

SISTEMA DE REHABILITACIÓN METACARPIANA MEDIANTE UN LABERINTO

METACARPAL REHABILITATION SYSTEM WITH A MAZE

Johan Ferney García Robayo; David Andrés Ortiz Sierra.

Resumen: *En este documento se describe la realización de un proyecto destinado a ayudar a las personas que requieran realizar ejercicios de extensión, flexión, abducción y aducción, o en general de estiramiento en la articulación metacarpiana, como personas con síndrome de túnel metacarpiano, distrofia muscular de Duchenne, enfermedad de Parkinson, entre otras. Para ello se realizan movimientos de la articulación metacarpiana que accionan un sistema de hardware, los movimientos se verán reflejados en el desplazamiento de una imagen para navegar en un laberinto diseñado virtualmente y visualizado en una interfaz de computador. La interfaz adicionalmente cuenta con una base de datos de las personas que utilizan el sistema, con almacenamiento de tiempos, distancias y fecha de cada sesión de entrenamiento para así lograr ver los progresos resolviendo el laberinto.*

Palabras clave: Sistema, interfaz, base de datos, acelerómetro, túnel metacarpiano.

Abstract: In this document the completion of a project to help people requiring exercises extension, flexion, abduction and adduction, or generally stretch in the metacarpal joint, such as people with carpal tunnel syndrome described, muscular dystrophy Duchenne, Parkinson's disease, among others. For this metacarpal joint movements that trigger a hardware system are made, the movements will be reflected

in the displacement of an image to navigate a maze designed virtually and displayed on a computer interface. The interface additionally has a database of people who use the system, with storage times, distances and date of each training session to see the progress achieve solving the maze.

Key Words: System interface, database, accelerometer, carpal tunnel.

1. INTRODUCCIÓN

En Colombia el síndrome de túnel metacarpiano es un padecimiento no tratado comúnmente ni en forma debida, ya que si no es ignorado se recomiendan ejercicios que el sujeto realiza de forma inadecuada porque no se tiene supervisión continua, son ejercicios monótonos y poco atractivos para realizar. En ocasiones se emplean inyecciones o cirugías para liberar el nervio [1]. La situación descrita se repite en pacientes que presentan distrofia muscular de Duchenne, enfermedad de Parkinson, y en general enfermedades que requieren ejercicios de rehabilitación física.

Por lo anterior, se propone la implementación de una herramienta interactiva para realizar los ejercicios mediante un sistema de rehabilitación muscular con interfaz de laberinto, el cual captura los movimientos de extensión, flexión, abducción y aducción en la articulación metacarpiana y permite navegar en un laberinto, también el almacenamiento en una base de datos de tiempo y distancia recorrida que emplea el usuario al resolverlo.

El sistema de rehabilitación muscular realizado es un sistema en donde el usuario y sujeto que requiere sesiones de entrenamiento físico en la articulación metacarpiana podrá acceder fácilmente a realizar ejercicios, ver sus avances y eliminar algunas sesiones en donde se crea que se hizo un ejercicio inadecuado para el tratamiento, con esta facilidad se abre el acceso a cualquier persona para realizar dicho tratamiento. El proyecto cuenta con una interfaz hecha en Python 2.7, una raspberry pi y una mbed para la parte de hardware las cuales muestran un laberinto sencillo de resolver pero con un recorrido largo para asegurar

que se realicen los ejercicios repetidamente.

2. MARCO TEÓRICO

2.1. MARCO CONCEPTUAL DE LA CIENCIA BIOLÓGICA

2.1.1. DISTROFIA MUSCULAR DE DUCHENNE: PERSPECTIVAS DESDE LA REHABILITACIÓN

La Distrofia Muscular de Duchenne (DMD) es una enfermedad hereditaria de tipo neuromuscular progresiva, que se presenta durante la infancia; afecta aproximadamente 1 de cada 3500 a 6000 niños varones nacidos vivos.

El gen DMD que codifica para la distrofia, una proteína estructural del músculo, se localiza en el brazo corto del cromosoma X, puede sufrir diferentes mutaciones ocasionando la pérdida parcial o total de la proteína lo que origina la DMD y otras enfermedades relacionadas como la Distrofia Muscular de Becker, que es una distrofia de forma más leve.

Desde el punto de vista de la rehabilitación, cuyo objetivo es retrasar la progresión de la enfermedad, se han venido desarrollado una serie de estrategias de tratamiento y manejo que involucran grupos multidisciplinarios; éstas van desde trabajo en gimnasio con terapeutas para estiramientos hasta prescripción de sillas de ruedas y formulación de ventilación mecánica no invasiva, que si bien se sabe no modifican las bases genéticas de la enfermedad, sí mejoran de forma importante la independencia del paciente, facilitan los cuidados por parte de la familia y los cuidadores y permiten una mejor calidad de vida.

La distrofia muscular de Duchenne como se dijo, es una enfermedad hereditaria con un patrón de herencia de tipo recesivo ligado al cromosoma X, por lo que se manifiesta en hombres y las mujeres sólo son transmisoras de la enfermedad, pero no presentan los síntomas. Al producirse la mutación, la célula muscular degenera, porque ya no hay contacto entre la matriz y la lámina basal de la célula. [2]

2.1.2. ENFERMEDAD DEL TÚNEL CARPIANO

El túnel carpiano es un canal o corredor en la cara anterior o flexora de la muñeca, situado entre los huesos de la muñeca y el ligamento anular del carpo, por el cual pasan los tendones flexores de los dedos y el nervio mediano.

El síndrome del túnel carpiano, también conocido como STC o neuropatía mediana de la muñeca, es una enfermedad que produce daño del nervio mediano. Si existe, por cualquier causa, un aumento de la presión dentro de ese túnel carpiano se puede producir una lesión en dicho nervio, lo que produce un fuerte dolor que puede ser tratado por ejercicios de flexión y extensión. Esta condición médica resulta en parestesias (hormigueo), debilidad muscular y entumecimiento en la mano. La mayoría de los tipos de STC son idiopáticos, es decir, de origen desconocido. Ciertos factores genéticos pueden poner a una persona en un cierto aumento del riesgo y el uso del brazo y otros factores también están siendo objeto de debate.

