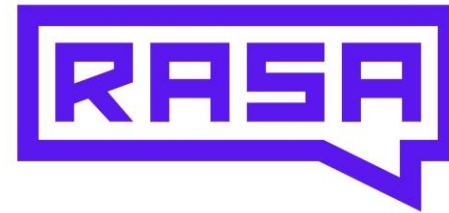
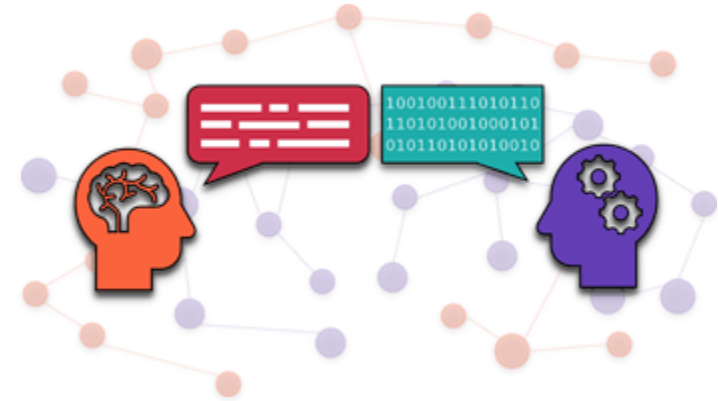


:: U3 ::



Procesamiento del Lenguaje Natural



3. Uso de modelos lingüísticos

Desarrollo de chatbots

Curso 2023-24

Tabla de contenidos

1. Introducción

2. Técnicas y modelos lingüísticos

a. Bag-of-words

b. TF-IDF

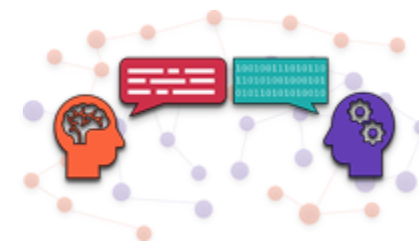
c. Topic Modeling

3. Modelos generativos

a. Word Embeddings

b. LangChain: integración con LLMs (Large Language Models)

c. Desarrollo de chatbots (RASA)



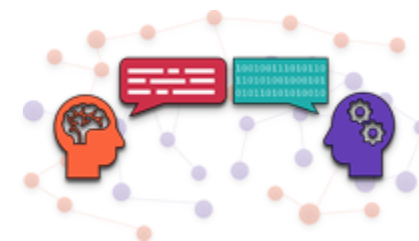
3c. Modelos generativos

1. ¿Qué es RASA?
2. Instalación y ejecución mediante comandos
3. Estructura de un proyecto
4. Configuración del chatbot: "Pipeline" y "Policies"
5. Gestión de desvíos en la conversación
6. Acciones ("Custom Actions")
7. Formularios ("Forms")



3c. Modelos generativos

1. ¿Qué es RASA?
2. Instalación y ejecución mediante comandos
3. Estructura de un proyecto
4. Configuración del chatbot: "Pipeline" y "Policies"
5. Gestión de desvíos en la conversación
6. Acciones ("Custom Actions")
7. Formularios ("Forms")



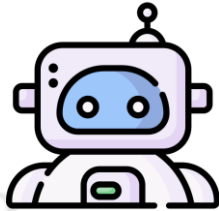
1. ¿Qué es **RASA** ?



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



Me gustaría reservar una habitación



De acuerdo, ¿para qué día?



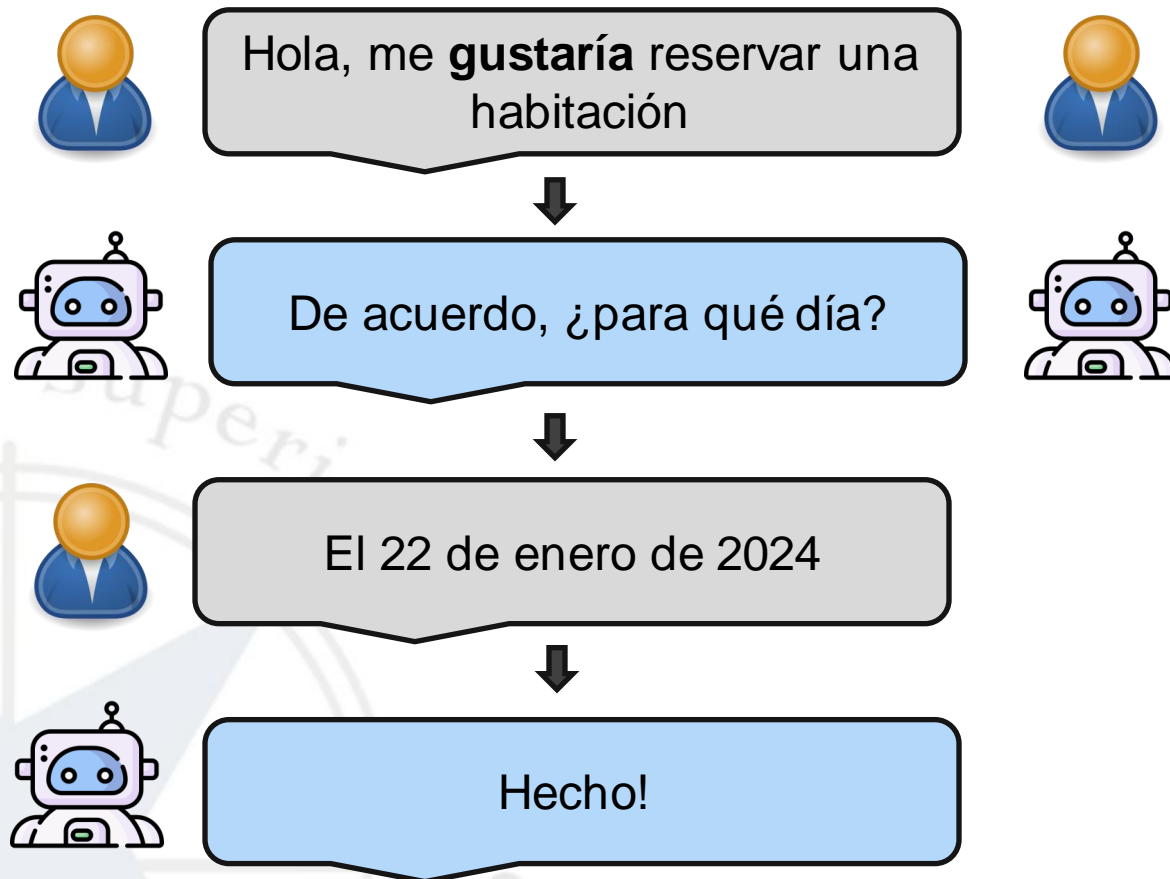
El 22 de enero de 2024



Hecho!



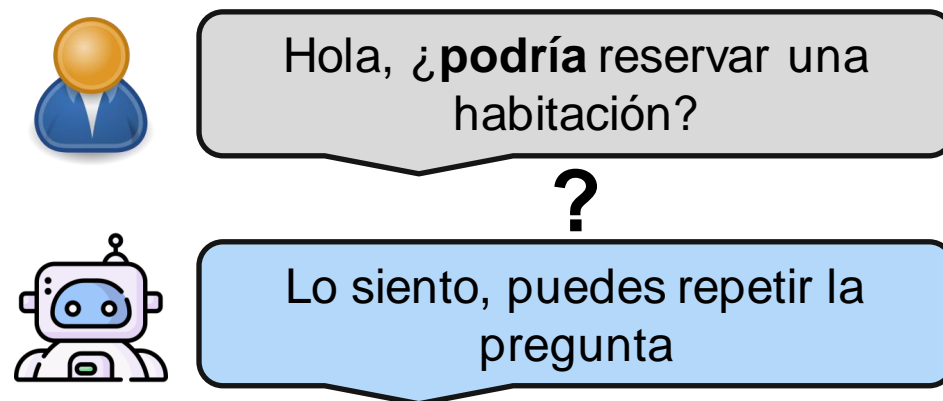
1. ¿Qué es **RASA** ?



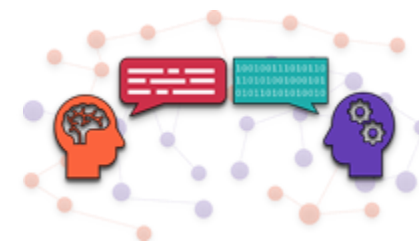
Hola, ¿**podría** reservar una habitación?



1. ¿Qué es **RASA** ?



¡En los chatbots de primera generación las locuciones debían ser "literales" o bien contemplar todas las posibilidades!



1. ¿Qué es **RASA** ?

- Los chatbots **antiguos** ...
 - a menudo utilizaban técnicas simples de coincidencia de patrones o palabras clave para interpretar las consultas de los usuarios.
 - a menudo ofrecían respuestas predefinidas y limitadas, lo que resultaba en interacciones rígidas y poco naturales.
 - carecían de capacidades de aprendizaje automático/profundo y, por lo tanto, no podían mejorar con el tiempo o con la retroalimentación del usuario.



1. ¿Qué es **RASA** ?

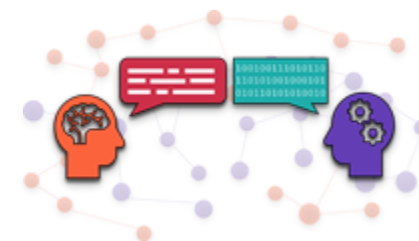
- Rasa es un framework para la creación de asistentes y chatbots, escrito en Python y de código abierto.
- Basa su funcionamiento en dos componentes principales:

> NLU

Parte encargada de **tomar el texto, analizarlo y descomponerlo** de tal manera que el bot comprenda el contenido del mensaje.

