

# GPLSI FieroBoT: Asistente virtual para la captación de insultos

## GPLSI FieroBoT: A virtual assistant for collecting insults

**Resumen de la aplicación:** FieroBoT es un asistente virtual orientado a la recopilación de insultos desde la aplicación de mensajería Telegram. A través del sistema de mensajería de Telegram los usuarios pueden libremente tener una conversación con el robot donde se incita a que lo insultemos. El proceso es el siguiente, en primer lugar, se nos preguntará nuestro sexo y la edad y a continuación podremos insultar al robot. Solo se almacenará el contenido del mensaje y los datos demográficos genéricos como rango de edad y sexo con el fin de organizar el contenido. Los insultos recolectados servirán para la creación de un recurso de palabras y expresiones violentas que podrá ser utilizado para distintos estudios relacionados con el lenguaje popular.

**Summary:** FieroBoT is a virtual assistant aimed at collecting insults from users of messaging application Telegram. Users can have an anonymous and gratis conversation with the robot whereby, they are prompted to insult it. The process involves asking the user their gender and age after which, they are free to insult the robot. Only the content of the message and generic demographic data such as age range and gender will be stored to categorize the content. The collected insults will be used to create a resource of violent words and expressions that can be used for different studies related to vernacular insults.

### **Documentación de instalación y mantenimiento del servicio:**

#### **Tecnología:**

- Docker
- Docker-compose
- Git
- SSH

#### **Detalles técnicos:**

Docker es un sistema de encapsulación de procesos, comúnmente como un sistema de nacionalización rápido y liviano. Esta basado en contenedores creados a partir de imágenes de sistemas. Con Docker-Compose, por medio de la configuración del archivo docker-compose.yml podemos especificar los requerimientos y acciones del sistema. Importante entender que con Docker-Compose, la máquina no tiene persistencia de datos. Cuando detengamos la máquina,

perderemos cualquier cambio o modificación, por lo que se utilizan volúmenes, que son carpetas compartidas con el host, donde almacenamos los datos que necesitan persistencia.

## Fuente

<https://www.digitalocean.com/community/tutorials/como-instalar-y-usar-docker-en-ubuntu-18-04-1-es>

## Proceso de Instalación

### Install.sh

```
#!/bin/bash
# -*- ENCODING: UTF-8 -*-
echo "Updating"
apt-get update
echo "installing nodeJS"
apt-get install nodejs -y
echo "intalling npm"
apt-get install npm -y
echo "installing fs"
npm install fs
echo "installing node-telegram-bot-api"
npm install node-telegram-bot-api
echo "installing csv-writer"
npm install csv-writer
echo "start"
node fierobot.js
exit
```

## Crear Bot (se necesita cuenta de Telegram)

### Obtener token de chatbot

Abrir Telegram y buscar @BotFather en contactos para iniciar una conversación enviando los siguientes mensajes:

- `/start` (o pulsar el botón Iniciar)
- `/newbot`
- (Nombre del bot)
- (Alias del bot, acabado en "bot")

@BotFather responderá lo siguiente al último mensaje:

Done! Congratulations on your new bot. You will find it at

**t.me/ALIAS\_DE\_TU\_BOT**

You can now add a description, about section and profile picture for your bot, see /help for a list of commands.

By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it.

Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

**00000000:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX**

Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page: <https://core.telegram.org/bots/api>

Hay que copiar el token que nos ofrece @BotFather, y copiarlo en el archivo **JSON/token.json** así:

```
{ "token": "00000000:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" }
```

Si se desea, se puede volver a @BotFather, enviar `/setDescription` en un primer mensaje, y la descripción deseada en un segundo mensaje.

El último paso antes de instalar paquetes y ejecutar el bot, es crear un Super Usuario.

Para añadir al primer Super Usuario es necesario acceder manualmente a **JSON/SuperUser.json** y añadir nuestro ID al array que se encuentra en el archivo.

Para obtener este ID hay que abrir una conversación en Telegram con @userinfobot. Al hacerlo, nos dará nuestra información de usuario donde, entre otras cosas, estará nuestro ID.

## Instalar paquetes y ejecutar el bot

Ejecutar **install.sh** para instalar todos los paquetes necesarios y poner el bot en ejecución.

Nota: Si se desea ejecutar desde consola primero aplicar permisos de ejecución al archivo. `$ chmod +x install.sh`

Para Ubuntu 16.04 LTS o superior

Esto pondrá el bot en ejecución. Lo primero que hay que hacer es añadir respuestas siendo Super Usuario. De lo contrario el bot recibirá los insultos pero no será capaz de responder a ellos.

## Organización de archivos

Este repositorio contiene:

- *install.sh* (instala paquetes y ejecuta el bot)
- *package.json* (referente a Node.js)
- *package-lock.json* (referente a Node.js)
- *fierobot.js* (contiene el código principal del bot, con todas sus funciones brevemente explicadas)  
Se recomienda no introducir en "*fierobot.js*" ninguna cadena de texto que trate explícitamente el token, nombre o alias del bot.
- *JSON/token.json* (contiene el token del bot)
- *JSON/SuperUser.json* (contiene los ID de los Super Usuarios)
- *JSON/StopWords.json* (contiene las palabras que no aportan un significado especial a la hora de tratar el lenguaje)
- *JSON/Insultos.json* (contiene los insultos recogidos y su frecuencia)
- *JSON/RespuestasGenericas.json* (contiene una lista de respuestas cuando no podemos identificar la frase que nos envía el usuario)
- *JSON/RespuestasPredefinidas.json* (contiene una lista con una frase o parte de una frase que el usuario envía, y su respuesta asociada siendo clave-valor, frase-respuesta)
- *JSON/Historial.json* (contiene los datos recibidos en cada mensaje)
- *JSON/Hombres.json* (historial filtrado por hombres)
- *JSON/Mujeres.json* (historial filtrado por mujeres)
- *JSON/Mayores.json* (historial filtrado por mayores de edad)
- *JSON/Menores.json* (historial filtrado por menores de edad)
- *JSON/HombresMayores.json* (historial filtrado por hombres mayores de edad)
- *JSON/MujeresMayores.json* (historial filtrado por mujeres mayores de edad)
- *JSON/HombresMenores.json* (historial filtrado por hombres menores de edad)
- *JSON/MujeresMenores.json* (historial filtrado por mujeres menores de edad)
- *CSV/ \** (contiene un archivo .csv por cada uno de los 12 archivos .json anteriores)
- *Docker/ \** (contiene un archivo README.md detallando los archivos que se encuentran en ese directorio, en caso de querer ejecutar el bot utilizando Docker)