La mayoría de las personas que desarrollan el síndrome del túnel carpiano experimentan un aumento gradual de los síntomas con el tiempo. Los primeros síntomas a menudo ocurren cuando la persona está durmiendo y por lo general incluyen una sensación de hormigueo y sensación de ardor y entumecimiento en los dedos y las palmas de las manos. Los síntomas más a menudo se presentan en la noche debido a que una persona tiende a doblar la muñeca cuando se duerme, lo que conduce a una mayor compresión del túnel carpiano.

En las primeras etapas, el síndrome del túnel carpiano a menudo puede confundirse con problemas de circulación debido a la “alfileres y agujas” sensación de la sangre. Con un caso crónico del síndrome del túnel carpiano, puede haber un real desgaste de los músculos que conectan con el pulgar y puede ser difícil llevar el pulgar fuera de la mano.

Los casos leves pueden evolucionar favorablemente con tratamiento mediante inmovilización de la muñeca o con la inyección de corticoides. En casos más avanzados, la solución siempre es quirúrgica, liberando el nervio atrapado en el túnel del carpo [3]. En ambos casos posteriormente se realizan terapias de rehabilitación física.

EJERCICIOS DE REHABILITACIÓN

Estudios de investigación médica recientes indican que una persona que realiza ejercicios de rehabilitación puede reducir la necesidad de cirugía en más del 40%. Se han dispuesto una serie de ejercicios para ayudar en la rehabilitación del síndrome del túnel carpiano, algunos se enuncian a continuación.

EXTENSIÓN: Extender el brazo delante con la palma hacia arriba. Doblar la muñeca con la mano apuntando hacia el suelo. Con la otra mano, se dobla suavemente la muñeca hasta que sentir un estiramiento leve y moderado en el antebrazo. Mantener esta posición durante al menos 15 a 30 segundos. Repetir de 2 a 8 veces.

FLEXIÓN: Extender el brazo delante con la palma hacia abajo. Doblar la muñeca con la mano apuntando hacia el suelo. Con la otra mano, doblar suavemente la muñeca más hasta sentir un estiramiento leve y moderado en el antebrazo. Mantener esta posición durante al menos 15 a 30 segundos. Repetir de 2 a 8 veces [4].

ESTIRAMIENTO: Estirar ambas muñecas hacia adelante y relajar los dedos y mantener durante 5 segundos.

EMPUÑO: Hacer un puño congelado y girar hacia la izquierda, mantener pulsado durante 5 segundos. En posición puño congelado, doblar suavemente cada muñeca hacia abajo 5 veces en cada lado. Enderezar las muñecas y los dedos relajados, agitar suavemente las muñecas 5 veces en cada muñeca.

2.1.3. ENFERMEDAD DE PARKINSON

La enfermedad de Parkinson (EP) es un trastorno degenerativo del sistema nervioso central que pertenece a un grupo de afecciones conocidas como trastornos del movimiento. Es a la vez crónica, es decir, que persiste durante un extenso período de tiempo, y progresiva, lo que significa que sus síntomas empeoran con el tiempo. A medida que las células nerviosas (neuronas) en partes del cerebro se deterioran o mueren, se puede empezar a notar problemas con el movimiento, temblores, rigidez en las extremidades o en el tronco, o problemas de equilibrio. Al volverse estos síntomas más pronunciados, las personas pueden

tener dificultad para caminar, hablar o completar otras tareas sencillas. Si se tiene uno o más de estos síntomas, no necesariamente quiere decir que se tiene la enfermedad de Parkinson, ya que los síntomas aparecen también en otras enfermedades.

Se desconoce la causa exacta de la enfermedad de Parkinson, aunque algunos casos son hereditarios y se pueden deber a mutaciones genéticas específicas. Sin embargo, la mayoría de los casos son esporádicos, lo que quiere decir que la enfermedad generalmente no es hereditaria. Se cree que la enfermedad de Parkinson probablemente es el resultado de una combinación de susceptibilidad genética y exposición a uno o más factores ambientales desconocidos que desencadenan la enfermedad.

El fundamento de la terapia para la enfermedad de Parkinson es el medicamento levodopa (conocido también como L-dopa). Las células nerviosas utilizan la levodopa para fabricar dopamina y reponer el suministro reducido del cerebro. El tratamiento mediante L-dopa, que es muy eficaz en los primeros años, da lugar a la aparición de complicaciones, entre las que predominan las fluctuaciones del efecto y las descíñeseis (movimientos involuntarios), seguidas más tarde de un deterioro cognitivo. En paralelo, los signos axiales se acentúan y se vuelven poco sensibles a la L-dopa, tras lo que la enfermedad se enriquece con nuevos signos durante su evolución, en especial con trastornos no motores.

Los distintos tratamientos consiguen mitigar muchos de los problemas y síntomas inherentes y más típicos de la enfermedad como son el temblor, la rigidez en los músculos, la dificultad en los movimientos y en la coordinación. La rehabilitación se orienta teniendo en cuenta los síntomas motores señalados.

Ante la gran variedad de síntomas, la descripción de la terapia rehabilitadora obedece a la evolución de la enfermedad [5]. Sin embargo se han propuesto una serie de ejercicios generales de estiramiento para todo el cuerpo, por ejemplo para manos y dedos se propone:

1. Abrir y cerrar cada mano.
2. Extender y cerrar los dedos de cada mano.
3. Separar alternativamente los dedos de cada mano.

4. Rotar las manos sobre las muñecas.

5. Flexionar y extender la mano abierta sobre la muñeca.

2.2. MARCO INGENIERIL

2.2.1. ACELERÓMETRO MMA8451Q

Los acelerómetros o sensores de aceleración son dispositivos que miden aceleración o vibración, proporcionan una señal eléctrica según la variación física, en este caso la variación física es la aceleración o la vibración. Existen diferentes modelos dependiendo de la tecnología utilizada, se tienen acelerómetros piezoresistivos, acelerómetros piezoeléctricos y acelerómetros capacitivos.

