

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**



**NIVEL LICENCIATURA
EXAMEN DE GRADO**

TRABAJO DE APLICACIÓN

**“DISEÑO DE PROTOTIPO DE UN BRAZO ROBÓTICO CONTROLADO POR
VOZ EN JAVA”**

Postulante: Yecid Alvarez Peralta

La Paz – Bolivia

2016

DEDICATORIA.

Para mi familia por su apoyo, a mi madre Viqui Peralta Aquise, mi padre Santiago Alvarez Vallejos y a mi novia Marlene y a mis hermanos por sus consejos, comprensión, amor, ayuda en los momentos difíciles, y nunca dejarme bajar los brazos brindándome siempre su apoyo incondicional. Me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos y hacer de este sueño una hermosa realidad.

AGRADECIMIENTO.

Expreso nuestro más grande y sincero agradecimiento en primer lugar a Dios por habernos guiado durante nuestra preparación profesional alcanzando el objetivo que nos planteamos al inicio de nuestros estudios, a todos los Licenciados e Ingenieros de la carrera. Por habernos trasmitido sus conocimientos y asesoramiento para la elaboración del presente trabajo.

RESUMEN

El problema observado es el peligro de manipular objetos, materiales o sustancias, nocivas que afectan al ser humano, la solución tecnológica plateada es la construcción de un brazo robótico controlado por voz en java de bajo costo en comparación de otros que existen en el mercado reduciendo notablemente los gastos económicos, el cual ayudara a la manipulación de componentes químicos nocivos para la salud del ser humano, la metodología a emplear es la investigación tecnológica el cual me ayuda para cumplir el objetivo general y específicos los cuales son, diseñar e implementar un brazo robótico controlado por voz en java para manipular sustancias peligrosas, Examinar algoritmos de procesamiento digital de voz que permitan un tratamiento sencillo de información de señales de voz, Diseñar la estructura del brazo robótico, Implementar un circuito para la manipulación del brazo robótico y por ultimo Obtener la interacción automática entre humano y computadora por medio de un sistema digital, se tiene un tiempo de ejecución del proyecto de tres meses mínimos requerido y el presupuesto a implementar tiene un monto de 1500 Bs el cual involucra materiales como herramientas, componentes electrónicos, arduino, recursos humanos y otros, para la construcción de brazo robótico.

TABLA DE CONTENIDO

DEDICATORIA	I
AGRADECIMIENTO	II
RESUMEN	III
TABLA DE CONTENIDO	IV
ÍNDICE DE FIGURAS	VI
INTRODUCCIÓN	VII
CAPÍTULO I	1
1.1 PLANTEAMIENTO DE PROBLEMA	1
1.2 OBJETIVOS	2
1.2.1 OBJETIVO GENERAL	2
1.2.2 OBJETIVOS ESPECÍFICOS	2
1.3 JUSTIFICACIÓN	2
1.4 DELIMITACIÓN	3
1.4.1 TEMPORAL	3
1.4.2 ESPACIAL	3
1.4.3 TEMÁTICA	3
1.5 METODOLOGÍA	4
CAPÍTULO II	5
2 MARCO TEÓRICO	5
2.1 CONCEPTOS GENERALES	5

2.1.1	ACTUADOR	5
2.1.2	ALGORITMO	5
2.1.3	RECONOCIMIENTO Y SÍNTESIS DE LA VOZ	5
2.1.4	RECONOCIMIENTO	5
2.1.5	SÍNTESIS.....	6
2.1.6	RECONOCIMIENTO DE VOZ	7
2.1.7	NETBEANS	9
2.1.8	CONCEPTO GENERAL ARDUINO	10
2.1.9	ARTICULACIÓN	11
2.1.10	BRAZO ROBÓTICO.....	11
2.1.11	GRADOS DE LIBERTAD.....	12
2.1.12	MODULACIÓN POR ANCHO DE PULSOS	12
2.1.13	CONCEPTO GENERAL SERVO MOTORES	13
2.2	ELEMENTOS DEL SISTEMA	14
2.2.1	ARDUINO UNO.....	14
2.2.2	SERVOMOTOR.....	15
2.2.3	ALUMINIO	16
2.2.4	TALKING JAVA SDK	17
	CAPÍTULO III.....	19
	3 INGENIERIA DE PROYECTO	19
3.1	DESCRIPCIÓN DEL PROYECTO.....	19
3.2	DIAGRAMA ESQUEMÁTICO DEL CIRCUITO CON ARDUINO UNO.....	20
3.3	DIAGRAMA EN BLOQUES DEL SISTEMA	21
3.4	SISTEMA DE RECONOCIMIENTO DE VOZ	23
3.4.1	CÓDIGO DE NETBEANS DEL RECONOCIMIENTO DE VOZ	23
3.5	ESTABLECIMIENTO DE COMUNICACIÓN SERIAL	26
3.5.1	CÓDIGO DE NETBEANS DE COMUNICACIÓN SERIAL	27
3.6	ETAPA DE CONTROL DEL ARDUINO	29
3.6.1	CÓDIGO FUENTE ARDUINO	29
	4 CONCLUSIONES Y RECOMENDACIONES	31
	5 BIBLIOGRAFIA.....	33
	ANEXOS.....	34
	FOTOS DEL BRAZO ROBÓTICO	34
	CÓDIGO FUENTE ARDUINO.....	35

ÍNDICE DE FIGURAS

Figura 1. Sistema Reconocedor.....	7
Figura 2. Arduino Uno.....	10
Figura 3. Brazo robótico uarm.....	11
Figura 4: Conexión de Arduino y servomotores.....	20
Figura 5: Diagrama en bloques.....	21
Figura 6: Diagrama en bloque de elementos del sistema.....	22
Figura 7: Interfaz de usuario.....	28
Figura 8: Interfaz de Reconocimiento de voz.....	28

INTRODUCCIÓN

El presente proyecto consiste en el diseño, construcción y de un brazo robótico dedicado a la manipulación de objetos. Se abarcan temas relacionados con el diseño mecánico, electrónico y programación de controladores con el fin de interactuar las expresiones del hombre y la Tecnología a través del reconocimiento de voz, se implementa un prototipo de un sistema de control electrónico mediante comandos de voz para un robot secuencial, se abordan los procesos de diseño, programación, construcción y control de un brazo robótico de cinco grados de libertad y reconocimiento de comandos de voz, orientado a la detección, clasificación y traslado de objetos definidos dentro de un espacio de trabajo, al realizar la implementación del prototipo se utilizó la tarjeta de Arduino que está diseñado con entradas y salidas analógicas y digitales, entradas y salidas PWM, para controlar los servomotores del brazo robótico. En el primer capítulo se describe el problema de investigación, explicando las causas y consecuencias que lo originan, y por último los objetivos de la investigación. El segundo capítulo presenta los antecedentes y se realiza una introducción a todo el marco teórico necesario para el desarrollo del proyecto. Se trata de una forma general los conceptos relacionados con la robótica y la teoría referida a las herramientas y etapas de la visión artificial y el reconocimiento de comandos de voz.

CAPÍTULO I

1.1 PLANTEAMIENTO DE PROBLEMA

El problema planteado es el peligro de manipular objetos, materiales o sustancias, nocivas que afectan al ser humano a la hora de su manipular dichas sustancias, y se tiene la necesidad de trabajar con estas sustancias en diferentes ámbitos como ser para la elaboración fármacos, en el ámbito de salud, minería, hidrocarburos y etc.

La magnitud del problema puede desembocar en enfermedades ya sean terminales como las cancerígenas y hasta enfermedades más peligrosas.

Causas

- Existencia de sustancias peligrosas
- La mala manipulación de químicos
- Riesgos de sufrir accidentes imprevistos
- La creciente elaboración de fármacos
- La necesidad de utilizar estos compuestos en la industria.

Efectos

- Enfermedades
- Puede provocar discapacidad
- Trastornos

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

- Diseñar un brazo robótico controlado por voz en java para manipulación de sustancias y objetos peligrosos.

1.2.2 OBJETIVOS ESPECÍFICOS

- Implementar algoritmos de procesamiento digital de voz que permitan un tratamiento sencillo de información de señales de voz.
- Diseñar la estructura del brazo robótico.
- Implementar un circuito para la manipulación del brazo robótico.
- Obtener la interacción automática entre humano y computadora por medio de un sistema digital.

1.3 JUSTIFICACIÓN

En el área de Robótica se cuenta con algunas investigaciones relacionadas con el desarrollo de manipuladores que reproducen algunos movimientos de extremidades superiores humanas. Con este proyecto se busca incrementar dicho conocimiento en esta línea de investigación, con la implementación de reconocimiento de voz. Su beneficio es aportar diferentes alternativas de solución para problemas específicos que involucran la integración de las áreas de Electrónica, Mecánica y Computación desde un punto de vista mecatrónico. En este proyecto se abordan los problemas que surgen en el estudio de los movimientos del brazo humano.

1.4 DELIMITACIÓN

Alcance: Diseñar un sistema emulador de algunos de los movimientos de las partes humanas con reconocimiento de voz: mano, brazo y antebrazo. El brazo robótico debe de contar con 5 GDL, así como poder manipular una carga de 0.5kg. en su efector final, así como obtener la cinemática involucrada en el sistema.

