

“PROYECTO FINAL- CODIGO QR”

MANUAL DE USUARIO

TECNOLOGIAS COMPUTACIONALES II

PROFESOR: BENITO SAMUEL LOPEZ RAZO

ERIK DAMIAN OLMEDO MENDEZ

JIMENA BASILIO CEDEÑO

GRUPO: I8

UNIVERSIDAD AUTONOMA DEL ESTADO DE MÉXICO

Índice

“PROYECTO FINAL- CODIGO QR”	1
Índice	2
Introducción al manual de usuario	3
Código y su funcionamiento.....	4
Registro y envío de mensaje 1926598	12
Usuarios Registrados	14

Introducción al manual de usuario

Este proyecto final se tendrá que hacer que nuestro código sea capaz de poder leer el código QR desde nuestra cámara web de la laptop y al momento de leer ese código te envíe un código de WhatsApp con el mensaje USUARIO REGISTRADO EN TECNOLOGIAS COMPUTACIONALES II PARA PROYECTO FINAL con el día en que se envió y también su fecha.

El entorno en donde se desarrollo nuestro programa fue en Jupyter notebook de Python, pero se usó la extensión en Visual Studio Code para mayor facilidad al momento de subir nuestro proyecto en GitHub.

A continuación, en este manual de usuario se verá todo el código del funcionamiento de este proyecto, mostrando capturas del código y su explicación como así también ejemplos de uso y corroboración de que el código funciona correctamente y que el usuario le sea de mucha ayuda este manual de usuario.

Código y su funcionamiento

Para empezar nuestro código se deberá de crear una base de datos mediante Excel la cual nosotros la llamaremos **proyectorfinalqr** es muy importante tener en cuenta la ruta de nuestra base de datos mediante Excel ya que se ocupara en nuestro código.

Los datos que se ocuparon en nuestra base de datos son:

	A	B	C	D	E	F
1	id_clave	Nombre	Apellido	Padre	Numero	Codigo QR
2	1926598	ERIK	MENDEZ	ANTONIO	5618503856	RUTA1
3	1973155	ALE	SUAREZ	JOSE	5613138133	RUTA2
4	2126812	FER	SUAREZ	ALE	5626680186	RUTA3
5	1926486	JIMENA	BASILIO	OMAR	5583233548	RUTA4
6	416300	ALEJANDRA	FLORES	FIDENCIO	5626323120	RUTA5

Ruta de acceso a nuestra BD de Excel:

C:\Users\suaba\Desktop\PROYECTOFINALQR\proyectorfinalqr.xlsx

También es muy importante crear una carpeta en donde se guardaran nuestros códigos QR generados por nuestro código y también se debe de tener presente la ruta de acceso de esa carpeta ya que de igual manera se utilizara en el código.

Ruta de acceso a nuestra carpeta donde se guardarán los códigos QR:

C:\Users\suaba\Desktop\PROYECTOFINALQR\codigosqr

Una vez creado todo lo anterior procedemos a abrir nuestro entorno de programación que en este caso será Visual Studio Code que en realidad con una extensión a Jupyter notebook podremos trabajar con Python en este entorno.