Los rangos de medida son diversos, desde 1 g hasta los miles de g's. Respecto al rango de frecuencia disponible, hay acelerómetros que parten de 0 Hz, para medida de bajas frecuencias, acelerómetros que llegan hasta los miles de Hz para altas frecuencias de vibración, otros modelos de muy alta sensibilidad con bajo rango de frecuencia, etc.

El acelerómetro MMA8451Q es un circuito diseñado por la compañía Freescale que detecta el cambio de posicionamiento y envía datos tipo FLOAT (de 0 a 1) a una tarjeta mbed, esta comunicación está regida bajo el protocolo de comunicación I2C (SDA y SCL) en donde el acelerómetro MMA8451Q es el esclavo y el micro controlador de la mbed es el maestro. Su lenguaje de programación es C. [6]. Las características del sensor son:

- Voltaje de alimentación de 1.95V a 3.6V
- Voltaje lógico de 1.6V a 3.6V
- Selección de escala de trabajo $\pm 2g/\pm 4g/\pm 8g$
- Tasa de salida de datos (ODR) de 1.56 Hz a 800 Hz
- Salida digital de 8 y 14-bits
- Interfaz digital I2C (TWI)
- 2 interrupciones programables

- Filtro pasa altas disponible
- Ruido $99 \mu\text{g}/\sqrt{\text{Hz}}$

2.2.2 SQLite

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (275 kiB) biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

El autor de SQLite ofrece formación, contratos de soporte técnico y características adicionales como compresión y cifrado. [7]

2.2.3. PYTHON

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible [8].

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Python usa tipado dinámico y conteo de referencias para la administración de memoria.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa (también llamado enlace dinámico de métodos).

Otro objetivo del diseño del lenguaje es la facilidad de extensión. Se pueden escribir nuevos módulos fácilmente en C o C++. Python puede incluirse en aplicaciones que necesitan una interfaz programable.

Aunque la programación en Python podría considerarse en algunas situaciones hostiles a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia Lisp como puede ser Scheme [9].

2.2.4. PYGAME

Pygame es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos en dos dimensiones de una manera sencilla. Está orientado al manejo de sprites.

Gracias al lenguaje, se puede prototipar y desarrollar rápidamente. Esto se puede comprobar en las competiciones que se disputan online, donde es cada vez más usado. Los resultados pueden llegar a ser profesionales.

También puede utilizarse para crear otros programas multimedia o interfaces gráficas de

usuario. Pygame está basado en la librería SDL 1.2, una alternativa más actual de SDL en Python podría ser Py-SDL2, que implementa varias mejoras respecto a Pygame. Funciona como interfaz de las bibliotecas SDL. [10]

2.2.5. MBED FRDM KL25Z

El Freescale FRDM KL25Z (Figura 1), es una placa de desarrollo ultraligera y de bajo costo para la familia de MCU Kinetis L Serie KL1x (KL14 / 15) y KL2x (KL24 / 25) construida sobre el procesador ARM Cortex.

Entre sus características destaca el fácil acceso a MCU I / O, operación de bajo consumo de batería y un control de gestión integrado en la interfaz para depuración para la programación flash. El FRDM-KL25Z se apoya en una serie de Freescale y desarrollo de software de terceros, una opción muy práctica es utilizar mbed.org con pleno acceso al SDK en línea, herramientas de código reutilizable sin costo alguno, además de contar con una comunidad activa de desarrolladores [11].



Figura 1. Mbed FRDM KL25Z [12]

2.2.6. RASPBERRY Pi

La raspberry pi (Figura 2), es una tarjeta que contiene un computador en una sola placa

desarrollada por la fundación Raspberry Pi en el Reino Unido para facilitar el aprendizaje de países en desarrollo, esta tarjeta se fabrica bajo acuerdos de licencia con Newark Element14, RS Components y Egoman en donde comparten el hardware.

Raspberry Pi 2 modelo B fue lanzado en febrero de 2015, cuenta con una Broadcom y SOC (Sistema de Chip) que tienen compatibilidad con los CPU ARM y SoC (Sistema de procesamiento gráfico), tiene un rango de velocidad de 700MHz y 1,2 GHz, con una capacidad de memoria de 512MB y maneja como sistema operativo Linux (Raspbian), este sistema se almacena en una memoria SD, las placas tienen entre uno y cuatro puertos USB, un puerto HDMI y video compuesto de salida y una clavija de audio de 3,5 mm para audio, con varios puertos GPIO que manejan varios protocolos de comunicación como I2C, serial entre otros. [13]

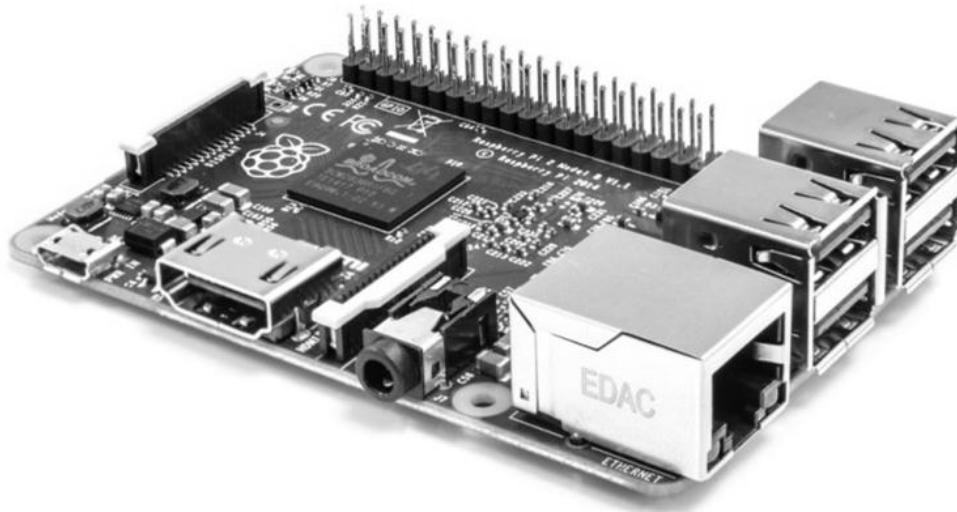


Figura 2. Raspberry Pi Modelo B+ [14]

3. DESARROLLO DEL PROYECTO

3.1. DIAGRAMA DE BLOQUES

En la figura 3 se aprecia el diagrama del sistema implementado para desarrollar el sistema capaz de brindar un medio para hacer ejercicios de rehabilitación. Se tienen cuatro bloques

generales: control del sistema, procesamiento, visualización y base de datos. Cada uno de ellos realiza una tarea específica de tal forma que la interconexión permitió lograr el dispositivo de rehabilitación cumpliendo los objetivos propuestos.

