



# FACULTAD DE CIENCIAS INTELIGENCIA ARTIFICIAL

---

## Chatbot con ChatterBot

---

Equipo: Skynet Scribes

Número de practica: 01

**Carlos Daniel Cortés Jiménez**  
420004846

**Sarah Sophía Olivares García**  
318360638

**Marco Silva Huerta**  
316205326

**Juan Daniel Barrera Holan**  
417079372

**Laura Itzel Tinoco Miguel**  
316020189

Profesora: Cecilia Reyes Peña

Ayudante teoría: Karem Ramos Calpulalpan

Ayudante laboratorio: Tania Michelle Rubí Rojas

**Semestre 2024-2**

Fecha de entrega:  
**14 de Febrero del 2024**

# 1. Investigación

## ChatterBot

La biblioteca ChatterBot de Python está diseñada para responder de manera automatizada a las entradas que de él usuario, con la ayuda de varios algoritmos de aprendizaje automático que generan una variedad de respuestas, lo que permite que al desarrollar chatbots se puedan dar respuestas apropiadas en diferentes contextos. El chatbot también va a poder mejorar su rendimiento conforme aprende con el paso del tiempo, presenta independencia de lenguaje, de esta manera se puede entrenar para hablar en cualquier idioma

### 1.1. Funcionamiento

Un chatbot no entrenado de ChatterBot inicia sin conocimiento de cómo debe comunicarse con el usuario, se pueden introducir frases y respuestas para que el chatbot aprenda qué debe responder en cada ocasión, cada vez que el usuario ingresa una entrada el chatbot guarda las entradas y respuestas con las que va tratando y con esos mismos datos genera respuestas automatizadas relevantes, cuando recibe una nueva entrada, compara la nueva entrada con datos anteriores, así chatbot puede seleccionar una respuesta que esté vinculada a la entrada conocida más cercana posible.

### 1.2. Capacidades

Como se mencionó anteriormente, algunas de sus capacidades es son el aprendizaje automático así como la independencia de lenguaje. Entre otras de sus capacidades están:

- **Almacenamiento y consulta de datos**

ChatterBot cuenta con con módulos llamados storage adapters los cuales permiten interactuar con el sistema de almacenamiento y proporcionan una abstracción, en su version 1.0.8 tiene los siguientes storage adapters; mongodb, sql\_storage, django\_storage. De esta manera permite mantener un registro de las interacciones y facilita la gestión de datos.

- **Procesamiento de lenguaje natural(NLP siglas en ingles)**

ChatterBot utiliza técnicas de procesamiento de lenguaje natural para poder entender y responder a los mensajes de los usuarios, de esta manera analiza las preguntas y comentarios de los usuarios para generar respuestas relevantes y coherentes.

- **Personalización**

Permite al desarrollador personalizar las respuestas y entrenar al chatbot con datos específicos para que responda de manera mas precisa y personalizada.

### 1.3. Limitaciones

**Falta de inteligencia emocional:** Los chatbot presentan dificultad para comprender las emociones humanas, sarcasmo... Lo cual puede representar un problema a la hora de interpretar el objetivo de las preguntas.

**Dependencia de los datos de entrenamiento:** La calidad y variedad de los datos de entrenamiento de ChatterBot influyen en gran medida con la calidad de las respuestas que genere. Los datos sesgados o incompletos pueden dar lugar a respuestas incorrectas o sesgos involuntarios en el comportamiento del chatbot. Por

lo cual depende mucho de un entrenamiento adecuado para que brinde respuestas mas precisas.

**Incapacidad para afrontar escenarios inesperados:** Al igual que otros chatbots, ChatterBot funciona siguiendo reglas y patrones predefinidos, cuando se enfrenta a situaciones que no se espera, lo mas probable es que no sepa como responder de manera adecuada, de respuestas fuera de contexto e incluso que responda con información poco precisa, por lo cual la base de datos del ChatterBot requiere actualizarse de manera frecuente.