> Core

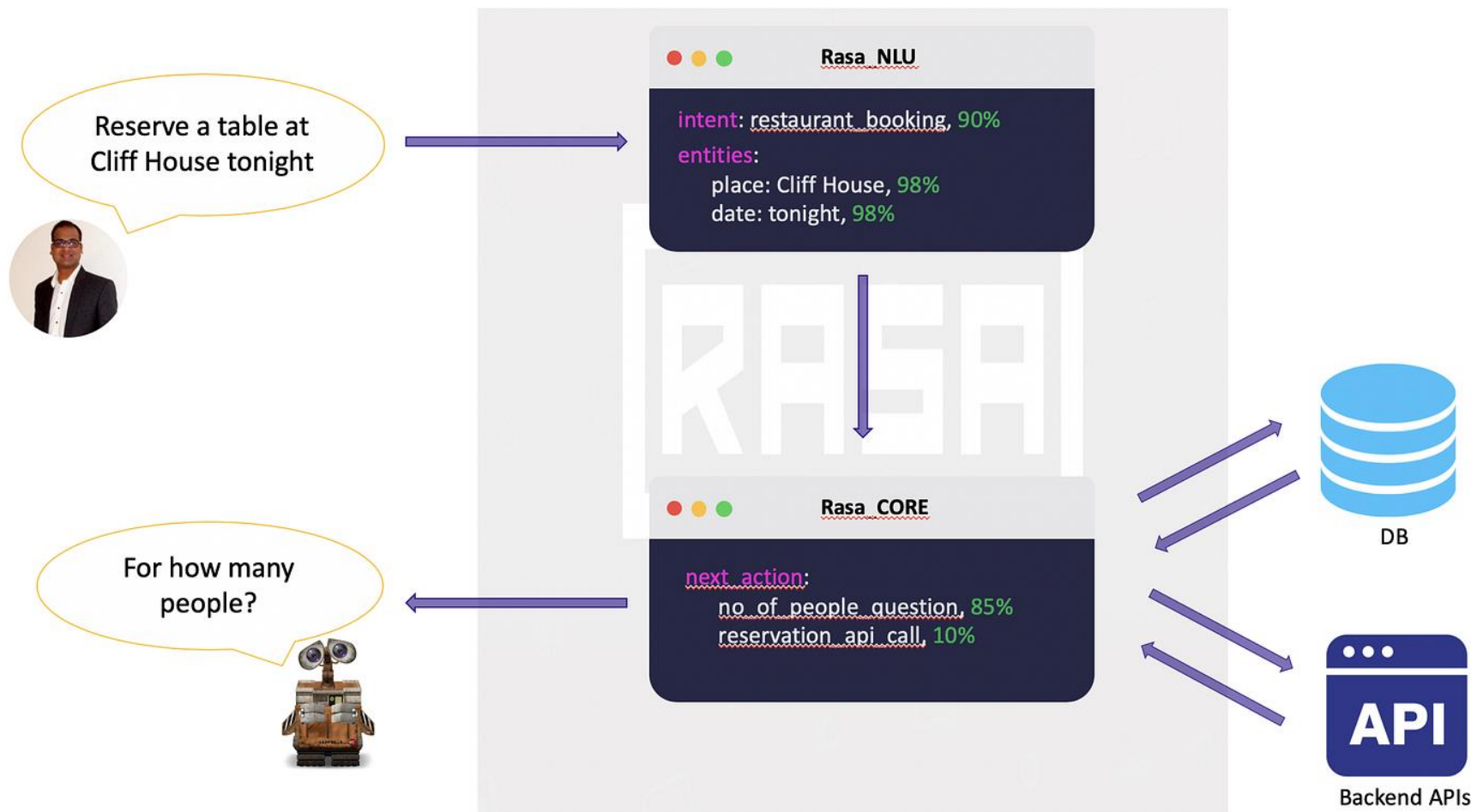
Parte encargada de **tomar decisiones y dar seguimiento a la conversación** según lo que NLU provee.



1. ¿Qué es **RASA** ?



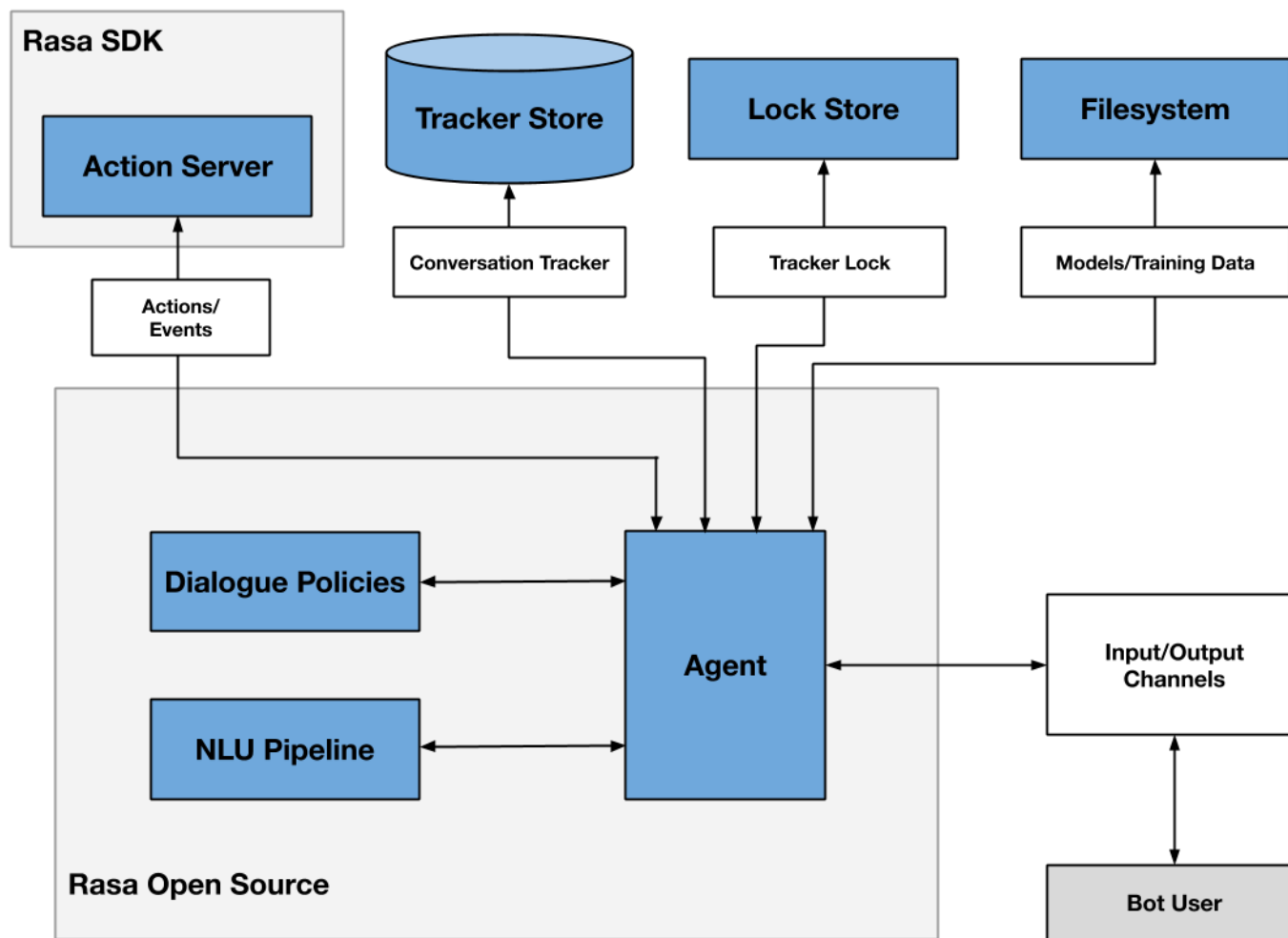
CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



1. ¿Qué es **RASA** ?



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



1. ¿Qué es **RASA** ?



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

**Rasa
Platform**



Rasa Pro

powered by Rasa Open Source

+



Rasa X/Enterprise

Rasa Pro license is required



Rasa Open Source



2. Instalación y ejecución

¿Cómo se instala **RASA**?

Instalación local

```
# sudo mkdir rasa_prueba  
# cd rasa_prueba  
# python3 -m venv /home/mia/rasa_prueba/  
# source /home/mia/rasa_prueba/bin/activate
```

+

```
# sudo python -m pip install --upgrade pip rasa
```



2. Instalación y ejecución

¿Cómo se ejecuta **RASA** ?

rasa init

Inicialización del entorno

rasa shell [nlu | core] [--debug]

Ejecución del entorno

rasa train [nlu | core]

Entrenamiento del modelo

rasa run actions

Manejo de acciones
personalizadas

rasa interactive

Entrenamiento interactivo



2. Instalación y ejecución

¿Cómo se instala **RASA**?