Limitaciones: La construcción del sistema diseñado estuvo condicionada a los recursos materiales con los que se contaban. Otra limitación es que el brazo robótico tiene cinco grados de libertad, de los cuales sólo tres de ellos están en acoplamiento directo y los restantes son a base de bandas y poleas dentadas, esto con el objetivo de que con la transmisión mencionada se pueda obtener un par mayor.

1.4.1 TEMPORAL

El prototipo del proyecto se realizó en un periodo de 3 meses tiempo en el cual dura la elaboración, diseño, programación y construcción del brazo robótico controlado por voz.

1.4.2 ESPACIAL

El presente proyecto se llevara a cabo en Instituciones y empresas, en el área de laboratorio e investigación.

1.4.3 TEMÁTICA

- Área académica: física y electrónica
- Línea de investigación: Programación y sistemas electrónicos

- Sub línea de investigación: Robótica

1.5 METODOLOGÍA

El tipo de investigación que se desarrolló en el proyecto fue la investigación Aplicada, pues busca la aplicación y utilización de conocimientos adquiridos para la solución de problemas, dirigidas a conseguir mejoras de procesos o productos en el ámbito industrial o sector empresarial incrementando la productividad, también se realizó la Investigación Experimental porque se realizaron pruebas para el desarrollo del proyecto, consiguiendo cumplir todos los objetivos planteados,



CAPÍTULO II

2 MARCO TEÓRICO.

2.1 Conceptos generales

2.1.1 Actuador

En todo sistema con capacidad de movimiento, el actuador es todo dispositivo que puede transformar la energía hidráulica, eólica, eléctrica, etc. en movimiento. (Mena, 2011). En este proyecto se utilizan motores de corriente continua.

2.1.2 Algoritmo

Es un conjunto de reglas que se encuentran en cierta secuencia, que sirven para cumplir con una función específica (Mena, 2011). En este caso la programación del sistema posee una serie de algoritmos que llevan a un funcionamiento completo del sistema.

2.1.3 Reconocimiento y síntesis de la voz

En el campo de la informática se han implementado diversas interfaces humano/computador; una de estas interfaces es la que combina el acceso y la transferencia de información a través de la voz.

2.1.4 Reconocimiento

En la informática el proceso de reconocimiento de voz se empieza a situar en el año 1870, cuando Alexander Graham Bell quiso desarrollar un dispositivo capaz de hacer visible el habla a personas con problemas auditivos, y producto de esto surgió el Teléfono.

En 1950, 80 años después, se realizó el primer intento por crear una máquina que reconociera la voz, fue desarrollada por los laboratorios AT&T Bell que

requería un extenso reajuste a la voz de una persona, pero una vez logrado tenía un 99% de certeza de reconocer los números del 1 al 9.

Entre los años 60"s y 70"s, los investigadores que trabajaban en la área, se dieron cuenta de lo complejo que era realizar un software en el campo del reconocimiento de voz, por lo tanto empezaron a realizar pequeños software con vocabularios dependientes del locutor y con pausas entre palabras y frases.

Durante los años 80"s y 90"s el reconocimiento de voz se favoreció por el incremento de computadoras personales, el apoyo de ARPA (Advance Research Projects Agency) y la reducción de costos en aplicaciones comerciales; el mayor interés durante la época fue la creación de grandes vocabularios. Un vocabulario de 100 palabras se consideraba grande, sin embargo en el año de 1985 existió un vocabulario con más de 20.000 palabras.

Los vocabularios extensos empezaron a volverse normales, las aplicaciones se centraron en ser independientes del locutor y se buscaba el flujo continuo, es decir que no hubiera pausas significativas entre palabras o frases.

2.1.5 Síntesis

Los primeros intentos para producir la voz humana por una máquina se realizaron en la segunda mitad del siglo XVIII, se había logrado la producción de vocales, por medio de tubos de resonancia conectados a un órgano, trabajo desarrollado por CH. G. Kratzenstein, profesor de filosofía de Copenhague.

Durante estos años se realizaron máquinas que trataban de simular la voz, pero no fue hasta el año de 1970, donde la síntesis de voz se relacionó estrechamente con la informática. Sin embargo, no fueron suficientes los circuitos integrados para simular la voz natural, se necesitaron elementos de software, por lo que los computadores permitieron explotar y

utilizar la síntesis de voz con propósitos prácticos, pero estos sonaban muy robóticos y eran a menudo inteligibles.

En el estado actual de la técnica de síntesis, los límites donde se podría diferenciar un sonido natural de uno artificial ha sido rebasado.

2.1.6 RECONOCIMIENTO DE VOZ

El reconocimiento de voz generalmente es utilizado como una interfaz humano/computador para algún software particular. Para lograr esto existen diferentes técnicas / algoritmos que reconocen la voz. Pero todas estas deben cumplir tres tareas

- **Pre procesamiento:** Convierte la entrada de voz a una forma en la cual el motor de reconocimiento la pueda procesar.
- **Reconocimiento:** Identifica lo que se ha dicho, generalmente traduciendo la señal a texto.
- **Comunicación:** Envía la señal traducida a texto, al sistema que lo requiera.

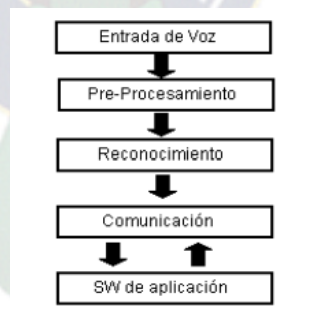


Figura 1: Sistema Reconocedor
Fuente de <http://monografias.com>

Los sistemas de reconocimiento de voz se enfocan en palabras y sonidos que distinguen una palabra de otra, éstos son los fonemas, por ejemplo, “tapa”, “capa”, “mapa”, “napa”, son palabras distintas puesto que su sonido inicial se reconocen como fonemas diferentes.

Existen diversas maneras para analizar y describir la voz. Los enfoques más comúnmente usados son:

- **Articulación:** Análisis de cómo el humano produce los sonidos de la voz.
- **Acústica:** Análisis de la señal de voz como una secuencia de sonidos.
- **Percepción auditiva:** Análisis de cómo el humano procesa la voz.

A demás de esto, generalmente los reconocedores de voz siguen una serie de pasos que se listan continuación:

- **Diseño de la gramática:** En la gramática se definen las palabras que se pueden emitir por parte del usuario y los patrones con los que se deben pronunciar. Una gramática debe ser creada y activada por el reconocedor para saber lo que se debe identificar en el audio entrante.
- **Procesamiento de señales:** Analizar el espectro de frecuencia, identificando las características del audio entrante.
- **Reconocimiento de fonemas:** Comparar los patrones encontrados en el análisis del espectro de frecuencia con los patrones de los fonemas de la lengua en la cual se está reconociendo.
- **Reconocimiento de palabras:** Comparar la secuencia de los fonemas probables contra las palabras y los patrones de las palabras especificadas en la gramática.

2.1.6.1 JSAPI (Java Speech API)

El Java Speech API especifica una interfaz de software multiplataforma estándar para el apoyo y control de reconocedores de voz, sistemas de dictado y sintetizadores de voz. Aunque JSAPI solo define una interfaz, existen varias implementaciones creadas por terceros, en otras palabras, JSAPI no es más que una especificación, una estructura de clases, interfaces, métodos y reglas de

funcionamiento para el API de reconocimiento y síntesis de voz, dejando implementaciones reales en manos de terceros. Con esto se logra que JSAPI sea implementado independientemente del motor utilizado, siempre y cuando el motor obedezca al estándar de JSAPI.

2.1.6.2 JSGF (Java SpeechGrammar)

La gramática utilizada por JSAPI, es JSGF o Java SpeechGrammar Format, es una especificación independiente de la plataforma para el uso del reconocimiento de voz, para determinar que debe escuchar y así determinar qué es lo que el usuario dijo. JSGF adopta el estilo y convenciones del lenguaje de programación Java, además del uso de notaciones de gramáticas tradicionales.

JSGF soporta gramáticas dinámicas, esto significa que la gramática que utiliza una aplicación puede ser modificada en tiempo de ejecución, según las necesidades de la aplicación; esto permite aumentar la flexibilidad del sistema, así como optimizar las aplicaciones que requieran reconocimiento de voz, adaptando la gramática a cada situación.

2.1.7 NetBeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE² es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. SunMicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un

archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

2.1.8 Concepto General Arduino

Arduino es una herramienta para crear computadoras que interactúan con el mundo exterior con mucha más facilidad que una computadora de escritorio. Es una plataforma tipo código abierto que se basa en un microcontrolador y un entorno de programación para escribir software.

(Arduino Introduction, s.f.). Las tarjetas Arduino tienen una gran versatilidad a la hora de emplearlas en un proyecto de este tipo, ya que el tipo de conexión es sencillo y permite tanto leer como escribir variables digitales y analógicas. Además, en lo concerniente a la programación, Arduino dispone de una gran cantidad de librerías que permiten controlar de manera más eficiente dispositivos como motores servo, motores a paso, etc.