Una vez abriendo nuestro entorno se procederá a instalar nuestras librerías que utilizaremos las cuales serán:

```

[55] ✓ 2.0s Python
... Requirement already satisfied: pyautogui in c:\users\suaaba\appdata\roaming\python\python39\site-packages (0.9.54)
Requirement already satisfied: pytweneing>=1.0.4 in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyautogui) (1.0.4)
Requirement already satisfied: pymsgbox in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyautogui) (1.0.4)
Requirement already satisfied: mouseinfo in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyautogui) (0.1.3)
Requirement already satisfied: pygetwindow>=0.0.5 in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyautogui) (0.0.5)
Requirement already satisfied: pyscreeze>=0.1.21 in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyautogui) (0.1.21)
Requirement already satisfied: pyrect in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pygetwindow>=0.0.5) (0.0.5)
Requirement already satisfied: pyscreenshot in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyscreeze>=0.1.21) (0.1.21)
Requirement already satisfied: Pillow>=9.2.0 in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyscreeze>=0.1.21) (9.2.0)
Requirement already satisfied: pyperclip in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from mouseinfo>=0.1.3) (1.8.2)
Requirement already satisfied: entrypoint2 in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyscreenshot>=0.1.21) (0.1.21)
Requirement already satisfied: mss in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyscreenshot>=0.1.21) (1.10.0)
Requirement already satisfied: EasyProcess in c:\users\suaaba\appdata\roaming\python\python39\site-packages (from pyscreenshot>=0.1.21) (0.1.21)
Note: you may need to restart the kernel to use updated packages.
```

```

[56] ✓ 2.0s Python
... Requirement already satisfied: pandas in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (2.1.2)Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy<2, >=1.22.4 in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (from pandas) (1.22.4)
Requirement already satisfied: pytz>=2020.1 in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (from pandas) (2022.7)
Requirement already satisfied: tzdata>=2022.1 in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (from pandas) (2023.1)
Requirement already satisfied: six>=1.5 in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (from python-dateutil>=2.8.2) (1.16.0)

[57] ✓ 1.7s Python
... Requirement already satisfied: openpyxl in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (3.1.2)Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: et-xmlfile in c:\users\suaaba\conda\envs\imagen nv\lib\site-packages (from openpyxl) (1.1.0)

[58] ✓ 1.8s Python
... Requirement already satisfied: pyzbar in c:\users\suaaba\appdata\roaming\python\python39\site-packages (0.1.9)Note: you may need to restart the kernel to use updated packages.
```

Una vez instaladas se procederá a importar nuestras librerías en la utilización de nuestro proyecto.

```

[59] ✓ 0.0s Python
... #AQUI SE MANDAN A IMPORTAR LAS LIBRERIAS

import cv2
import openpyxl
import qrcode
import os
import tkinter as tk
from tkinter import ttk
from tkinter import Label
import pandas as pd
from pyzbar.pyzbar import decode # Para decodificar Los códigos QR
import sqlite3
```

Hacemos una comprobación de nuestra versión de OpenCV

```
print("OpenCV Version:", cv2.__version__, openpyxl.__version__)
```

[60] ✓ 0.0s

... OpenCV Version: 4.7.0 3.1.2

+ Code + Markdown

Teniendo esto se procede a poner la ruta de nuestra base de datos en Excel y también el directorio de nuestra carpeta donde se guardara nuestros códigos QR generados.

```
# RUTA DEL DIRECTORIA DE NUESTRA BASE DE DATOS EN EXCEL
bd_excel = r"C:\Users\suaaba\Desktop\PROYECTOFINALQR\proyectorfinalqr.xlsx"

# NOMBRE DE HOJA DE NUESTRO EXCEL
nombre_hoja = "registros"

# CARPETA DONDE SE GUARDARAN LOS CODIGOS QR
directorio_destino = r"C:\Users\suaaba\Desktop\PROYECTOFINALQR\codigosqr"

# ABRIR EXCEL
workbook = openpyxl.load_workbook(bd_excel)

# SELECCION DE HOJA
hoja = workbook[nombre_hoja]
oracion = f"Se está trabajando con el archivo '{bd_excel}' en la hoja '{nombre_hoja}' del documento./ '{hoja}' "
```

[61] ✓ 0.0s Python

... Se está trabajando con el archivo 'C:\Users\suaaba\Desktop\PROYECTOFINALQR\proyectorfinalqr.xlsx' en la hoja 'registros' del d

