

Presentación

DESARROLLO DE UN CHATBOT

Ayrton Tomas Valdes

Ulises Palazzo

Jeremias Verde



Introducción:

En el siguiente informe se detalla el funcionamiento de dos tipos de chatbots. Tres corresponden a distribuidoras mayoristas y tres a minoristas. Estos serán capaces de tomar pedidos, actualizar stock, permitir consultas sobre disponibilidad de productos e interactuar con una persona o chatbot para distintas tareas. Esto se realiza con el framework RASA como medio de codificación.

Qué es un chatbot:

Los chatbots o bots conversacionales son programas informáticos diseñados para simular conversaciones con personas a través del teclado o mediante la voz. Tienen capacidad para actuar simulando el comportamiento humano y pueden mantener conversaciones mucho más fluidas que un contestador automático.

Desarrollo de los chatbots:

En un comienzo se planteó un esquema de un mayorista tomando como ejemplo a Jaguar S.A. Tomando esas bases se realizó una pequeña charla donde el chatbot se presenta al usuario y preguntado que quería realizar, entre estas opciones se encontraban:

- Consultas precios
- Realizar pedidos
- Asistencia
- Consultar pedido

Devolviendo a esto mensajes simples, pero sin mucha complejidad.

Luego se implementó un sistema de registro y login utilizando slots que nos permitiría tomar datos de los usuarios para su posterior uso en las charlas planificadas y se mejoró la consulta con respecto a los precios de los distintos productos tomando expresiones regulares ya según “fideos”, “sopas”, etc. El siguiente paso fue realizar un sistema de compras que permitiera registrar la compra y dar distintas formas de pago ya sea efectivo o crédito. Una vez realizado a través de funciones en Python se crea un proceso por el cual mediante un pedido se obtiene un ticket que después se utiliza para realizar el seguimiento de la compra e indica la fecha de entrega del pedido.

A su vez también se unió el ticket mediante slots y funciones al sistema de consultas de envíos donde en base al ticket emitido se informa cuando llega el pedido si el usuario lo consulta. También se implementó un sistema por el cual se recordaba el método de compras utilizado por última vez y permitir su posterior uso sin la necesidad de cargar datos extras y facilitando la compra.

Terminados y establecidos los cimientos de este chatbot se modelan los posteriores chatbots tomando de base este mismo.

En el caso de los chatbots minoristas se utilizó como ejemplo al supermercado Monarca y tomando de base al mayorista se realizaron cambios pertinentes, a su vez se implementó un sistema de stock para ambos tipos de chatbots, donde se permite ver la cantidad que se tiene antes y después de una compra o venta de los distintos productos que se ofrecen.

Ejemplos de charlas:

A continuación, se presentarán ejemplos de cómo se realizan las charlas entre el chatbot y el alumno. El término “**H**” hace referencia al humano mientras que “**B**” al chatbot. El primero corresponde a un chatbot mayorista y el segundo ejemplo A un minorista

Ejemplo 1:

B: Bienvenido a distribuidora Jaguar S.A por favor regístrese o inicie sesión

H: loguearse: Juan 1234

B: Buenas Juan en qué puedo ayudarte

- Realizar pedido
- Consultas pedido
- Consultar stock
- Ayuda o contacto

H: Consultar stock

B: Sobre qué producto deseas consultar

- Aderezos
- Fideos
- Papel Higiénico
- Sopas

H: Sopas

B: Quick Knorr: Individual 27,8, Bulto por 24 unidades 624,22. Desea saber algo más?

B: ¡Desea saber algo más?

H: No. Muchas Gracias

B: Hasta luego. Gracias por su consulta.

Ejemplo 2:

B: Bienvenido a Supermercados Monarca por favor regístrese o lóguese

H: loguearse: Juan 1234

B: Buenas Juan en que puedo ayudarte

- Realizar pedido
- Consultas pedido
- Consultar stock
- Ayuda o contacto

H: Realizar pedido

B: Sobre que producto/s deseas realizar un pedido

- Aderezos
- Fideos
- Papel Higiénico
- Sopas
- Alfajores
- Caramelos

- Papel de baño

H: Alfajores

B: Alfajores águila Por un valor de 80 pesos ¿Cuántos desea llevar?

H: 5

B: Perfecto. ¿Desea cargar algo más?

H: No por el momento

B: Okey. Sería un total de 400 pesos. ¿Retira en local o domicilio?

H: A domicilio

B: Cargue calle, número y un número de contacto para llamarlo cuando esté en el lugar el repartidor.

H: Calle falso 123 2494111111

B: Correcto, su pedido llegará en el transcurso del día. Muchas Gracias por usar nuestro servicio.