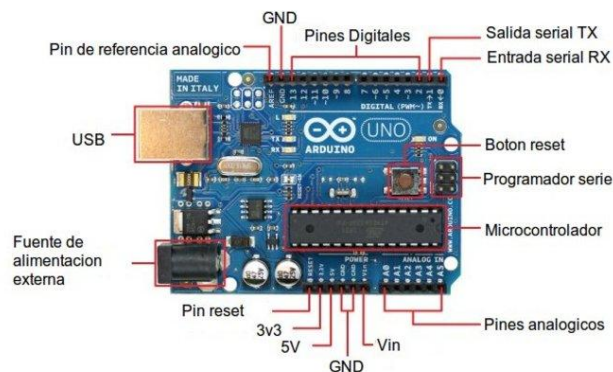


Figura 2: Arduino Uno

Fuente de <http://comohacer.eu>

2.1.9 Articulación

Una articulación es la parte de la estructura del robot mediante los cuales se unen los eslabones y permiten un movimiento relativo entre los mismos. Por lo general cada articulación que se aumenta en el robot, incrementa también un grado de libertad en el mismo (Romero, s.f).

2.1.10 Brazo Robótico

Se puede definir como el conjunto de elementos electromecánicos que propician el movimiento de un elemento terminal (gripper o herramienta), sea para cumplir una función o solo para manipular un objeto (Mena, 2011). La Figura 3 muestra una estructura simple de un brazo robótico.



Figura 3. Brazo robótico uarm
Fuente de <http://amazon.es>

El sistema de un brazo robótico está compuesto por una estructura mecánica, transmisiones, actuadores, sensores, elementos finales y un controlador. Su constitución física es similar a la anatomía de los brazos de un ser humano y por lo general se hace referencia a los componentes del robot con los nombres de su parte correspondiente en la extremidad de una persona. Como por ejemplo hombro, codo, brazo, muñeca, etc. (Romero, s.f.).

Existen cuatro configuraciones básicas para un brazo robótico. La primera es la configuración cartesiana, que consta de tres articulaciones prismáticas. La segunda configuración es la cilíndrica, la cual tiene dos articulaciones prismáticas y una de rotación. La configuración polar o esférica está formada por dos articulaciones rotacionales y una prismática.

Finalmente, la configuración angular tiene tres articulaciones rotacionales (Ollero, 2001).

2.1.11 Grados de libertad

Los grados de libertad son la cantidad de parámetros independientes que determinan la posición del elemento terminal del brazo robótico, el número de grados de libertad por lo general coincide con el número de eslabones de la cadena cinemática (Ollero, 2001).

Para posicionar y orientar un objeto en el espacio de cualquier manera deseada, se necesitan seis parámetros, tres para la posición y tres para la orientación. Es por esto que por lo general los brazos robóticos industriales en su mayoría tienen seis grados de libertad. Cuando el número de grados de libertad del robot excede los necesarios para que cumpla con su tarea se suele decir que un robot es redundante (Romero, s.f.).

2.1.12 Modulación por ancho de pulsos

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de *pulse-width modulation*) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

$$D = \frac{t}{T}$$

D es el ciclo de trabajo

t es el tiempo en que la función es positiva (ancho del pulso)

T es el período de la función

La construcción típica de un circuito PWM se lleva a cabo mediante un comparador con dos entradas y una salida. Una de las entradas se conecta a un oscilador de onda diente de sierra, mientras que la otra queda disponible para la señal moduladora. En la salida la frecuencia es generalmente igual a la de la señal diente de sierra y el ciclo de trabajo está en función de la portadora.

La principal desventaja que presentan los circuitos PWM es la posibilidad de que haya interferencias generadas por radiofrecuencia. Estas pueden minimizarse ubicando el controlador cerca de la carga y realizando un filtrado de la fuente de alimentación (Wikipedia).

2.1.13 Concepto general Servo motores

Los servos son un tipo especial de motor de c.c. que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su intervalo de operación. Para ello, el servomotor espera un tren de pulsos que se corresponde con el movimiento a realizar. Están generalmente formados por un amplificador, un motor, un sistema reductor formado por ruedas dentadas y un circuito de realimentación, todo en una misma caja de pequeñas dimensiones. El resultado es un servo de posición con un margen de operación de 180° aproximadamente.

Se dice que el servo es un dispositivo con un eje de rendimiento controlado ya que puede ser llevado a posiciones angulares específicas al enviar una señal codificada. Con tal de que exista una señal codificada en la línea de entrada, el servo mantendrá la posición angular del engranaje. Cuando la señal codificada cambia, la posición angular de los piñones cambia. En la práctica, se usan servos para posicionar elementos de control como palancas, pequeños ascensores y timones. También se usan en radio-control, marionetas y, por supuesto, en robots. Los Servos son sumamente útiles en robótica. Los motores son pequeños. Un motor como el de las imágenes superiores posee internamente una circuitería de control y es sumamente potente para su tamaño. Un servo normal o estándar como el HS-300 de Hitec proporciona un par de 3 kg-cm a 4.8 V, lo cual es bastante para su tamaño, sin consumir mucha energía. La corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cual es la corriente que consume. Eso no significa mucho si todos los servos van a estar moviéndose todo el tiempo. La corriente depende principalmente del par, y puede exceder un amperio si el servo está enclavado.

2.2 Elementos del sistema

2.2.1 Arduino UNO

Arduino es una placa con un microcontrolador de la marca Atmel y con toda la circuitería de soporte, que incluye, reguladores de tensión, un puerto USB (En los últimos modelos, aunque el original utilizaba un puerto serie) conectado a un módulo adaptador USB-Serie que permite programar el microcontrolador desde cualquier PC de manera cómoda y también hacer pruebas de comunicación con el propio chip.

Resumen de características Técnicas

Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Límite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin	3.3V 50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

2.2.2 Servomotor

Dentro de los servos económicos el MG995 Tower Pro destaca por su gran torque, engranajes metálicos y gran robustez. Funciona con la mayoría de tarjetas electrónicas con microcontroladores y además con la mayoría de los sistemas de radio control comercial. Es utilizado principalmente en proyectos de robótica y modelismo de mediano tamaño.

El servo Tower Pro MG995 tiene un conector universal tipo “S” que encaja perfectamente en la mayoría de los receptores de radio control incluyendo los Futaba, JR, GWS, Cirrus, Hitec y otros. Los cables en el conector están distribuidos de la siguiente forma: : **Rojo** =Alimentación (+), **Cafe** = Alimentación (–) o tierra, **Naranja**= Señal PWM.

Características

- Dimensiones: 40mm x 19mm x 43mm
- Ángulo de rotación : 360 grados máximo
- Peso neto : 69g ; Peso del paquete : 75g
- Velocidad de funcionamiento :
 - 0.17sec / 60 grados (4.8V)
 - 0.13sec / 60 grados (6.0V)
- Par:
 - 13 kg - cm en 4.8V
 - 15 kg - cm en 6V
- Voltaje de funcionamiento: 4.8 - 7.2Volts
- El conector del cable: (30cm)

2.2.3 Aluminio

Detalle superficial (55×37 mm) de una barra de aluminio (pureza $\geq 99,9998$ %). La superficie ha sido pulida mediante medios químicos con ácido (etching) para evidenciar a simple vista la estructura de las cristalitas metálicas.

Características físicas

El aluminio es un elemento muy abundante en la naturaleza, solo aventajado por el oxígeno y el silicio. Se trata de un metal ligero, con una densidad de 2700 kg/m³, y con un bajo punto de fusión (660 °C). Su color es blanco y refleja bien la radiación electromagnética del espectro visible y el térmico. Es buen conductor eléctrico (entre 35 y 38 m/(Ω mm²)) y térmico (80 a 230 W/(m·K)).

Características mecánicas

Es un material blando (escala de Mohs: 2-3-4) y maleable. En estado puro tiene un límite de resistencia en tracción de 160-200 N/mm² (160-200 MPa). Todo ello le hace adecuado para la fabricación de cables eléctricos y láminas delgadas, pero no como elemento estructural. Para mejorar estas propiedades se alea con otros metales, lo que permite realizar sobre él operaciones de fundición y forja, así como la extrusión del material. También de esta forma se utiliza como soldadura.

Características químicas

Estructura atómica del aluminio.

La capa de valencia del aluminio está poblada por tres electrones, por lo que su estado normal de oxidación es III. Esto hace que reaccione con el oxígeno de la atmósfera formando con rapidez una fina capa gris mate de alúmina Al₂O₃, que recubre el material, aislándolo de ulteriores corrosiones. Esta capa puede disolverse con ácido cítrico. A pesar de ello es tan estable que se usa con frecuencia para extraer otros metales de sus óxidos. Por lo demás, el aluminio se disuelve en ácidos y bases. Reacciona con facilidad con el ácido clorhídrico y el hidróxido sódico.

2.2.4 Talking Java SDK

Talking Java SDK es una librería desarrollada por *Cloud Garden* que realiza una implementación de JSAPI para la plataforma Windows.

Dicha librería necesita unos requisitos mínimos tanto software como hardware para su correcto funcionamiento:

Requisitos software:

- ✓ Sun's JDK/JRE: JDK (Java Development Kit) es un software que provee herramientas de desarrollo para la creación de programa en Java. Puede ser instalado en un ordenador local o en una unidad de red. JRE (Java Runtime Environment) es un conjunto de utilidades que permite la ejecución de programas Java. Un usuario sólo necesita el JRE para ejecutar aplicaciones desarrolladas en Java, mientras que para

desarrollar nuevas aplicaciones en dicho lenguaje es necesario el JDK, que además de incluir el JRE, incluye un compilador para Java. Para trabajar con aplicaciones relacionadas con el reconocimiento del habla se recomienda utilizar la versión 1.3 o mayor.

