



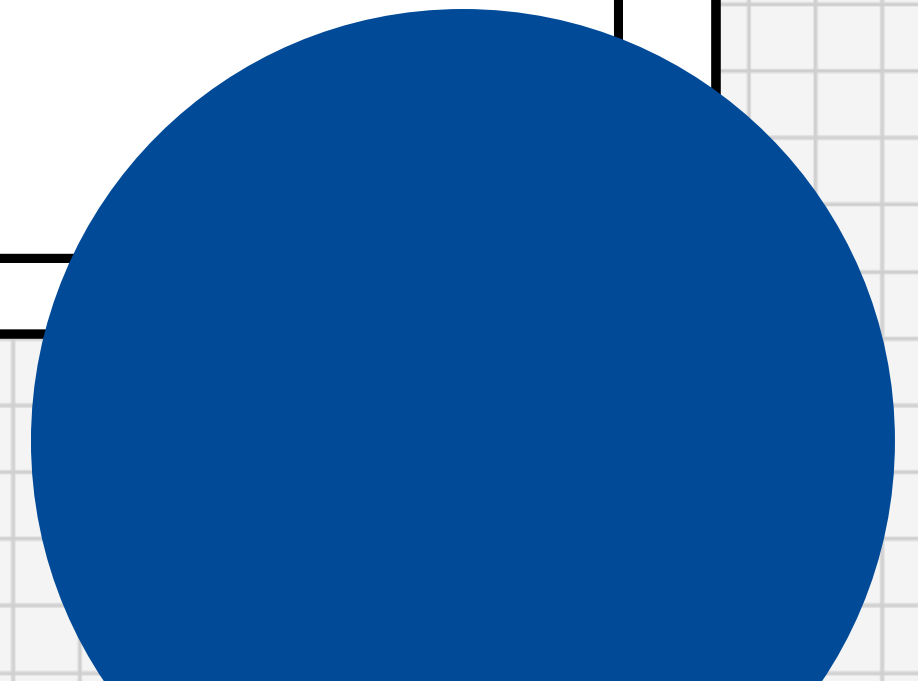
Por Aitana Menárguez Box

RAH



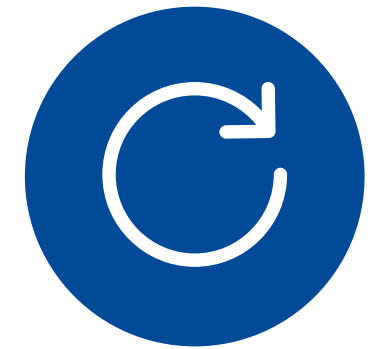
# Chatbot con RASA

Recomendación de películas



# Objetivos principales

motivación



1

- Elección de **película** de forma rápida

2

- Criterios de selección **personalizados**

3

- Aplicación **práctica** de tecnologías del habla

4

- Exploración de **posibilidades** que ofrece **RASA**

5

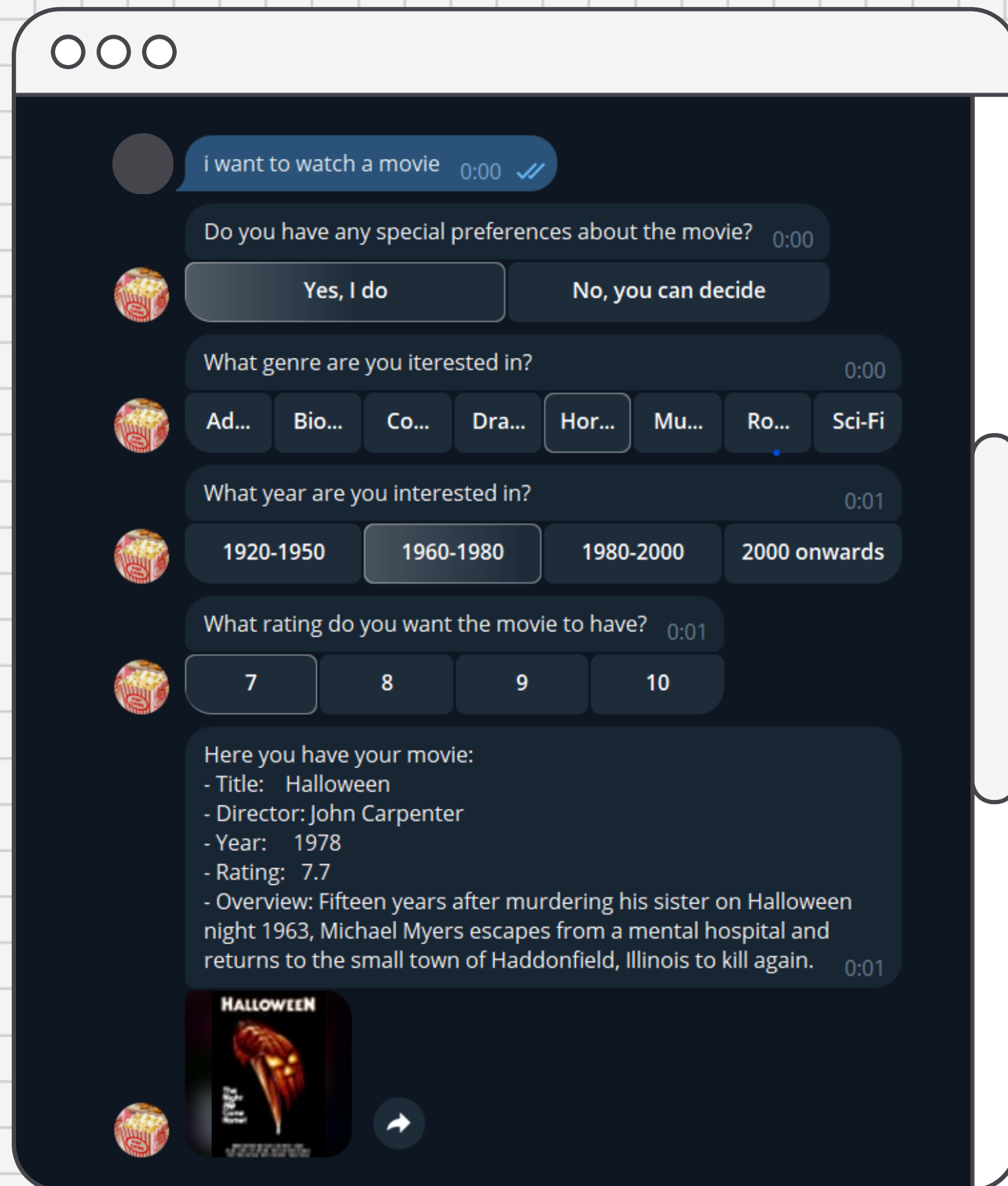
- Uso en plataformas externas: **Telegram**