A continuación, se presenta una captura de una charla con el chatbot donde se realiza un pedido y se identifica la sesión iniciada:

```
Your input -> cual es mi usuario?
Has iniciado sesion como juan
Your input -> quiero realizar un pedido
Que le gustaria comprar?
Your input -> quiero comprar 2 mayonesa helmanns y 3 sopa knorr
Digame que metodo de pago desea utilizar para realizar la compra, por favor.
Your input -> tarjeta de credito
Por favor, ingrese al siguiente link y complete el formulario con sus datos para completar la compra: https://www.urlparacompletarpago.com
El numero de pedido es 1548.
Su pedido llegara el 31/10/2021
Valor total (incluye envio): 8545
Muchas gracias por su compra.
Your input -> gracias
Adios, juan, espero haberte ayudado :)
Your input -> 
```

A su vez también se presenta una consulta al chatbot por el estado de un pedido:

```
Your input -> hola
Hola, soy el asistente automatizado de Jaguar S.A
En que puedo ayudar?
Your input -> cuando llega el pedido 3243?
El pedido con numero 3243, esta programado para llegar el dia 15/12/2021
Your input -> gracias, tambien quiero saber cuando son los horarios de atencion al cliente
Los horarios de atencion al cliente son de 8:00 a 17:00
Si quieries contactarte con nosotros, nuestro numero de telefono es: 0-800-rasahelp
Your input -> 
```

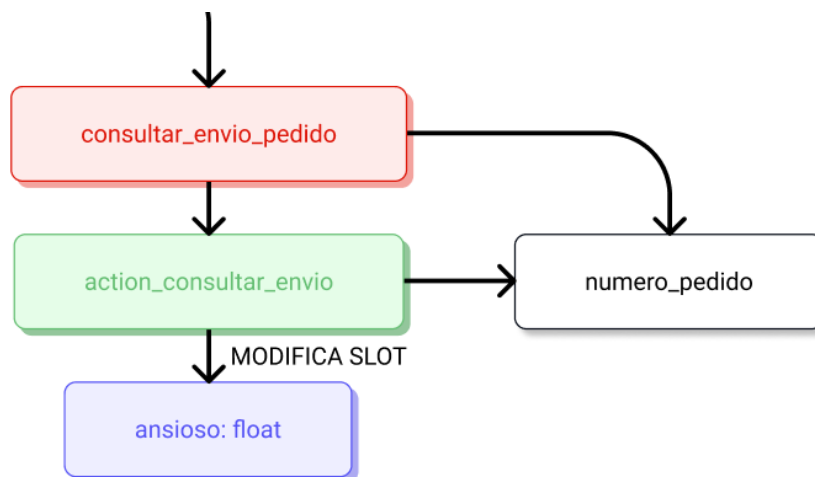
Chatbot y perfiles:

Para los distintos perfiles de los usuarios implementamos algunas maneras de interactuar en el chatbot, para darle una tonalidad personalizada, según el ánimo, o el tipo de usuario que está tratando en el momento del diálogo.

Nos enfocamos en **poder calmar a un usuario** en el caso de estar **ansioso** por la demora de una compra realizada.

Lo implementamos de la siguiente manera:

Para poder comprender el diagrama recomendamos revisar el [Anexo](#).