- ✓ Sun's Java Media Framework: Es una extensión de Java que permite la programación de tareas multimedia en dicho lenguaje de programación.
- ✓ Es recomendable la versión 2.1.1 o superior.
- ✓ Uno o más motores de voz compatibles: Por ejemplo SAPI 5.0 aunque hay muchos de libre distribución.
- ✓ Sistemas operativos compatibles: Windows 98, Me, NT4, 2000 o XP.

Requisitos hardware:

Los sistemas Pentium II de 400MHz con 256 Mb de memoria RAM proporcionan un buen rendimiento.

Los sistemas Pentium II de 133MHz con 64Mb también ofrecen un rendimiento satisfactorio para gramática de comandos aunque son lentos para realizar dictados.

No es necesaria una tarjeta de sonido especial, la mayoría de ellas funcionan correctamente.

TalkingJava SDK es de libre distribución para un uso no comercial. Si su utilización se realiza en compañías o instituciones es necesario adquirir una licencia.

No es necesaria una tarjeta de sonido especial, la mayoría de ellas funcionan correctamente. TalkingJava SDK es de libre distribución para un uso no comercial. Si su utilización se realiza en compañías o instituciones es necesario adquirir una licencia

CAPÍTULO III

3 INGENIERIA DE PROYECTO

3.1 Descripción del proyecto

El prototipo de brazo robótico controlado por voz permitirá a los usuarios poder controlar el movimiento de los actuadores por medio de comandos de voz que son:

- Base izquierda, Base derecha
- Brazo subir, Brazo bajar
- Mano subir, mano bajar
- Girar garra izquierda, girar garra derecha
- Abrir garra, Cerrar garra

El mecanismo nos permitirá poder tener secuencias de giros que realizara el robot repetida mente el cual permitirá poder realizar la manipulación de compuestos químicos tóxicos para el ser humano y también permitirá agarrar objetos contaminados para ser movidos de un lugar a otro, que las personas no lo pueden hacer porque corren el riesgo de sufrir un accidente o daño físicos por ser altamente peligrosos.

Mediante una computadora se mandara comandos de voz el cual los procesara y lo volverá texto para q luego el programa los reconozca el texto y lo envíe por el puerto usb de la pc que está conectado a un arduino mediante una comunicación