**Limitaciones en el contexto:** Dado que ChatterBot en su versión 1.0.8, solo trabaja con el contexto de la entrada anterior esto puede ser un problema para seguir el hilo de una conversación por ejemplo.

## 2. Propósito

El propósito principal del chatbot es simular una veterinaria, brindando un servicio educativo e informativo sobre animales domésticos, principalmente de perros. Busca ser una herramienta que ayude a los usuarios a comprender mejor cómo cuidar adecuadamente a sus mascotas.

Continuando con la idea de simular una veterinaria tenemos las siguientes características en nuestro programa:

- Información sobre perros
- Adopta un gato
- Información sobre la veterinaria
- Datos curiosos sobre animales en el mundo
- Pequeño cuestionario para una vacante en nuestra veterinaria

### Funcionalidades básicas:

- **Mensaje de saludo:** El chatbot debe ser capaz de saludar a los usuarios de manera amigable, utilizando una variedad de saludos para crear una experiencia acogedora desde el inicio de la interacción.
- **Descripción de servicios:** Proporciona una descripción clara y concisa de los servicios que ofrece nuestra veterinaria
- **Datos curiosos:** Proporciona datos curiosos sobre animales de la vida salvaje con el fin que puedas descubrir un nuevo mundo
- **Datos de Veterinaria:** Horario y servicios y precios
- **Datos sobre Perros:** Información sobre perros (cuidados, entrenamiento, alimentación).
- **Información para adoptar un gato:** Simulación para la adopción de un gato y conocer sobre sus cuidados
- **Empleo en la Veterinaria:** Pequeña simulación sobre una entrevista
- **Despedida cordial:** Al finalizar la interacción manda un mensaje de despedida cordial, agradeciéndoles por utilizar el chatbot.

### 3. Documentación

#### 3.1. Organización del proyecto

Dentro del archivo .zip, el proyecto se compone de la siguiente forma

■ Carpeta Principal: PRACTICA01	Carpeta Principal: PRACTICA01
• Carpeta: src	Archivos ejecutables para el bot
○ Carpeta: venv	Contiene la configuración del entorno
○ Carpeta: templates	Contiene el front del chatbot
◇ HTML: index.html	HTML: index.html
○ Carpeta: conversaciones	Archivos .json de cada funcionalidad
○ PYTHON: chat.py	Lógica del programa
○ BD : db.sqlite3	Guarda el <i>aprendizaje</i> del bot
• PDF : practica01.pdf	PDF : practica01.pdf

#### 3.2. Como ejecutar Chat Dog

Antes de iniciar a conversar con Chat Dog necesitamos esto en nuestra computadora:

1. Iniciamos nuestra terminal
2. Revisamos la versión de python para evitar conflictos con el entorno.

```
1 Skynet : $ python --version
2 Skynet : $ Python 3.7.9
```

3. Navegar hasta la carpeta donde esta el archivo bot.py

```
1 Skynet : $ cd Practica01/src
2 Skynet : /src $ ls
3 bot.py  conver.json  templates  venv
```

4. Vamos a instalar flask para que nuestro bot se vea correctamente en la web

```
1 Skynet : /src $ pip install flask
```

- (Se descargarán los archivos necesarios)

5. Lo que sigue es activar nuestro entorno, primero crearemos la carpeta

- Para Windows

```
1 Skynet : /src $ python -m venv venv
```

- Para Linux

```
1 Skynet : /src $ python -m venv venv
```

6. Ahora vamos a activar el entorno

- Para Windows

```
1 Skynet : /src $ venv\Scripts\activate
2 (venv) Skynet : /src $
```

■ Para Linux

```
1 Skynet : /src $ source_venv/bin/activate
2 (venv) Skynet : /src $
```

■ Sabremos que esta activo el entorno al ver del lado izquierdo: (venv)

## 7. Ahora instalaremos CharrterBot

■ Para Windows

```
1 (venv) Skynet : /src $ python -m pip install chatterbot==1.0.4 pytz
2 (venv) Skynet : /src $
```