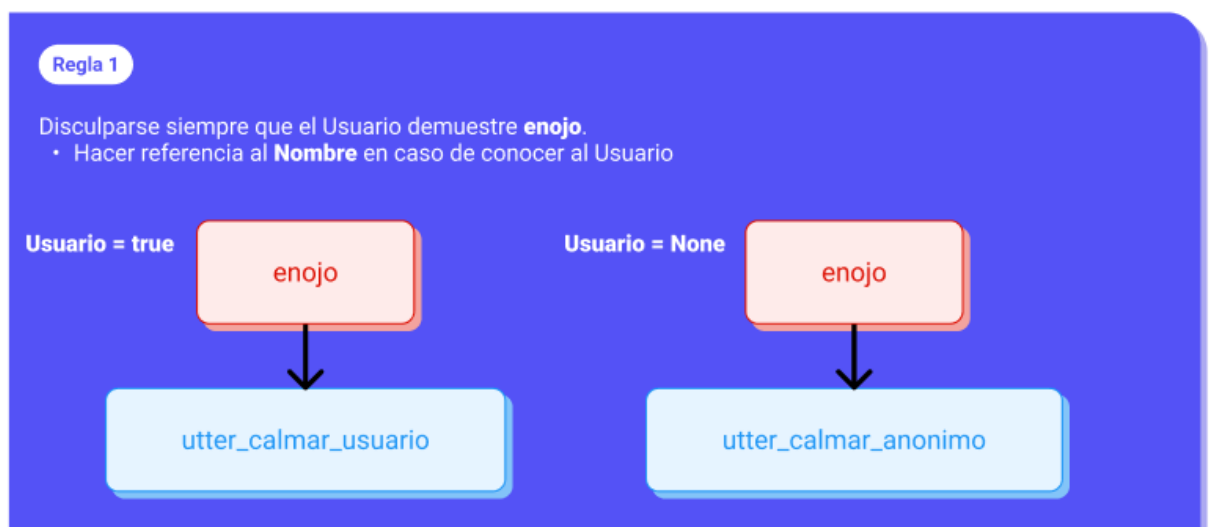
1. El bot identifica el Intent “*consultar_envio_pedido*”
2. El bot realiza la acción “*action_consultar_envio*”
 - a. La acción busca el “*numero_pedido*” en la base de datos y en caso de encontrarlo informa al usuario el día de entrega.
 - b. Se identifica el slot “*ansioso*” y se le suma 1 al valor que contenga.
 - c. Si “*ansioso*” es múltiplo de 5, el bot intenta calmar al usuario.
 - d. Si se intenta calmar al usuario, se modifica la fecha de entrega del pedido(llega antes) y se le ofrece un descuento al cliente en su próxima compra.

Otros perfiles de usuario que consideramos, fueron:

- Usuario enojado.
- Usuario agradecido.
- Usuario olvidadizo.

Para poder personalizar el diálogo considerando estos usuarios, de la forma menos invasiva que creemos para el curso de diálogo actual, implementamos Reglas:

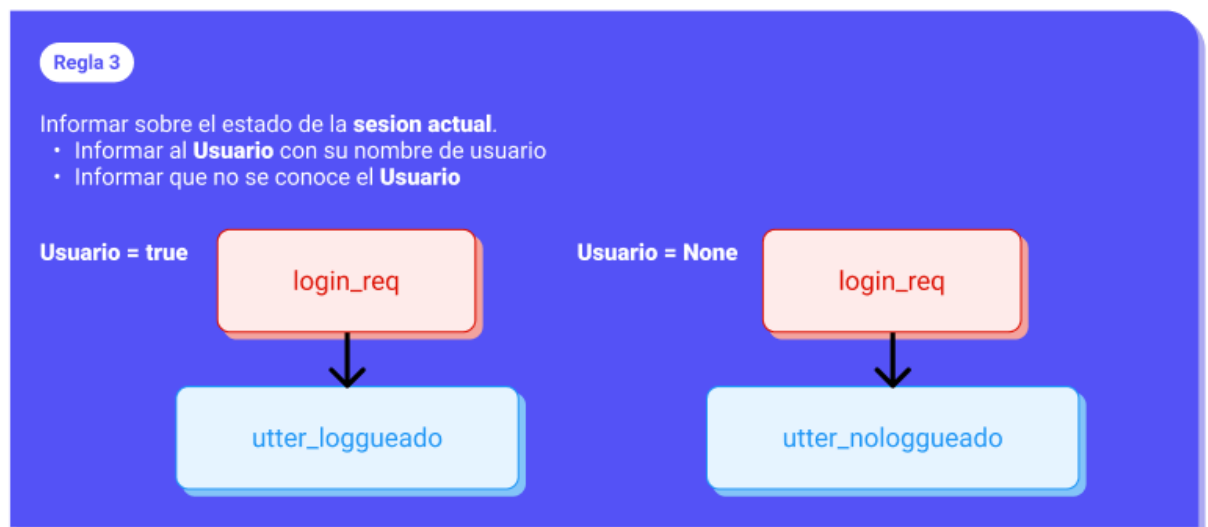
1. **Usuario enojado:**



2. Usuario agradecido:



3. Usuario despistado:



Caso:

H: Cuando llega mi pedido.

B: Durante el transcurso de la semana llegará su pedido.

H: No puede llegar más rápido.

B: Por 1800 pesos más IVA, podemos adelantar su pedido para el martes a la mañana

H: Bien, los pago.

B: Correcto. Cargue los datos para realizar el pago

Chatbot funcionamiento:

En el funcionamiento de nuestro bot, intentamos incorporar todas las herramientas que nos ofrece el framework Rasa y para demostrarlo, vamos a explicar el diseño y desarrollo de la función más compleja que implementamos en el bot.