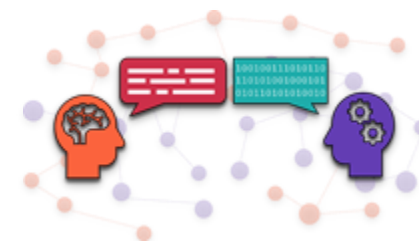
Instalación en Docker

Creación del contenedor para el proyecto

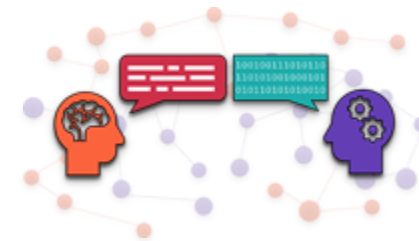
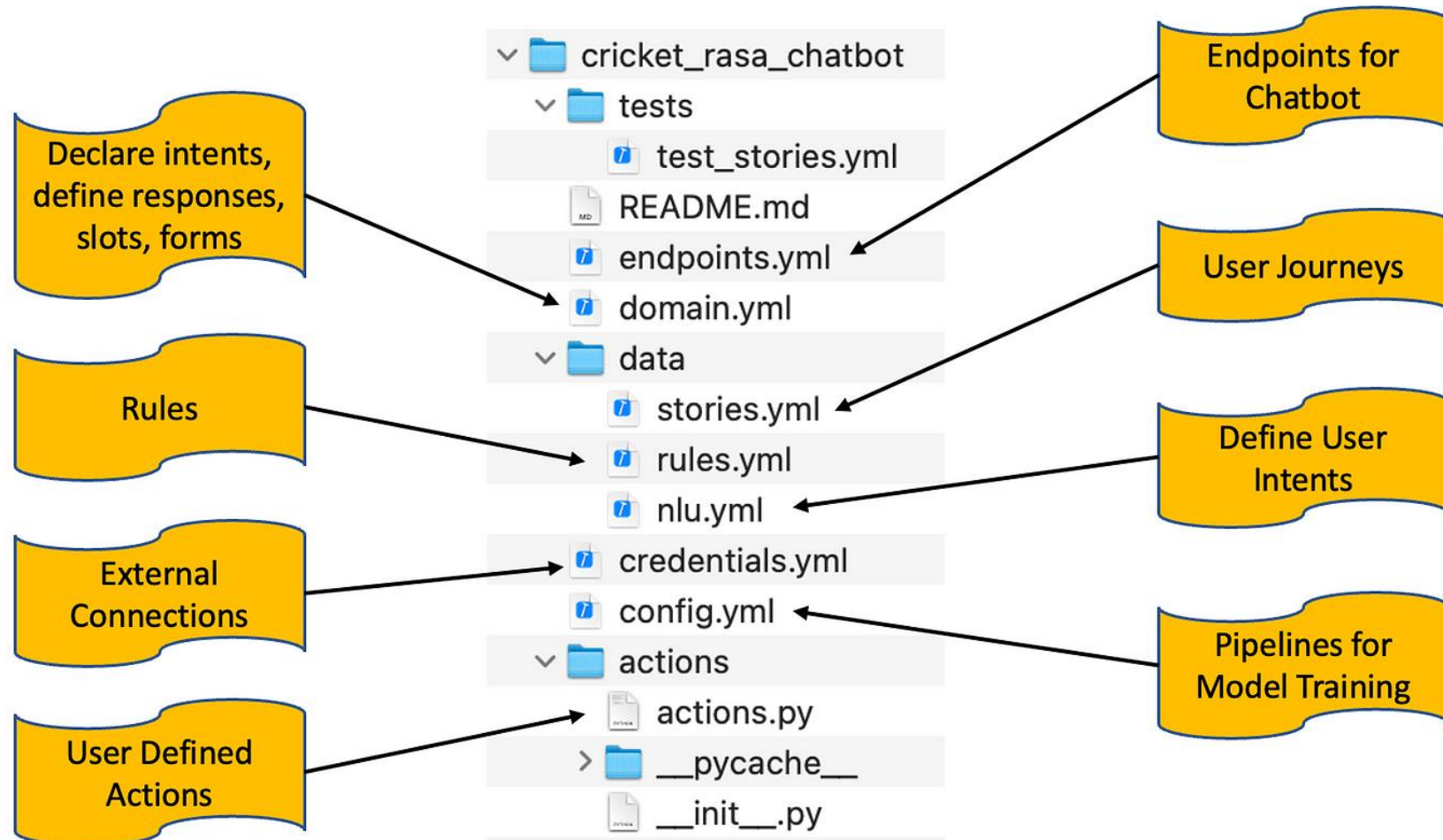
```
# mkdir rasa_docker  
# cd rasa_docker  
# sudo docker run -u $(id -u) -v $(pwd):/app  
rasa/rasa:3.0.0-full init --no-prompt
```

Ejecución del contenedor

```
# sudo docker run -it -v $(pwd):/app rasa/rasa:3.0.0-  
full shell
```



3. Estructura de un proyecto



3. Estructura de un proyecto

nlu.yml

- Define las intenciones ("intents")
- Una intención es una entrada por parte del usuario
- Las intenciones alimentan con datos los algoritmos de reconocimiento de texto.

```
version: "3.1"

nlu:
- intent: greet
  examples: |
    - Hey
    - Hi
    - hey there [Sara](name)

- intent: faq/language
  examples: |
    - What language do you speak?
    - Do you only handle english?
```



3. Estructura de un proyecto

stories.yml

- Una historia representa una conversación entre un usuario y un asistente de IA.
- Se compone de una secuencia de intenciones y acciones de respuesta.
- Se pueden utilizar para entrenar modelos que son capaces de generalizar a rutas de conversación invisibles.

```
version: "3.1"

stories:
- story: greet and faq
  steps:
  - intent: greet
  - action: utter_greet
  - intent: faq
  - action: utter_faq
```



3. Estructura de un proyecto



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

rules.yml

- Las reglas son fragmentos de conversación que siguen siempre el mismo curso en la conversación.
- A diferencia de las historias, el chatbot aplica siempre las reglas definidas.

```
version: "3.1"

rules:
- rule: Greet user
  steps:
  - intent: greet
  - action: utter_greet
```



3. Estructura de un proyecto

domain.yml

- El dominio define el universo en el que opera el asistente.
- Especifica las intenciones, entidades, ranuras, respuestas, formularios y acciones que el chatbot debe conocer.
- También incluye datos de configuración de las sesiones de conversación.

```
version: "3.1"

intents:
- affirm
- deny
- greet
- thankyou
- goodbye
- search_concerts
- search_venues
- compare_reviews
- bot_challenge
- nlu_fallback
- how_to_get_started

responses:
  utter_greet:
    - text: "Hey there!"
  utter_goodbye:
    - text: "Goodbye :("
  utter_default:
    - text: "Sorry, I didn't get that, can you rephrase?"
  utter_youarewelcome:
    - text: "You're very welcome."
  utter_iamabot:
    - text: "I am a bot, powered by Rasa."
  utter_get_started:
    - text: "I can help you find concerts and venues. Do you like music?"
  utter_awesome:
    - text: "Awesome! You can ask me things like \"Find me some concerts\" or \"What's a good venue\""

session_config:
  session_expiration_time: 60 # value in minutes
  carry_over_slots_to_new_session: true
```



4. Configuración del chatbot

config.yml

- El archivo de configuración define los componentes y políticas que su modelo utilizará para hacer predicciones basadas en la entrada del usuario.

```
language: en

pipeline:
- name: WhitespaceTokenizer
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100

policies:
- name: MemoizationPolicy
- name: TEDPolicy
  max_history: 5
  epochs: 10
- name: RulePolicy
```



4. Configuración del chatbot

```
language: en
```

```
pipeline:
```

- name: WhitespaceTokenizer
 - name: RegexFeaturizer
 - name: LexicalSyntacticFeaturizer
 - name: CountVectorsFeaturizer
 - name: CountVectorsFeaturizer
- ```
analyzer: char_wb
min_ngram: 1
max_ngram: 4
- name: DIETClassifier
epochs: 100
```

```
policies:
```

- name: MemoizationPolicy
  - name: TEDPolicy
- ```
max_history: 5  
epochs: 10  
- name: RulePolicy
```

El procesamiento de las
entradas de usuario se realiza
en base a:

← Pipeline ("flujo")

← Políticas ("policies")



4. Configuración del chatbot

```
language: en

pipeline:
- name: WhitespaceTokenizer
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100

policies:
- name: MemoizationPolicy
- name: TEDPolicy
  max_history: 5
  epochs: 10
- name: RulePolicy
```

El procesamiento de las entradas de usuario se realiza en base a:

← Pipeline ("flujo")

- Componentes que se usan para procesar y analizar los mensajes de entrada en un modelo de NLU
- Se organizan en un flujo secuencial, pasando a través de cada uno de ellos en orden.



4. Configuración del chatbot

```
language: en
```

```
pipeline:
```

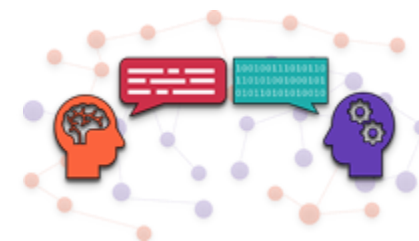
```
- name: WhitespaceTokenizer  
- name: RegexFeaturizer  
- name: LexicalSyntacticFeaturizer  
- name: CountVectorsFeaturizer  
- name: CountVectorsFeaturizer  
  analyzer: char_wb  
  min_ngram: 1  
  max_ngram: 4  
- name: DIETClassifier  
  epochs: 100
```

```
policies:
```

```
- name: MemoizationPolicy  
- name: TEDPolicy  
  max_history: 5  
  epochs: 10  
- name: RulePolicy
```

Un "**pipeline**" suele incluir algunos de estos componentes:

- **Tokenization:** Divide el texto en unidades más pequeñas llamadas "tokens".
- **Featurization:** Convierte los tokens en representaciones numéricas
- **Intent Classification:** Determina la intención detrás del mensaje del usuario.
- **Entity Recognition:** Identifica las entidades específicas dentro del mensaje, como fechas, ubicaciones, nombres, etc.
- ...



4. Configuración del chatbot

Ejemplo de pipeline basado en **SpaCy**:

```
language: "es"

pipeline:
  - name: "SpacyNLP"
  - name: "SpacyTokenizer"
  - name: "SpacyFeaturizer"
  - name: "CRFEntityExtractor"
  - name: "EntitySynonymMapper"
  - name: "SklearnIntentClassifier"
```



4. Configuración del chatbot

```
language: en

pipeline:
- name: WhitespaceTokenizer
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100

policies:
- name: MemoizationPolicy
- name: TEDPolicy
  max_history: 5
  epochs: 10
- name: RulePolicy
```

El procesamiento de las entradas de usuario se realiza en base a:

Políticas ("policies")

- Determinan cómo el modelo de RASA debe seleccionar la próxima acción en función del estado del diálogo actual y la historia pasada de interacciones.



4. Configuración del chatbot

```
language: en

pipeline:
- name: WhitespaceTokenizer
- name: RegexFeaturizer
- name: LexicalSyntacticFeaturizer
- name: CountVectorsFeaturizer
- name: CountVectorsFeaturizer
  analyzer: char_wb
  min_ngram: 1
  max_ngram: 4
- name: DIETClassifier
  epochs: 100

policies:
- name: MemoizationPolicy
- name: TEDPolicy
  max_history: 5
  epochs: 10
- name: RulePolicy
```

Algunas **políticas** disponibles:

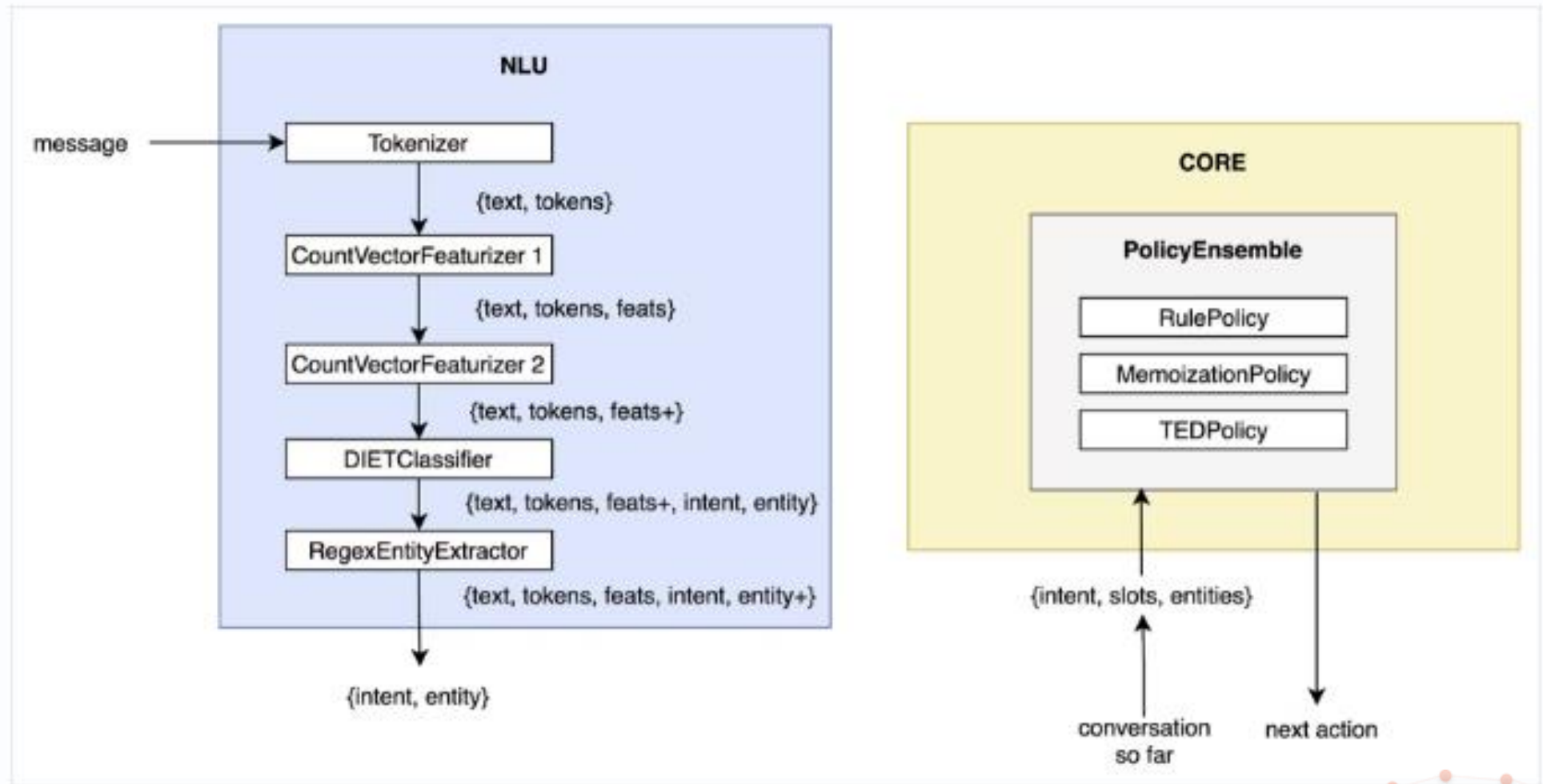
- **Memoization Policy:** Utiliza información pasada para predecir la próxima acción en función de su similitud
- **TED Policy:** Utiliza modelos basados en *transformers* para mejorar la toma de decisiones en el diálogo en función del contexto.
- **Rule Policy:** Define comportamientos basados en patrones de entrada para casos simples y específicos.
- **Fallback Policy:** Gestiona situaciones de incertidumbre
- ...



4. Configuración del chatbot



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



4. Configuración del chatbot



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

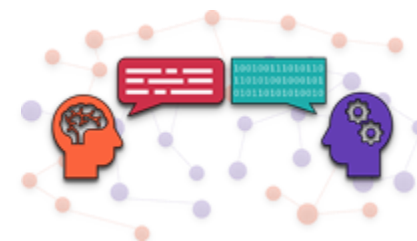
Realiza los siguientes ejercicios relacionados con la puesta en marcha de RASA:

- **Ejercicio 1: primera toma de contacto con RASA**
- **Ejercicio 2: creación de un chatbot (en español)**



5. Gestión de desvíos

- En el diseño de chatbots se hace necesario contemplar casos en los que las intervenciones del usuario se desvíen de lo esperado por el contexto.
- **RASA** incluye dos mecanismos para afrontar desvíos en las conversaciones:
 - a) **Mensajes "fuera de contexto"**
 - b) **Recuperaciones ("fallbacks")**



5. Gestión de desvíos

> Mensajes "fuera de contexto"

- Se refiere a situaciones en las que el sistema no puede entender la entrada del usuario porque carece de contexto o referencia relevante.
- Uso de **reglas**: tienen prioridad sobre los modelos de aprendizaje automático y se aplican sólo en casos donde ciertos desvíos deben manejarse de manera específica

PASOS PARA MANEJAR ESTOS CASOS

1. Crear una intención ("intent") asociada
2. Definir un mensaje de respuesta
3. Crear una regla asociada



5. Gestión de desvíos

1 Crear una intención ("intent") en **nlu.yml**

```
- intent: out_of_scope
  examples: |
    - quiero un helado
    - qué hora es?
    - hoy hace una caca de tiempo
    - quién es el presidente de Filipinas?
    - cuántas patas tiene un gato?
    - dos más dos suma cuatro
```



5. Gestión de desvíos



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

2 Definir un mensaje de respuesta en **domain.yml**



```
utter_out_of_scope:  
- text: No estoy seguro de entender. ¿Puedes reformular o proporcionar más información?
```



5. Gestión de desvíos

3 Crear una regla en **rules.yml**



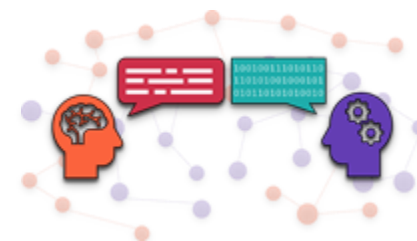
```
rules:  
- rule: out-of-scope  
  steps:  
  - intent: out_of_scope  
  - action: utter_out_of_scope
```



5. Gestión de desvíos

> Recuperaciones ("fallbacks")

- Mecanismo diseñado para manejar situaciones en las que el modelo NLU tiene poca confianza en la clasificación de un mensaje de usuario.
- Cuando un mensaje recibe una confianza baja al ser clasificado, el "fallback" entra en juego para garantizar que el chatbot maneje la situación de manera adecuada.

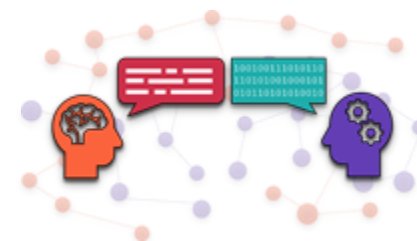


5. Gestión de desvíos

> Recuperaciones ("fallbacks")

PASOS PARA MANEJAR ESTOS CASOS

1. Agregar "FallbackClassifier" al "pipeline"
2. Crear una acción asociada al "fallback"
3. Añadir una regla asociada al "fallback"

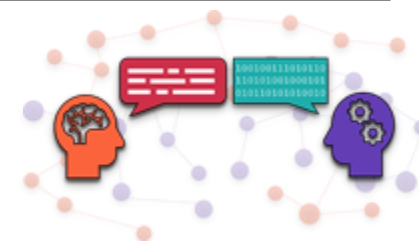


5. Gestión de desvíos

1 Agregar "FallbackClassifier" a **config.yml**

```
pipeline:  
  # other components  
  - name: FallbackClassifier  
    threshold: 0.7
```

- Umbral para mensajes de baja confianza en NLU
- En el ejemplo, cualquier clasificación por debajo de **0,7** activa el Fallback
- Intento de clasificación de intención desconocida se asocia a "**nlu_fallback**"



5. Gestión de desvíos

2 Definir una respuesta en **domain.yml**



```
utter_repetir_entrada:  
- text: No estoy seguro de entender. ¿Puedes reformular o proporcionar más información?
```



5. Gestión de desvíos

3 Añadir una regla en **rules.yml**



```
rules:  
- rule: Pedir al usuario para entradas con una confianza NLU baja  
  steps:  
  - intent: nlu_fallback  
  - action: utter_repetir_entrada
```

NOTA: En este ejemplo se supone que existiría la intención **nlu_fallback** previamente definida



5. Gestión de desvíos

```
Google™  
2024-01-05 12:59:48 INFO      rasa.core.processor - Loading model models/20240105-125342-blocky-  
document.tar.gz...  
NLU model loaded. Type a message and press enter to parse it.  
Next message:  
odio la fruta  
{  
  "text": "odio la fruta",  
  "intent": {  
    "name": "nlu_fallback",  
    "confidence": 0.5  
  },  
  "entities": [],  
  "text_tokens": [  
    0,  
    4  
  ],  
  ...  
},  
"intent_ranking": [  
  {  
    "name": "nlu_fallback",  
    "confidence": 0.5  
  },  
  {  
    "name": "despedida",  
    "confidence": 0.3064098060131073  
  },  
  {  
    "name": "aburrimiento",  
    "confidence": 0.1602540761232376  
  },  
],
```

Si entramos en modo interactivo (**# rasa shell nlu**) podemos comprobar en qué casos se activa "nlu_fallback"



5. Gestión de desvíos

... resultado al interactuar con el chatbot

```
2024-01-05 13:03:36 INFO      root - Starting Rasa server on http://0.0.0.0:5005
2024-01-05 13:03:36 INFO      rasa.core.processor - Loading model models/20240105-125342-blocky-
document.tar.gz...
2024-01-05 13:04:53 INFO      root - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> hola
¡Hola! ¿Cómo estás?
Your input -> odio la fruta
No estoy seguro de entender. ¿Puedes reformular o proporcionar más información?
Your input ->
```



5. Gestión de desvíos



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

... junto al mecanismo "FallbackClassifier" puede ser interesante definir una política expresamente que prediga la siguiente acción en base a la confianza ...



5. Gestión de desvíos

1

Se añaden los siguientes parámetros a "RulePolicy" en **config.yml**

```
polices:  
- name: RulePolicy  
  core_fallback_threshold: 0.4  
  core_fallback_action_name: "action_default_fallback"  
  enable_fallback_prediction: True
```



2

Se añade una respuesta "utter_default" en **domain.yml**

```
responses:  
  utter_default:  
    - text: Disculpa, ¿puedes ser más claro?
```

¡La acción desencadena esta respuesta!