■ Para Linux

```
1 (venv) Skynet : /src $ python -m pip install chatterbot==1.0.4 pytz
2 (venv) Skynet : /src $
```

## 8. El siguiente paso es ejecutar el bot con el comando: `python bot.py`

```
1 (venv) Skynet : /src $ ls
2 bot.py  conver.json  templates  nenv
3 (venv) Skynet : /src $ python bot.py
```

■ A continuación veremos varias lineas como estas

```
1 (venv) Skynet : /src $ python bot.py
2 [nltk_data] Downloading package averaged_perceptron_tagger to
3 [nltk_data] C:\Users\youruser\AppData\Roaming\nltk_data...
4 List Trainer: [#####] 100%
5 * Serving Flask app 'bot'
6 * Debug mode: on
```

■ La siguiente linea es importante porque es el puerto donde verás a Chat-Dog

```
1 * Running on http://127.0.0.1:5000
```

■ Tal vez en tu computadora sea otro puerto, pero así debe vers. Copia y pega en tu navegador de preferencia (`http://127.0.0.1:5000`)

```
1 * Running on http://127.0.0.1:5000
2 [nltk_data] Package stopwords is already up-to-date!
3 List Trainer: [#####] 100%
4 List Trainer: [#####] 100%
5 List Trainer: [#####] 100%
6 List Trainer: [#####] 100%
7 List Trainer: [#####] 100%
8 * Debugger is active!
```

## 9. Esto te deberá permitir ver la interfaz de Chat-Dog

## 10. Y listo podrás empezar a conversar con él

## 11. Para terminal el proceso basta con que en la terminal ejecutes: `Ctrl+C`

```
1 * Running on http://127.0.0.1:5000
2 [nltk_data] Package stopwords is already up-to-date!
3 List Trainer: [#####] 100%
4 List Trainer: [#####] 100%
5 * Debugger is active!
6
7 Ctrl+C
8 (venv) Skynet : /src $
```

## 4. Funcionamiento

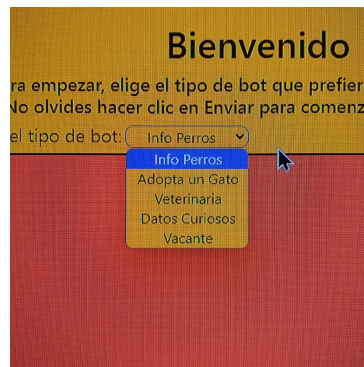
### 4.1. Info Perros

Este bot se le han cargado datos informativos sobre los perros, un ejemplo de ejecución es el siguiente con las líneas:

- Hola
- ¿Cuánto ejercicio necesita mi cachorro?
- ¿Cuál es la mejor forma de entrenar a mi perro?



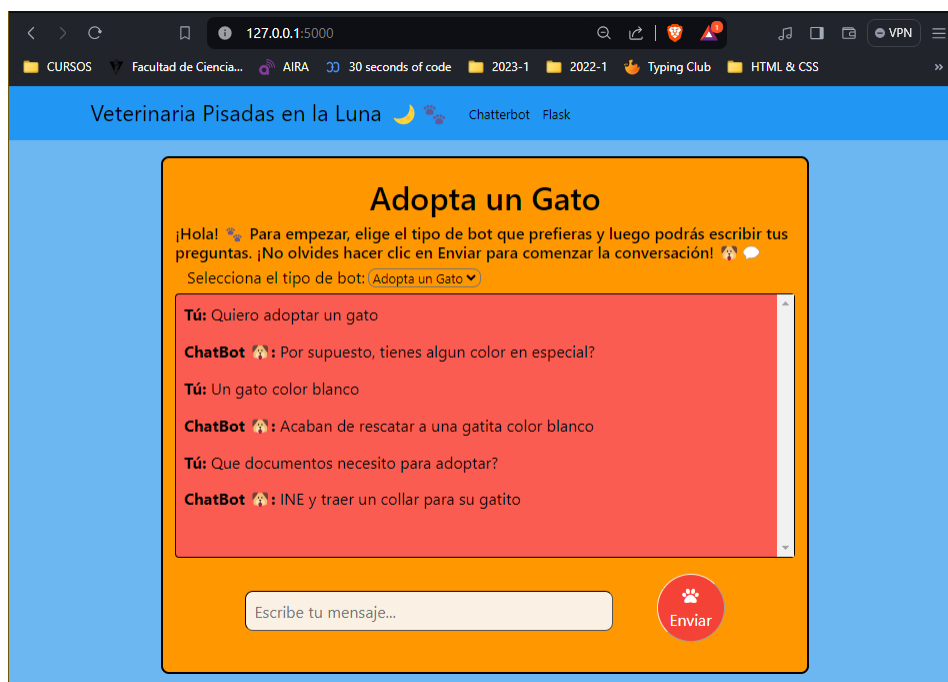
Podemos notar que la pagina tiene un menú seleccionable.  
Dependiendo de con que bot queramos conversar, será el que seleccionemos.  
Al cambiar de bot se va a reiniciar el char



