

ESCUELA POLITÉCNICA NACIONAL

INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROCESAMIENTO DIGITAL DE SEÑALES

ALUMNOS: Contreras Marialejandra

Dávila Melanny

SEMESTRE: 2020B

PARALELO: GR1

FECHA: 04 de enero de 2021

PROFESOR: DR. ROBIN ÁLVAREZ

TEMA: Proyecto Bimestral 1

SUBTEMA: Deber Práctico

Contenido

[MARCO TEORICO 3](#_Toc60818216)

[OBJETIVOS 3](#_Toc60818217)

[SENSOR DE FUERZA O PRESION MF01 3](#_Toc60818218)

[ARDUINO UNO 3](#_Toc60818219)

[ELECTROMIOGRAMA 4](#_Toc60818220)

[MATERIALES 4](#_Toc60818221)

[CONEXIÓN 5](#_Toc60818222)

[CODIGO ARDUINO 6](#_Toc60818223)

[DISEÑO INTERFAZ GRAFICA 6](#_Toc60818224)

[INTERFAZ DE INICIO 6](#_Toc60818225)

[INTERFAZ DE INICIO – CODIGO 6](#_Toc60818226)

[INTERFAZ DE PRESENTACION DE SEÑALES 8](#_Toc60818227)

[INTERFAZ DE GRAFICA-CODIGO 9](#_Toc60818228)

[RESULTADOS OBTENIDOS 12](#_Toc60818229)

[CONCLUSIONES 12](#_Toc60818230)

[IMPLEMENTACION PRACTICA 13](#_Toc60818231)

[LINK VIDEO DEMOSTRATIVO 14](#_Toc60818232)

[REFERENCIAS 14](#_Toc60818233)

## MARCO TEORICO

### OBJETIVOS

* Diseñar e implementar un circuito sencillo que funcione como un electromiograma.
* Conocer a profundidad la utilidad y el funcionamiento del sensor MF01.
* Consolidar los conocimientos aprendidos en clases acerca del procesamiento de señales mediante el uso de la placa Arduino Uno.
* Realizar la creación de una interfaz gráfica que permita observar las señales capturadas por medio de la diferencia de tensión medida en el circuito.

### SENSOR DE FUERZA O PRESION MF01

El sensor de fuerza o presión MF01 autoadherible para Arduino es ideal para detectar una fuerza aplicada en la membrana. Básicamente, al detectar una flexión en la membrana el sensor cambia su resistencia interna, por lo que si se hace uso de un divisor de voltaje se puede medir la salida de voltaje.

Está compuesto por dos capas, separadas entre sí por un espaciador. Al momento de aplicar presión al dispositivo, existen más puntos de elemento activo que se juntan al semiconductor, lo que ocasiona que su resistencia disminuya.



Fig. 1 Sensor de presión MF01

##### *VENTAJAS DEL SENSOR*

* Este sensor de fuerza es de bajo costo, funciona con cualquier tarjeta de desarrollo o microcontrolador con etapa de ADC.
* Tiene una extensión que termina en dos puntas que se pueden soldar cómodamente.
* Es muy sencillo de utilizar y puede tener multitud de aplicaciones en muchos proyectos electrónicos.
* Sirve para la mayoría de las aplicaciones sensibles al tacto.
* Puede utilizar cualquier fuente de alimentación ya que utiliza menos de 1 mA de corriente.

#### DESVENTAJAS DEL SENSOR

Estos sensores son raramente precisos. Así que, básicamente, cuando se utiliza este sensor sólo se debe esperar para obtener “rangos” de respuesta. Tal sensor de fuerza puede detectar el peso, pero son una mala elección para detectar exactamente cuántas libras de peso hay en ellos.

### ARDUINO UNO

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas.

Incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteado. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

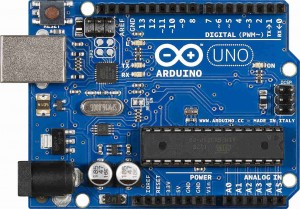


Fig. 2 Arduino UNO

### ELECTROMIOGRAMA

El electromiograma se usa para estudiar el sistema nervioso periférico y los músculos que inerva, y permite diagnosticar enfermedades neuromusculares, así como determinar su intensidad y origen. La prueba consiste esencialmente en registrar mediante electrodos especiales las corrientes eléctricas que se forman en los nervios y músculos al producirse contracciones.

Con esta prueba se puede identificar si las alteraciones neuromusculares se deben al músculo o a las fibras nerviosas. Para ello analiza básicamente:

* La amplitud de las corrientes eléctricas.
* El número de fibras musculares que se contraen.
* El tiempo que tardan en contraerse.
* El tiempo que se mantienen contraídas

## MATERIALES

Para el desarrollo del siguiente proyecto que permitirá la implementación de un electromiograma con el fin de adquirir señales al momento de cerrar o abrir la mano, se necesitan los siguientes materiales:

* Arduino UNO
* Sensor MF01
* Protoboard
* Resistencias
* Cables para protoboard
* Matlab
* IDE de Arduino

## CONEXIÓN

A continuación, se presenta el circuito esquemático el cual será utilizado para la adquisición de señales de la palma de la mano de una persona mediante el uso de la placa de Arduino y el sensor MF01.

Tal y como se presenta en la figura 3, se hace uso de los materiales mencionados anteriormente, sobretodo el sensor y la resistencia con el fin de medir la diferencia de voltaje entre los terminales del dispositivo y de esta manera capturar las señales de datos mediante el uso de la placa.

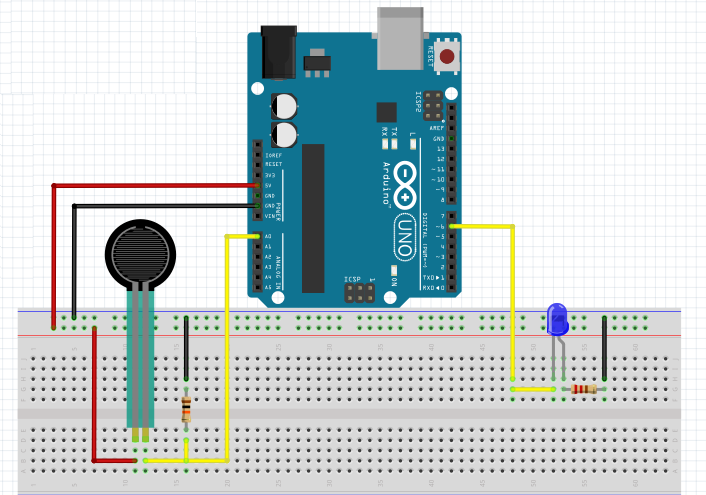


Fig. 3 Circuito esquemático

El código que se presenta a continuación, es el que será utilizado en la placa de Arduino cuyo objetivo es capturar las señales de datos para eso se hace uso de una resistencia antes mencionada con el fin de capturar la diferencia de tensión durante un cierto intervalo de tiempo, que en este caso es de 100 mili segundos.

### CODIGO ARDUINO

int AnalogPin = 0; // Sensor conectado a Analog 0

int ResRead; // La Lectura de la Resistencia por División de Tensión

void setup()

{

Serial.begin(9600); // Enviaremos la información de depuración a través del Monitor de Serial

}

void loop()

{

ResRead = analogRead(AnalogPin); // La Resistencia es igual a la lectura del sensor (Analog 0)

Serial.print("Lectura Analogica = ");

Serial.println(ResRead);

delay(100); //Cien “ms” de espera en cada lectura

}

Código 1. Script utilizado en el IDE de Arduino

## DISEÑO INTERFAZ GRAFICA

### INTERFAZ DE INICIO

Como primera parte de la interfaz gráfica del proyecto se diseñó una interfaz de inicio la cual se presenta en la figura 4 y su código se describe a continuación. Para proceder con la captura de las señales de datos se debe presionar el botón “Siguiente” mostrado en la figura 4.