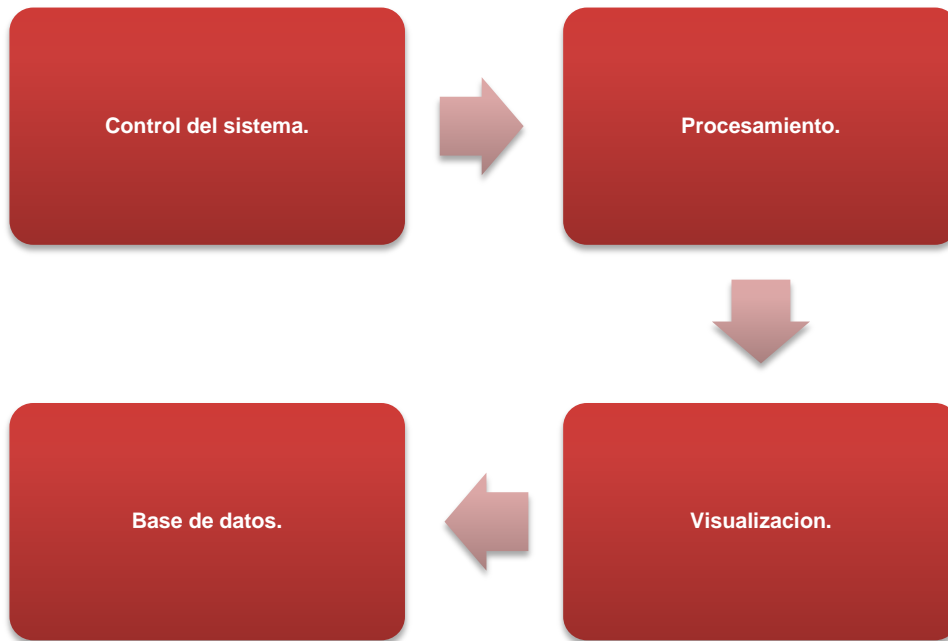


Figura 3. Diagrama del desarrollo del proyecto

3.1.1. CONTROL DEL SISTEMA

El control del sistema está principalmente compuesto por una tarjeta de desarrollo Mbed, la cual posee un acelerómetro que tomará el papel de control para el desarrollo del sistema de rehabilitación. Este acelerómetro envía pulsos los cuales dirán en cuál de los ejes del plano cartesiano se está produciendo el movimiento (X o Y). Para saber en cuál de los ejes se ubica se determina un rango definido de números el cual envía un carácter (“i” si es izquierda, “d” si es derecha, “a” si es arriba y “b” si es abajo) y así se identifica el sentido del movimiento. Gracias a la explicación y modificación del acelerómetro se logró llegar al código que se muestra en la Figura 4 para capturar dichos movimientos.

```
1 #include "mbed.h"
2 #include "MMA8451Q.h"
3
4 #if defined (TARGET_KL25Z) || defined (TARGET_KL46Z)
5   PinName const SDA = PTE25;
6   PinName const SCL = PTE24;
7 #elif defined (TARGET_KL05Z)
8   PinName const SDA = PTB4;
9   PinName const SCL = PTB3;
10 #elif defined (TARGET_K20D50M)
11   PinName const SDA = PTB1;
12   PinName const SCL = PTB0;
13 #else
14   #error TARGET NOT DEFINED
15 #endif
16
17 #define MMA8451_I2C_ADDRESS (0x1d<<1)
18
19 Serial pc(USBIX, USBRX);
20
21 int main(void)
22 {
23   MMA8451Q acc(SDA, SCL, MMA8451_I2C_ADDRESS);
24   PwmOut rled(LED1);
25   PwmOut gled(LED2);
26   PwmOut bled(LED3);
27   pc.baud(9600);
28   DigitalOut d(D0);
29   DigitalOut i(D1);
30   DigitalOut a(D2);
31   DigitalOut b(D3);
32
33   printf("MMA8451 ID: %d\n\r", acc.getWhoAmI());
34
35   while (true) {
36     float x, y, z;
37     x = acc.getAccX();
38     y = acc.getAccY();
39     z = acc.getAccZ();
40     rled = 1.0f - x;
41     gled = 1.0f - y;
42     bled = 1.0f - z;
43     wait(0.1);
44
45     if (y > 0.77 and y < 0.97){
46       i=1;
47       d=0;
48       a=0;
49       b=0;
50     }
51     if (x > -0.65 and x < -0.55){
52       i=0;
53       d=0;
54       a=0;
55       b=1;
56     }
57     if (x > 0.78 and x < 0.98){
58       i=0;
59       d=0;
60       a=1;
61       b=0;
62     }
63     if (y > -0.96 and y < -0.76 ){
64       i=0;
65       d=1;
66       a=0;
67       b=0;
68     }
69
70     printf("esperando \n\r");
71   }
72 }
```