## 4.2. Adopta un gato

Este bot se le han cargado datos para que un usuario por medio del color de gato que escoja pueda adoptar un gato y conocer sobre su nueva mascota. Un ejemplo de ejecución es el siguiente con las líneas:

- Quiero adoptar un gato
- Un gato color blanco
- Que le doy de comer a mi gato?



## 4.3. Veterinaria

Este bot se encarga de dar datos relevantes sobre la veterinaria. Un ejemplo de ejecución es el siguiente con las líneas:

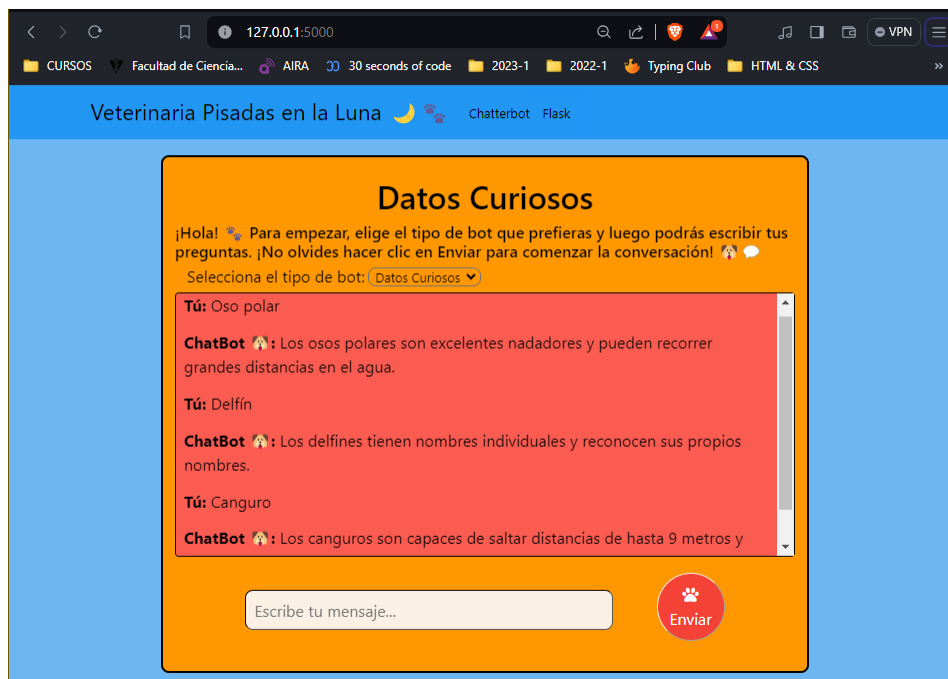
- ¿Que servicios ofrecen?
- ¿Cuáles son sus costos?
- ¿Costo de vacunas?



