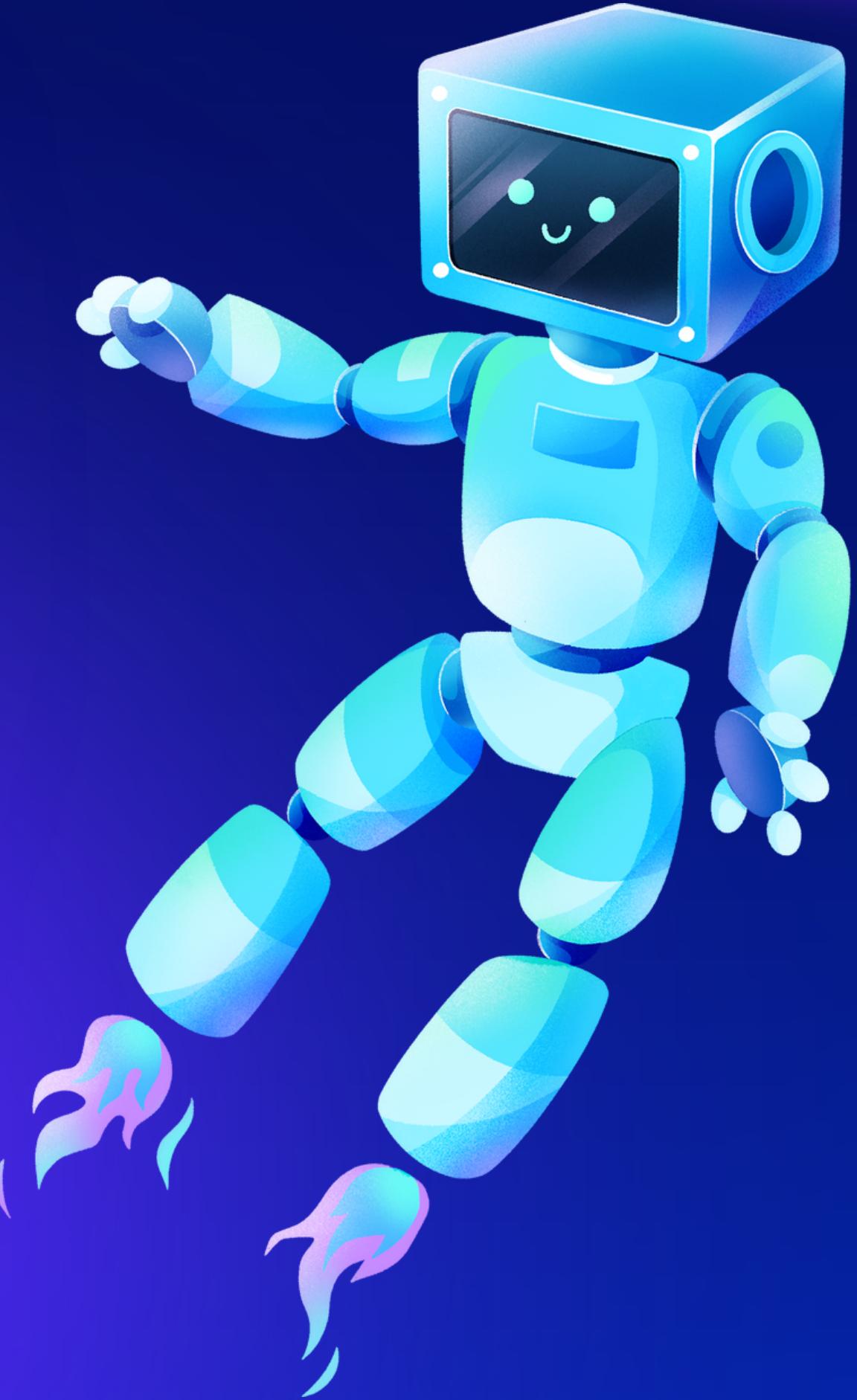


CHAT BOT DENTISTA ARTEMAI

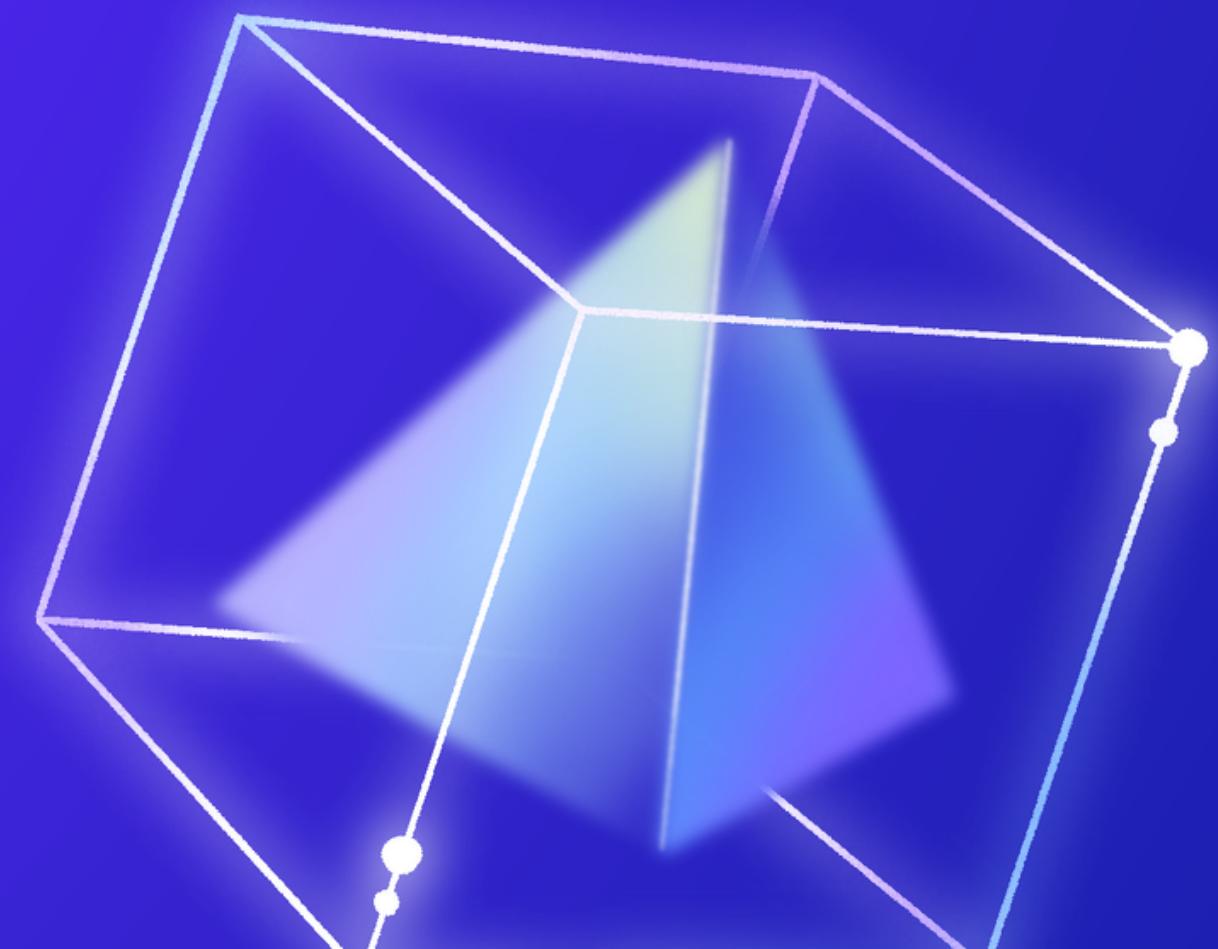
Presentación realizada por Juan José Martínez Águila





CONTENIDO

- Introducción
- ¿Qué puede hacer?
- Creación del bot
- Diseño base de datos
- Ficheros auxiliares
- Código principal del bot
- Curiosidades
- Bibliografía