Figura 4. Código Mbed para captura de movimientos

En la figura 5 se observan comandos implementados en la creación del juego laberinto.

```
pygame.display.set_caption("Aye! sir") : creacion de pantalla con el nombre de aye sir
pantalla = pygame.display.set_mode((1360,770)) : creacion de pantalla con especificaciones de pixeles
fuentel= pygame.font.SysFont("Arial", 30, True, False) : fuente para hacer algun texto
info=fuentel.render("Sistema de ", 1200, (255,0,0)) : definicion de texto y color con las especificaciones de fuentel
GPIO.setup(4,GPIO.IN) : inicializacion del GPIO para detectar algun pulso
pygame.mixer.music.load("/home/pi/Desktop/Tutorial 2/Fairy_Tail_Songs-Epic.mp3"): carga de la musica de fondo
pygame.mixer.music.play(3) : reproduccion de la musica de fondo y un parametro de que se repita 3 veces
pygame.mixer.music.fadeout(3000) : detencion de musica de fondo luego de 3 segundos
imagen1 = pygame.image.load("/home/pi/python_games/Hp4040.png") : carga de imagenes
colorect=(0,0,0) : creacion de colores en RGB (RED, GREEN, BLUE)
rectanguloa = pygame.Rect(0,0,1090,25) : Creacion de rectangulos como primer parametro coordenada y segundo largo y ancho
spritel= pygame.sprite.Sprite() : creacion de sprite
spritel.image=imagen1 : indicacion que dibuje el sprite debajo de la imagen1
spritel.rect=imagen1.get_rect() : obtencion de dimensiones de la imagen1
spritel.rect.top=25 : coordenada en 'y' para dibujarlo
spritel.rect.left=10 : coordenada en 'x' para dibujarlo
oldx=spritel.rect.left : almacenamiento de la coordenada anterior en 'x'
oldy=spritel.rect.top : almacenamiento de la coordenada anterior en 'y'
spritel.rect.move_ip(vx,vy) : movimiento de sprites con parametros de movimiento vx y vy
pantalla.blit(spritel.image,spritel.rect) : dibujo del sprite en pantalla
Tiempo = pygame.time.get_ticks()/1000 : obtencion de un cronometro
pantalla.blit(imagen2,(1050,700)) : dibujo de una imagen obteniendo como parametro la coordenada en donde se dibujara
pantalla.blit(info,(1110,120)): impresion en pantalla de algun texto con la informacion llamada info
pygame.draw.rect(pantalla,colorect2,rectanguloa) : dibujo de rectangulo en pantalla
GPIO.input(4) : indicacion de entrada de pulso en el GPIO 4
spritel.rect.colliderect(rectanguloa) : deteccion de colisiones entre el spritel y el rectanguloa
```

Figura 5. Algunos comandos básicos de pygame.

3.1.2. PROCESAMIENTO

La parte del procesamiento está a cargo de una tarjeta Raspberry Pi, la cual es el cerebro de todo el sistema, recibe los datos de la Mbed y los transforma en comandos de ordenamiento para un código que maneja toda la parte de software. Gracias a esto es posible darle un orden a cada cosa en nuestro sistema para así generar una interfaz gráfica agradable al usuario.

En la raspberry Pi B+ se realizó la interfaz con pygame, el procesamiento de pulsos enviados por la mbed y la visualización tanto del cursor como del laberinto. Para el procesamiento se reciben los datos provenientes de la mbed de manera digital en los GPIO de la raspberry y se procesan para dar el movimiento a la imagen cursor que se muestra en la interfaz.

3.1.3 VISUALIZACIÓN

De la Raspberry Pi se pasa a un monitor, donde se muestra el laberinto en la interfaz gráfica, el menú y la base de datos. Se puede interactuar con un menú que permite visualizar los

datos almacenados de sesiones previas, eliminar los datos menos importantes y acceder al laberinto para realizar el tratamiento.

Para el desarrollo de la interfaz del menú se utilizó la clase Menú de python y una serie de 'def' los cuales cumplen la tarea de separar todo el código del programa y poder ser llamados dependiendo de la selección del usuario, además inicializan las opciones y el cursor de selección. En el "def" llamado "actualizar" se determina e inicia una variable llamada 'k' donde se pregunta constantemente la tecla accionada y así conocer la selección de la opción a ejecutar. Se realiza una acción dependiendo de la opción seleccionada, también se llama a la opción asociada que está en selección para pintarla de un color diferente y por último se realiza una restricción para que no se salga del rango de las opciones permitidas.

El 'def' llamado "imprimir" determina la posición y la apariencia de cada opción en el menú, por último "main" determina que cosas se le van a colocar al menú, tales como fondo de pantalla, texto y música de fondo, en la figura 6 se puede apreciar la interfaz desarrollada para el menú.

En resumen, el menú da las opciones de:

1. **Jugar y realizar ejercicios:** Brinda acceso al juego del laberinto.
2. **Ver Datos:** Permite ver los registros almacenados en la base de datos.
3. **Eliminar Dato:** Permite borrar cualquier registro erróneo.
4. **Salir**



Figura 6. Interfaz del menú del juego.

En **Jugar y realizar ejercicios** se muestra el laberinto y se puede hacer uso de él. Esta parte de visualización es recrear y utilizar pygame de manera adecuada, para esto inicialmente se aprendió y se hizo una pantalla aparte con el nombre de JUEGO LABERINTO en donde se visualizan todos los movimientos, rectángulos y datos que se vayan tomando a lo largo del juego. Luego de tener la pantalla o el espacio donde se realizarán los ejercicios, se colocó una imagen de fondo de piedras las cuales reflejan un realismo de desierto, luego se colocó alrededor de 55 rectángulos que componen el laberinto y cumplen el papel de obstáculos para realizar el tratamiento, después se procedió a colocar la imagen-cursor y la imagen-meta que consta de un gato azul (imagen-cursor) tratando de llegar a su novia (imagen-meta).