3.3 Diagrama en bloques del sistema

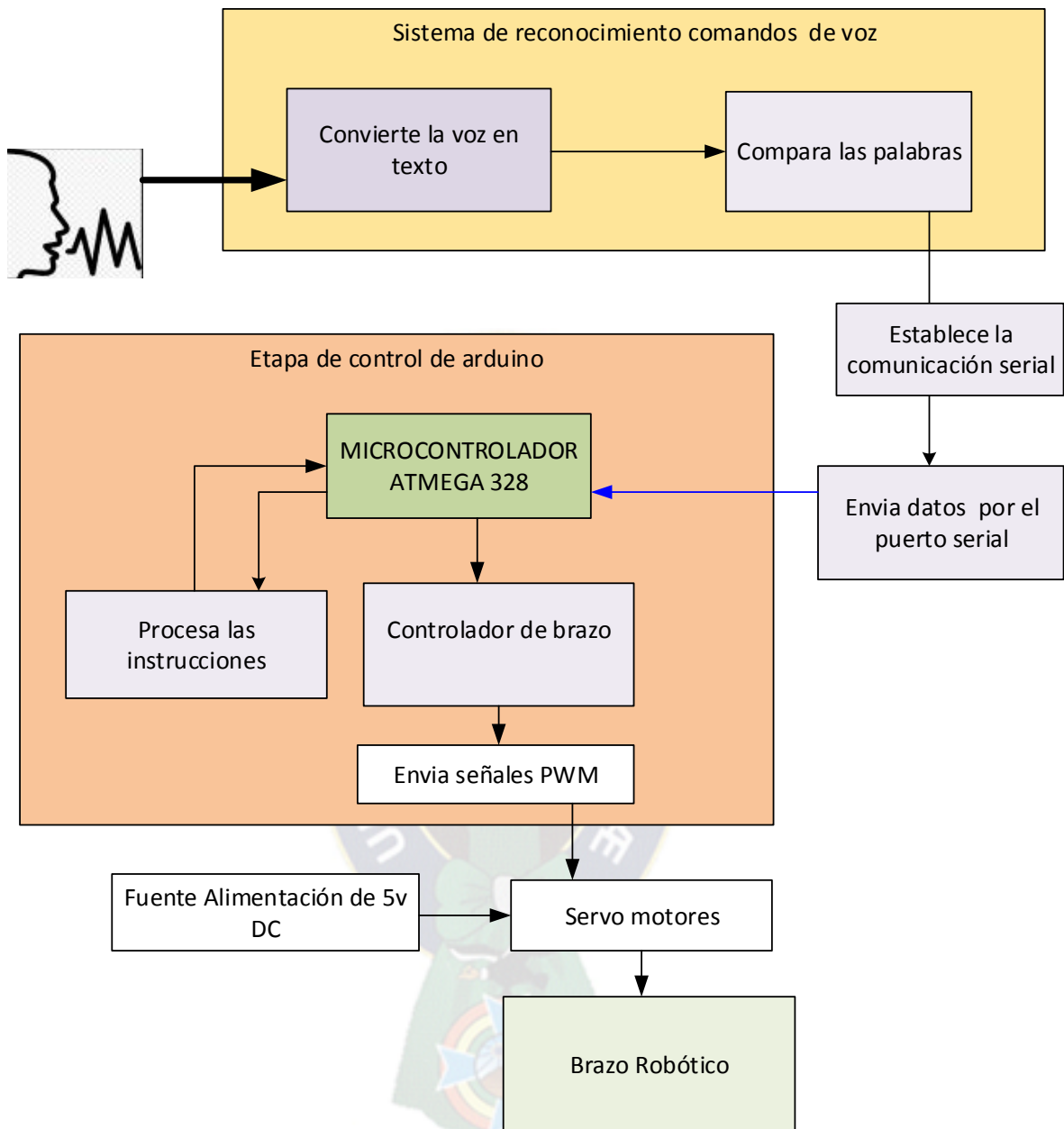


Figura 5: Diagrama en bloques

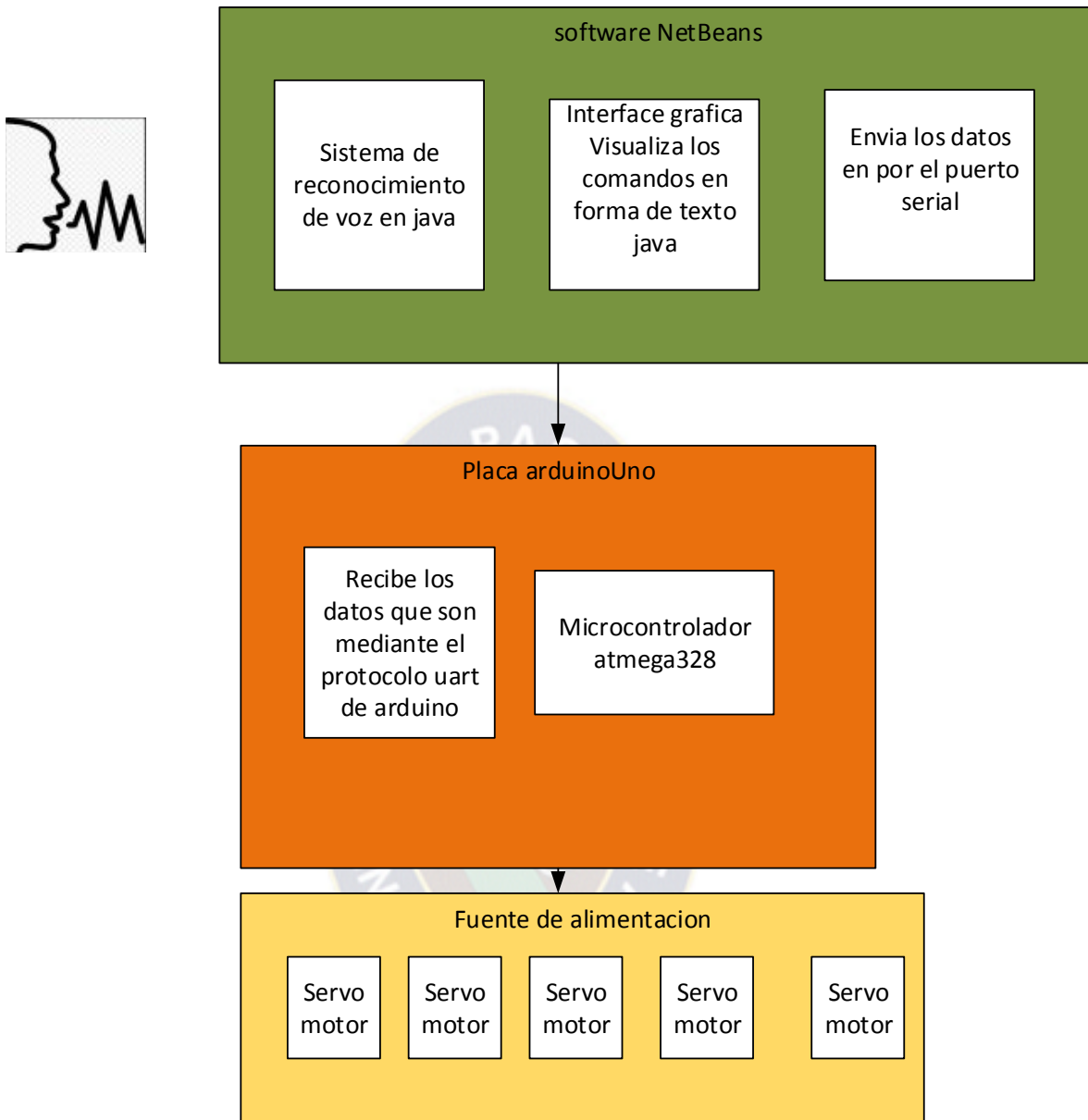


Figura 6: Diagrama en bloque de elementos del sistema

3.4 Sistema De Reconocimiento de Voz

Mediante una computadora recibirá señales de voz, que serán procesados por el motor de Microsoft que utiliza la librería de reconocimiento de voz llamado talking java que se utilizó en el programa NetBeans.

Convierte la voz en texto: Una vez obtenido los fonemas este es convertido a texto y es almacenado en una variable llamada palabra de tipo String, gracias a talking java.

Compara las palabras: Después de obtener los fonemas compara con el archivo de texto si son los fonemas correctos a utilizar después es comparado con el uso de condicionales pre establecidos en el programa anterior mente para verificar si corresponde la palabra mencionada, si son iguales este me permite enviar un número por pantalla cada vez que reciba una señal de voz.

Una vez obtenido el número por pantalla se toman los datos y con la estructura selectiva verifica si coincide con las instrucciones que se necesita y este permite enviar datos de tipo ASCII o números por el puerto serial mediante la utilización de una jar (librería) Arduino de java.

3.4.1 Código de Netbeans del Reconocimiento de Voz

```
import java.io.FileReader;
import java.util.Locale;
import javax.speech.Central;
import javax.speech.EngineModeDesc;
import javax.speech.recognition.*;

/**
 * Esta es la clase principal, la clase que se encarga de comprar los fonemas y
 * compararlos con el carchivo ("c:/SimpleGrammarES2.txt"), donde se establecen
 * las palabras que se utilizaran en el fonema.....
 */
public class ReconocimientoVoz extends ResultAdapter implements Runnable {

    BrazoRoboticoVoz brazorobotico;
    /**
     * Permite manipular la pantalla desde esta clase, y ejecutar las
```

```

* acciones segun el comando de voz*
*/
Recognizer reconecedorVoz;
/**
* Reconoce el comando de voz del Usuario
*/
String palabra;

/**
* Palabra que el usuario a pronunciado
*/
public ReconocimientoVoz(BrazoRoboticoVoz brazorobotico) {
    this.brazorobotico = brazorobotico;
}

/**
* Captura el fonema(palabra), y la compara con un conjunto de palabras y
* ejecuta una accion establecida.
*/
@Override
public void resultAccepted(ResultEvent re) {
    try {
        Result res = (Result) (re.getSource());
        ResultToken tokens[] = res.getBestTokens();
        for (int i = 0; i < tokens.length; i++) {
            palabra = tokens[i].getSpokenText();
            if (palabra.equals("izquierda")) {
                this.brazorobotico.escribirEnPantalla("1");
            }
            if (palabra.equals("derecha")) {
                this.brazorobotico.escribirEnPantalla("2");
            }
            if (palabra.equals("base")) {
                this.brazorobotico.escribirEnPantalla("3");
            }
            if (palabra.equals("brazo")) {
                this.brazorobotico.escribirEnPantalla("4");
            }
            if (palabra.equals("mano")) {
                this.brazorobotico.escribirEnPantalla("5");
            }
            if (palabra.equals("girar")) {
                this.brazorobotico.escribirEnPantalla("6");
            }
            if (palabra.equals("garra")) {
                this.brazorobotico.escribirEnPantalla("7");
            }
        }
    }
}

```

```

    }
    if (palabra.equals("abrir")) {
        this.brazorobotico.escribirEnPantalla("8");
    }
    if (palabra.equals("cerrar")) {
        this.brazorobotico.escribirEnPantalla("9");
    }
    if (palabra.equals("secuencia")) {
        this.brazorobotico.escribirEnPantalla("10");
    }
    if (palabra.equals("subir")) {
        this.brazorobotico.escribirEnPantalla("11");
    }
    if (palabra.equals("bajar")) {
        this.brazorobotico.escribirEnPantalla("22");
    }
    if (palabra.equals("tres")) {
        this.brazorobotico.escribirEnPantalla("33");
    }
    if (palabra.equals("parar")) {
        this.brazorobotico.escribirEnPantalla("13");
    }
    if (palabra.equals("reset")) {
        this.brazorobotico.escribirEnPantalla("21");
    }
    if (palabra.equals("robot")) {
        //borra pantalla
        this.brazorobotico.borrarTodo();
    }
}
} catch (Exception ex) {
    System.out.println("Error: " + ex.getMessage());
}
}

```

```

/**
 * Carga el archivo ("c:/SimpleGrammarES2.txt") y para poder empezar un
 * proceso de comparacion gramatical y hacer los procesos de
 * reconocimiento
 *
 * Importante: Se debe establecer un archivo ("c:/SimpleGrammarES2.txt"),
 * con un conjunto de expresiones gramaticales(Palabras) para poder generar
 * un proceso correcto de reconocimiento
 */
public void reconocerVoz() {
    try {

```

```

reconocedorVoz = Central.createRecognizer(new
EngineModeDesc(Locale.ROOT));
reconocedorVoz.allocate();

FileReader gramatica = new FileReader("c:/SimpleGrammarES2.txt");

RuleGrammar reglaGramatica = reconocedorVoz.loadJSGF(gramatica);
reglaGramatica.setEnabled(true);

reconocedorVoz.addResultListener(new
ReconocimientoVoz(brazorobotico));
reconocedorVoz.commitChanges();
reconocedorVoz.requestFocus();
} catch (Exception e) {
System.out.println("Error en " + e.toString());
}
}

@Override
public void run() {
this.reconocerVoz();
}
}

```

3.5 Establecimiento de comunicación serial

En él, estableceremos una comunicación con Arduino a través de Java. Para ello, utilizaremos la librería RXTX, con el fin de poder utilizar el puerto serie.

La explicación del proyecto es muy sencilla. Lo que hacemos es abrir el puerto desde el cual podemos utilizar Arduino y definirlo en nuestro código Java.

El enviando datos de Java a Arduino

Para ello utilizaremos un método similar a **EnviarDatos(): RecibirDatos()**.

Para poder utilizar este método necesitamos modificar nuestra función **ArduinoConnection()** y declarar algunas variables.

Los datos que enviamos de Arduino a Java se envían a través de una variable **OutputStream** llamada **Output**.

3.5.1 Código de netbeans de Comunicación Serial

```
import Arduino.Arduino;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class BrazoRoboticoVoz extends javax.swing.JFrame implements Runnable
{

    Arduino Arduino = new Arduino();
    public ReconocimientoVoz escuchar;
    public Thread hiloEscucha;

    public BrazoRoboticoVoz() {
        initComponents();

        try {
            Arduino.ArduinoTX("COM3", 2000, 9600);
            //nombre del puerto , tiempo de sailda, velocidad de transmision baudios

        } catch (Exception ex) {
            Logger.getLogger(BrazoRoboticoVoz.class.getName()).log(Level.SEVERE,
null, ex);
        }
        this.setLocationRelativeTo(null); //centrea pantalla
        this.setVisible(true);
//        this.presionadoAccion = false;
        this.txtNumero.setEditable(false);
    }
    public void borrarTodo() {
        this.txtNumero.setText("");
    }

    private void iniciarReconocimieto() {
        hiloEscucha = new Thread(escuchar = new ReconocimientoVoz(this));
        hiloEscucha.start();
    }
//escribe en pantalla

    public void escribirEnPantalla(String numero) {
        this.txtNumero.setText(this.txtNumero.getText() + numero);
        System.out.println(this.txtNumero.getText());
//garra abrir
        if (Integer.parseInt(this.txtNumero.getText()) == 78) {
            try {
                Arduino.SendData("A");
            }
        }
    }
}
```

```

    } catch (Exception ex) {

Logger.getLogger(BrazoRoboticoVoz.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
//garra cerrar
if (Integer.parseInt(this.txtNumero.getText()) == 79) {
    try {
        Arduino.SendData("a");
    } catch (Exception ex) {

Logger.getLogger(BrazoRoboticoVoz.class.getName()).log(Level.SEVERE, null,
ex);
    }
}
}

```



Figura 7: Interfaz de usuario

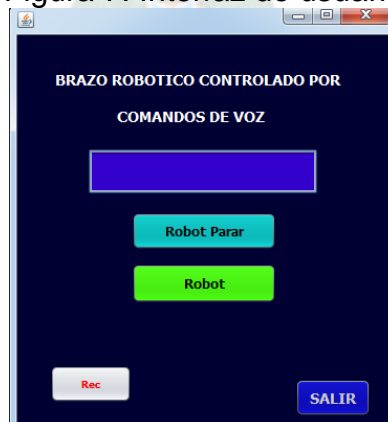


Figura 8: Interfaz de Reconocimiento de voz

3.6 Etapa De Control del Arduino

Una vez recibido los datos por el puerto serial estos son procesados por el Microcontrolador atmega328 el cual está programado por el IDE arduino, esto permite controlar las cinco señales PWM, mediante la utilización de la librería PWM.