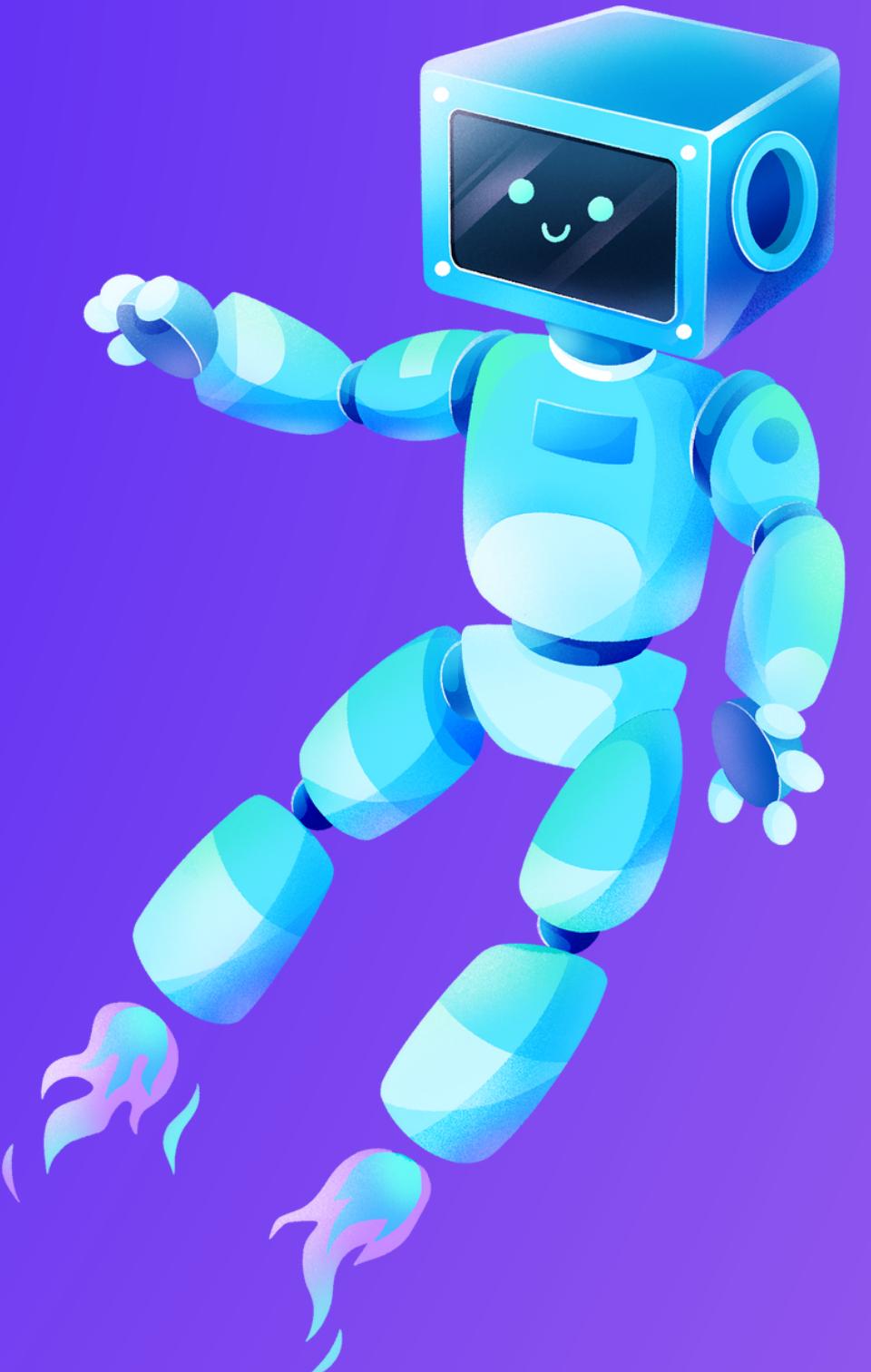


INTRODUCCIÓN

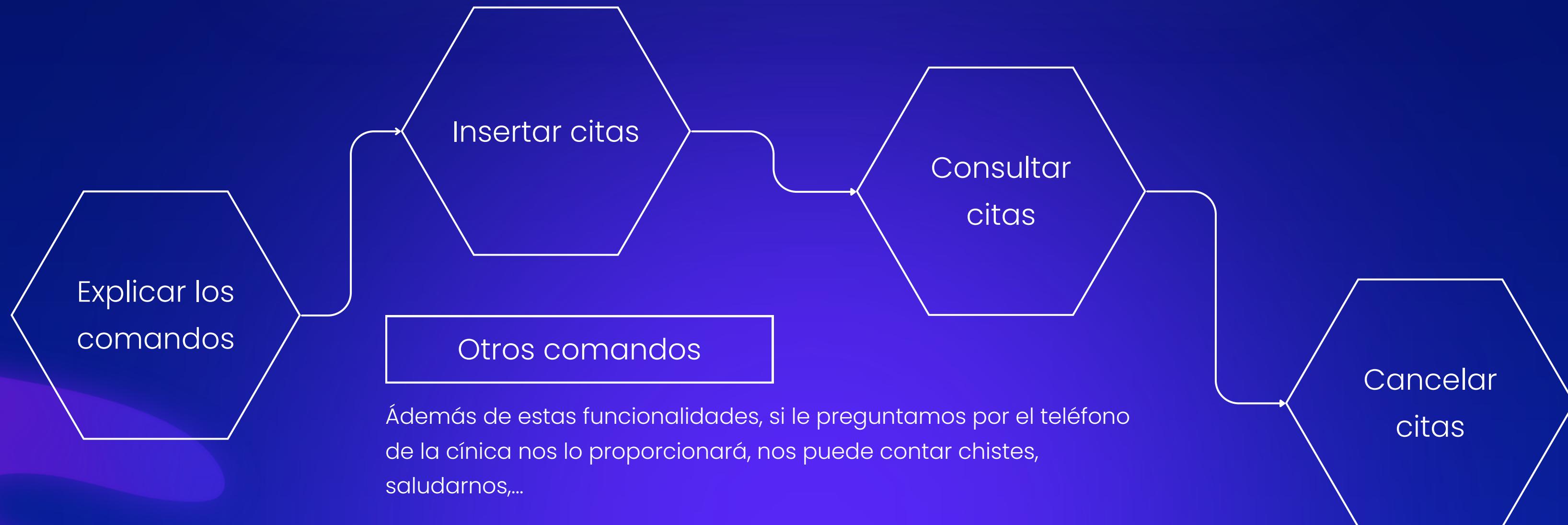
Una clínica llamada Artemai compuesta por varios dentistas nos ha contratado para automatizar el registro, las cancelaciones y la información a la reserva de citas de los clientes, Además las citas solo se podrán almacenar de media hora en media hora y la clínica estará abierta de 9 de la mañana a 14 y de 16:30 hasta las 21:00.

Solución

Ante este problema la mejor solución que se me ha ocurrido es programar un bot de telegram que se encargue de efectuar todas aquellas funcionalidades de manera autónoma.



QUE PUEDE HACER ESTE BOT



• CREACIÓN DEL BOT



Pasos

- 1.- Iniciar el chat predeterminado que hay en telegram llamado BotFather.
- 2.- Pulsar iniciar y utilizamos el comando /newbot y seguir los pasos que nos indica.
- 3.- Configurar nuestro bot en caso de que sea necesario por nuestro problema.

DISEÑO DE LA BASE DE DATOS

Para el diseño de la base de datos habrá dos tablas importantes, Cita y Usuario.

La tabla Usuario almacenará el chat id del cliente y su nombre de usuario y la tabla Cita guardará la información correspondiente a las citas.

```
mysql> describe Usuario;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| chat_id | varchar(100) | YES | UNI  | NULL    |          |
| user    | varchar(500)  | NO  | PRI  | NULL    |          |
+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

```
mysql> describe Cita;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| ID    | int    | NO   | PRI  | NULL    | auto_increment |
| usuario | varchar(500) | NO  | MUL  | NULL    |          |
| fecha | date   | NO   |      | NULL    |          |
| hora  | time   | YES  |      | NULL    |          |
+-----+-----+-----+-----+-----+
```

Nuestra base de datos también contendrá un evento para que se ejecutará cada media hora para que se borren las citas a la fecha y hora menores a la actual:

```
--EVENTO PARA QUE SE BORREN TODAS LAS FECHAS ANTERIORES AL DÍA Y HORA ACTUAL
CREATE EVENT eliminar_citas_antiguas
ON SCHEDULE EVERY 30 MINUTE
DO
    DELETE FROM Cita
    WHERE hora < NOW() and fecha < NOW();
```

También ejecutaremos un trigger para cada vez que se almacene una cita compruebe que la fecha y hora son mayores a la actual.

