



PROYECTO FINAL

Estructura de datos e Base de datos

Integrantes:

- Luis Alvarez Medina
- Carlos Flores Paucara
- Elvin Cussi Aranibar
- Nilber Mayta Cuno

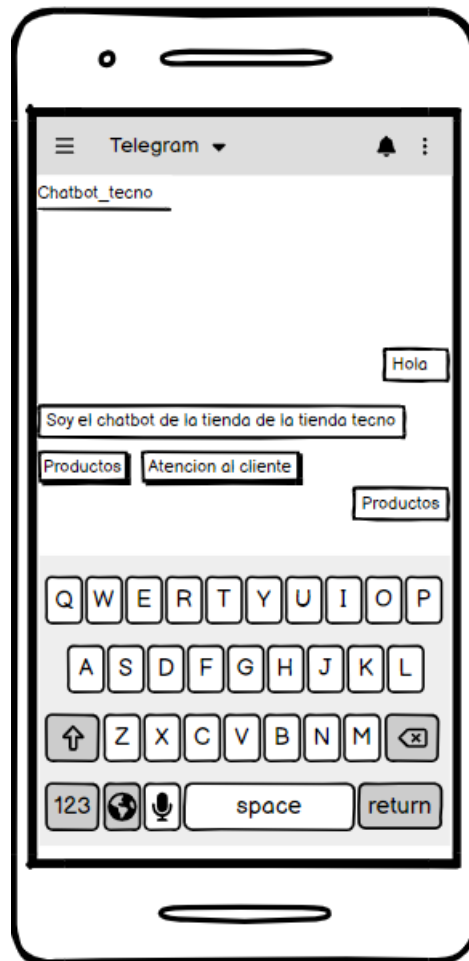
Ingeniero: **WILLIAM RODDY BARRA PAREDES**

2023



1. Introducción

El objetivo del proyecto es implementar un chat Bot que muestre los productos de la tienda tecno para que pueda responder consultas de clientes sobre los productos sin necesidad de una persona. Es decir, el cliente podrá hacer consultas sobre detalles de productos de una manera remota



2. Análisis y diseño del sistema

- a. Nombre del proyecto: Dada la situación de buscar detalles de productos nuestro chat Bot tendrá el nombre de "Tecno"
- b. **Definición de las plataformas:** Identificar cuales son las clases, métodos librerías necesarias para desarrollar el sistema.
 - 1) Python
 - 2) Telegram
 - 3) Telebot

- 4) NITK
- 5) Token Telegram (Bot Father)
- 6) Etiquetas
- 7) @bot.message_handler()
- 8) @bot.callback_query_handler()
- 9) Listas
- 10) Funciones
- 11) Diccionarios

Python	Un lenguaje de programación de alto nivel que lo usamos para el desarrollo de nuestro chat ya que este lenguaje de aplicación.
Telegram	Aplicación de mensajería instantánea que permite la implementación de bots.
BotFather	Un Bot de la aplicación de telegram que nos permite generar un Bot.
Telebot	Es una librería que nos proporciona funciones para interactuar con la API de telegram.
NLTK	Es una librería que nos permite convertir palabras claves en tokens.
@bot.message_handler()	Utilizado para recibir un parámetro, es decir, encuentre una similitud en los tokens que existe y realiza una comparación.
@bot.callback_query_handler()	Iteración de botón para la activación de una función.
Listas	Conjunto de elementos.
Funciones	Es un bloque de proceso reutilizable.
Diccionario	Permite almacenar palabras claves que contiene un valor.

c. Diseño de clases

```

class {
    "deep": {
        "nombre": "deep",
        "imagen": "https://coolboxpe.vtexassets.com/arquivos/ids/178831-800-auto?v=617478186687400000&width=800&height=800",
        "descripcion": "Capacidad máx.: 17\nPuertos USB: 2\nVentiladores: 3\nPosturas: 5 Niveles de Altura Ajustable\nMaterial: Malla Metálica + Plástico\nLuces: si\nUso: para Laptop\n¿Qué incluye la caja?: cable USB A USB, cooler\n",
    },
    "emaxx": {
        "nombre": "emaxx",
        "imagen": "https://coolboxpe.vtexassets.com/arquivos/ids/198341-800-auto?v=617789871675400000&width=800&height=800",
        "descripcion": "Capacidad máx.: 15.6\nPuertos USB: 2\nVentiladores: 3\nPosturas: 5\nMaterial: Plástico y malla de hierro\nLuces: si\nUso: para Laptop\nDimensiones del ventilador: 7 x 7 x 1.5 cm (4 ventiladores)\n",
    },
}

