente.

- Siguiente: Esta acción permite posicionarse en un carácter posterior al actual bien sea una letra, palabra, o párrafo los cuales cada uno cuenta con un atajo de teclado "CTR + L", "CTR + T", "CTR + P" respectivamente.

- Inicio/Fin de lectura automática: Esta acción nos proporciona el inicio o finalización de la lectura del texto de manera automática la cual puede ser accedida a través del menú o con su atajo de teclado "CTR + P"

- Ayuda: La opción de ayuda engloba acciones que podrían ayudar al usuario con dudas frecuentes ante alguna eventualidad bajo las sub-opciones siguientes:
  - Información: Para los usuarios que requieran una breve explicación de la aplicación esta opción será suficiente de tal manera que se logre entender la funcionalidad de la aplicación y a quien está dirigida.

  - Contacto: Para diversas dudas se generó la opción de contacto que detalla los nombres de los integrantes del desarrollo, así como también el código del software alojado en GitHub.

2. Diseñar e implementar un módulo de reconocimiento de posicionamiento y transición inteligente del cursor entre caracteres alfanuméricos presentados en torno a la lectura

Este módulo muestra donde está posicionado el cursor dentro del cuadro diálogo realizado con Balsamiq Mockups de esta forma si hay una persona asistiendo a la persona con discapacidad visual sabe el lugar exacto de la lectura y evitando que se pierda en la lectura, a medida que va avanzando la lectura se representa el carácter en Braille en la parte inferior del software.

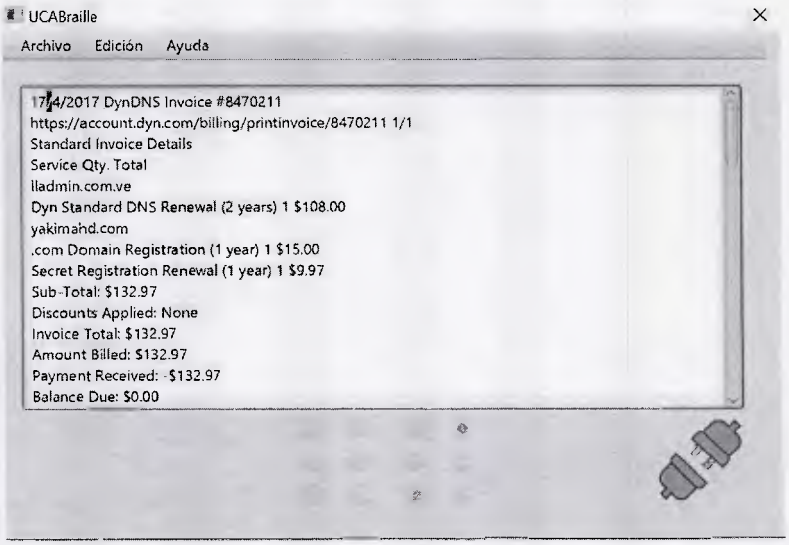


Figura 19 Posición Inteligente del cursor. [Fuente Propia].

Con los pulsadores conectados a la baquelita, la persona con discapacidad visual puede avanzar o retroceder en la lectura tantas veces como lo desee, con la intención de que la lectura sea lo más fluida posible y evitar un poco la dependencia de un asistente.

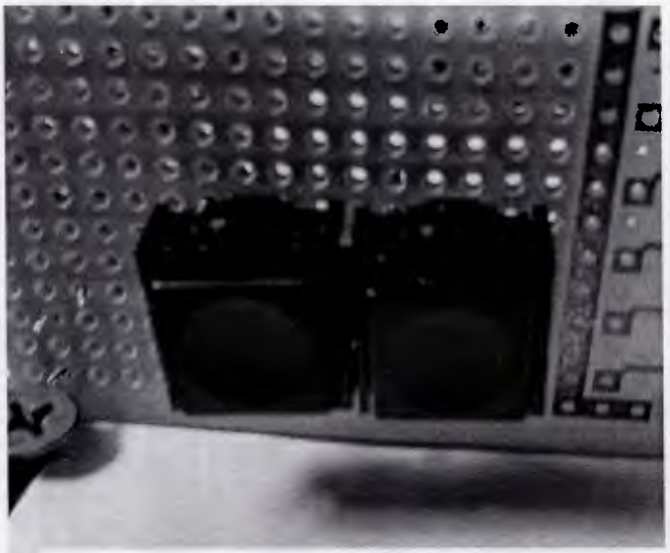


Figura 20 Pulsadores [Fuente Propia].