Hacemos la verificación y hacemos la conexión de nuestra base de datos para que verifiquemos si el valor es verdad y se muestra el resultado

```
# SOLO SE CORRE PARA GENERAR EL QR DE EL REGISTRO QUE SE NECESITA

valor_buscado = int(input("ingrese el dato de busqueda"))

#CARGAR EXCEL
excel_file = 'proyectorfinalqr.xlsx'
df = pd.read_excel(excel_file, engine='openpyxl')

#CONEXION BD

conn = sqlite3.connect('RegistroBD.db')

# Agrega Los datos del DataFrame a La tabla de La base de datos
df.to_sql('Usuarios', conn, if_exists='replace', index=False)

print(df)
resultado = df.loc[df['id_clave'] == valor_buscado, 'id_clave']

# Verifica si se encontró el valor y muestra el resultado

valor_en_columna_F = resultado.values[0]
print(f'El valor numero de id_clave de "{valor_buscado}" es: {valor_en_columna_F}')
```

[62] ✓ 6.2s Python

Dádonos como resultado:

```
...      id_clave      Nombre Apellido      Padre      Numero      Codigo QR
0      1926598      ERIK      MENDEZ      ANTONIO      5618503856      RUTA1
1      1973155      ALE      SUAREZ      JOSE      5613138133      RUTA2
2      2126812      FER      SUAREZ      ALE      5626680186      RUTA3
3      1926486      JIMENA      BASILIO      OMAR      5583233548      RUTA4
4      416300      ALEJANDRA      FLORES      FIDENCIO      5626323120      RUTA5
El valor numero de id_clave de "2126812" es: 2126812
```

En el siguiente apartado podremos ver que al momento de estar compilando nuestro código nos pedirá que ingresemos un valor y buscara para ver si esta para que se genere nuestro código QR mediante ese valor el cual esta en nuestra base de datos

```
# BUSCA EN LA COLUMNA DE ID_CLAVE Y ME DA EL NUMERO DE TELEFONO(el Folio)
# USANDO DF

#CARGAR BD EXCEL
excel_file = 'proyectofinalqr.xlsx'
df = pd.read_excel(excel_file, engine='openpyxl')

# CONEXION BD
conn = sqlite3.connect('RegistroBD.db')

df.to_sql('Usuarios', conn, if_exists='replace', index=False)

# Muestra el DataFrame con todos los datos
print(df)

# Busca la fila en la que el valor de 'Column2' sea igual a 'B'
fila_seleccionada = df.loc[df['id_clave'] == valor_buscado]

if not fila_seleccionada.empty:
    columna_folio = fila_seleccionada['id_clave'].values[0]
    print(f"Fila seleccionada: \n{fila_seleccionada}")
    print(f"Valor de la columna 0: {columna_folio}")
else:
    print("No se encontraron filas con el valor 'B' en 'Column2'")
```

Cerramos la conexión a nuestra Base de datos ya anteriormente mencionada y nos da como resultado la búsqueda de nuestro id que le pusimos que buscara

```
#CIERRE CONEXION BD
conn.close()

[63] ✓ 0.0s

...
  id_clave  Nombre Apellido  Padre  Numero  Codigo QR
0    1926598    ERIK    MENDEZ    ANTONIO  5618503856    RUTA1
1    1973155     ALE    SUAREZ     JOSE  5613138133    RUTA2
2    2126812     FER    SUAREZ     ALE  5626680186    RUTA3
3    1926486    JIMENA  BASILIO     OMAR  5583233548    RUTA4
4    416300  ALEJANDRA  FLORES  FIDENCIO  5626323120    RUTA5
Fila seleccionada:
  id_clave  Nombre Apellido Padre  Numero  Codigo QR
2    2126812     FER    SUAREZ     ALE  5626680186    RUTA3
Valor de la columna 0: 2126812

columna_folio=str(columna_folio)

[64] ✓ 0.0s
```

