

```

import logging
import re

from telegram import ForceReply, Update
from telegram.ext import Application, CommandHandler, ContextTypes,
MessageHandler, filters

    # Enable logging
    logging.basicConfig(format="%(asctime)s - %(name)s - %(levelname)s -
%(message)s", level=logging.INFO)
    logger = logging.getLogger(__name__)

    # Definir patrones de expresiones regulares
    patron_origen_destino_fecha = r"pedir un servicio de (\w+) a (\w+) para
el (\d{1,2} de \w+)"
    patron_precio = r"cuánto cuesta un servicio de (\w+) a (\w+)"
    patron_ida_vuelta = r"un Rappifavor de ida y vuelta de (\w+) a (\w+)"
    #PROPUESTA
    patron_precio_menor = r"muestra comidas con precio menor de (\d+)"

    async def start(update: Update, context: ContextTypes.DEFAULT_TYPE) ->
None:
        """Send a message when the command /start is issued."""
        user = update.effective_user
        await update.message.reply_html(
            rf"Hi {user.mention_html()}!",
            reply_markup=ForceReply(selective=True),
        )

    async def help_command(update: Update, context: ContextTypes.DEFAULT_TYPE) -
> None:
        """Send a message when the command /help is issued."""
        await update.message.reply_text("Help!")

    async def process_flight_query(update: Update, context:
ContextTypes.DEFAULT_TYPE) -> None:
        """Process user queries about flights using regular expressions."""
        message_text = update.message.text

        # Check if the message matches any of the regular expression patterns
        if re.search(patron_origen_destino_fecha, message_text):
            origen_destino_fecha = re.search(patron_origen_destino_fecha,
message_text)
            origen = origen_destino_fecha.group(1)
            destino = origen_destino_fecha.group(2)

```

```

        fecha = origen_destino_fecha.group(3)
        await update.message.reply_text(f"Pedido de {origen} a {destino}
para la {fecha}")

    elif re.search(patron_precio, message_text):
        precio = re.search(patron_precio, message_text)
        origen = precio.group(1)
        destino = precio.group(2)
        await update.message.reply_text(f"Consultar precio de pedido de
{origen} a {destino}")

    elif re.search(patron_ida_vuelta, message_text):
        ida_vuelta = re.search(patron_ida_vuelta, message_text)
        origen = ida_vuelta.group(1)
        destino = ida_vuelta.group(2)
        await update.message.reply_text(f"RappiFavor de ida y vuelta de
{origen} a {destino}")

    # PROPUESTA AGREGADA
    elif re.search(patron_precio_menor, message_text):
        precio_menor = re.search(patron_precio_menor, message_text)
        precio_max = precio_menor.group(1)
        await update.message.reply_text(f"Mostrar comidas con precios
menores de {precio_max}")

    else:
        await update.message.reply_text("Lo siento, no puedo entender tu
consulta.")

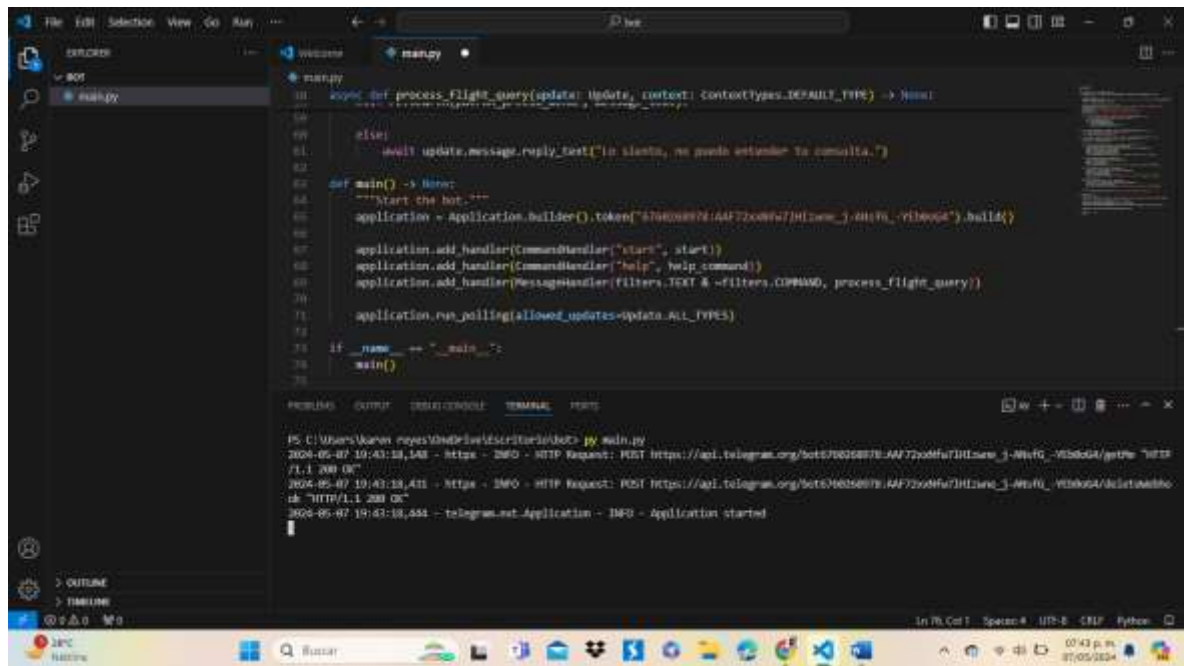
def main() -> None:
    """Start the bot."""
    application =
Application.builder().token("6760268978:AAF72xxNfw7lHIzwne_j-ANsfG_-
YEb0oG4").build()

    application.add_handler(CommandHandler("start", start))
    application.add_handler(CommandHandler("help", help_command))
    application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND,
process_flight_query))

    application.run_polling(allowed_updates=Update.ALL_TYPES)

if __name__ == "__main__":
    main()

```



```
File Edit Selection View Go Run ...
manqy
EXPLORER
src
  bot
    manqy.py
Webview
manqy
10 async def process_flight_query(update: Update, context: ContextTypes.DEFAULT_TYPE) -> None:
11     """Process flight query"""
12     pass
13
14 else:
15     await update.message.reply_text("in silent, no need answer to consulta.")
16
17 def main() -> None:
18     """Start the bot"""
19     application = Application.builder().token("4768088978:AAF72x0fu7JH1zaw_3-0HsF0_Vib60s4").build()
20
21     application.add_handler(CommandHandler("start", start))
22     application.add_handler(CommandHandler("help", help_command))
23     application.add_handler(MessageHandler(filters.TEXT & ~filters.COMMAND, process_flight_query))
24
25     application.run_polling(allowed_updates=Update.ALL_TYPES)
26
27 if __name__ == "__main__":
28     main()
29
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\Karen Reyes\OneDrive\Documents> py main.py
2024-05-07 19:43:18.543 - https - INFO - HTTP Request: GET https://api.telegram.org/bots4768088978:AAF72x0fu7JH1zaw_3-0HsF0_Vib60s4/getMe "HTTP/1.1 200 OK"
2024-05-07 19:43:18.411 - https - INFO - HTTP Request: POST https://api.telegram.org/bots4768088978:AAF72x0fu7JH1zaw_3-0HsF0_Vib60s4/setWebhook "HTTP/1.1 200 OK"
2024-05-07 19:43:18.454 - telegram.bot.Application - INFO - Application started
```

Nombre: Karen Adilene García Reyes

No. Control: 21200231