3. Construir un dispositivo de salida que permita representar textos alfanuméricos en el sistema alfabético Braille.

Con este dispositivo la salida representada por solenoides, permite a la persona con discapacidad visual realizar la lectura del documento, con las parametrizaciones correspondientes como lo son velocidad lectura si la misma es colocada de forma automática, mediante los pulsadores para avanzar o retroceder en la lectura bien sea el caso, o mediante la combinación de teclas del computador para avanzar o retroceder en la lectura, esta última debe ser asistida por una persona que no presenta discapacidad visual.

4. Investigar y analizar los distintos tipos de microcontroladores a utilizar en el dispositivo con salida al sistema alfabético Braille para el transporte y conexión de la información contenida en el computador.

El microcontrolador utilizado como salida al sistema alfabético Braille fue el Arduino, ya que este representó muchas ventajas significativas, como son su precio accesible en el mercado con respecto a otras placas microcontroladoras por ejemplo el Raspberry Pi (es mucho más costoso), posee la cantidad suficiente de puertos (i/o) para realizar las múltiples conexiones con las resistencias, se programó en un lenguaje de alto nivel (JAVA) junto con librerías (PANAMAHITEK_ARDUINO), además de la alta disponibilidad de documentación sobre Arduino en internet, todo esto junto permite la conversión de la información del software del computador hacia el microcontrolador y este a su vez genera las salidas correspondientes levantando el solenoide correspondiente para la representación del carácter en Braille.

5. Crear un dispositivo traductor que permita la transformación de las señales digitales a código Braille.

Como cada carácter tiene su representación en código Braille, se crea una matriz de 2x6, junto con la conexión del Arduino gracias a la librería PANAMAHITEK_ARDUINO y los patrones de diseño composite (para el desglose de los textos en párrafos, los párrafos en palabras y las palabras en letras) y singletone (para la garantizar que solo exista una conexión con el Arduino desde el computador), da como resultado el dispositivo traductor de señal digital a código Braille.

6. Diseñar e implementar un módulo de reconocimiento de frontera de contenido.

Con la implementación de este módulo de frontera el software reconoce la posición del cursor en el contenido, el mismo válida si se encuentra al inicio o final del documento, para que se activen las notificaciones correspondientes en este caso un cuadro diálogo, con esto la persona que asiste a la persona con discapacidad visual sabe si se encuentra al principio o al final del documento a leer.

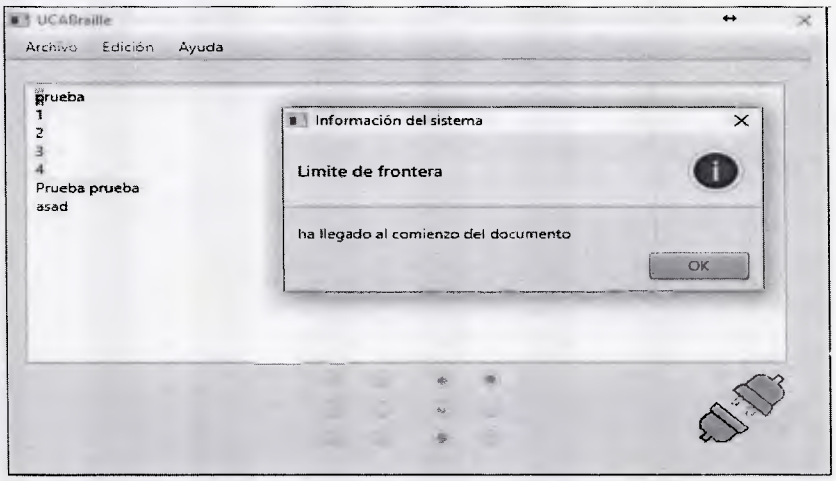


Figura 21 Límite de frontera inicial. [Fuente propia].