#### 4.4. Datos Curiosos

Con este bot te llenaras de información sobre curiosidades de animales al redor del mundo. Un ejemplo de ejecución es el siguiente con las líneas y basta con poner el nombre del animal

- Oso polar
- Delfín
- Canguro

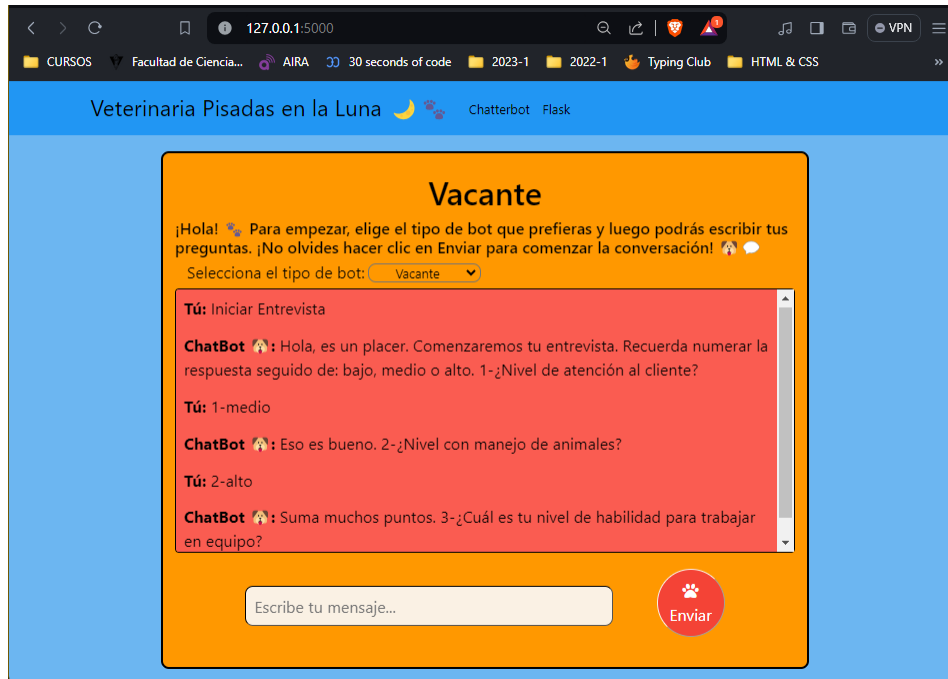




## 4.5. Vacante

Ahora simularemos una entrevista para entrenar al equipo de trabajo de la veterinaria, para comenzar escribir: *Iniciar Entrevista* y para cada pregunta se debe numerar, seguido un guion medio y una de estas tres respuestas según sea el caso: *bajo, medio o alto*. Un ejemplo de ejecución es el siguiente con las líneas:

- Iniciar Entrevista
- 1-medio
- 2-alto



## Referencias

- [Cox21] Gunther Cox. *ChatterBot*. <https://chatterbot.readthedocs.io/en/stable/>. 2021.
- [Pat21] Maria do Patrocinio Alfonso Folhento. «Aplicacions multimedia per e-learning». En: (2021).
- [Bre22] Martin Breuss. *ChatterBot: Build a Chatbot With Python*. <https://realpython.com/build-a-chatbot-python-chatterbot/>. 2022.
- [Dyo22] Abdelhadi Dyourri. *How To Make a Web Application Using Flask in Python 3*. <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3>. 2022.
- [Lin23] Pohan Lin. *How to Develop Your Own Chatbot With Python and ChatterBot from Scratch*. <https://www.codemotion.com/magazine/ai-ml/develop-chatbot-with-python-and-chatterbot/>. 2023.
- [Sah23] Kiran Kumar Sahu. *Benefits and Limitations of Using AI chatbots*. <https://medium.com/@kiran2002sk/benefits-and-limitations-of-using-ai-chatbots-7eed242b7a5f>. 2023.
- [Nav24] Avinash Navlani. *Building a Chatbot using Chatterbot in Python*. <https://www.codemotion.com/magazine/ai-ml/develop-chatbot-with-python-and-chatterbot/>. 2024.
- [Ove24] Overleaf. *Code listing*. [https://www.overleaf.com/learn/latex/Code\\_listing](https://www.overleaf.com/learn/latex/Code_listing). 2024.