5. Gestión de desvíos



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Realiza el siguiente ejercicio:

- **Ejercicio 3: Manejar desvíos en la conversación**



6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

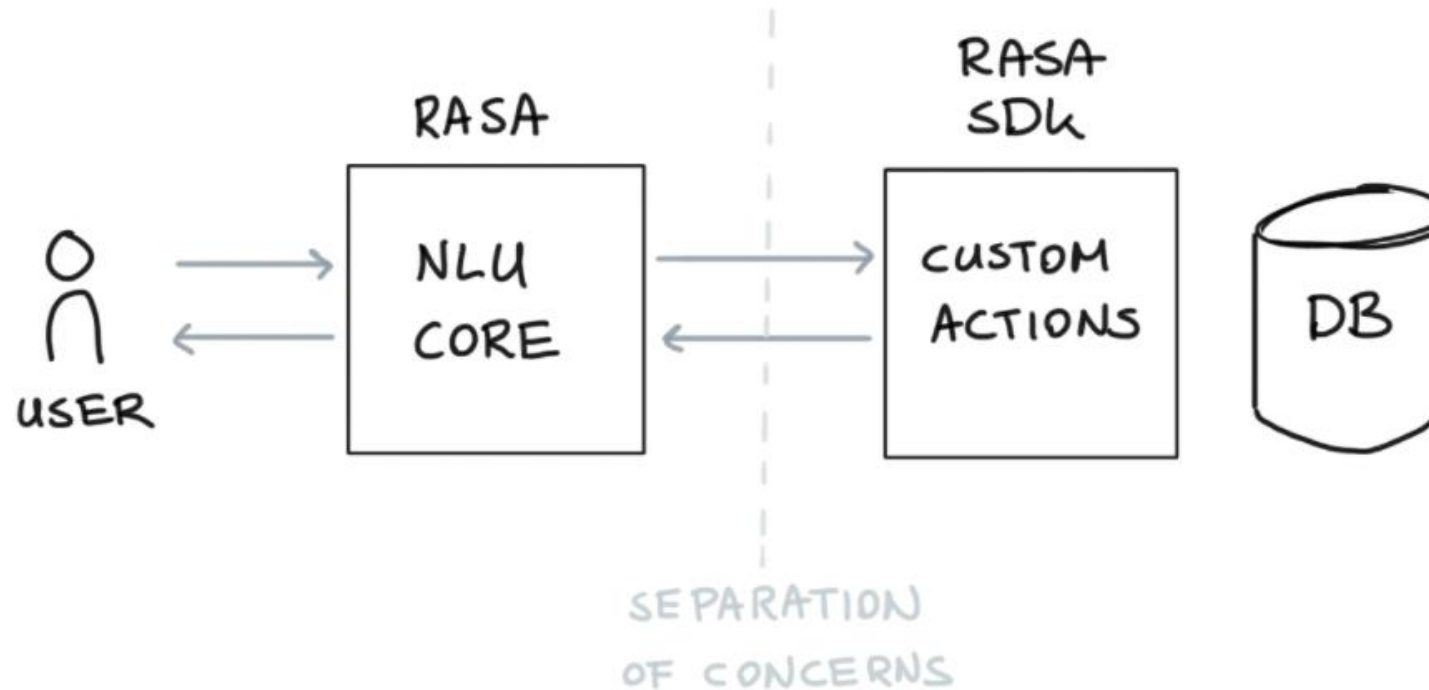
- Una **acción** personalizada permite ejecutar código Python como parte del flujo de la conversación.
- El código puede incluir llamadas API, consultas de bases de datos, etc.
- El límite es la imaginación: encender luces en un sistema IoT, añadir un evento a un calendario, comprobar el saldo bancario de un usuario, ...



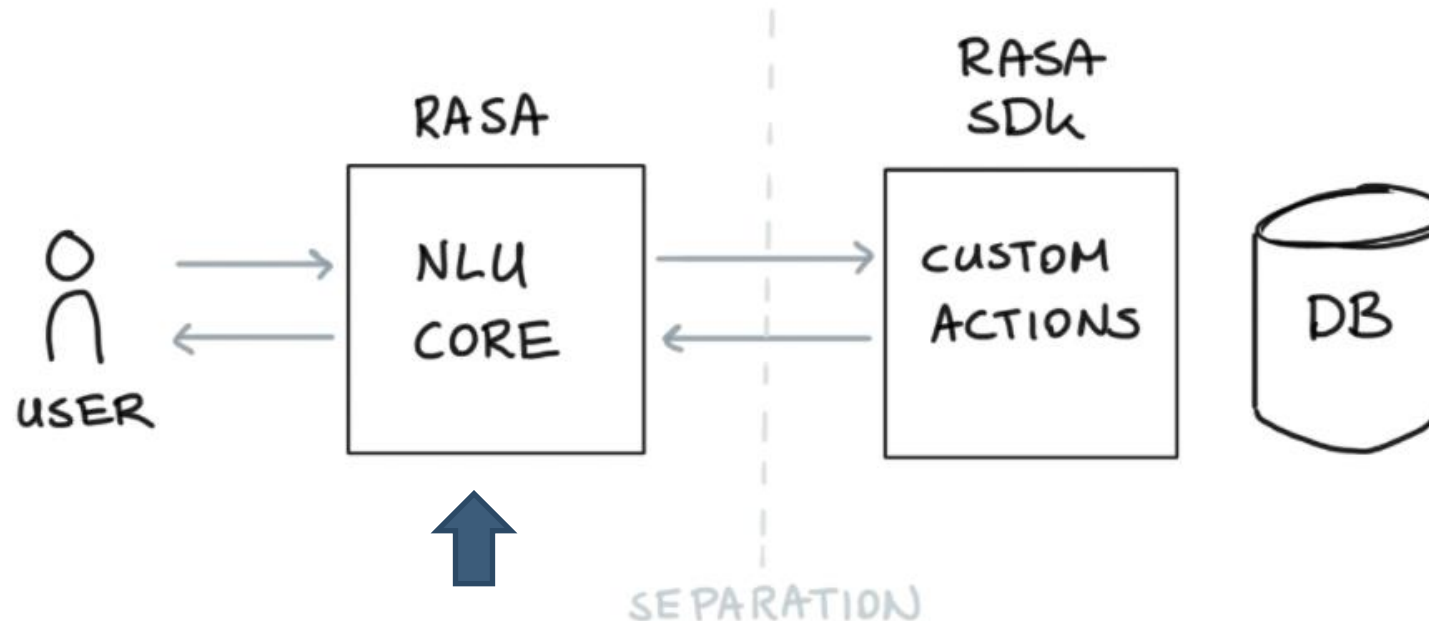
6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



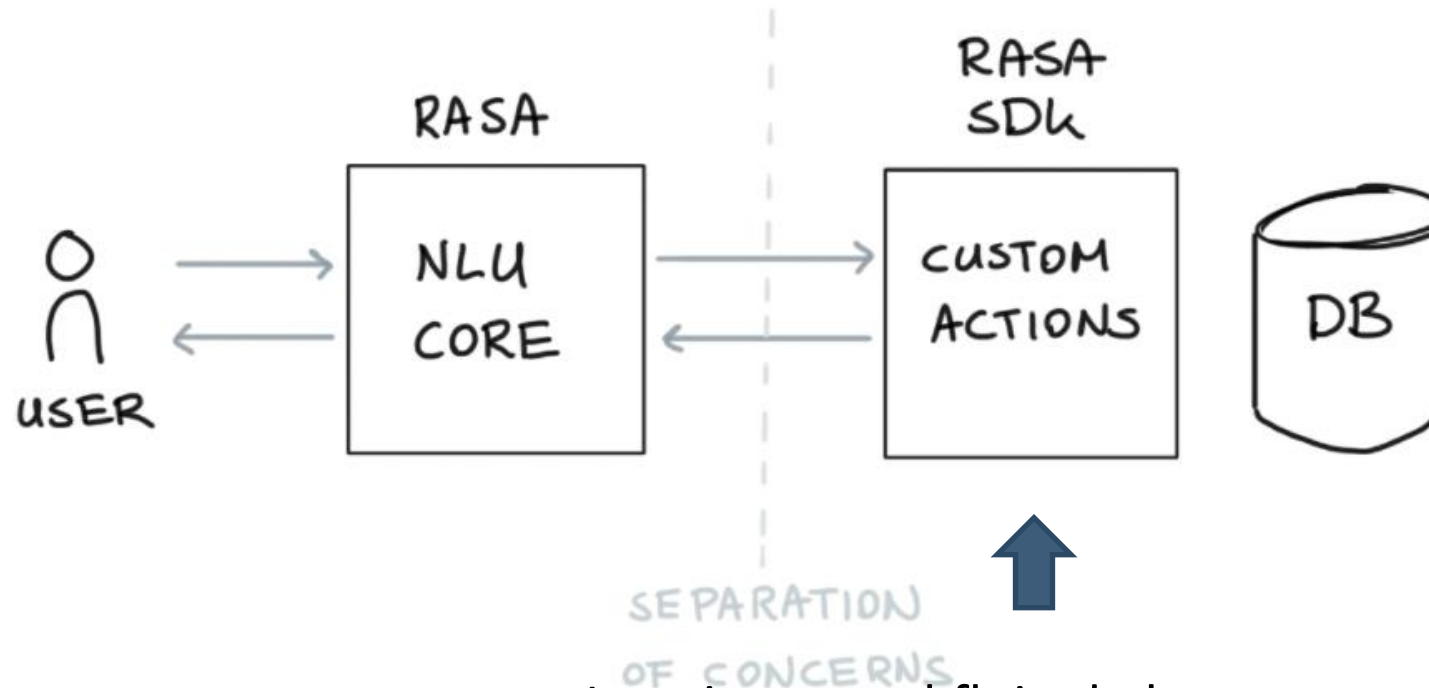
6. Acciones ("Custom Actions")



- Se encarga de procesar el lenguaje natural y extraer *intenciones* y entidades de los mensajes de los usuarios.
- Utiliza técnicas de procesamiento de lenguaje natural y aprendizaje automático para comprender el significado detrás de las palabras.