7. Diseñar e implementar un módulo de notificación.

En la implementación del módulo de notificación se genera un audio (archivo .mp3), estos sonidos fueron tomados del portal web [35] que permite la conversión de texto a audio, en esa misma página se generó lo sonidos para dar la bienvenida a la aplicación, notificar inicio o finalización del texto e inicio o finalización de la lectura automática.

Software de lectura y conversión Braille

Bajo el cumplimiento de los objetivos anteriormente descritos se obtuvo como resultado general un software de carga y conversión de documentos de texto plano a su correspondiente representación al alfabeto Braille.

Dispositivo de lectura Braille

Un dispositivo que permite la representación de caracteres Braille en relieve y el control de la lectura digital a través de dos botones o mediante acciones emitidas por el software de lectura Braille para el desplazamiento de las palabras, obteniendo una reducción del costo de producción de $ 65.92 o a su equivalente a una tasa DICOM al 18 de Octubre del 2017 de Bs 11.311,00 por dólar, generando un total de Bs 752.238,37.

Tabla 5 Materiales en divisa externa. [Fuente propia].

Artículo	Cantidad	Costo Unitario	Moneda	Total
Arduino Due	1	$ 32.11	Dolar	$ 32.11
Solenoides	12	$ 2.38	Dolar	$ 28.56
Baquelita	1	$ 2.65	Dolar	$ 2.65
Transistores	12	$ 0.18	Dolar	$ 2.12
Resistencias	14	$ 0.04	Dolar	$ 0.48
Total	**-**	**-**	**Dolar**	**$ 65.92**

Tabla 6 Materiales en divisa interna. [Fuente propia].

Artículo	Cantidad	Costo	Moneda	Total
Arduino Due	1	Bs 363.196,21	Bolívares	Bs 363.196,21
Solenoides	12	Bs 26.920,18	Bolívares	Bs 323.042,16
Baquelita	1	Bs 30.0000	Bolívares	Bs 30.000
Transistores	12	Bs 2.000	Bolívares	Bs 24.000
Resistencias	14	Bs 500	Bolívares	Bs 12.000
Total	**-**	**-**	**Bolívares**	**Bs 752.238,37**

Manuales de uso

Con la finalidad de brindar un óptimo uso del dispositivo de lectura, se realizó un manual con las instrucciones de cómo se tiene que utilizar indicando el tipo de cable a utilizar para la conexión del dispositivo con el computador, como es la funcionalidad de los botones a la hora de inicio, pausa y detención de la lectura, que programa tiene que ejecutar la persona ayudante a la hora de la carga del archivo y cómo debe realizar la carga del mismo, así mismo el software tiene integrado una referencia de su uso adecuado en caso tal de que el manual se extravíe.

Manuales de programación

En el manual de programación se describe los estándares implementados para el desarrollo, así como también todo el manejo técnico necesario para su entendimiento y que pueda servir para futuras mejoras.

Diagramas de clases

Para el desarrollo de UCABraille se diseñó el diagrama de clases (UML) el cual describe la estructura del sistema, clases, métodos y funciones con sus respectivos atributos, además de la relación entre los objetos.

Capítulo VI

Conclusiones y recomendaciones

Conclusiones

Como conclusión de este trabajo de grado, se puede afirmar que es posible la construcción y desarrollo de un dispositivo de lectura con salida al sistema Braille para personas con discapacidad visual, a un costo total de $ 65.92 lo que equivale a una reducción del 96% aproximado al costo del dispositivo de lectura Braille más económico actualmente en el mercado con un valor aproximado de $1.995.

Aunque el dispositivo de lectura va orientado a personas con discapacidad visual, es necesario la asistencia de una persona sin discapacidad visual para realizar la carga del documento; A partir de este punto la persona que presenta discapacidad visual puede desenvolverse de manera independiente en la lectura.