3.6.1 Código fuente Arduino

```
int input;
#include <Servo.h> // incluimos la biblioteca Servo

Servo miservo1; // creamos un objeto servo para controlar el motor
Servo miservo2; // creamos un objeto servo para controlar el motor
Servo miservo3; // creamos un objeto servo para controlar el motor
Servo miservo4; // creamos un objeto servo para controlar el motor
Servo miservo5; // creamos un objeto servo para controlar el motor
void setup(){

pinMode(13, OUTPUT); // Declaramos que utilizaremos el pin 13 como salida

Serial.begin(9600);

miservo1.attach(5); // liga el servo al pin 5 base 180
miservo2.attach(6); // liga el servo al pin 6 brazo 360
miservo3.attach(9); // liga el servo al pin 9 mano 360
miservo4.attach(10); // liga el servo al pin 10 muñeca 180
miservo5.attach(11); // liga el servo al pin 11 garra 180
}
void loop(){

input=Serial.read(); // girar garra izquierda
//servo pin 11 garra 360
if (input=='A'){ //cerrar
    miservo5.write(10); // manda al servo la posicion
    delay(410); // espera unos milisegundos para que el servo
llegue a su posicion
}
miservo5.write(90); // manda al servo la posicion
delay(20);
if (input=='a'){ //abrir
    miservo5.write(170); // manda al servo la posicion
    delay(400); // espera unos milisegundos para que el servo
llegue a su posicion
```

```

}
miservo5.write(90);      // manda al servo la posicion
  delay(20);

//servo pin 10 muñeca 180
if (input=='B'){        //cerrar
  miservo4.write(70);    // manda al servo la posicion
  delay(1000);           // espera unos milisegundos para que el servo
  llegue a su posicion
}
miservo4.write(90);     // manda al servo la posicion
  delay(20);
if (input=='b'){        //abrir
  miservo4.write(110);   // manda al servo la posicion
  delay(1000);           // espera unos milisegundos para que el servo
  llegue a su posicion
}
miservo4.write(90);     // manda al servo la posicion
  delay(20);
//servo pin 9 mano 360
if (input=='C'){
  miservo3.write(10);    // manda al servo la posicion
  delay(500);             // espera unos milisegundos para que el servo
  llegue a su posicion
}
miservo3.write(90);     // manda al servo la posicion
  delay(200);
if (input=='c'){
  miservo3.write(170);   // manda al servo la posicion
  delay(500);             // espera unos milisegundos para que el servo
  llegue a su posicion
}
miservo3.write(90);     // manda al servo la posicion
  delay(200);

//servo pin 6 brazo 360
if (input=='D'){
  miservo2.write(10);    // manda al servo la posicion
  delay(600);             // espera unos milisegundos para que el servo
  llegue a su posicion
}
miservo2.write(90);     // manda al servo la posicion
  delay(200);
if (input=='d'){
  miservo2.write(170);   // manda al servo la posicion
  delay(600);             // espera unos milisegundos para que el servo
  llegue a su posicion
}

```



```

}
miservo2.write(90);          // manda al servo la posicion
  delay(200);
//servo pin 5 base 180
if (input=='E'){

  for(int angulo = 10; angulo < 70; angulo += 1) { // un ciclo para mover el servo
entre los 0 y los 180 grados
    miservo1.write(angulo);          // manda al servo la posicion
    delay(20);                       // espera unos milisegundos para que el servo
llegue a su posicion
  }
}
if (input=='e'){

  for(int angulo = 70; angulo >= 10; angulo -= 1) { // un ciclo para mover el servo
entre los 180 y los 0 grados
    miservo1.write(angulo);          // manda al servo la posicion
    delay(20);                       // espera unos milisegundos para que el servo
llegue a su posicion
  }
}
}.

```

4 CONCLUSIONES Y RECOMENDACIONES

Al final de este proyecto y con base en presentado y los objetivos inicialmente planteados se puede concluir que:

- Se cumplió con el objetivo general que era desarrollar un brazo robótico controlado por voz en java para manipular sustancias peligrosas.

Mirando los objetivos específicos se concluye que:

- Mediante la librería TalkingJava se logró utilizar algoritmos de procesamiento digital de voz que permitan un tratamiento sencillo de información de señales de voz.
- Se Diseñó la estructura del brazo robótico utilizando piezas de aluminio modificándolos según lo requerido.

- En cuanto a la Implementación del circuito para la manipulación del brazo robótico electrónico todo funciona de manera adecuada, el Arduino procesa de manera correcta tanto las señales enviadas por el puerto serial, y trabaja muy bien con los límites establecidos. Esto nos permite concluir que para esta clase de proyectos o sistemas el Arduino es una muy buena opción como sistema de control, cumple muy bien con lo solicitado y de manera eficaz.
- También se interactuó de forma automática entre humano y computadora por medio de un sistema digital.

En cuanto al reconocimiento de voz se concluye que:

- En el proceso del reconocimiento de voz hay demasiadas variables que desde el desarrollo no se pueden controlar, produciendo fallas en el proceso, la voz de las personas puede variar según su estado de ánimo, lo que hace más difícil la tarea de reconocimiento, por lo tanto un reconocedor que es abierto a cualquier locutor, puede equivocarse más que los reconocedores que son específicos para un locutor.

Las posibilidades de una aplicación usando reconocimiento de voz para controlar el brazo robótico son muchas, este sólo es un ejemplo de lo que se puede llegar a realizar.

Para aplicaciones en las que se necesite mayor torque en las articulaciones del robot, utilizar motores DC con acondicionamientos como engranajes de diferente tamaño para amplificar o reducir el torque.

5 BIBLIOGRAFIA

Barrientos, B.; Peñín, L.; Balaguer, C. y Aracil, R. Fundamentos de robótica.

CLOUD GARDEN. TalkingJava SDK (en línea). Disponible en

<http://www.cloudgarden.com/JSAPI/index.html>.

<http://arduino.cc/en/Guide/Introduction>

Ollero, A. (2001). Robótica, manipuladores y robots móviles. Barcelona, España: Marcombo.

<HTTPS://OPENWEBINARS.NET/BLOG/TUTORIAL-ARDUINO-COMUNICACION-SERIE/>

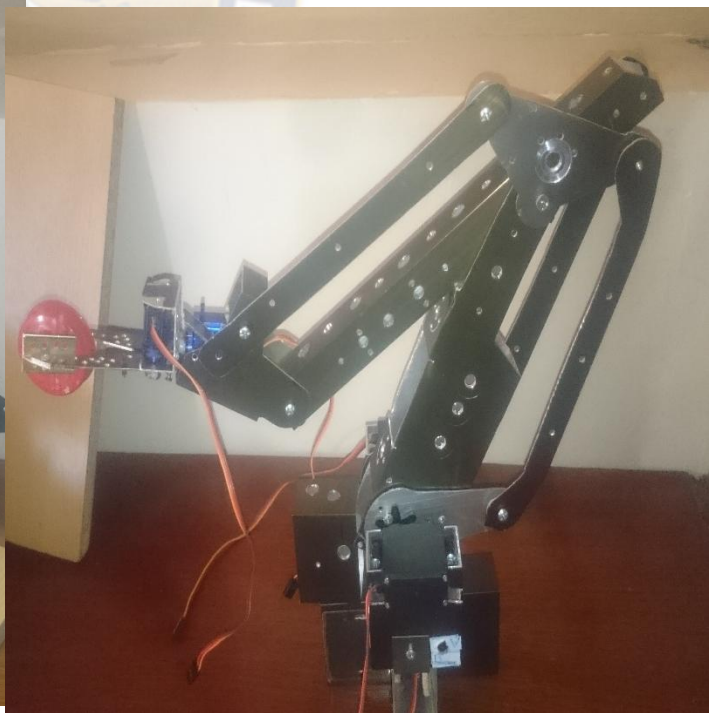
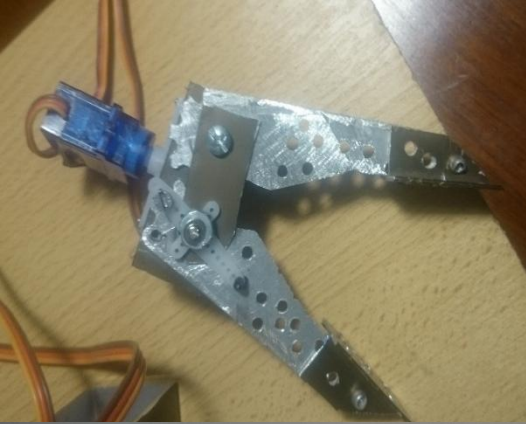
<http://www.electroensaimada.com/servomotor.html>