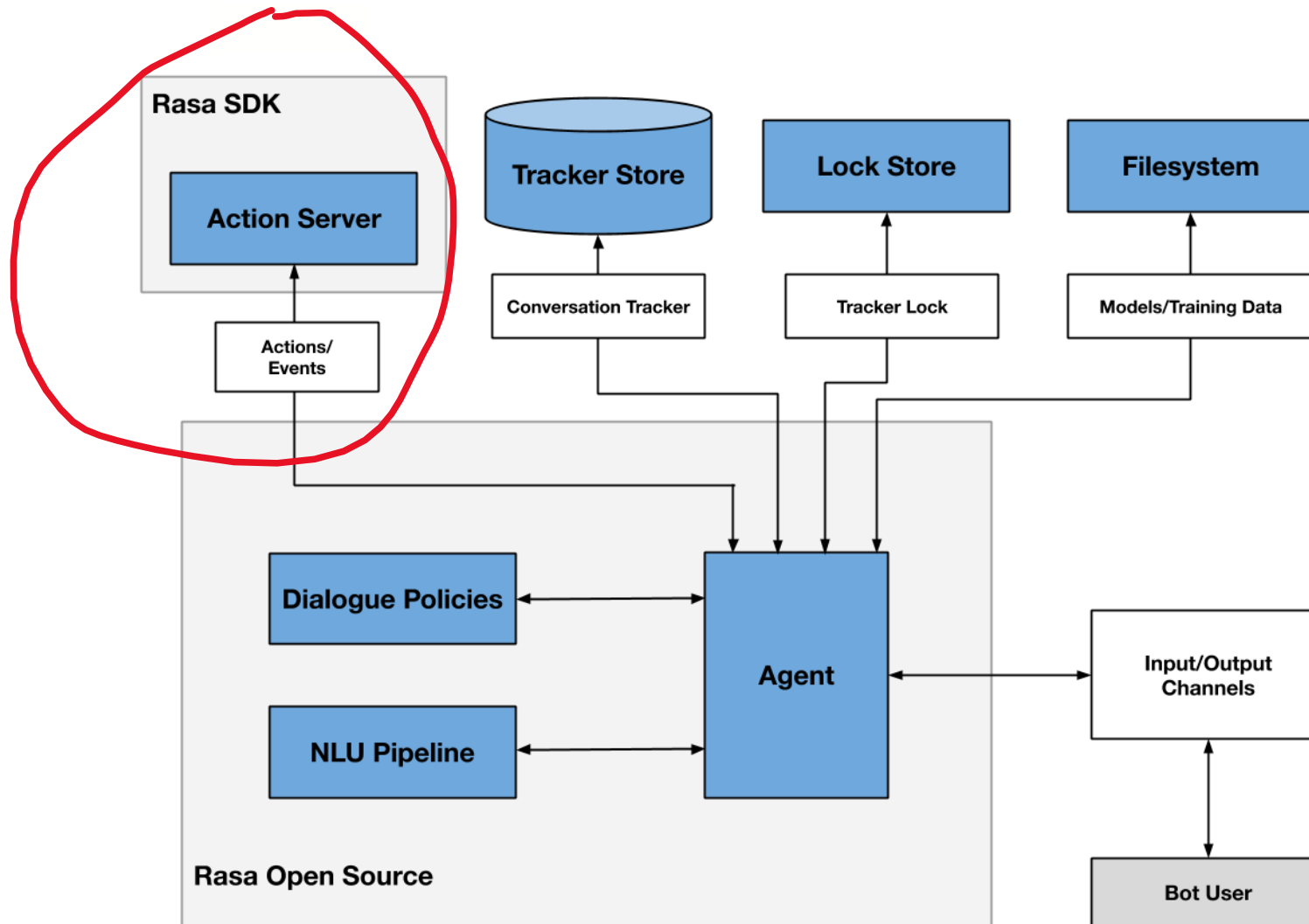
6. Acciones ("Custom Actions")



- Interviene en el flujo de la conversación decidiendo cómo responder en función de las intenciones y entidades identificadas por Rasa NLU



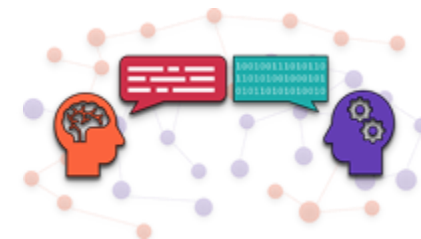
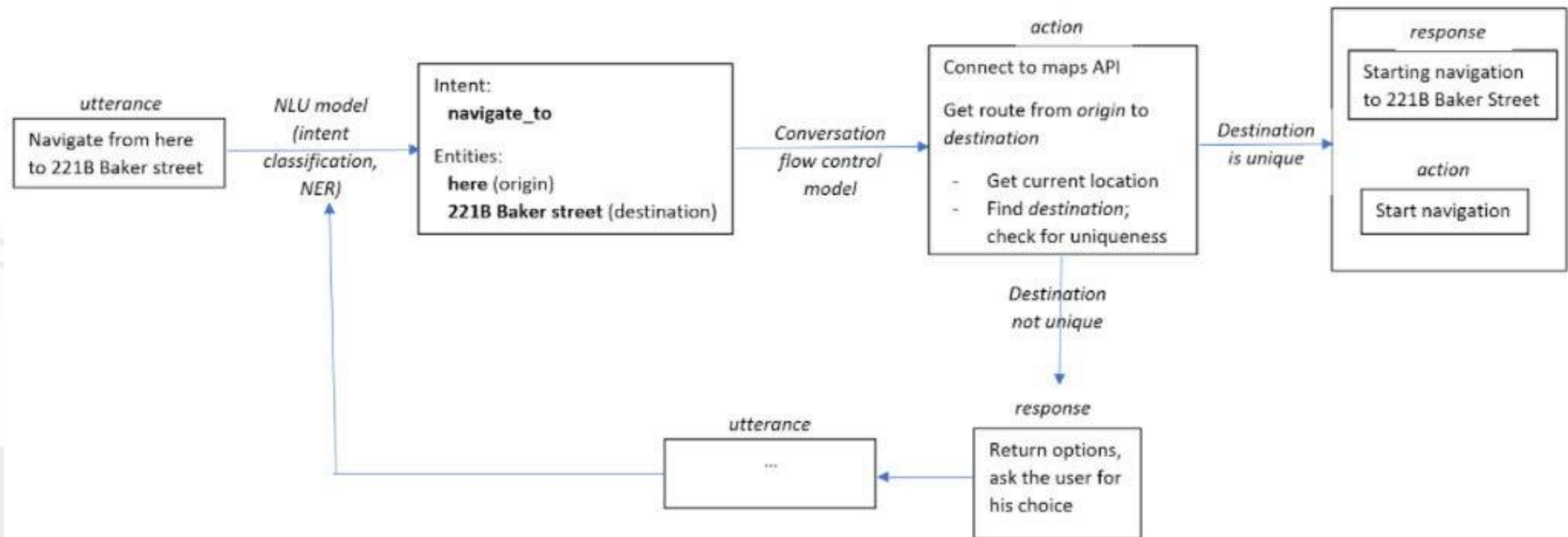
6. Acciones ("Custom Actions")



6. Acciones ("Custom Actions")



Ejemplo :: acceso a API de Google Maps para obtener una dirección



6. Acciones ("Custom Actions")



actions.py

```
from typing import Any, Text, Dict, List

from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectingDispatcher

class ActionHelloWorld(Action):

    def name(self) -> Text:
        return "action_show_time"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        print("Custom code goes here!")

        return []
```



6. Acciones ("Custom Actions")



actions.py

```
from typing import Any, Text, Dict, List

from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectingDispatcher

class ActionHelloWorld(Action):

    def name(self) -> Text:
        return "action_show_time"

    def run(self, dispatcher: CollectingDispatcher,
            tracker: Tracker,
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:

        print("Custom code goes here!")

        return []
```



6. Acciones ("Custom Actions")



actions.py

```
from typing import Any, Text, Dict, List
```

```
from rasa_sdk import Action, Tracker
```

```
from rasa_sdk.executor import CollectingDispatcher
```

```
class ActionHelloWorld(Action):
```

```
    def name(self) -> Text:  
        return "action_show_time"
```

```
    def run(self, dispatcher: CollectingDispatcher,  
            tracker: Tracker,  
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
```

```
        print("Custom code goes here!")
```

```
        return []
```

- Define el nombre de la acción
- Código asociado a la acción
- Recibe de y envía datos a la conversación



6. Acciones ("Custom Actions")



actions.py

```
from typing import Any, Text, Dict, List
```

```
from rasa_sdk import Action, Tracker
```

```
from rasa_sdk.executor import CollectingDispatcher
```

```
class ActionHelloWorld(Action):
```

```
    def name(self) -> Text:  
        return "action_show_time"
```

```
    def run(self, dispatcher: CollectingDispatcher,  
            tracker: Tracker,  
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:
```

```
        print("Custom code goes here!")
```

```
        return []
```

- Permite enviar mensajes al usuario
- Permite extraer datos de la conversación: intenciones, entidades, etc.
- Acceso al fichero **domain.yml**



6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Ejecutar acciones requiere **2 pasos** más:

- 1) Habilitar el punto de estucha (puerto) del servidor en el fichero "**endpoints.yml**":

```
action_endpoint:  
  url: "http://localhost:5055/webhook"
```

- 2) Lanzar el servidor RASA SDK:

```
# rasa run actions
```

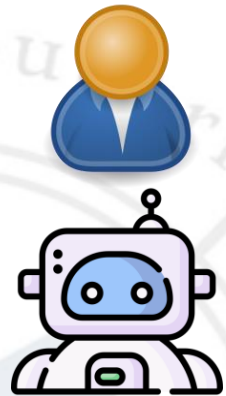


6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Ejemplo :: La hora en diferentes ciudades del mundo



¿Qué hora es en Valencia ahora mismo

Ahora son las 16:32h

Custom Action

Recibe la ciudad como "entidad" y devuelve el valor con la hora

Entidad: lugar



6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

nlu.yml

```
- intent: inquire_time
  examples: |
    - what time is it?
    - what time is it in [Amsterdam](place)?
    - what time is it in [London](place)?
    - tell me the time in [Lisbon](place)
    - what is the current time in [Berlin](place)
    - what time is it in [amsterdam](place)[amsterdam](place)
- lookup: place
  examples: |
    - brussels
    - zagreb
    - london
    - lisbon
    - amsterdam
    - seattle
```



6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

rules.yml

```
● ● ●  
  
- rule: Tell the time  
  steps:  
    - intent: inquire_time  
    - action: action_tell_time
```

domain.yml

```
● ● ●  
  
intents:  
  ...  
  - inquire_time  
  ...  
  
entities:  
  - place  
  
actions:  
  - action_tell_time
```



6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

config.yml

```
pipeline:
  - name: WhitespaceTokenizer
  - name: LexicalSyntacticFeaturizer
  - name: CountVectorsFeaturizer
  - name: CountVectorsFeaturizer
    analyzer: char_wb
    min_ngram: 1
    max_ngram: 4
  - name: DIETClassifier
    entity_recognition: False
    epochs: 100
  - name: RegexEntityExtractor
    use_lookup_tables: True
```