Durante el desarrollo del circuito se percibió un alto consumo de corriente, lo cual fue solventado mediante una fuente externa que proporciona la cantidad necesaria de amperaje para el funcionamiento correcto del circuito, los transformadores normalmente tienen una salida de 12 voltios, 2 Amp por lo que se tuvo que recurrir a una fuente no convencional con mayor corriente de salida ya que el circuito requiere un consumo máximo de 24 Amp debido a que contiene 12 solenoides en paralelo con un consumo unitario de 12 v 1.75 amp.

El software facilita la carga y lectura de documentos de texto plano como PDF (Adobe Reader) , Txt (Documento de texto) y Doc (Archivo de Word) mediante la transcripción de caracteres digitales a una salida de representación Braille a través de una conexión serial con el dispositivo de representación Braille desarrollado.

Recomendaciones

1. Para darle un buen uso al dispositivo se recomienda usar una diferencia de potencial de 12V y con una corriente de +19Amp, ya que el circuito consta de 12 solenoides en paralelo y esto representa un consumo individual de 12 V, 1.75 Amp dando un total de 18 Amp, ya que el máximo de puntos Brailles representados es de 9.
2. Para futuras mejoras se recomienda emplear solenoides de menor consumo o sustituirlos por polímeros electroactivos ya que esto representa un beneficio en la reducción del consumo de amperaje lo que garantiza que exista menos recalentamiento en los componentes del circuito. Espacio entre cada punto Braille debido a que los solenoides de menor consumo son más pequeños.
3. Para una mejor experiencia de lectura es preciso acoplar los puntos en relieve al estándar de separación entre puntos Braille, lo cual puede ser realizado con solenoides de menor consumo debido a que éstos son más pequeños.
4. Las notificaciones pueden estar acompañadas con un motor para generar vibraciones de diferentes niveles para cada información, de esta forma el usuario pueda centrar su atención de forma completa en el tacto sin requerir de algún otro sentido, lo que permitiría la integración de personas con discapacidad visual y auditiva.
5. El trabajo de grado puede ser empleado como mecanismo didáctico para la enseñanza y aprendizaje de la representación de caracteres alfanuméricos en el sistema alfabético Braille.

Referencias Bibliográficas

[1] Organización Mundial de la salud. En el mundo hay unos 45 millones de ciegos, y la cifra va en aumento [En línea]. (9 octubre 2003). [Último acceso: 20 de diciembre del 2015]. Obtenido de: http://www.who.int/mediacentre/news/releases/2003/pr73/es/

[2] Universidad de alicante. Teclados Braille [En línea]. (s.f.). [Último acceso: 11 de enero del 2016]. Obtenido de: http://accesibilidadweb.dlsi.ua.es/?menu=tec-Braille1

[3] Universidad de alicante. Líneas Braille [En línea]. [Último acceso: 11 de enero del 2016]. Obtenido de: http://accesibilidadweb.dlsi.ua.es/?menu=lin-Braille

[4] Acens Technologies. Línea Braille: los invidentes ya pueden leer con sus manos en Internet [En línea]. (22 Julio 2012). [Último acceso: 18 de enero del 2016].Obtenido en: http://www.acens.com/blog/linea-Braille-los-invidentes-ya-pueden-leer-con-sus-manos-en-internet.html

[5] Tecno Accesible. Línea Braille [En línea]. (s.f.). [Último acceso: 11 de enero del 2016]. Obtenido de: http://www.tecnoaccesible.net/category/tecnolog%C3%ADa/perif%C3%A9rico-de-salida/l%C3%ADnea-Braille

[6] Python. Pyserial [En línea]. (s.f.). [Último acceso: 17 de enero del 2016]. Obtenido de: https://pypi.python.org/pypi/pyserial

[7] Python. OS Miscellaneous operating system interfaces [En línea]. (08 enero 2016). [Último acceso: 17 de enero del 2016]. Obtenido de: https://docs.python.org/2/library/os.html