Para la parte de colisiones con los rectángulos se utilizó Sprites los cuales básicamente dibujan un rectángulo que sigue la imagen y así poder utilizar la orden colliderect (colisión entre rectángulos) en donde se guardan dos variables llamadas oldx y oldy y allí se almacena

la posición anterior para cuando se detecte la colisión vuelva a su posición anterior. Luego se procedió a la captura del primer dato (Tiempo) en el cual se utilizó la librería time y se implementó un contador que se inicia cuando la línea init se ejecute, para capturar el segundo dato (distancia) lo que se uso fue un contador en donde a través de la medida de un monitor de resolución (1366x768) se calcula la distancia teniendo en cuenta la posición inicial de la imagen-cursor y el primer obstáculo en centímetros y pixeles, para este caso es de 8.4 cm y 320 pixeles, también se tiene en cuenta la velocidad de 20 pixeles*pulso que se le ha asignado, lo que da un total de 0.56cm por pulso detectado. Cuando se detecta colisión se genera un pulso que no afecta la distancia. Se adiciono texto en la pantalla de interfaz de laberinto para dar a conocer el contador de la distancia y del tiempo, para cuando se llegue a la imagen-meta y por último se le agrego música de fondo. Con todo esto la interfaz del juego laberinto quedo como lo muestra la figura 7.



Figura 7. Interfaz final del laberinto.

La opción **Ver Datos** permite acceder a la tabla creada por el Sqlite3, esta se crea cada vez que un usuario completa el laberinto, esta tabla permite ver código, tiempo, distancia, fecha e

identificación del usuario.

La opción **Eliminar dato** permite borrar alguno de los registros guardados en la tabla generada. Adelante se describe detalladamente la implementación y funcionamiento de las opciones Ver Datos y Eliminar Datos.

La opción **Salir** permite cerrar el programa sin haber iniciado la sesión.

3.1.4 BASE DE DATOS

Una de las partes fundamentales del sistema es almacenar los datos del usuario y su desempeño en el juego, para ello se implementó un software conocido como sqlite3. Este software fue instalado en la Raspberry pi b+ a través de internet mediante un cable Ethernet, directamente al router y en la terminal de la Raspberry digitando el comando: Sudo apt get-install sqlite3

Con sqlite3 ya instalado se procedió a realizar la base de datos, primero creando un archivo llamado base1 con la extensión 'sqlite3' luego creando el programa en la plataforma python 2.7 llamado crear-tabla.py y empleando el código de la figura 8.

```
import sqlite3

conexion=sqlite3.connect("/home/pi/Desktop/prueba_juego_completol/base1.sqlite3")

consulta=conexion.cursor()

sql="""
CREATE TABLE IF NOT EXISTS test(
Sesion INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
ident VARCHAR(50) NOT NULL,
Tiempo INTEGER NOT NULL,
distancia FLOAT NOT NULL,
fecha DATE NOT NULL)"""

if (consulta.execute(sql)):
    print("Tabla creada con exito. ")
else:
    print("Ha ocurrido un error. ")

consulta.close()

conexion.commit()

conexion.close()
```

Figura 8. Código creación de tabla en sqlite3.

La lógica y la sintaxis se basa en establecer conexión creando unas variables llamadas conexión y consulta, para así conectar con el archivo base1.sqlite3 y crear la tabla llamada test, almacenando los datos allí y especificándole al programa que columnas que se utilizaron.

TIPOS DE VARIABLES QUE SE UTILIZARON EN EL SISTEMA

Las variables utilizadas en la programación son de diferentes tipos, según el tipo de los datos a tratar, a continuación se muestran los tipos utilizados:

- Sesión llave primaria incrementable de enteros: Tipo de dato INTEGER en donde se incrementa y se almacena cada dato, todos los otros datos giran en entorno a esta variable
- Identificación dato varchar de máximo 50 caracteres numéricos: Tipo de dato para cualquier tipo de número con un rango de caracteres de 50.
- Tiempo: dato integer ya que solo se maneja segundos exactos
- Distancia dato float (coma flotante): el cual permite capturar la distancia lo más exacta posible
- Fecha: dato datetime donde este posee su propia librería para importar especifica la fecha en la cual se realizó la sesión.

Las columnas y datos que componen la base de datos se anexan al programa principal (donde está el juego completo), haciendo uso de líneas de código pertinentes, tales como el establecer la conexión con la extensión connect, consulta.

Cuando se soluciona el laberinto y se llega a la meta con la instrucción INTO IN test (ident, Tiempo, distancia, datetime ()) los valores más recientes de identificación, tiempo, distancia y fecha son almacenados dentro de la tabla test.

Para ver y eliminar datos de la base de datos nuevamente se realiza la conexión con base1.sqlite3 con las variables consulta y conexión. Para cargar todos los datos se crea un vector en donde se asignan los datos almacenados y se imprimen posición a posición.

3.1.5 REQUERIMIENTOS ADICIONALES

Los códigos usados en la programación de las diferentes tarjetas para realizar el sistema están basados en los lenguajes de programación Python y C. Las figuras son capturas de pantallas del entorno gráfico.

En esta sección se presentan las instrucciones de edición para las figuras, tablas, abreviaturas y acrónimos.

Cada tarjeta de programación tiene asignado un lenguaje de programación, tabla 1.

Tarjeta	Lenguaje
Raspberry Pi	Python
Mbed	C

Tabla 1. Identificación de los lenguajes en las tarjetas. **Fuente:** Elaboración propia de los autores.

4. RESULTADOS

Para la prueba inicial del funcionamiento del sistema se realizó la prueba con 5 personas en las cuales se registraron datos similares aún con la persona que sufre túnel metacarpiano, esto es debido a que esta persona encontró la solución al laberinto de manera más rápida. En la resolución del laberinto influyen aspectos como la agilidad mental del usuario y la rapidez con la que encuentre el camino correcto para resolverlo. Los datos ubicados en el

número 5 de la base de datos corresponden a la persona que sufre del túnel carpiano, sus resultados fueron similares a los de las personas sanas ya que ella encontró con mucha más facilidad el camino correcto a la meta, en la tabla 2 se muestran el tiempo y la distancia que se obtuvieron en la primera prueba del sistema, prueba que fue realizada para ver el comportamiento del mismo. Esta tabla muestra el código, el tiempo y la distancia recorrida.

Al igual que en las pruebas anteriores influyen varios factores que determinaron los resultados, ya que se debía iniciar el tiempo acorde al del sistema, la agilidad mental para resolver el laberinto jugaba un papel importante sin contar la correcta posición de la Mbed al momento de ejecutar el juego.