```
--TRIGGER PARA EVITAR QUE SE ALMACENEN FECHAS Y HORAS ACTUALES
delimiter //
CREATE TRIGGER no_fecha_anterior
BEFORE INSERT ON Cita
FOR EACH ROW
BEGIN
    IF CONCAT(NEW.fecha, ' ', NEW.hora) < NOW() THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'No se pueden almacenar fechas y horas anteriores a la actual';
    END IF;
END;
```

FICHEROS AUXILIARES

Constara de dos ficheros auxiliares importantes que guardarán las estructuras de datos necesarias para transformar en lenguaje más legible las fechas y horas del día que el formato de estas en mysql. El de las fechas es YYYY-MM-DD y el de las horas es HH:MM:SS.

Estos serán tipos de datos de tipo diccionario que dada una hora o una fecha en el día te dará esa fecha pero en un formato de texto y viceversa.

Dada la fecha 2023-05-19 y la hora 12:30:00 utilizando los distintos diccionarios obtendriamos esa fecha y hora como el día 19 de mayo del año 2023 a las 12 y media del mediodía

CÓDIGO PRINCIPAL DEL BOT

Funciones auxiliares:

Crearemos varias funciones auxiliares como comprobar si una fecha o hora está reservada a la hora de guardar una cita, otra para comprobar si la hora insertada está entre las horas donde está abierta la clínica, también añadiremos una función con la biblioteca signal para cuando se aborte el programa se haga un commit en la base de datos y estas entre otras tantas funciones.

Estas funciones se llamarán en nuestra función principal del bot, handle(msg) que es la que recibirá los mensajes de los usuarios y la que le responderá a dichas peticiones.

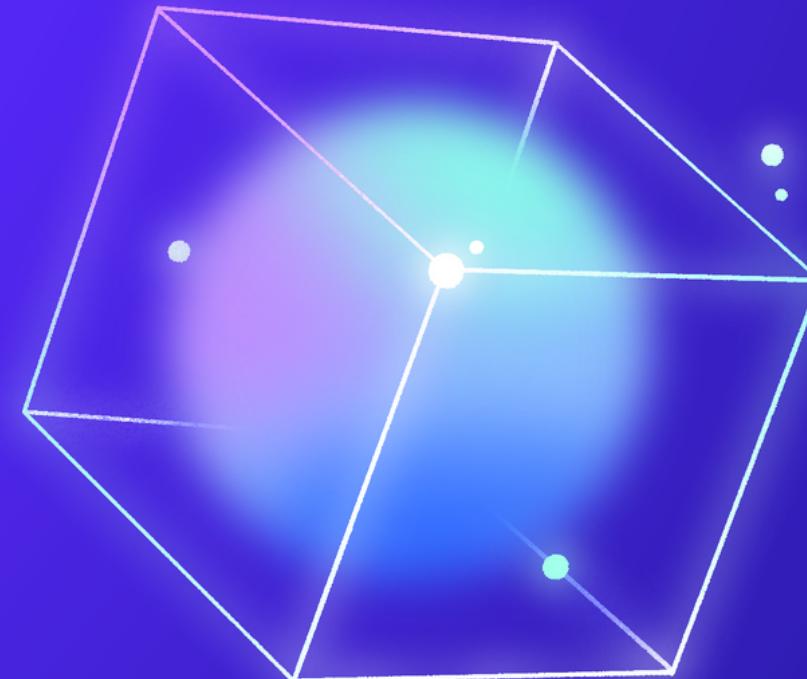


CÓDIGO PRINCIPAL DEL BOT

Función handle(msg):

Para responder o analizar el mensaje por el usuario tendrá una estructura de if/else y según una serie de palabras clave responderá con la información oportuna.

- /start
- /time
- /consultarCitas
- /insertarCita
- /cancelarCita
- /comandos
- hola
- me das el móvil
- cuéntame un chiste



CÓDIGO PRINCIPAL DEL BOT

Lo primero que haremos en esta función handle será con datos que pueden sernos útiles, como el texto del mensaje que envía el usuario (antes pasando todo el mensaje a mayúsculas), nickname de telegram,...

Para comunicarnos con él con sentencias simples como un saludo o pedirle el teléfono el bot buscará en el mensaje de texto distintas palabras clave y según cuales sean responderá al mensaje de la manera correspondiente.

