

```

import json
import logging
from telegram import Update
from telegram.ext import ApplicationBuilder, CommandHandler, ContextTypes

logging.basicConfig(level=logging.INFO)

with open("dados_plantio.json", "r", encoding="utf-8") as f:
    dados_plantio = json.load(f)

def formatar_resposta(fazenda):
    resposta = f"[INFO] *Informações para: {fazenda.title()}*\n\n"
    encontrados = False
    for item in dados_plantio:
        if fazenda.lower() in item["fazenda"].lower():
            encontrados = True
            resposta += (
                f"[FAZENDA] *Fazenda:* {item['fazenda']}\n"
                f"[DATA] *Data do plantio:* {item['data_plantio']}\n"
                f"[CULTURA] *Cultura:* {item['cultura']}\n"
                f"[PIVO] *Pivô:* {item['pivo']}\n"
                f"[AREA] *Área:* {item['area']:.2f} ha\n\n"
            )
    if not encontrados:
        resposta = "Nenhuma informação encontrada para essa fazenda."
    return resposta

async def start(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text(
        "Olá! Envie o comando /info seguido do nome da fazenda para obter informações.\nEx: /info Catules"
    )

async def info(update: Update, context: ContextTypes.DEFAULT_TYPE):
    if not context.args:
        await update.message.reply_text("Por favor, informe o nome da fazenda. Ex: /info Catules")
        return

    fazenda = " ".join(context.args)
    resposta = formatar_resposta(fazenda)
    await update.message.reply_markdown(resposta)

if __name__ == '__main__':
    import os
    TOKEN = os.getenv("TELEGRAM_TOKEN")
    app = ApplicationBuilder().token(TOKEN).build()

    app.add_handler(CommandHandler("start", start))
    app.add_handler(CommandHandler("info", info))

    app.run_polling()

```