Código	Tiempo (min) Programa	Distancia (cm)
1	5.9	145.8
2	8.2	140.4
3	8.8	148.2
4	12.9	208.2
5	11.4	168.6

Tabla 2. Comparación de tiempos. **Fuente:** Elaboración propia de los autores

Las pruebas siguientes se hicieron con tres personas, una de ellas con la enfermedad del túnel del carpio (sujeto con el código 6, en la base de datos correspondiente al número de cedula 1026579088) (figura 13), se tomaron los tiempos que arrojaban el programa y el cronometro digital de un celular para así saber el porcentaje de error que tiene el sistema de apoyo metacarpiano.

La tabla 3 muestra los valores capturados, comparados y el porcentaje de error en cada una de las pruebas además del porcentaje promedio del sistema.

La visualización de la tabla que muestra los usuarios del sistema se realiza a través de la consola del programa (Figura 9), donde se muestran las variables obtenidas en cada una de las sesiones realizadas por cada uno de los pacientes. Estas variables son Tiempo, Identificación, Código, Distancia y fecha.

Cedula	Tiempo del sistema	Tiempo cronometro	% de error
1022408500	415 seg	414 seg	0.24%
1026579088	1119 seg	1116 seg	0.26%
1018470573	772 seg	768 seg	0,51%
Promedio			0.33%

Tabla 3. Datos para obtener el porcentaje de error. **Fuente:** Elaboración propia de los autores.

Se realizaron cinco sesiones realizadas se tomaron los datos de una persona que sufre del túnel carpiano y cuatro personas sanas en este sentido, para ello se tomaron datos en tiempo real y se compararon con los tiempos que arrojaba el sistema de rehabilitación. Se pueden ver estos resultados en la figura 9.

Como se puede ver en la figura 9 las personas con dicho padecimiento registraron los mayores tiempos, estas personas se les dificultó más el movimiento del dispositivo y por ende los pulsos del acelerómetro no tenían una emisión correcta, también influyen los caminos que tomaron para solucionar el laberinto pero en estas pruebas las personas conocían con anterioridad la solución al laberinto y de esta manera se puede tener en cuenta los tiempos que tomaba cada persona al solucionar el laberinto.

El sistema presenta un margen de error del 0.33% como se podrá ver en el cálculo que se realizara más adelante, esto se debe a la capacidad del procesador mientras procesa los datos una vez terminado el tratamiento.

```
Python 2.7.9 (default, Mar 8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
Warning (from warnings module):
  File "/home/pi/Desktop/prueba_juego_completol/menuysistema2.py", line 65
    def comenzar_nuevo_juego():
SyntaxWarning: import * only allowed at module level
>>> ===== RESTART =====
>>>
Selecciona la opción 'Ver Datos.'.
Función que muestra otro menú de opciones.
Estas en la opcion ver Datos!
```

Codigo	identificacion	Tiempo (seg)	Distancia (cm)	Fecha
1	1018470573	356	145.8	2016-04-04
2	1022408500	497	140.4	2016-04-04
3	52125620	532	148.2	2016-04-04
4	99111900228	775	208.2	2016-04-04
5	51751682	685	168.6	2016-04-04

```
>>>
```

Figura 9. Tabla de base de datos

Tomando el dato de tiempo a una nueva persona con síndrome de túnel metacarpiano y a dos personas que ya habían realizado la prueba con cronometro se pretende demostrar el error que tiene el sistema. Los datos usados para el cálculo del error son de los sujetos 6, 7 y 8 de la tabla que se muestra en la figura 10. Los tiempos cronometrados para cada persona se muestran en la tabla (tabla de datos cronometrados)...

```
Estas en la opcion ver Datos!
```

Codigo	identificacion	Tiempo (seg)	Distancia (cm)	Fecha
1	1018470573	356	145.8	2016-04-04
2	1022408500	497	140.4	2016-04-04
3	52125620	532	148.2	2016-04-04
4	99111900228	775	208.2	2016-04-04
5	51751682	685	168.6	2016-04-04
6	1026579088	1119	226.8	2016-04-05
7	1018470573	772	178.2	2016-04-05
8	1022408500	415	114.0	2016-04-05

```
>>>
```

Figura 10. Tabla de datos para encontrar el porcentaje de error.

El tiempo de la persona identificada con el documento 1018470573 fue de 12:48 (Ver figura 11) que en segundos equivale a 768 segundos y el tiempo cronometrado en la base de

datos correspondiente al código 7 fue de 772 segundos ahora se procederá a realizar el porcentaje de error (Ecuación 1) para el sujeto 1018470573.

$$\%Error = \frac{|772 \text{ segundos} - 768 \text{ segundos}|}{772 \text{ segundos}} = 0.51 \%$$

(1)

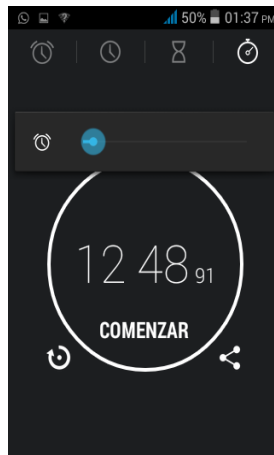


Figura 11. Tiempo de 1018470573

El tiempo de 1022408500 fue de 06:54 (Ver figura 12) que en segundos equivale a 414 segundos y el tiempo cronometrado en la base de datos correspondiente al código 8 fue de 415 segundos ahora se procederá a realizar el porcentaje de error (Ecuación 2) para el sujeto 1022408500.

$$\%Error = \frac{|415 \text{ segundos} - 414 \text{ segundos}|}{415 \text{ segundos}} = 0.24 \%$$

(2)

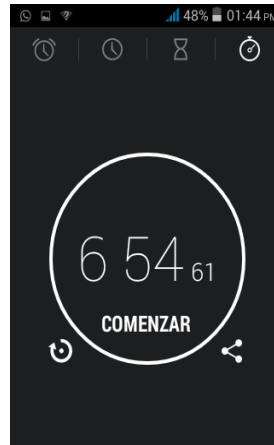


Figura 12. Tiempo de 1022408500

El tiempo de 1026579088 fue de 18:36 (Ver figura 13) que en segundos equivale a 1116 segundos y el tiempo cronometrado en la base de datos correspondiente al código 6 fue de 1119 segundos ahora se procederá a realizar el porcentaje de error (Ecuación 3) para el sujeto 1026579088.