```
elif command.find("CHISTE") != -1:  
    num = random.randint(1, 80)  
    print(num)  
    sql="SELECT mensaje FROM Chiste WHERE id=%s"  
    cursor.execute(sql, (num,))  
    mensaje = cursor.fetchall()  
    bot.sendMessage(chat_id, mensaje[0][0])
```

CÓDIGO PRINCIPAL DEL BOT

Función para insertar una cita

```
elif command.find("/INSERTARCITA") != -1:
    if(estaUsuario(nickname, chat_id) == False):
        print("ENTROOOOOO")
        anadirUsuario(chat_id, nickname)

    mensaje = command.split(" ")
    print(len(mensaje))
    if(len(mensaje) == 1):
        bot.sendMessage(chat_id, "Ha olvidado escribir el dia y la fecha al lado, \n P.Ejemplo: /insertarcita el 19 de mayo de 2001 a las seis de la tarde")
    elif len(mensaje) != 13 and len(mensaje) != 15:
        bot.sendMessage(chat_id, "El formato que ha introducido es incorrecto, el formato correcto es el DD de (mes escrito en texto) de YYYY a las (hora en letra) de la (parte del dia que desee)")

    else:
        if len(mensaje) == 13:
            if(len(mensaje[2]) == 1):
                mensaje[2] = '0' + mensaje[2]
            fecha = mensaje[6] + '-' + meses_numero[mensaje[4]] + '-' + mensaje[2]
            formato_hora = mensaje[7] + " " + mensaje[8] + " " + mensaje[9] + " " + mensaje[10] + " " + mensaje [11] + " " + mensaje[12]
            formato_hora = formato_hora.lower()
            print(formato_hora)
        else:
            if(len(mensaje[2]) == 1):
                mensaje[2] = '0' + mensaje[2]
            fecha = mensaje[6] + '-' + meses_numero[mensaje[4]] + '-' + mensaje[2]
            formato_hora = mensaje[7] + " " + mensaje[8] + " " + mensaje[9] + " " + mensaje[10] + " " + mensaje [11] + " " + mensaje[12] + " " + mensaje[13] + " " + mensaje[14]
            formato_hora = formato_hora.lower()
            print(formato_hora)

        try:
            hora = horas_mysql[formato_hora]
        except:
            bot.sendMessage(chat_id, "Ha ocurrido un error el formato recuerde ingresar su cita como en este ejemplo: el 7 de abril de 2024 a las cinco de la tarde y recuerde que los saltos se dan de media hora en media hora por ejemplo a las cinco, a las cinco y media, a las seis,... esto")
            comprobacion = comprobar_hora(hora)
            print(hora)
            print(comprobacion)

        if len(comprobacion) == 2:
            bot.sendMessage(chat_id, comprobacion[1])
        else:
            sql = "INSERT INTO Cita(usuario,fecha,hora) VALUES(%s, %s, %s)"

            if(citaDisponible(fecha, hora)):
                try:
                    cursor.execute(sql,(nickname, fecha, hora))

                    bot.sendMessage(chat_id, "La cita ha sido ingresada con éxito")
                except:
                    bot.sendMessage(chat_id, "Lo siento ha ocurrido algún error, seguramente sea que ha intentado insertar una fecha que es anterior a la actual")
            else:
                bot.sendMessage(chat_id, "Ese dia y a esa hora ya esta ocupada")
```



CÓDIGO PRINCIPAL DEL BOT

Función para cancelar una cita