[8] Currey,M. Solenoid Breackout board V2 [En línea]. (03 de noviembre del 2013) [Último acceso: 01 de mayo del 2016]. Obtenido en: http://www.martyncurrey.com/solenoid-breakout-board-v2/

[9] Berger. M, Nanotechnology Braille display could make Web accessible to the blind [En línea]. (16 de marzo del 2012). [Último acceso: 18 de enero del 2016]. Obtenido en: http://www.nanowerk.com/spotlight/spotid=24618.php

[10] HumanWare. Brailliant BI 32 (NEW generation). [En línea]. (s.f.) [Último acceso: 07 de junio del 2016]. Obtenido en: http://store.humanware.com/hus/brailliant-bi-32-new-generation.html

[11] Amazon. Braille Sense U2-Querty Keyboard Notetaker. [En línea]. (s.f.). [Último acceso: 07 de junio del 2016]. Obtenido en: http://www.amazon.com/Braille-Sense-U2-Keyboard-Notetaker/dp/B00D6K9CXC/ref=sr_1_29_a_it?ie=UTF8&qid=1465065306&sr=8-29&keywords=humanware

[12] HumanWare. BrailleNote Apex BT 32 Braille Notetaker. [En línea]. (s.f.). [Último acceso: 07 de Junio del 2016].Obtenido en: http://store.humanware.com/hus/Braillenote-apex-bt-32-Braille-notetaker.html

[13] Amazon. SyncBraille Portable 20 Cell Braille Display. [En línea]. (s.f.). [Último acceso: 07 de junio del 2016]. Obtenido en: http://www.amazon.com/SyncBraille-Portable-20-Cell-Braille-Display/dp/B003TM1AEQ

[14] Magnifying Aids. SuperVario2- 32 Portable Braille Display [En línea]. (20 de Febrero del 2014) [Último acceso: 07 Junio del 2016].Obtenido en: http://www.magnifyingaids.com/SuperVario2-32-Portable-Braille-Display

[15] Magnifying Aids. Supervario2- 40 Portable Braille Display [En línea]. (20 de febrero del 2016) [Último acceso: 07 de junio del 2016]. Obtenido en: http://www.magnifyingaids.com/SuperVario2-40-Portable-Braille-Display.

[16] Amazon. Push Pull Open Frame Solenoid Electromagnet [En línea] (10 de junio del 2016) [Último acceso: 07 de junio del 2016]. Obtenido en: https://www.amazon.com/gp/product/B00HR8H38A/ref=ox_sc_act_title_6?ie=UTF8&psc=1&smid=AO2R8RP2MLB3M

[17] Amazon. DIY DC Electromagnet Magnet Solenoid [En línea]. (10 de junio del 2016) [Último acceso: 07 de junio del 2016]. Obtenido en: https://www.amazon.com/gp/product/B018HX0R2W/ref=ox_sc_act_title_5?ie=UTF8&psc=1&smid=A1THAZDOWP300U

[18] Amazon. Arduino Leonardo [En línea] (23 de abril del 2015) [Último acceso: 07 de junio del 2016]. Obtenido en: https://www.amazon.com/Arduino-A000057-Leonardo-with-