Como siguiente paso se empezara con la creación de nuestro código QR en donde se mostrara de qué color se pondrá nuestro código QR y también nuevamente el directorio donde guardaremos nuestro código

```
#CREACION DE CODIGO QR
qr = qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)

# DATOS FOLIO
qr.add_data(columna_folio)
qr.make(fit=True)

#IMAGEN QR
img = qr.make_image(fill_color="black", back_color="white")

import os

# DIRECTORIO IMGENES QR
directorio_destino = r"C:\Users\suaaba\Desktop\PROYECTOFINALQR\codigosqr"

#GUARDAR IMAGEN QR EN CARPETA ESPECIFICA
nombre_archivo_qr = f"codigo_qr_{columna_folio}.png"

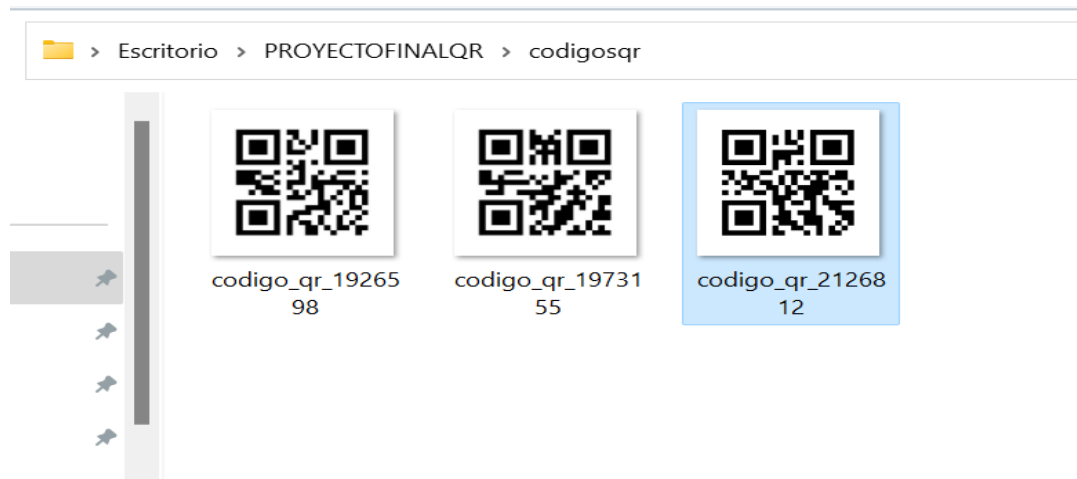
# Eliminar espacios en blanco al final del nombre del archivo
nombre_archivo_qr = nombre_archivo_qr.strip()

ruta_completa = os.path.join(directorio_destino, nombre_archivo_qr)

img.save(ruta_completa)
img.show()

✓ 3.6s
```


Una vez echo este paso en nuestra carpeta comprobaremos si se creo nuestro código QR



Ya comprobando que si se crearon se hará la lectura de la cámara para que lea nuestro código QR y poder enviar nuestro mensaje en WhatsApp

```
#LECTURA DE CAMARA PARA LEER EL CODIGO QR
import cv2
import numpy as np
from pyzbar.pyzbar import decode

cap= cv2.VideoCapture(0)

while True:
    success, img = cap.read()

    if not success:
        break

    for code in decode(img):

        decoded_data = code.data.decode("utf-8")

        print(decoded_data)

        rect_pts = code.rect

        if decoded_data:
            pts = np.array([code.polygon], np.int32)
            cv2.polylines(img, [pts], True,(255,0,0),3)
            cv2.putText(img, str(decoded_data),(rect_pts[0], rect_pts[1]),cv2.FONT_HERSHEY_COMPLEX_SMALL,1,(0,255, 0),)
```

Donde al momento de que se abra la cámara de nuestra laptop pondremos nuestro código una vez leído se cerrara la cámara y abriéndonos una captura de pantalla con el código escaneados y mostrando de que id se refiere nuestro código QR.

```

# CIERRE DE CAMARA
cap.release()

#VENTANA PARA EL CODIGO QR
cv2.imshow("QR Code Content", img)
cv2.waitKey(0) #PRESIONA CUALQUIER TECLA
decoded_data = int(decoded_data)
#CIERRA LAS VENTANAS
cv2.destroyAllWindows()
# PROCEDE A LEER EL DATO Y LO QUE VA A BUSCAR ES UN NUMERO DE TELEFONO
# BUSCA EN LA COLUMNA DE ID_CLAVE Y ME DA EL NUMERO DE TELEFONO