6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

actions.py

```
from typing import Any, Text, Dict, List

import arrow
import dateparser
from rasa_sdk import Action, Tracker
from rasa_sdk.events import SlotSet
from rasa_sdk.executor import CollectingDispatcher

city_db = {
    'brussels': 'Europe/Brussels',
    'zagreb': 'Europe/Zagreb',
    'london': 'Europe/Dublin',
    'lisbon': 'Europe/Lisbon',
    'amsterdam': 'Europe/Amsterdam',
    'seattle': 'US/Pacific'
}
```



6. Acciones ("Custom Actions")

```
class ActionTellTime(Action):  
  
    def name(self) -> Text:  
        return "action_tell_time"  
  
    def run(self, dispatcher: CollectingDispatcher,  
            tracker: Tracker,  
            domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:  
        current_place = next(tracker.get_latest_entity_values("place"), None)  
        utc = arrow.utcnow()  
  
        if not current_place:  
            msg = f"It's {utc.format('HH:mm')} utc now. You can also give me a place."  
            dispatcher.utter_message(text=msg)  
            return []  
  
        tz_string = city_db.get(current_place, None)  
        if not tz_string:  
            msg = f"It's I didn't recognize {current_place}. Is it spelled correctly?"  
            dispatcher.utter_message(text=msg)  
            return []  
  
        msg = f"It's {utc.to(city_db[current_place]).format('HH:mm')} in {current_place} now."  
        dispatcher.utter_message(text=msg)  
  
        return []
```



6. Acciones ("Custom Actions")



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Realiza el siguiente ejercicio:

- **Ejercicio 4: Acceso a una API**

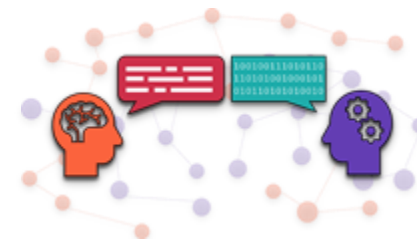


7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- En las conversaciones es necesario a veces memorizar datos para recuperarlos en otros momentos.
- RASA cuenta con las **ranuras** o "slots" como mecanismo de almacenamiento de datos

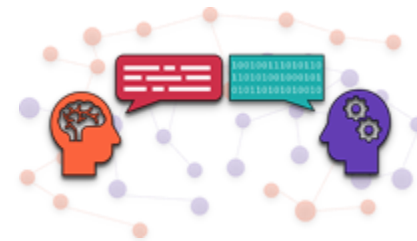


7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

¿Cómo
funcionan las
ranuras
("slots")?



7. Formularios

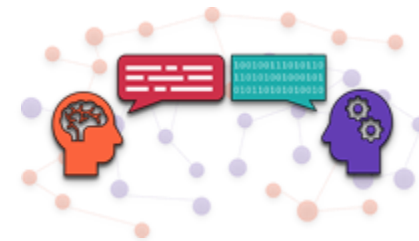


CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



Quiero comprarme una camisa

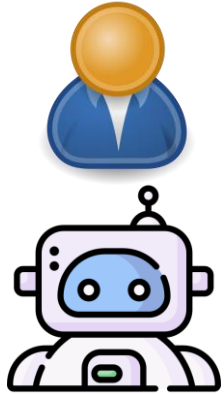
¿Cómo
funcionan las
ranuras
("slots")?



7. Formularios



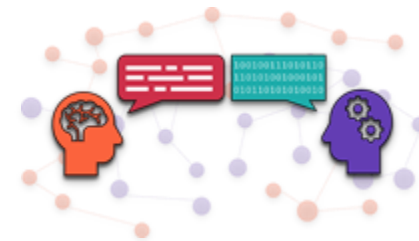
CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



Quiero comprarme una camisa

De acuerdo, ¿qué talla usas?

¿Cómo
funcionan las
ranuras
("slots")?



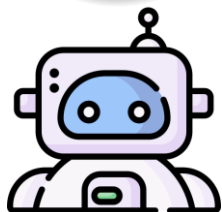
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



Quiero comprarme una camisa



De acuerdo, ¿qué talla usas?



Mediana



Guardaremos
"mediana"
como valor
para la ranura
"talla de la
camisa"



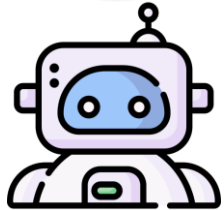
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



Quiero comprarme una camisa



De acuerdo, ¿qué talla usas?



Mediana

...



¿Qué talla tiene mi camisa?

... la conversación
continua y se
pregunta por la
talla
posteriormente...



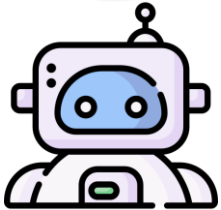
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



Quiero comprarme una camisa



De acuerdo, ¿qué talla usas?

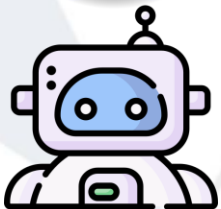


Mediana

...



¿Qué talla tiene mi camisa?



Tu camisa es de talla **mediana**

... pudiendo
recuperar el
valor
almacenado
anteriormente.



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

nlu.yml

intent: pedir_ayuda_camisa

examples: |

- necesito ayuda con mi camisa
- quiero darte la talla de mi camisa
- quiero comprarme una camisa
- necesito una camiseta

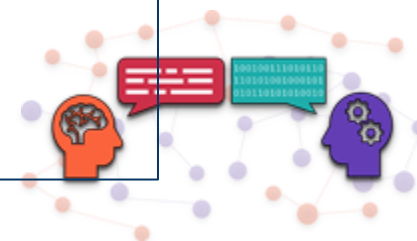
...

- **intent:** repetir_talla_camisa

examples: |

- recuerda la talla de camisa
- repite mi talla de camisa
- ¿puedes recordarme cuál era la talla de mi camisa?
- recordarme la talla de mi camisa
- ¿cuál es la talla de mi camisa?

...



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

nlu.yml

...

- **intent**: dar_talla_camisa

examples: |

- talla **[mediana]**(talla_camisa)

- talla **[pequeña]**(talla_camisa)

- talla **[grande]**(talla_camisa)

- **[pequeña]**(talla_camisa)

- **[mediana]**(talla_camisa)

- **[grande]**(talla_camisa)



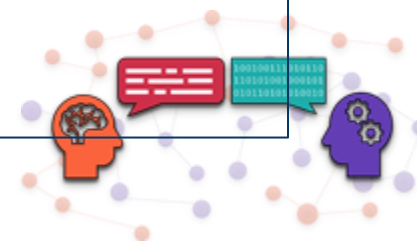
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

stories.yml

- **story:** pedir ayuda con talla de camisa
- steps:**
 - **intent:** pedir_ayuda_camisa
 - **action:** utter_pedir_talla_camisa
 - **intent:** dar_talla_camisa
- entities:**
 - **talla_camisa:** grande
- **slot_was_set:**
 - **talla_camisa:** grande
- **action:** utter_recordar_talla_camisa
- **intent:** repetir_talla_camisa
- **action:** action_dar_talla_camisa
- **intent:** despedir
- **action:** utter_despedir



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

stories.yml

- **story**: recordar talla camisa
- steps**:
 - **intent**: saludar
 - **action**: utter_saludar
 - **intent**: repetir_talla_camisa
 - **action**: action_dar_talla_camisa
 - **intent**: despedir
 - **action**: utter_despedir



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

domain.yml

intents:

- saludar

...

- dar_talla_camisa

- pedir_ayuda_camisa

entities:

- talla_camisa



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Otras opciones útiles de "type" serían "bool", "float", "list", "any"

domain.yml

slots:

talla_camisa:

type: text

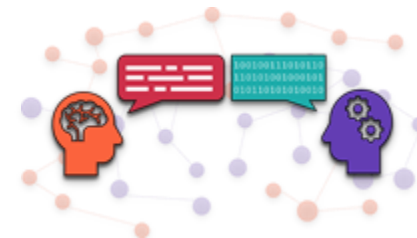
influence_conversation: true

mappings:

- type: from_entity

entity: talla_camisa

En el caso de "mappings", otras opciones útiles de "type" serían "from_intent" y "from_text",



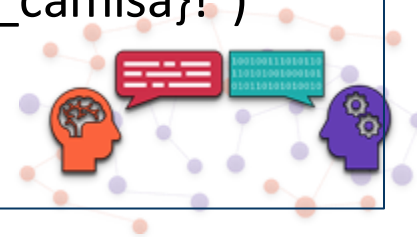
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

actions.py

```
class ActionDarTallaCamisa(Action):  
  
    def name(self) -> Text:  
        return "action_dar_talla_camisa"  
  
    def run(self, dispatcher: CollectingDispatcher,  
            tracker: Tracker, domain: Dict[Text, Any]) -> List[Dict[Text, Any]]:  
  
        talla_camisa = tracker.get_slot("talla_camisa")  
        if not talla_camisa:  
            dispatcher.utter_message(text="No sé cuál es la talla de tu camisa.")  
        else:  
            dispatcher.utter_message(text=f"La talla de tu camisa es {talla_camisa}!")  
        return []
```



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

domain.yml

responses:

...

utter_pedir_talla_camisa:

- text: Claro, ¿puedes decirme la talla de tu camisa?

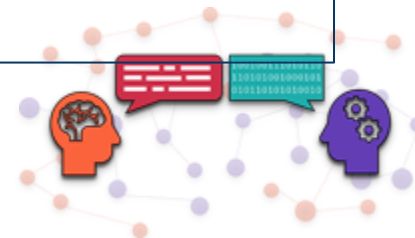
utter_recordar_talla_camisa:

- text: ¡De acuerdo, apunto la talla!

actions:

...

- action_dar_talla_camisa



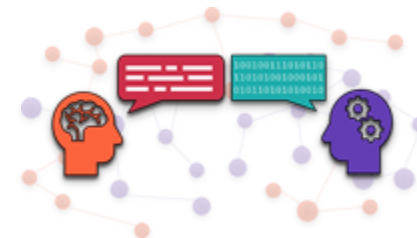
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

stories.yml

- **story**: recordar talla camisa
- steps**:
 - **intent**: saludar
 - **action**: utter_saludar
 - **intent**: repetir_talla_camisa
 - **action**: **action_dar_talla_camisa**
 - **intent**: despedir
 - **action**: utter_despedir



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

... probamos el chatbot

Your input -> hola

¡Hola! ¿Cómo estás?

Your input -> necesito comprarme una camisa

Claro, ¿puedes decirme la talla de tu camisa?

Your input -> mediana

De acuerdo, apunto la talla!