La funcionalidad de *hacer un pedido de compra*, es la más compleja y desarrollada de todo el bot, en esta funcionalidad interactuamos con las bases de datos de la empresa mayorista, tanto la que contiene las cuentas de los usuarios (Cuentas.json) como la de los productos de la empresa (Productos.json), diseñamos una story con ciertas restricciones que logra hacer que la compra sea más formal y personalizada, dependiendo de el tipo de cliente que esté realizando la compra. En la siguiente imagen se muestran las Stories que participan.

```
- story: comprar_path_no_logged
  steps:
  - intent: greet
  - action: utter_greet
  - intent: pedidos
  - slot_was_set:
    |   - usuario: null
  - action: utter_nologgeado

- story: comprar_path_logged
  steps:
  - intent: greet
  - action: utter_greet
  - intent: pedidos
  - slot_was_set:
    |   - usuario
  - action: utter_pedidos_logged
  - intent: carrito_compra
  - action: action_realizar_pedido
  - slot_was_set:
    |   - compra_aceptada: true
  - slot_was_set:
    |   - nuevo: false
  - action: action_metodo_pago
  - checkpoint: check_respuesta

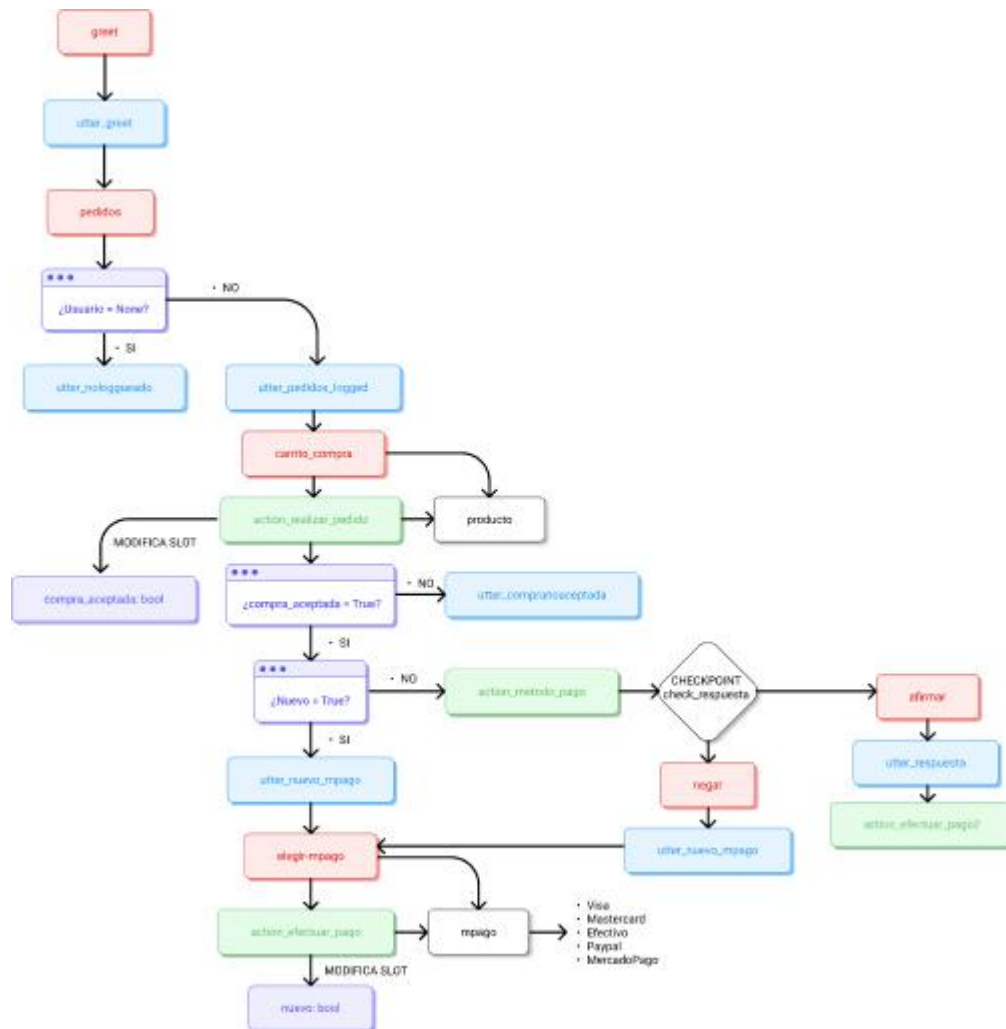
- story: respuesta_afirmacion
  steps:
  - checkpoint: check_respuesta
  - intent: afirmar
  - action: utter_respuesta
  - action: action_efectuar_pago2

- story: respuesta_negacion
  steps:
  - checkpoint: check_respuesta
  - intent: negar
  - action: utter_nuevo_mpago
  - intent: elegir_mpago
  - action: action_efectuar_pago

- story: comprar_path_logged_no_aceptado
  steps:
  - intent: greet
  - action: utter_greet
  - intent: pedidos
  - slot_was_set:
    |   - usuario
  - action: utter_pedidos_logged
  - intent: carrito_compra
  - action: action_realizar_pedido
  - slot_was_set:
    |   - compra_aceptada: false
  - action: utter_compranoaceptada

- story: comprar_path_logged2
  steps:
  - intent: greet
  - action: utter_greet
  - intent: pedidos
  - slot_was_set:
    |   - usuario
  - action: utter_pedidos_logged
  - intent: carrito_compra
  - action: action_realizar_pedido
  - slot_was_set:
    |   - compra_aceptada: true
  - slot_was_set:
    |   - nuevo: true
  - action: utter_nuevo_mpago
  - intent: elegir_mpago
  - action: action_efectuar_pago
```