### INTERFAZ DE INICIO – CODIGO

function varargout = inicio(varargin)

% INICIO MATLAB code for inicio.fig

% INICIO, by itself, creates a new INICIO or raises the existing

% singleton\*.

% Last Modified by GUIDE v2.5 27-Dec-2020 15:53:01

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @inicio\_OpeningFcn, ...

'gui\_OutputFcn', @inicio\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% UIWAIT makes inicio wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = inicio\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

handles.pushbutton1;

close inicio

Proyecto\_1\_PDS

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before inicio is made visible.

function inicio\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to inicio (see VARARGIN)

% Choose default command line output for inicio

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes inicio wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = inicio\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

handles.pushbutton1;

close inicio

Proyecto\_1\_PDS

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

handles.pushbutton1;

close inicio %cierre de la caratula

Proyecto\_1\_PDS %inicio de la pestana para leer señal del sensor

Código 2. Script que permite la creación de la primera interfaz gráfica



Fig. 4 Interfaz de inicio

### INTERFAZ DE PRESENTACION DE SEÑALES

La siguiente ventana que se desplegará será la interfaz mostrada en la figura 5, donde una vez que se presione el botón ‘’Iniciar’’, se comenzará a capturar datos. Mientras que al presionar el botón ‘’Salir’’ el programa desplegará una ventana donde se debe confirmar que se sea salir, la misma que se muestra en la figura 6.

INTERFAZ DE GRAFICA-CODIGO

function varargout = Proyecto\_1\_PDS(varargin)

% PROYECTO\_1\_PDS MATLAB code for Proyecto\_1\_PDS.fig

% PROYECTO\_1\_PDS, by itself, creates a new PROYECTO\_1\_PDS or raises the existing

% singleton\*.

% Edit the above text to modify the response to help Proyecto\_1\_PDS

% Last Modified by GUIDE v2.5 27-Dec-2020 15:55:16

% Begin initialization code - DO NOT EDIT

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @Proyecto\_1\_PDS\_OpeningFcn, ...

'gui\_OutputFcn', @Proyecto\_1\_PDS\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% End initialization code - DO NOT EDIT

% --- Executes just before Proyecto\_1\_PDS is made visible.

function Proyecto\_1\_PDS\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to Proyecto\_1\_PDS (see VARARGIN)

% Creacion de Variables Globales

global abierta

global punio

global mano\_abierta

global mano\_cerrada

% Choose default command line output for Proyecto\_1\_PDS

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes Proyecto\_1\_PDS wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = Proyecto\_1\_PDS\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global abierta

global punio

global mano\_cerrada

global mano\_abierta

senal=zeros(1,1000);

A=arduino('COM6','Uno')

inicio=true

while inicio

senal=zeros(1,200);

for numero=1:200

acele=readVoltage(A,'A0');

senal(numero)=acele;

axes(handles.axes1)

plot(senal)

drawnow

if ~inicio

break

end

end

end

clear A

% --- Executes on button press in btn\_salir.

function btn\_salir\_Callback(hObject, eventdata, handles)