```

Vamos hacia nuestro penúltimo paso el cual consiste que por medio de nuestra conexión de la base de datos se busque la columna 5 la cual es en donde está el número de teléfono de nuestros registros y lo agarra para que en el último paso pueda enviar mensaje satisfactoriamente a WhatsApp

```

import pandas as pd
import sqlite3
#CARGAR NUESTRA BD DE EXCEL
excel_file = 'proyectofinalqr.xlsx'
df = pd.read_excel(excel_file, engine='openpyxl')

#CONEXION BD
conn = sqlite3.connect('RegistroBD.db')

# Agrega Los datos del DataFrame a La tabla de La base de datos
df.to_sql('Usuarios', conn, if_exists='replace', index=False)

fila_seleccionada = df.loc[df['id_clave'] == decoded_data]

if not fila_seleccionada.empty:
    valor_columna5 = fila_seleccionada['Numero'].values[0]
    print(f"Fila seleccionada: \n{fila_seleccionada}")
    print(f"Valor de la columna 5: {valor_columna5}")
else:
    print("No se encontraron filas con el valor 'B' en 'Column2'")

#CONEXION BD CERRADA
conn.close()

valor_columna5 = str(valor_columna5)

```

Como ultimo paso debemos de mandar a llamar a las Horas, Minutos y Segundos actuales para que el envío del mensaje sea en tiempo real, así como el mensaje que queremos que se envíe y con eso tendremos nuestro proyecto terminado al igual que nuestro manual de usuario.

```
▶ ▾ #CONEXION BD CERRADA
conn.close()

valor_columna5 = str(valor_columna5)

#LIBRERIAS
import pywhatkit as kit
from datetime import datetime
import time
import pyautogui as pg

#FECHA
fecha_hora_actual = datetime.now()

#SE EXTRAE LA FECHA, HORA Y MINUTOS
fecha_actual = fecha_hora_actual.strftime("%Y-%m-%d")
hora_actual = fecha_hora_actual.strftime("%H:%M")
minutos_actual = fecha_hora_actual.strftime("%M")

# IMPRIME LA FECHA, HORA Y MINUTOS
print("Fecha actual:", fecha_actual)
print("Hora actual:", hora_actual)
print("Minutos actuales:", minutos_actual)
```

```
phone_number = str("+52" + valor_columna5)
phone_number

message = "USUARIO REGISTRADO EN TECNOLOGIAS COMPUTACIONALES II PARA PROYECTO FINAL " + hora_actual + " hoy : " + fecha_
second = time.time()+60
date = datetime.fromtimestamp(second)

#for i in range(35):
kit.sendwhatmsg(phone_number,message, date.hour, date.minute)
pg.hotkey('enter')

time.sleep(30)
pg.hotkey('ctrl','w')
#PROYECTO FINALIZADO
```

Esperamos que este manual de usuario le sea de utilidad para comprobar que este código funciona correctamente, a continuación se crearon tarjetas con las fotos y el código QR de los registros, a continuación se mostrara un ejemplo de

que los mensajes lleguen correctamente a WhatsApp concluyendo satisfactoriamente este proyecto final de segundo parcial.

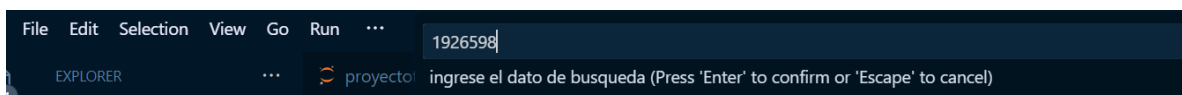
Registro y envío de mensaje 1926598



Erik Damian
Olmedo Mendez



Aquí nos pide que busquemos el dato y le damos el id del usuario que queremos que se genere su código QR