```

```
#GENERAR BOTONES
audfonos = {"Inalámbricos":inalambrico,"Cable ":concable,"Orejeras":cascos}
parlants = {"Parlantes": parlantes,"Microfonos":microfonos}
articuloPc = {"Teclado":teclados,"Mouse":mouses,"Cooler para Laptop":microfonos}
```

```
def botenesOpciones(chat_id):
    markup = types.InlineKeyboardMarkup()
    for opcion in opciones:
        button = types.InlineKeyboardButton(opcion,callback_data=opcion)
        markup.add(button)
    respuesta = "En que puedo ayudarte"
    bot.send_message(chat_id,respuesta,reply_markup=markup)
```

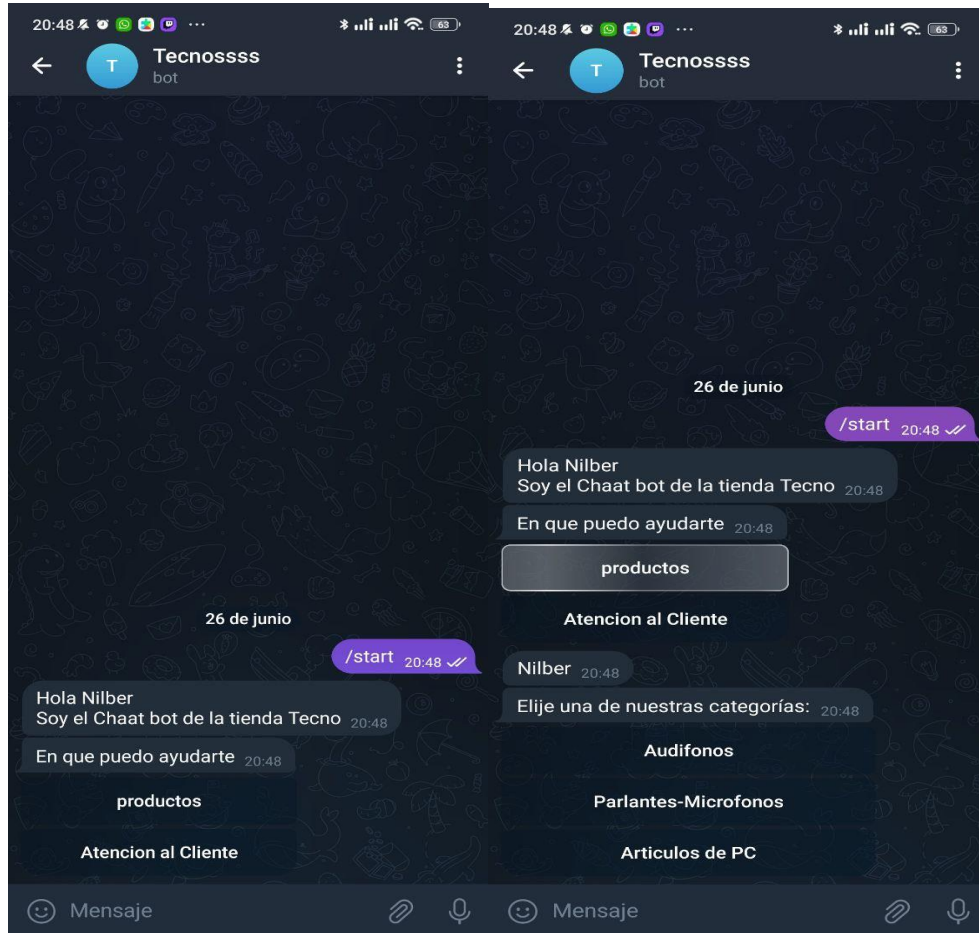
```
@bot.callback_query_handler(func=lambda call: call.data in opciones.keys())
def botenesDeOpciones(call):
    opcion = call.data
    atencionAlCliente = opciones[opcion]
    if(opcion == "Atencion al Cliente"):
        respuesta = atencionAlCliente
        bot.send_message(call.message.chat.id, respuesta)
    elif(opcion == "productos"):
        respuesta = "{}".format(call.from_user.first_name)
        bot.send_message(call.message.chat.id, respuesta)
        tiposProd(call.message.chat.id)
```