% hObject handle to btn\_salir (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

opc=questdlg('Desea Salir?','SALIR','SI','NO','NO');

if strcmp(opc,'NO')

return

end

clear,clc,close all;

clc

close all

clear all

senal=zeros(1,1000);

a=arduino('COM10','Uno');

figure

for i=1:5

senal=zeros(1,100);

for numero=1:500

emg=readVoltage(a,'A0')

senal(numero)=emg;

plot(senal)

title('Electromiograma')

xlabel('Tiempo')

ylabel('señal')

drawnow

end

end

clear a

global punio

global mano\_abierta

global mano\_cerrada

% Choose default command line output for Proyecto\_1\_PDS

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% UIWAIT makes Proyecto\_1\_PDS wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = Proyecto\_1\_PDS\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Creacion de variables globales

global abierta

global punio

global mano\_cerrada

global mano\_abierta

senal=zeros(1,1000); %creacion de vector de ceros

A=arduino('COM6','Uno') %lectura de datos de COM6

inicio=true

while inicio %lazo while que permite recorrer datos

senal=zeros(1,200);

for numero=1:200 %lazo for del vector de datos

acele=readVoltage(A,'A0');

senal(numero)=acele;

axes(handles.axes1)

plot(senal) %grafica de la señal

drawnow

if ~inicio

break

end

end

end

clear A

% --- Executes on button press in btn\_salir.

function btn\_salir\_Callback(hObject, eventdata, handles)

% hObject handle to btn\_salir (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

opc=questdlg('Desea Salir?','SALIR','SI','NO','NO');

if strcmp(opc,'NO')

return

end

clear,clc,close all;

clc

close all

clear all

senal=zeros(1,1000);

a=arduino('COM10','Uno');

figure

for i=1:5

senal=zeros(1,100);

for numero=1:500

emg=readVoltage(a,'A0')

senal(numero)=emg;

plot(senal)

title('Electromiograma')

xlabel('Tiempo')

ylabel('señal')

drawnow

end

end

clear a

drawnow

if ~inicio

break

end

end

end

clear A % Limpieza de la variable A

% --- Executes on button press in btn\_salir.

function btn\_salir\_Callback(hObject, eventdata, handles)

% hObject handle to btn\_salir (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

opc=questdlg('Desea Salir?','SALIR','SI','NO','NO'); %Salida de la interfaz

if strcmp(opc,'NO') %Si no se desea salir se regresa a la segunda interfaz

return

end

clear,clc,close all;

clc

close all

clear all

senal=zeros(1,1000); %creacion de un vector de ceros

a=arduino('COM10','Uno'); %lectura de los datos del Puerto COM10 de Arduino

figure

for i=1:5 %lazo for que recorre los datos

senal=zeros(1,100);

for numero=1:500 %lazo for que recorre los vectores de datos

emg=readVoltage(a,'A0') %lectura de datos

senal(numero)=emg;

plot(senal) % Grafica de la señal de datos

title('Electromiograma')

xlabel('Tiempo')

ylabel('señal')

drawnow

end

end

clear a % Limpieza de la variable a

Código 3. Script que permite la creación de la segunda interfaz gráfica

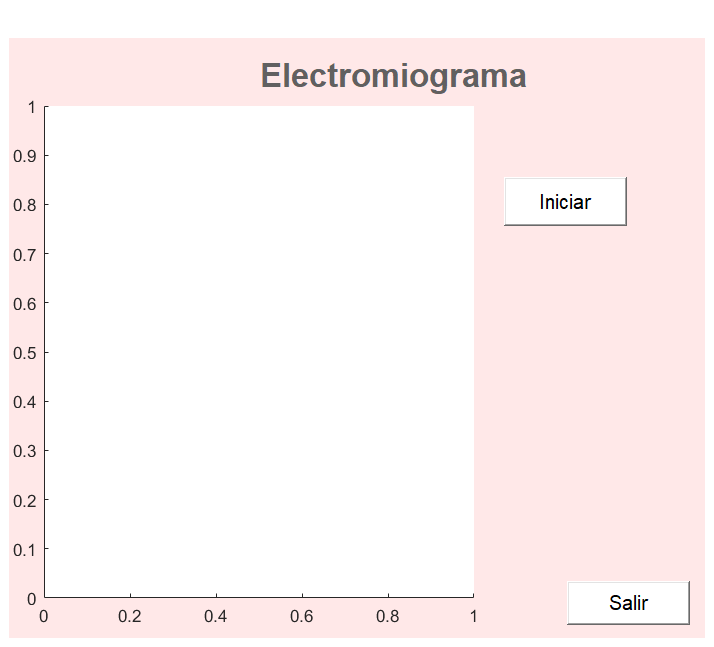


Fig. 5 Interfaz que permite graficar las señales de datos

Si no se desea salir de la interfaz, se debe presionar el botón ‘’NO’’ y se desplegará nuevamente la ventana mostrada en la figura 5, caso contrario el programa habrá finalizado.