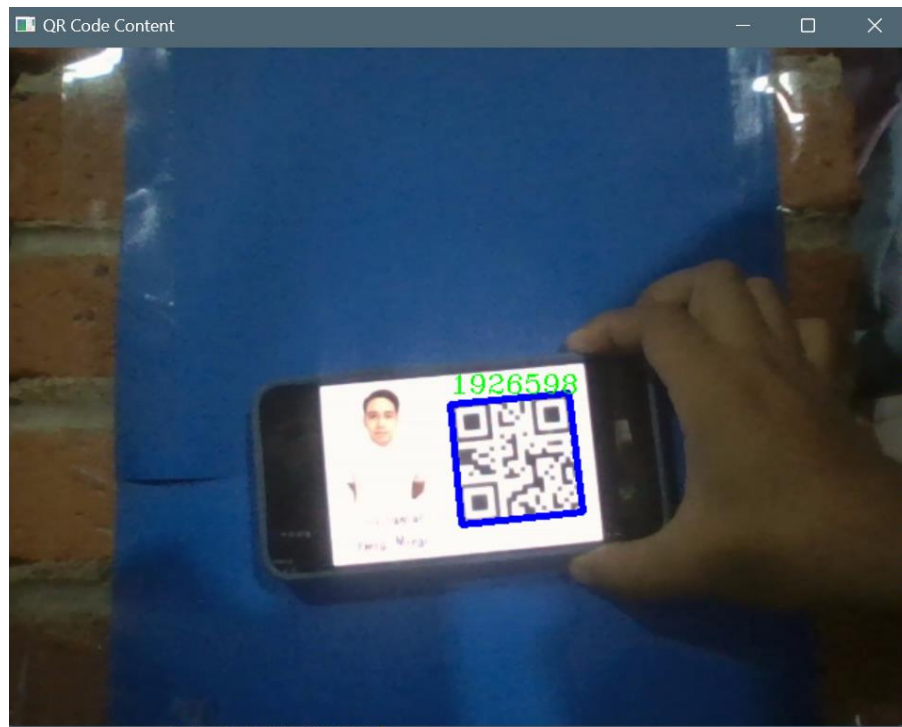


Codigo QR generado



qui

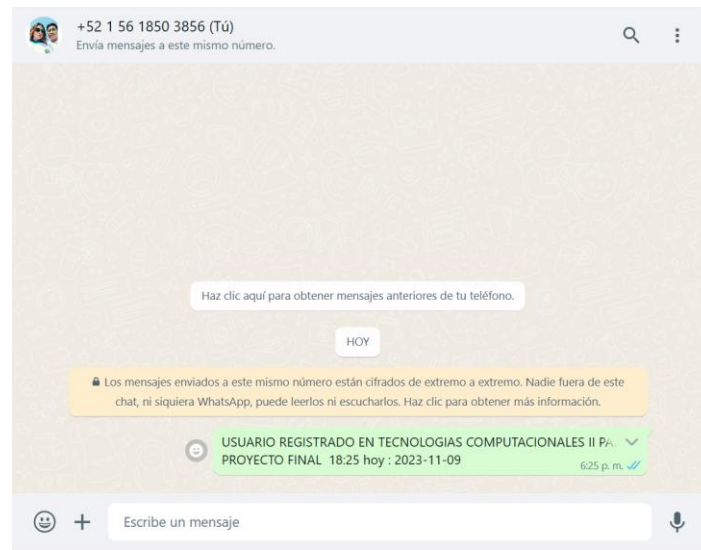
Lectura de nuestro código QR mediante nuestra webcam de la laptop



Comprobación de que todo fue correcto

```
... 1926598
Fila seleccionada:
  id_clave Nombre Apellido Padre Numero Codigo QR
0 1926598 ERIK MENDEZ ANTONIO 5618503856 RUTA1
Valor de la columna 5: 5618503856
Fecha actual: 2023-11-09
Hora actual: 18:22
Minutos actuales: 22
In 43 Seconds WhatsApp will open and after 15 Seconds Message will be Delivered!
```

Mensaje enviado automáticamente a nuestro numero de nuestra base de datos



Usuarios Registrados



Asi como se mostro el ejemplo, podemos hacerlo con los registros que se estan mostrando, con esto damos por terminado nuestro proyecto final y manual de usuario, esperamos que sea de su agrado, Gracias.