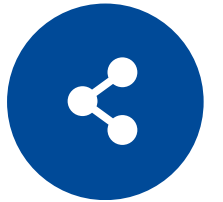
# Componentes básicos

Acciones y gestión de los datos





# Tipos de búsqueda



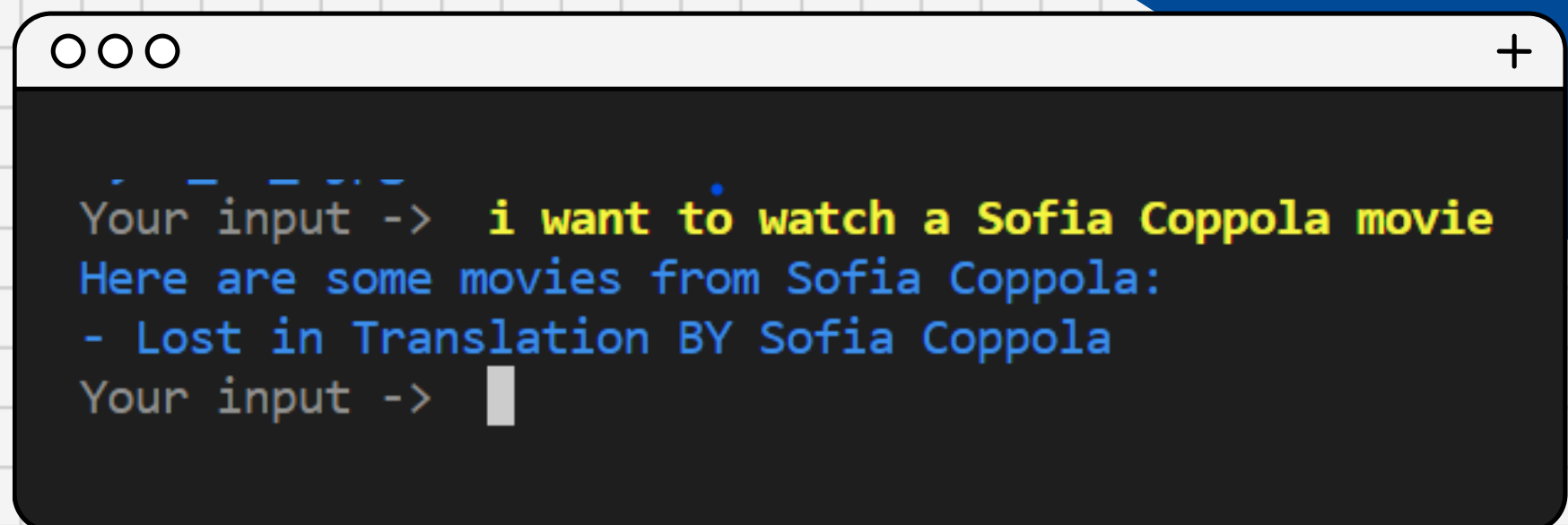
- Aleatoria



- Guiada por el *chatbot*



- Guiada por el usuario



nlu.yml - domain.yml



# Implementación en código

- Incorporación de **slots** a través de **entities** en los datos de *train*
  - Uso de **botones**
- **Stories** para **guiar** el tipo de búsqueda
- **actions.py** para recuperar la **información**

**Datos**

**¿De dónde sacar la información?**

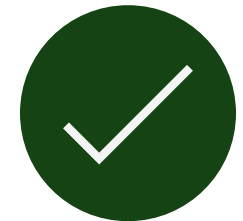


**API  
OMDb**  
[link](#)



**vs.**

**Fichero .csv  
IMDb dataset**  
[link](#)



Flexibilidad de la búsqueda, mantenimiento de datos

# Extractos del código

Implementación en Python

```
genre = tracker.get_slot("genre")
year = int(tracker.get_slot("year"))
rating = float(tracker.get_slot("rating"))
match_movies = []
for movie in MOVIESDB:
    mv_genre = movie[5]
    mv_year = int(movie[2])
```

```
- intent: genre_movie
examples: |
- Tell me some [action>{"entity": "genre", "value": "action"} movies
- I want to watch a [comedy>{"entity": "genre", "value": "comedy"} movie
- I want to watch a [drama>{"entity": "genre", "value": "drama"} movie
- I want to watch a [horror>{"entity": "genre", "value": "horror"} movie
- Any [romance>{"entity": "genre", "romance": "action"} movies?
```

```
intents:
- greet
- goodbye
- affirm
- deny
- mood_great
- mood_unhappy
- watch_movie
- genre_movie
- director_movie
- rating_movie
- year_movie
```

```
- story: genre 2
steps:
- intent: genre_movie
- slot_was_set:
- guided: False
- action: action_give_genre
```

```
actions:
- action_give_movie
- action_random_movie
- action_test
- action_give_genre
- action_give_year
- action_give_rating
- action_give_director
```

```
slots:
genre:
type: text
initial_value: null
mappings:
- type: from_entity
entity: genre
director:
type: text
initial_value: null
mappings:
- type: from_entity
entity: director
```