Headers/dp/B008A36R2Y/ref=sr_1_2?ie=UTF8&qid=1465567086&sr=8-2&keywords=arduino+leonardo
[19] Amazon. Arduino UNO R3 [En línea] (10 de febrero del 2015) [Último acceso: 07 de junio del 2016]. Obtenido en: https://www.amazon.com/Arduino-UNO-board-ATmega328P-A000066/dp/B008GRTSV6/ref=sr_1_3?ie=UTF8&qid=1465567428&sr=8-3&keywords=arduino+uno
[20] Amazon. Raspberry Pi 3 [En línea] (10 de junio del 2016) [Último acceso: 07 de junio del 2016]. Obtenido en: https://www.amazon.com/gp/product/B01C6Q4GLE/ref=ox_sc_act_title_1?ie=UTF8&psc=1&smid=A30ZYR2W3VAJ0A
[21] Amazon. Raspberry Pi 2 [En línea] (17 de agosto del 2017) [Último acceso: 17 de agosto del 2017]. Obtenido en: https://www.amazon.com/CanaKit-Raspberry-Complete-Starter-9-Items/dp/B008XVAVAW/ref=sr_1_18?ie=UTF8&qid=1508269281&sr=8-18&keywords=raspberry+pi+2
[22] Amazon. PIC16F877A-I/P DIP-40 Microchip Microcontroller [En línea] (03 de enero del 2016) [Último acceso: 07 de junio del 2016]. Obtenido en: https://www.amazon.com/gp/product/B011TPFZMM/ref=ox_sc_act_title_3?ie=UTF8&psc=1&smid=AEG73ZKU0CZ26
[23] Amazon. AT32UC3A3256-ALUR IC MCU 256KB FLASH 144LQFP 32UC3 AT32UC3 [En línea] (25 de noviembre del 2014) [Último acceso: 07 de junio del 2016]. Obtenido en: https://www.amazon.com/gp/product/B015VUJDC4/ref=ox_sc_act_title_2?ie=UTF8&psc=1&smid=A3DSOCXH7OD2TG
[24] Elissalde E. Alfabeto Braille [En línea]. (s.f.). [Último acceso: 30 de enero del 2016]. Obtenido en: http://www.fbu.edu.uy/informacion/alfabeto/alfabeto2.htm
[25] Wikipedia. Arduino Uno [En línea]. (s.f.). [Último acceso: 17 de agosto del 2017]. Obtenido de: https://es.wikipedia.org/wiki/Arduino

[26] Connexion Developments Ltd. How does a selenoid valve work. [En línea]. (s.f.). [Último acceso: 01 de mayo del 2016]. Obtenido en: https://www.solenoid-valves.com/theory/what-is-a-solenoid-valve.php
[27] Transistor TIP122 [En línea]. (s.f.) [Último acceso: 17 de agosto del 2017]. Obtenido en: http://www.electronicoscaldas.com/379-thickbox_default/transistor-tip122.jpg
[28] Sabelotodo. Código de colores de los resistores. [En línea]. (s.f.). [Último acceso: 17 de agosto del 2017]. Obtenido en: http://www.sabelotodo.org/electrotecnia/imagenes/codigocolores.jpg
[29] Alvarez R. El "Código espagueti" y los patrones avanzados de programación. [En línea] (s.f.) [Último acceso: 24 de septiembre del 2017]. Obtenido en: http://tecbolivia.com/index.php/articulos-y-tutoriales-microcontroladores/12-el-qcodigo-espaguetiq-y-los-patrones-avanzados-de-programacion
[30] Alvarez M. ¿Que es M.V.C.? [En línea] (2 de enero del 2014) [Último acceso: 12 de octubre del 2017]. Obtenido en: https://desarrolloweb.com/articulos/que-es-mvc.html
[31] Balsamiq. Balsamiq Mockups. [En línea]. (s.f.) [Último acceso: 02 de noviembre del 2016). Obtenido en: https://balsamiq.com/products/mockups/
[32] Gomez V. Patrones de diseño, patrón composite. [En línea]. (22 de enero del 2017). [Último acceso: 06 de octubre del 2017]. Obtenido en: https://instintobinario.com/patrones-de-diseno-patron-composite/

[33] Circuit Marker [En línea] (s.f.)[Último acceso: 02 de noviembre del 2016] Obtenido en: https://circuitmaker.com/
[34] Arduino. Language reference. [En línea]. (s.f.). [Último acceso: 10 de enero del 2017]. Obtenido en: https://www.arduino.cc/en/Reference/HomePage
[35] Text 2 MP3. Text To MP3 [En línea] (s.f.). [Último acceso: 10 de octubre del 2017]. Obtenido en: http://www.texttomp3.online/
[36] Git. Getting Started – Git Basics. [En línea] (s.f.) [Último acceso: 19 de octubre del 2017]. Obtenido en: https://git-scm.com/book/en/v2/Getting-Started-Git-Basics

[37] Brown K. ¿What is Github, and What is it used for? [En línea](21 de septiembre del 2016) [Último acceso: 19 de octubre del 2017]. Obtenido en: https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/