<http://www.prometec.net/servos/>



ANEXOS

Fotos del Brazo Robótico



Código fuente Arduino

```
String input;
#include <Servo.h> // incluimos la biblioteca Servo

Servo miservo1; // creamos un objeto servo para controlar el motor
Servo miservo2; // creamos un objeto servo para controlar el motor
Servo miservo3; // creamos un objeto servo para controlar el motor
Servo miservo4; // creamos un objeto servo para controlar el motor
Servo miservo5; // creamos un objeto servo para controlar el motor
void setup(){

pinMode(13, OUTPUT); // Declaramos que utilizaremos el pin 13 como salida

Serial.begin(9600);

miservo1.attach(5); // liga el servo al pin 5 base
miservo2.attach(6); // liga el servo al pin 6 brazo
miservo3.attach(9); // liga el servo al pin 9 mano
miservo4.attach(10); // liga el servo al pin 10 muñeca
miservo5.attach(11); // liga el servo al pin 11 garra
}

void loop(){

input=Serial.read();

if (input.equals("11"){
    miservo1.write(60); // manda al servo la posicion
    delay(200); // espera unos milisegundos para que el servo llegue a su
posicion
}

if (input.equals("22"){
    miservo1.write(80); // manda al servo la posicion
    delay(200); // espera unos milisegundos para que el servo llegue a su
posicion
}
if (input.equals("33"){
    miservo1.write(100); // manda al servo la posicion
    delay(200); // espera unos milisegundos para que el servo llegue a su
posicion
}
if (input.equals("44"){
    miservo1.write(140); // manda al servo la posicion
    delay(200); // espera unos milisegundos para que el servo llegue a su
posicion
}
```

```

miservo1.write(90);      // manda al servo la posicion
delay(20);

//segundo servo

if (input.equals("66")){
    miservo2.write(60);  // manda al servo la posicion
    delay(200);         // espera unos milisegundos para que el servo llegue a su
posicion
}

if (input.equals("77"){
    miservo2.write(80);  // manda al servo la posicion
    delay(200);         // espera unos milisegundos para que el servo llegue a su
posicion
}
if (input.equals("88"){
    miservo2.write(100); // manda al servo la posicion
    delay(200);         // espera unos milisegundos para que el servo llegue a su
posicion
}
if (input.equals("99"){
    miservo2.write(140); // manda al servo la posicion
    delay(200);         // espera unos milisegundos para que el servo llegue a su
posicion
}

miservo2.write(90);     // manda al servo la posicion
delay(20);

if (input.equals("78")
{
    for(int angulo = 0; angulo < 180; angulo += 1) { // un ciclo para mover el servo entre
los 0 y los 180 grados
        miservo4.write(angulo); // manda al servo la posicion
        delay(20);             // espera unos milisegundos para que el servo llegue a su
posicion
    }
}

if (input.equals("79")
{
    for(int angulo = 180; angulo >= 1; angulo -= 1) { // un ciclo para mover el servo entre
los 180 y los 0 grados
        miservo4.write(angulo); // manda al servo la posicion
        delay(20);             // espera unos milisegundos para que el servo llegue a su
posicion
    }
}

```

```

if (input.equals("671"){

    for(int angulo = 0; angulo < 180; angulo += 1) { // un ciclo para mover el servo entre
    los 0 y los 180 grados
        miservo5.write(angulo); // manda al servo la posicion
        delay(20); // espera unos milisegundos para que el servo llegue a su
    posicion
    }
}

if (input.equals("672")

{

    for(int angulo = 180; angulo >= 1; angulo -= 1) { // un ciclo para mover el servo entre
    los 180 y los 0 grados
        miservo5.write(angulo); // manda al servo la posicion
        delay(20); // espera unos milisegundos para que el servo llegue a su
    posicion
    }
}
}
}

```

Código de Netbeans del software en el computador

Formulario JFrame de INTERFACE

```

import Arduino.Arduino;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

public class INTERFACE extends javax.swing.JFrame implements Runnable {

    Arduino Arduino = new Arduino();
    // private double operando1;
    // private double operando2;
    // private double resultado;
    /**
     * Permite saber si se ejecuto una accion matematica
     */
    // private boolean presionadoAccion;
    /**
     * Permite saber cuan es la accion(Operacion) matematica que se ejecuto
     */
    // private byte accion;
    public ReconocimientoVoz escuchar;
    public Thread hiloEscucha;

    public INTERFACE() {
        initComponents();
    }
}

```

```

try {
    Arduino.ArduinoTX("COM3", 2000, 9600);

} catch (Exception ex) {
    Logger.getLogger(INTERFACE.class.getName()).log(Level.SEVERE, null, ex);
}
this.setLocationRelativeTo(null);
this.setVisible(true);
// this.presionadoAccion = false;
this.txtNumero.setEditable(false);
}

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new javax.swing.JPanel();
    btnparar = new javax.swing.JButton();
    btnReset = new javax.swing.JButton();
    btnBorrarTodo1 = new javax.swing.JButton();
    txtNumero = new javax.swing.JFormattedTextField();
    BtnSalir = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setBackground(new java.awt.Color(0, 0, 0));

    jPanel1.setBackground(new java.awt.Color(0, 0, 51));

    btnparar.setBackground(new java.awt.Color(0, 153, 153));
    btnparar.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
    btnparar.setText("Robot Parar");
    btnparar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnpararActionPerformed(evt);
        }
    });

    btnReset.setBackground(new java.awt.Color(51, 204, 0));
    btnReset.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
    btnReset.setText("Robot");
    btnReset.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            btnResetActionPerformed(evt);
        }
    });

    btnBorrarTodo1.setFont(new java.awt.Font("Tahoma", 1, 10)); // NOI18N
    btnBorrarTodo1.setForeground(new java.awt.Color(255, 0, 0));
    btnBorrarTodo1.setText("Rec");
    btnBorrarTodo1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {

```



```

        btnBorrarTodo1ActionPerformed(evt);
    }
});

txtNumero.setBackground(new java.awt.Color(51, 0, 204));
txtNumero.setForeground(new java.awt.Color(255, 255, 255));
txtNumero.setFormatterFactory(new javax.swing.text.DefaultFormatterFactory(new
javax.swing.text.NumberFormatter()));
txtNumero.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
txtNumero.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
txtNumero.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtNumeroActionPerformed(evt);
    }
});

BtnSalir.setBackground(new java.awt.Color(0, 0, 153));
BtnSalir.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
BtnSalir.setForeground(new java.awt.Color(204, 204, 204));
BtnSalir.setText("SALIR");
BtnSalir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BtnSalirActionPerformed(evt);
    }
});

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("COMANDOS DE VOZ");

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("BRAZO ROBOTICO CONTROLADO POR ");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addGap(31, 31, 31)
        .addComponent(btnBorrarTodo1,
javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(BtnSalir)
        .addGap(18, 18, 18)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(21, 21, 21)

```

```

        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 289,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 45, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
        .addGap(0, 0, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED_SIZE,
138, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnparar, javax.swing.GroupLayout.PREFERRED_SIZE, 138,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(108, 108, 108))
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(68, 68, 68)
        .addComponent(txtNumero, javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(30, 30, 30)
        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 289,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(36, Short.MAX_VALUE)))
);
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addGap(64, 64, 64)
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(txtNumero, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(btnparar, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(13, 13, 13)
        .addComponent(btnReset, javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
59, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup())
        .addComponent(BtnSalir, javax.swing.GroupLayout.PREFERRED_SIZE,
36, javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addContainerGap()
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addComponent(btnBorrarTodo1,
javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(19, 19, 19)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(29, 29, 29)
        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(314, Short.MAX_VALUE)))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
}

private void btnpararActionPerformed(java.awt.event.ActionEvent evt) {
    this.borrarTodo();
}

private void btnBorrarTodo1ActionPerformed(java.awt.event.ActionEvent evt) {
    this.iniciarReconocimiento();

    JOptionPane.showMessageDialog(rootPane, "Reconocimiento de voz iniciado"
    + "\nincorrectamente....!!", "Comando",
    JOptionPane.INFORMATION_MESSAGE);
}

```

```

private void txtNumeroActionPerformed(java.awt.event.ActionEvent evt) {
}

private void BtnSalirActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

public void borrarTodo() {
    this.txtNumero.setText("");
}

private void iniciarReconocimieto() {
    hiloEscucha = new Thread(escuchar = new ReconocimientoVoz(this));
    hiloEscucha.start();
}

public void escribirEnPantalla(String numero) {
    this.txtNumero.setText(this.txtNumero.getText() + numero);
    System.out.println(this.txtNumero.getText());
    // girar garra izquierda
    if (Integer.parseInt(this.txtNumero.getText()) == 671) {
        try {
            Arduino.SendData("A");
        } catch (Exception ex) {
            Logger.getLogger(INTERFACE.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    // girar garra derecha
    if (Integer.parseInt(this.txtNumero.getText()) == 672) {
        try {
            Arduino.SendData("a");
        } catch (Exception ex) {
            Logger.getLogger(INTERFACE.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    // garra abrir
    if (Integer.parseInt(this.txtNumero.getText()) == 78) {
        try {
            Arduino.SendData("B");
        } catch (Exception ex) {
            Logger.getLogger(INTERFACE.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    // garra cerrar
    if (Integer.parseInt(this.txtNumero.getText()) == 79) {
        try {
            Arduino.SendData("b");
        }
    }
}