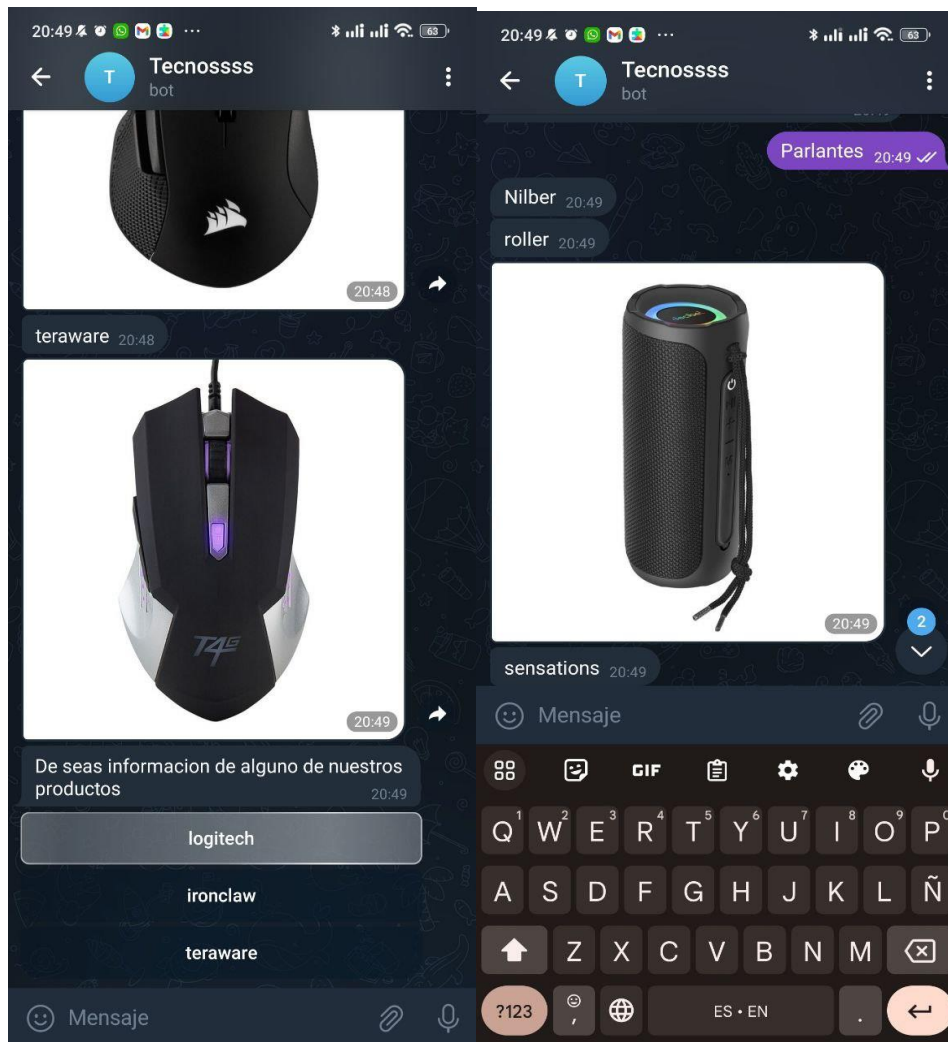
```
#hablado de por mensaje cuando escoja un cooler
@bot.message_handler(func=lambda message: any(token.lower() in coler for token in word_tokenize(message.text.lower()))
def mausess(message):
    mensajeRecuperado = message.text
    if(mensajeRecuperado in coolers):
        nombre = coolers[mensajeRecuperado]["nombre"]
        imagen_url = coolers[mensajeRecuperado]["imagen"]
        detalle = coolers[mensajeRecuperado]["descripcion"]
        bot.send_message(message.chat.id, nombre)
        bot.send_photo(message.chat.id, imagen_url)
        bot.send_message(message.chat.id, detalle)
    else:
        respuesta = "Ingrese El nombre Del producto De manera adecuada o el producto que ingreso no existe"
        bot.send_message(message.chat.id, respuesta)
```

```
#LIBRERIAS PARA EL CHATBOT-----
#pip install pytelegrambotapi
import telebot
from telebot import types
#pip install nltk
import nltk
#nltk.download('punkt')
from nltk.tokenize import word_tokenize
```

3. Usabilidad

- a. Adjuntar imágenes del sistema.





b. <https://youtu.be/36afbl9shIA>

4. Conclusiones

La combinación de Python y Telegram para la realización de un Bot ofrece eficiencia y facilidad de implementación, permitiendo un desarrollo rápido y depuración sencilla del código.

la elección de Python y Telegram proporciona un potencial de personalización y mejora continua para el Bot, gracias a la activa comunidad de Python, su amplia gama de bibliotecas y la constante evolución del ecosistema de Telegram, permitiendo adaptar el Bot a las necesidades cambiantes y ampliar su funcionalidad.

En conclusión, tanto las librerías de Telebot y la librería de NLTK nos sirvieron para que el proceso del desarrollo de Bot sea más sencillo para el desarrollo del chat Bot haciendo uso de sus decoradores y funciones.