```
elif command.find("/CANCELARCITA") != -1:
    mensaje = command.split(" ")
    if(len(mensaje) == 1):
        bot.sendMessage(chat_id, "Ha olvidado escribir el día y la fecha al lado, \n P.Ejemplo: /cancelarcita el 19 de mayo de 2001 a las cinco de la tarde")
    elif(len(mensaje) != 13 and len(mensaje) != 15):
        bot.sendMessage(chat_id, "El formato que ha introducido es incorrecto, el formato correcto es el DD de (mes escrito en texto) de YYYY a las (hora en letra) de la tarde/mañana")
    else:
        if len(mensaje) == 13:
            if(len(mensaje[2]) == 1):
                mensaje[2] = '0' + mensaje[2]
            fecha = mensaje[6] + '-' + meses_numero[mensaje[4]] + '-' + mensaje[2]
            formato_hora = mensaje[7] + " " + mensaje[8] + " " + mensaje[9] + " " + mensaje[10] + " " + mensaje [11] + " " + mensaje[12]
            formato_hora = formato_hora.lower()
        else:
            if(len(mensaje[2]) == 1):
                mensaje[2] = '0' + mensaje[2]
            fecha = mensaje[6] + '-' + meses_numero[mensaje[4]] + '-' + mensaje[2]
            formato_hora = mensaje[7] + " " + mensaje[8] + " " + mensaje[9] + " " + mensaje[10] + " " + mensaje [11] + " " + mensaje[12]+ " " + mensaje[13] + " " + mensaje[14]
            formato_hora = formato_hora.lower()

        print(formato_hora)
        try:
            hora = horas_mysql[formato_hora]

        except:
            bot.sendMessage(chat_id, "Ha ocurrido un error el formato recuerde ingresar su cita como en este ejemplo: el 7 de abril de 2024 a las cinco de la tarde y recuerde que los saltos se dan de media hora en media hora")
            comprobacion = comprobar_hora(hora)

        if len(comprobacion) == 2:
            bot.sendMessage(chat_id, comprobacion[1])
        else:
            sql = "DELETE FROM Cita WHERE fecha=%s AND usuario=%s AND hora=%s"
            if(citaOcupadaUsuario(fecha, nickname)):
                try:
                    cursor.execute(sql, (fecha, nickname, hora))
                    bot.sendMessage(chat_id, "Cita borrada con éxito")
                except:
                    bot.sendMessage(chat_id, "Ha ocurrido un error y no se ha podido borrar la cita")
            else:
                bot.sendMessage(chat_id, "No tiene ninguna cita ese día")
```



CÓDIGO PRINCIPAL DEL BOT

Función para consultar las citas

```
elif command == "/CONSULTARCITAS":  
    sql ="SELECT fecha,hora FROM Cita WHERE usuario = %s"  
    cursor.execute(sql,(nickname,))  
    datos = cursor.fetchall()  
    print(datos)  
  
    if(len(datos) == 0):  
        mensaje = "Lo siento no hay ninguna cita reservada a su nombre"  
    else:  
        mensaje = "Estas son sus citas: \n "  
  
        for dato in datos:  
            #lista_fecha = dato[0].split('-')  
            hora = str(dato[1])  
            fecha = str(dato[0])  
            print(fecha)  
            lista_fecha = fecha.split('-')  
            mensaje = mensaje + "- El día " + lista_fecha[2] + " de " + meses[lista_fecha[1]] + " del año " + lista_fecha[0] + " " + expresiones_horas[hora] + "\n"  
  
    bot.sendMessage(chat_id, mensaje)
```



CURIOSIDADES

Para almacenar los chistes he hecho webscrapping a una página web.

Esto lo he hecho con la librería beautifull soup.

Analizando el código html de la página puedes quedarte con las partes que te interesen.



BIBLIOGRAFÍA

- <https://atareao.es/tutorial/crea-tu-propio-bot-para-telegram/>
- Chatgpt
- <https://youtu.be/GEydlPTqp6E>
- <https://j2logo.com/python/web-scraping-con-python-guia-inicio-beautifulsoup/>

MUCHAS GRACIAS
POR VER ESTA PRESENTACIÓN