# Montaje de chatbot

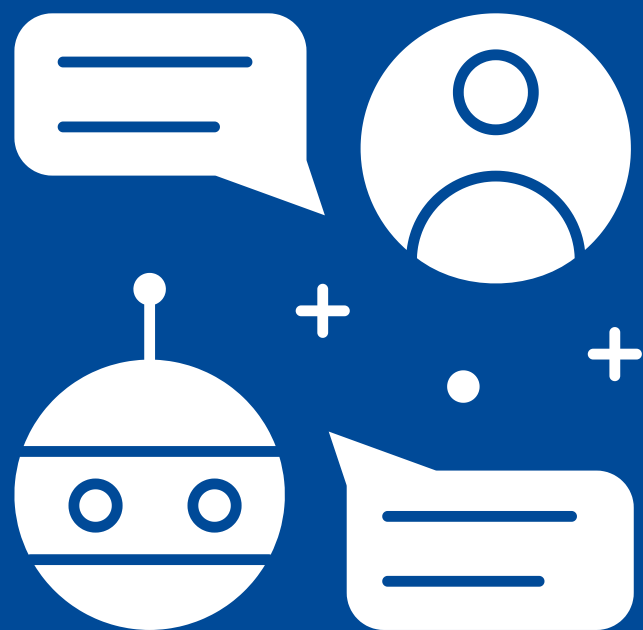
Puesta en marcha del modelo y uso





# Pasos a seguir

Infraestructura necesaria para poder utilizar el *chatbot*



***Entrenar modelo***

***Endpoint de acciones***

***Chatbot de Telegram***

**@BotFather**

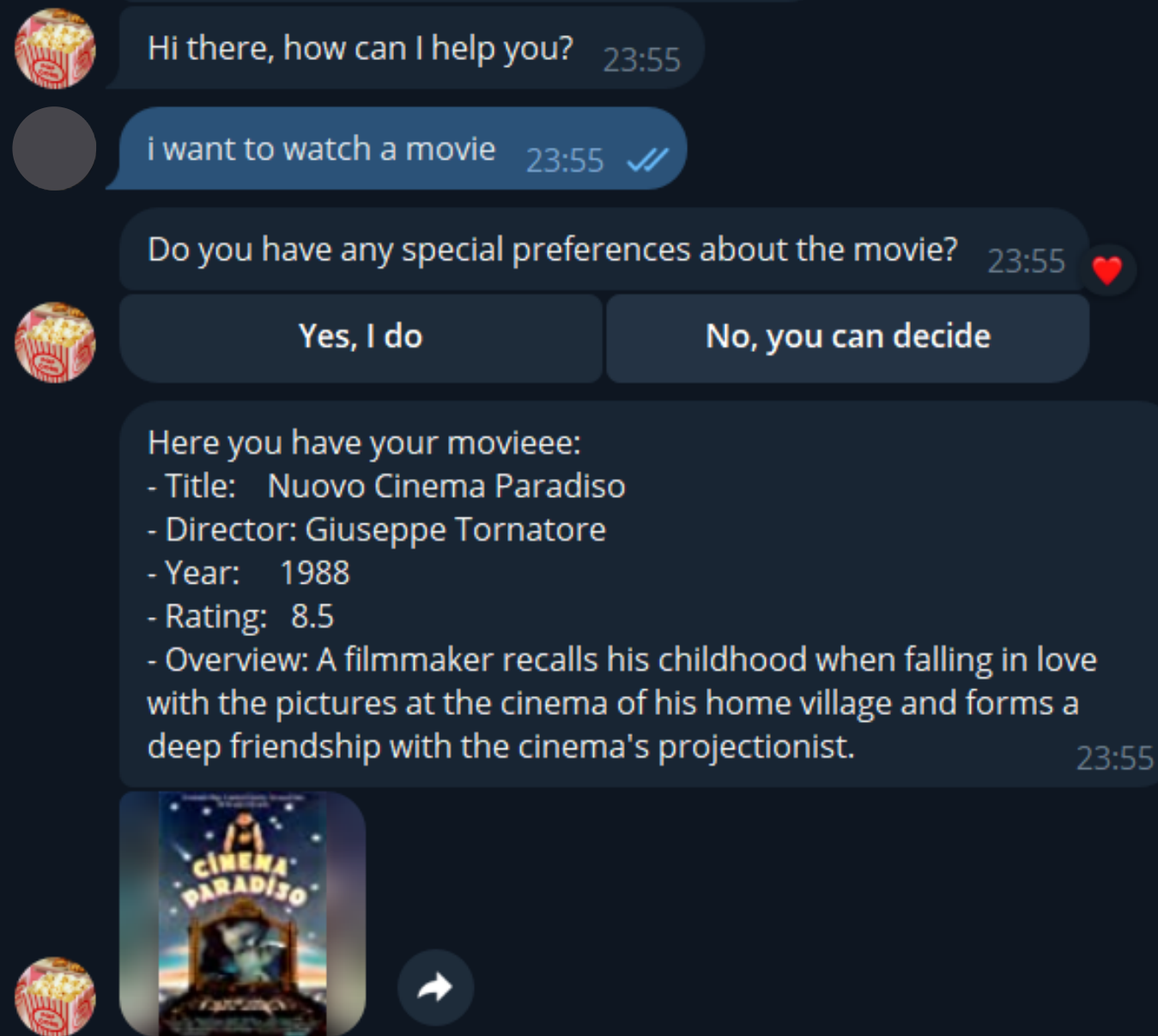
**Credentials.yml**

**Rasa run --enable-api**

**Entrar en [https://t.me/rah\\_movie\\_bot](https://t.me/rah_movie_bot)**

# Búsqueda aleatoria

Movie Chatbot 🔍



Ejemplo

# Búsqueda libre

Movie Chatbot 🔍



i want to watch a movie rated with a 8.5 0:26 ✓✓



Here are some movies rated with 8.5 or above:  
- Nuovo Cinema Paradiso BY Giuseppe Tornatore  
- The Godfather BY Francis Ford Coppola  
- The Matrix BY Lana Wachowski

0:26



i want to watch a movie from 1997 0:12 ✓✓



Here are some movies from year 1997:  
- Hana-bi BY Takeshi Kitano  
- Pâfekuto burû BY Satoshi Kon  
- L.A. Confidential BY Curtis Hanson

0:12

## Ejemplo



# Conclusión

Recapitulación y reflexión





## Conclusiones

- Incorporación de **diferentes tecnologías**
- Gran potencial de RASA
- Especialización por **ámbitos: equipo** de trabajo




## Trabajo futuro

- **Consulta** a una mejor base de datos, **API**
- **Funcionalidades:** plataformas de *streaming*
- Más **humanidad** al *chatbot*



## En definitiva

Aplicación práctica y útil de las tecnologías del habla



Gracias por vuestra  
atención :)