```

```

        } catch (Exception ex) {
            Logger.getLogger(INTERFACE.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
// Variables declaration - do not modify
private javax.swing.JButton BtnSalir;
private javax.swing.JButton btnBorrarTodo1;
private javax.swing.JButton btnReset;
private javax.swing.JButton btnparar;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPanel jPanel1;
private javax.swing.JFormattedTextField txtNumero;
// End of variables declaration

@Override
public void run() {
    throw new UnsupportedOperationException("Not supported yet.");
}
}

```

Class Main

```

public class Main {
    public static void main(String[] args) throws Exception {
        Menu calculadora = new Menu();
    }
}

```

Principal Formulario JFrame Menu

```

import javax.swing.JOptionPane;

public class Menu extends javax.swing.JFrame {

    public Menu() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel3 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txtUsuario = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        BtnSalir = new javax.swing.JButton();
    }
}

```

```

BtnIngresar = new javax.swing.JButton();
jPasswordField1 = new javax.swing.JPasswordField();
FONDO = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel3.setIcon(new
javax.swing.ImageIcon("C:\\Users\\Rex\\Desktop\\INTERFACES\\user.png")); // NOI18N
getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(30,
140, 250, 170));

jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setText("Usuario:");
getContentPane().add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 130, 110, 30));

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
jLabel2.setForeground(new java.awt.Color(255, 255, 255));
jLabel2.setText("Contraseña:");
getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 190, -1, 70));

txtUsuario.setBackground(new java.awt.Color(51, 0, 153));
txtUsuario.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
txtUsuario.setForeground(new java.awt.Color(255, 255, 255));
getContentPane().add(txtUsuario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 170, 180, 30));

jButton1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jButton1.setForeground(new java.awt.Color(0, 0, 153));
jButton1.setText("BORRAR");
getContentPane().add(jButton1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(280, 320, 120, 40));

BtnSalir.setBackground(new java.awt.Color(0, 0, 153));
BtnSalir.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
BtnSalir.setForeground(new java.awt.Color(204, 204, 204));
BtnSalir.setText("SALIR");
BtnSalir.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BtnSalirActionPerformed(evt);
    }
});
getContentPane().add(BtnSalir, new
org.netbeans.lib.awtextra.AbsoluteConstraints(440, 370, 100, 30));

BtnIngresar.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
BtnIngresar.setForeground(new java.awt.Color(51, 0, 153));
BtnIngresar.setText("INGRESAR");
BtnIngresar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        BtnIngresarActionPerformed(evt);
    }
});

```

```

    }
  });
  getContentPane().add(BtnIngresar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 320, 120, 40));

  jPasswordField1.setBackground(new java.awt.Color(51, 0, 153));
  jPasswordField1.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
  jPasswordField1.setForeground(new java.awt.Color(255, 255, 255));
  getContentPane().add(jPasswordField1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 250, 180, 30));

  FONDO.setForeground(new java.awt.Color(255, 255, 255));
  FONDO.setIcon(new
javax.swing.ImageIcon("C:\\Users\\Rex\\Desktop\\INTERFACES\\IMAGEN3.png")); //
NOI18N
  getContentPane().add(FONDO, new org.netbeans.lib.awtextra.AbsoluteConstraints(0,
0, 570, -1));

  pack();
} // </editor-fold>

private void BtnSalirActionPerformed(java.awt.event.ActionEvent evt) {
  // TODO add your handling code here:
  System.exit(0);
}

private void BtnIngresarActionPerformed(java.awt.event.ActionEvent evt) {
  // TODO add your handling code here
  char clave[]=jPasswordField1.getPassword();

String clavedef=new String(clave);

if (txtUsuario.getText().equals("admin") && clavedef.equals("12345")){

  this.dispose();

  JOptionPane.showMessageDialog(null, "Bienvenido\n"
+ "Has ingresado satisfactoriamente al sistema", "Mensaje de bienvenida",
JOptionPane.INFORMATION_MESSAGE);

  INTERFACE formformulario1 = new INTERFACE();

  formformulario1.setVisible(true);

} else {

  JOptionPane.showMessageDialog(null, "Acceso denegado:\n"

```

```
+ "Por favor ingrese un usuario y/o contraseña correctos", "Acceso denegado",
OptionPane.ERROR_MESSAGE);
```

```
    }
}

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Menu.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Menu().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JButton BtnIngresar;
private javax.swing.JButton BtnSalir;
private javax.swing.JLabel FONDO;
private javax.swing.JButton jButton1;
```



```

private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JTextField txtUsuario;
// End of variables declaration
}

```

Clase de Reconocimiento de Voz

```

import java.io.FileReader;
import java.util.Locale;
import javax.speech.Central;
import javax.speech.EngineModeDesc;
import javax.speech.recognition.*;

/**
 * Esta es la clase principal, la clase que se encarga de comprar los fonemas y
 * compararlos con el carchivo ("c:/SimpleGrammarES2.txt"), donde se establecen
 * las palabras que se utilizaran en el fonema.....
 *
 * @version 1.0.0.0 Educativa
 * @author Carlos Mario Montaña Lopera
 */
public class ReconocimientoVoz extends ResultAdapter implements Runnable {

    INTERFACE calculadora;
    /**
     * Permite manipular la calculadora desde esta clase, y ejecutar las
     * acciones segun el comando de voz*
     */
    Recognizer reconecedorVoz;
    /**
     * Reconoce el comado de voz del Usuario
     */
    String palabra;

    /**
     * Palabra que el usuario a pronunciado
     */
    public ReconocimientoVoz(INTERFACE calculadora) {
        this.calculadora = calculadora;
    }

    /**
     * Captura el fonema(palabra), y la compara con un conjunto de palabras y
     * ejecuta una accion establecida.
     *
     * Importante: yo e utilizado
     * <code>if</code>, por que mi version del jdk no permite la la utilizacion
     * de
     * <code>switch</code> con String

```

```

*
* @param re
*/
@Override
public void resultAccepted(ResultEvent re) {
    try {
        Result res = (Result) (re.getSource());
        ResultToken tokens[] = res.getBestTokens();
        for (int i = 0; i < tokens.length; i++) {
            palabra = tokens[i].getSpokenText();
            if (palabra.equals("izquierda")) {
                this.calculadora.escribirEnPantalla("1");
            }
            if (palabra.equals("derecha")) {
                this.calculadora.escribirEnPantalla("2");
            }
            if (palabra.equals("base")) {
                this.calculadora.escribirEnPantalla("3");
            }
            if (palabra.equals("brazo")) {
                this.calculadora.escribirEnPantalla("4");
            }
            if (palabra.equals("mano")) {
                this.calculadora.escribirEnPantalla("5");
            }
            if (palabra.equals("girar")) {
                this.calculadora.escribirEnPantalla("6");
            }
            if (palabra.equals("garra")) {
                this.calculadora.escribirEnPantalla("7");
            }
            if (palabra.equals("abrir")) {
                this.calculadora.escribirEnPantalla("8");
            }
            if (palabra.equals("cerrar")) {
                this.calculadora.escribirEnPantalla("9");
            }
            if (palabra.equals("secuencia")) {
                this.calculadora.escribirEnPantalla("10");
            }
            if (palabra.equals("uno")) {
                this.calculadora.escribirEnPantalla("11");
            }
            if (palabra.equals("dos")) {
                this.calculadora.escribirEnPantalla("22");
            }
            if (palabra.equals("tres")) {
                this.calculadora.escribirEnPantalla("33");
            }
            if (palabra.equals("parar")) {
                this.calculadora.escribirEnPantalla("13");
            }
            if (palabra.equals("reset")) {

```

```

        this.calculadora.escribirEnPantalla("21");
    }
    if (palabra.equals("robot")) {
        //borra pantalla
        this.calculadora.borrarTodo();
    }
}
} catch (Exception ex) {
    System.out.println("Error: " + ex.getMessage());
}
}

/**
 * Carga el archivo ("c:/SimpleGrammarES2.txt") y para poder empezar un
 * proceso de comparacion gramatical y hacer los procesos de
 * reconocimiento
 *
 * Importante: Se debe establecer un archivo ("c:/SimpleGrammarES2.txt"),
 * con un conjunto de expresiones gramaticales(Palabras) para poder generar
 * un proceso correcto de reconocimiento
 */
public void reconocerVoz() {
    try {
        reconocedorVoz = Central.createRecognizer(new
EngineModeDesc(Locale.ROOT));
        reconocedorVoz.allocate();
        FileReader gramatica = new FileReader("c:/SimpleGrammarES2.txt");
        RuleGrammar reglaGramatica = reconocedorVoz.loadJSGF(gramatica);
        reglaGramatica.setEnabled(true);
        reconocedorVoz.setResultListener(new ReconocimientoVoz(calculadora));
        reconocedorVoz.commitChanges();
        reconocedorVoz.requestFocus();
    } catch (Exception e) {
        System.out.println("Error en " + e.toString());
    }
}

@Override
public void run() {
    this.reconocerVoz();
}
}

```