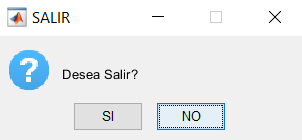


Fig. 6 Ventana de confirmación

## RESULTADOS OBTENIDOS

### CONCLUSIONES

* Si bien la precisión del sensor no es tan alta, permite obtener una gran aproximación en los resultados obtenidos en base al movimiento de la mano y la diferencia tensión de tensión medida.
* El uso del sensor en aplicaciones sensibles al tacto es de gran utilidad debido a que se puede trabajar con cualquier fuente de alimentación debido a que necesita menos de 1 mA de corriente.
* Se debe colocar el sensor en el antebrazo de una manera adecuada, sin presionar mucho y evitando movimientos mientras se realiza la medición; todo esto permitirá mayor precisión en los resultados.
* El electromiograma al ser una prueba segura, útil y sencilla de realizar, en muchos centros de atención médica se considera como una prueba de rutina.

### IMPLEMENTACION PRACTICA

A continuación, se presentan los resultados obtenidos mediante el uso del circuito mostrado en la figura 3.

Imagen que contiene computadora

Descripción generada automáticamente

Fig. 7 Implementación del sensor

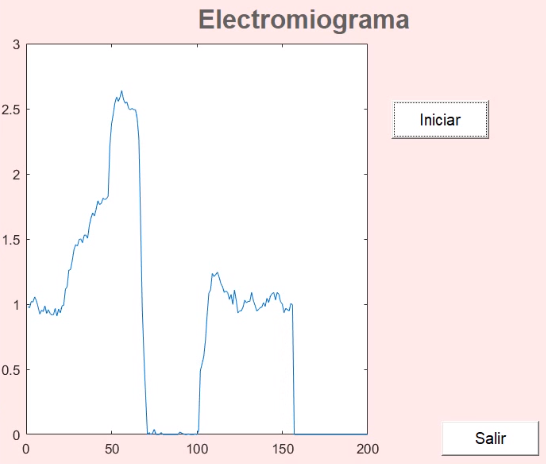


Fig. 8 Resultado obtenido mano cerrada

## LINK VIDEO DEMOSTRATIVO

<https://epnecuador-my.sharepoint.com/:v:/g/personal/marialejandra_contreras_epn_edu_ec/ETplKpwsOEdAguKnMLaZi4cBSzRgJA7RZ6YWMgq8sHpCSA?e=eZxGAM>

## REFERENCIAS

[1] Nares, C. (2015, February 16). Sensor de Fuerza o Presión MF01. HETPRO/TUTORIALES. <https://hetpro-store.com/TUTORIALES/sensor-de-fuerza-o-presion-mf01/>

[2] «Sensor MF01», BConsultors. <https://bconsultors.com/project/sensor-mf01/> (accedido dic. 28, 2020).

[3] «Sensor de Fuerza o Presión MF01», HETPRO/TUTORIALES, feb. 16, 2015. <https://hetpro-store.com/TUTORIALES/sensor-de-fuerza-o-presion-mf01/> (accedido dic. 28, 2020).

[4] PLACA ARDUINO UNO. (2016, January 21). <http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>

‌[5] «Electromiograma», oct. 10, 2013. <https://www.webconsultas.com/pruebas-medicas/electromiograma-12209> (accedido dic. 28, 2020)