Your input -> hola

¡Hola! ¿Cómo estás?

Your input -

> bien

¡Genial, sigue así!

Your input -> recuerda la talla de mi camisa

La talla de tu camisa es mediana!

Your input ->



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

... si necesitamos manejar más de
un dato, RASA dispone de
formularios ...



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

- **Pasos:**
 1. añadir "**forms**" a "domain.yml"
 2. definir las **preguntas** del bot asociadas al formulario en "domain.yml"
 3. asociar **reglas** en "rules.yml"
 4. **activar** el formulario en "stories.md"



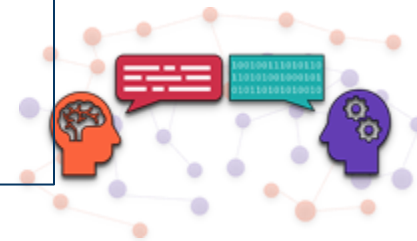
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

1. Añadir "forms" a "domain.yml"

```
...  
forms:  
  camisa_form:  
    required_slots:  
      - talla_camisa  
    ...  
  slots:  
    talla_camisa:  
      type: text  
      influence_conversation: true  
  mappings:  
    - type: from_entity  
      entity: talla_camisa  
  conditions:  
    active_loop: camisa_form
```



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

2. Definir preguntas del bot en "domain.yml"

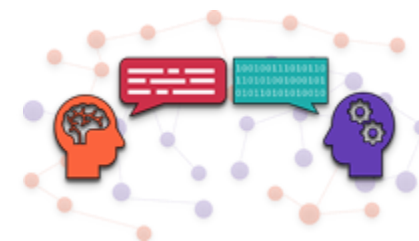
- Seguir el formato "**utter_ask_<slot>**"

responses:

...

utter_ask_talla_camisa:

- **text:** Claro, ¿puedes decirme la talla de tu camisa?



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

3. Asociar regla en "rules.yml"

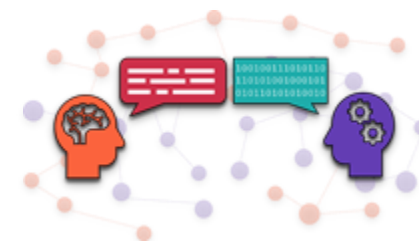
```
...  
rule: Activar formulario camisa  
steps:  
- intent: pedir_ayuda_camisa  
- action: camisa_form  
- active_loop: camisa_form  
  
rule: Enviar formulario camisa  
condition:  
- active_loop: camisa_form  
steps:  
- action: camisa_form  
- active_loop: null  
- slot_was_set:  
- requested_slot: null  
- action: utter_recordar_talla_camisa
```



Una regla para
activar el formulario



Otra regla para
desactivarlo



7. Formularios

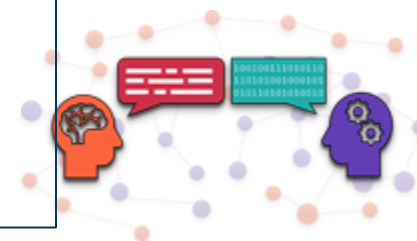


CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

4. Activar formulario en "stories.yml"

```
...  
- story: pedir ayuda con talla de camisa  
  steps:  
  - intent: pedir_ayuda_camisa  
  - action: camisa_form  
  - active_loop: camisa_form  
  - slot_was_set:  
    - requested_slot: talla_camisa  
    - slot_was_set:  
      - talla_camisa: grande  
    - slot_was_set:  
      - requested_slot: null  
    - active_loop: null  
  - action: utter_recordar_talla_camisa  
  - intent: repetir_talla_camisa  
  - action: action_dar_talla_camisa  
  - intent: despedir  
  - action: utter_despedir
```

Se da el valor "grande" como ejemplo, aunque será la interacción la que asigne el valor real al "slot"!



7. Practicando con formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Realiza en este orden los ejercicios relacionados con las ranuras y los formularios:

- **Ejercicio 5: Primera prueba con los “slots”**
- **Ejercicio 6: Ser profesor del curso de IA**



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

En el uso de **formularios** hay situaciones que pueden desviarse del flujo de interacción previsto inicialmente:

- 1) Referencia repentina a otras intenciones
- 2) Petición de cancelación del formulario

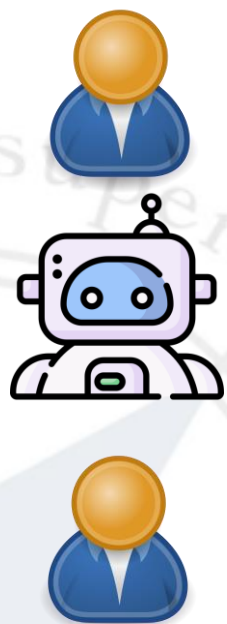


7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

1) Referencia repentina a otras intenciones



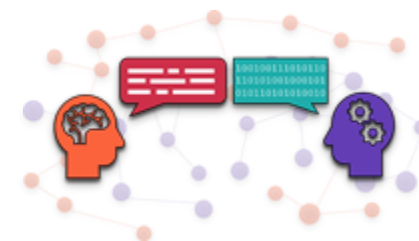
Quiero comprarme una camisa

De acuerdo, ¿qué talla usas?

Eres un humano?



Se realiza una
pregunta que
se sale del
contexto del
formulario



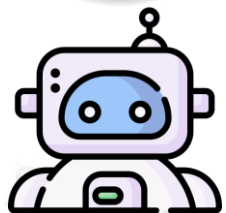
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



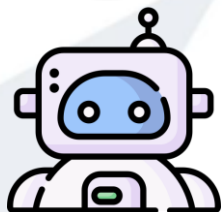
Quiero comprarme una camisa



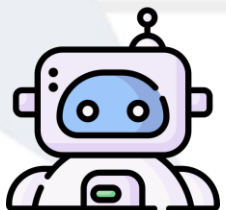
De acuerdo, ¿qué talla usas?



Eres un humano?

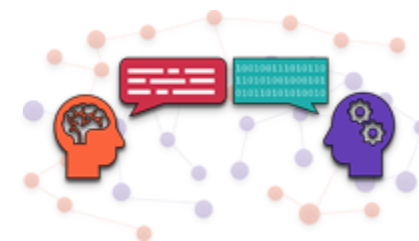


No, soy un chatbot



De acuerdo, ¿qué talla usas?

El chatbot
retoma el
formulario una
vez solventado
el desvío en la
conversación



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



```
- rule: Interrupción en formulario camisa
condition:
- active_loop: camisa_form
steps:
- intent: desafio_bot
- action: utter_soy_un_bot
- action: camisa_form
- active_loop: camisa_form
```

Solución

... para casos
puntuales, se
puede recurrir a
las reglas ...

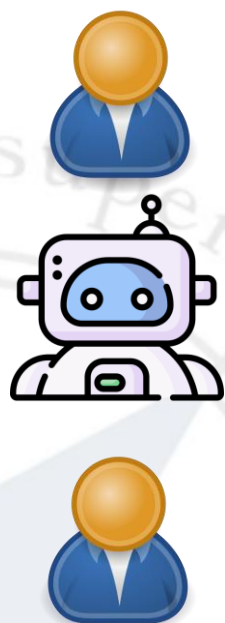


7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

2) Petición de cancelación del formulario



Quiero comprarme una camisa

De acuerdo, ¿qué talla usas?

**Al final no necesito una
camisa**



La entrada
persigue
cancelar la
interacción en
curso



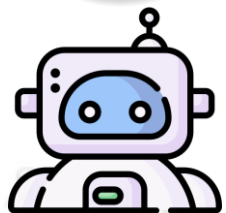
7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior



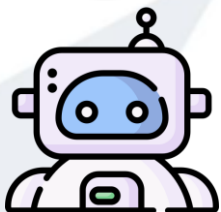
Quiero comprarme una camisa



De acuerdo, ¿qué talla usas?



Al final no necesito una
camisa



Estás seguro?



Sí

En este caso, el
chatbot
contempla una
posible petición
de cancelación



7. Formularios



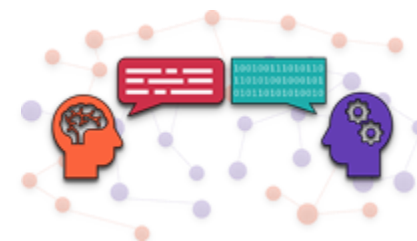
CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Forma de abordar este caso:

1. Uso de una o varias historias en lugar de reglas

2. Modificación de ficheros:

- **stories.yml**: añadir al menos una historia que dé respuesta a la petición de cancelación
- **nlu.yml**: añadir una intención que recoge la petición del usuario
- **domain.yml**: añadir confirmación de cancelación del diálogo



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

stories.yml

- Mediante "action_deactivate_loop" se controla la finalización del formulario
- Incluir diferentes historias permite a RASA generalizar variantes que se dan en la interacción

```
- story: Interrupción en formulario camisa 2
  steps:
  - intent: pedir_ayuda_camisa
  - action: camisa_form
  - active_loop: camisa_form
  - slot_was_set:
    - requested_slot: talla_camisa
  - intent: parar_dialogo_camisa
  - action: utter_continuar_camisa
  - or:
    - intent: afirmar
    - intent: negar
  - action: action_deactivate_loop
  - slot_was_set:
    - requested_slot: null
  - active_loop: null
  - action: despedir
```



7. Formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

nlu.yml



```
- intent: parar_dialogo_camisa
  examples: |
    - quiero abortar la conversación
    - quiero acabar la conversación
    - quiero dejar la conversación
    - quiero dejar la conversación
    - no quiero hablar más
    - no me preguntes sobre la camisa
    - cambiemos de tema
    - no quiero darte la talla
    - me voy
    - me largo|
```



7. Formularios

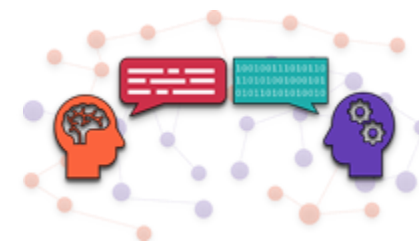


CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

domain.yml

```
● ● ●

intents:
  ...
  - parar_dialogo_camisa
|
responses:
  ...
  utter_continuar_camisa:
    - text: Abortamos la conversación?
```



7. Practicando con formularios



CIPFP Mislata
Centre Integrat Públic
Formació Professional Superior

Realiza en este orden los ejercicios relacionados con las ranuras y los formularios:

- **Ejercicio 7: Salir de un formulario**