$$\%Error = \frac{|1119 \text{ segundos} - 1116 \text{ segundos}|}{1119 \text{ segundos}} = 0.26 \%$$

(3)

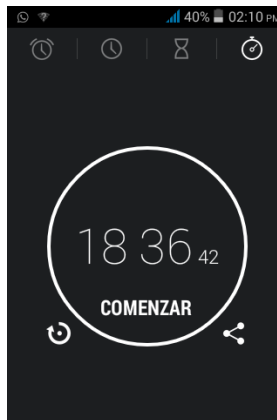


Figura 13. Tiempo de 1026579088

El porcentaje de error total del sistema se obtuvo realizando el promedio de los errores calculados (Ecuación 4).

$$0.51\% + 0.24\% + 0.26\% = 1.01809\% / 3 = 0.33\%$$

(4)

Luego de varias sesiones y un uso correcto del sistema, con los ejercicios se pretende que se ayudará para la rehabilitación los nervios del Carpio dando así una mayor tolerancia a los dolores presentados y a que los pacientes conlleven un estilo de vida mejor en cuanto a esta enfermedad se trata.

5 CONCLUSIONES

- Se logró un dispositivo electrónico e interactivo capaz de mostrar una serie de ejercicios para que el sujeto con síndrome de túnel metacarpiano pueda realizar ejercicios de manera interactiva mientras mejora y libera el nervio del Carpio.
- A pesar de no tener síndrome de túnel metacarpiano existen sujetos propensos a padecerlo en lo cual se demuestra en la tabla de resultados de la base de datos ya que obtuvieron tiempos y distancias prolongados.
- Con relación a los tiempos y distancias tomados por el sistema se tiene un rango de error del 0,33%, bajo esto se debe a la detección y no detección de pulsos en ciertas áreas del juego.
- La raspberry Pi b+ brinda bastantes ventajas para realizar este tipo de proyectos porque cuenta con el software adecuado para tratar datos, interfaz, menú, cálculos extensos entre otros.
- Pygame es un paquete muy completo para el desarrollo de videojuegos ya que brinda las herramientas necesarias y completas con lenguaje de programación muy sencillo para crear diversos videojuegos para cualquier finalidad.

6 REFERENCIAS

- [1] J. Domínguez, “Para que las manos no duelan”. Grupo Sura [Online], 15 de abril del 2015, disponible en: <http://www.sura.com/blogs/calidad-de-vida/tunel-carpiano.aspx>
- [2] Kliegman RM, Stanton BF, St. Geme J, Schor N, Behrman, “Distrofia Muscular de Duchenne”, [Online], Colombia, MedlinePlus, 2014, Disponible en: <http://www.nlm.nih.gov/medlineplus/spanish/ency/article/000705.htm>
- [3] Santiago Amillo Garayoa, “Diagnóstico y tratamiento del síndrome del túnel carpiano en la clínica”. Disponible en <http://www.cun.es/enfermedades-tratamientos/enfermedades/sindrome-tunel-carpiano>
- [4] comologia, “Ejercicios para mejorar el síndrome del túnel carpiano”, comologia [Online], 2015, Disponible en: <http://www.comologia.com/sindrome-del-tunel-carpiano-ejercicios-para-mejorar/>
- [5] Bleton, J.-P., Ziegler, M. “Rehabilitación de la enfermedad de Parkinson” [Online], Colombia, Universidad Distrital, Facultad Tecnológica, 2012, Disponible en: [http://tcna.primo.hosted.exlibrisgroup.com/primolibrary/libre/action/display.do?tabs=detailsTab&ct=display&fn=search&doc=TN_sciversesciencedirect_elsevierS1293-2965\(12\)60864-0&indx=3&reclDs=TN_sciversesciencedirect_elsevierS1293-2965\(12\)60864-0&reclDxs=2&elementId=2&renderMode=poppedOut&display](http://tcna.primo.hosted.exlibrisgroup.com/primolibrary/libre/action/display.do?tabs=detailsTab&ct=display&fn=search&doc=TN_sciversesciencedirect_elsevierS1293-2965(12)60864-0&indx=3&reclDs=TN_sciversesciencedirect_elsevierS1293-2965(12)60864-0&reclDxs=2&elementId=2&renderMode=poppedOut&display)
- [6] R. S. Linares, A. M. Arenas, Tutorial Mbed #9 Acelerómetro MMA8451Q [Online], YouTube, 16 de marzo 2015, disponible en: https://www.youtube.com/watch?v=srpgVnp_xmc
- [7] Wikipedia, “SQLite”, Wikipedia [Online], 2 de febrero de 2016, disponible en: <https://es.wikipedia.org/wiki/SQLite>
- [8] L. A. Luengas, Laboratorio virtual de destilación apoyado en un sistema de captura de movimiento - Universidad Distrital Francisco José de Caldas. Bogotá: Editorial UD,

2013.

- [9] Wikipedia, “Python”, Wikipedia[online], 18 de febrero de 2016, disponible en:
<https://es.wikipedia.org/wiki/Python>
- [10] Wikipedia, “Pygame”, Wikipedia[Online], 9 de diciembre de 2015, disponible en:
<https://es.wikipedia.org/wiki/Pygame>
- [11] Miguel Alonso, “Freescale Freedom Development Platform for Kinetis”, 2015,
Disponible en: <https://alonsodub.wordpress.com/category/frdm-kl25z/>
- [12] SwitchScience, “Freescale FRDM-KL25Z”, SwitchScience [Online], disponible en:
<https://www.switch-science.com/catalog/1287/>
- [13] Wikipedia, “Raspberry Pi”, Wikipedia[Online], 20 de abril de 2016, disponible en:
https://en.wikipedia.org/wiki/Raspberry_Pi
- [14] Raspberry pi 2, blog, disponible en: <https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale/>