Anexos

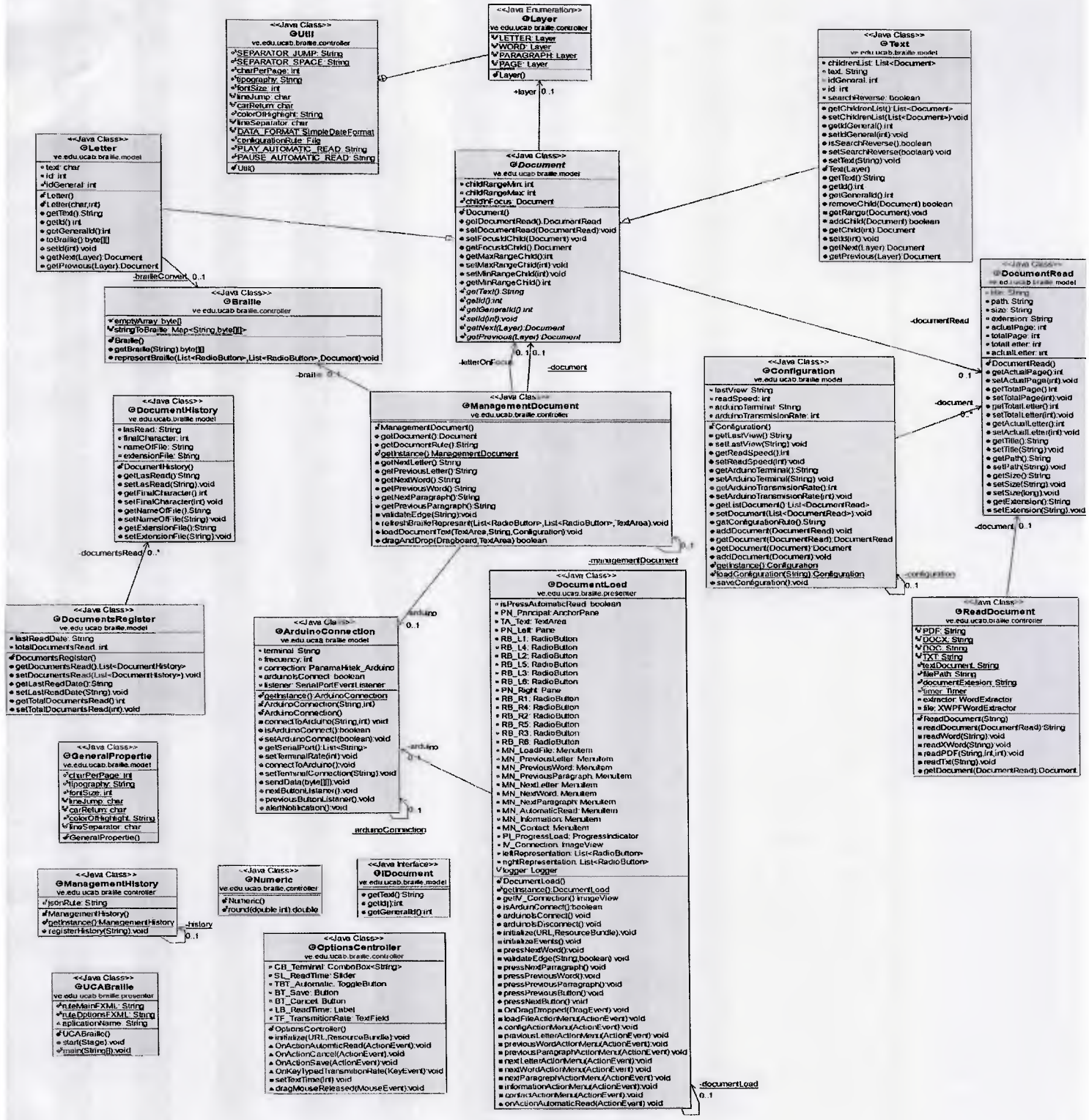


Diagrama de clase. [Fuente propia].