A continuación, mediante un diagrama, unificamos los distintos caminos de las Stories anteriores, para demostrar el diseño del dialogo que creemos mas optimo para realizar un pedido de compra. *Para poder comprender el diagrama recomendamos revisar el [Anexo](#).*



Definimos 3 criterios que se deben cumplir para que la compra sea realizada:

1. El usuario debe verificar su identidad para realizar una compra.
2. La compra debe ser válida.
 - a. El producto debe existir.
 - b. La cantidad de producto debe ser superior o igual a 1.
 - c. No se debe solicitar el mismo producto más de una vez por pedido.
3. ¿Este cliente ya ha realizado compras antes? Este criterio está implementado únicamente con fines de personalización y probar más herramientas de Rasa.
 - a. (True) Si ya se han realizado compras antes, se le consulta al cliente si desea utilizar el método de pago que ya ha utilizado anteriormente.
 - b. Mediante un *checkpoint* dividimos el diálogo en caso de que el cliente acepte o rechace la consulta anterior.

- c. En caso de aceptar, se reutiliza el método de pago anterior, si rechaza la consulta, se va a False.
- d. (False) Si no ha realizado compras antes, se le da la opción de elegir uno de los métodos de pago que generalmente utiliza la empresa.

Si bien la funcionalidad explicada es la más elaborada del bot, este también cuenta con las funciones de:

- Registrar un usuario en la base de datos.
- Identificar (o loguear) a un usuario registrado en las bases de datos.
- Almacenar todas las compras realizadas, en forma de comprobante de compra.
- Interactuar con la base de datos de los productos de la empresa.
 - Marcar un producto para abastecer en caso de que tenga poco stock.
 - Modificar el stock de los productos cuando estos son comprados.
- Funcionalidad de consultas generales.
 - Horarios de atención.
 - Catálogo de productos.
 - Historial de compras.

Empresa minorista:

Para poder lograr la interacción entre bots implementamos un chatbot que representa una empresa PYME o empresa minorista. Considerando la funcionalidad de realizar pedidos de compra que contiene la empresa mayorista, esta también sirve como funcionalidad para una empresa minorista, en caso de que requiera automatizar esa acción.

El bot no está orientado a que el usuario sea un cliente, si no, un empleado de la compañía que desea realizar labores de mantenimiento al stock de los productos.

Suponiendo el caso de que en una empresa minorista, la persona encargada de controlar el stock tiene que realizar su labor, el bot de esta empresa automatiza la tarea de este encargado.

El bot cuenta con una base de datos en la cual se almacena el stock de los productos que vende la empresa minorista, divididos según su proveedor. Estos valores de stock se modifican según se van rellenando.

Este chatbot recibirá las órdenes del usuario y, si es necesario, establecerá comunicación con un bot Mayorista. A la hora de hacer la interacción, internamente, el bot analiza los valores de “*usuario*”, “*stock*”, “*historial*”, “*horarios*” y “*precios*”. Estos valores booleanos son slots, los cuales determinan qué acciones se ha detectado que debe cumplir durante su interacción con el bot del Mayorista. Estas acciones pueden ser iniciar sesión, consultar los horarios de atención al cliente, los precios de los productos, el historial de compras realizado con el mayorista en cuestión o realizar un pedido de compra, esta última consulta es la más importante, ya que es la encargada de rellenar el stock de la empresa.

En el siguiente diagrama se puede observar los caminos de decisión que tiene esta funcionalidad principal.

Interacción entre bots:

Para poder hacer que los bots se comuniquen entre ellos debemos tener 4 terminales abiertas, dos para cada bot.

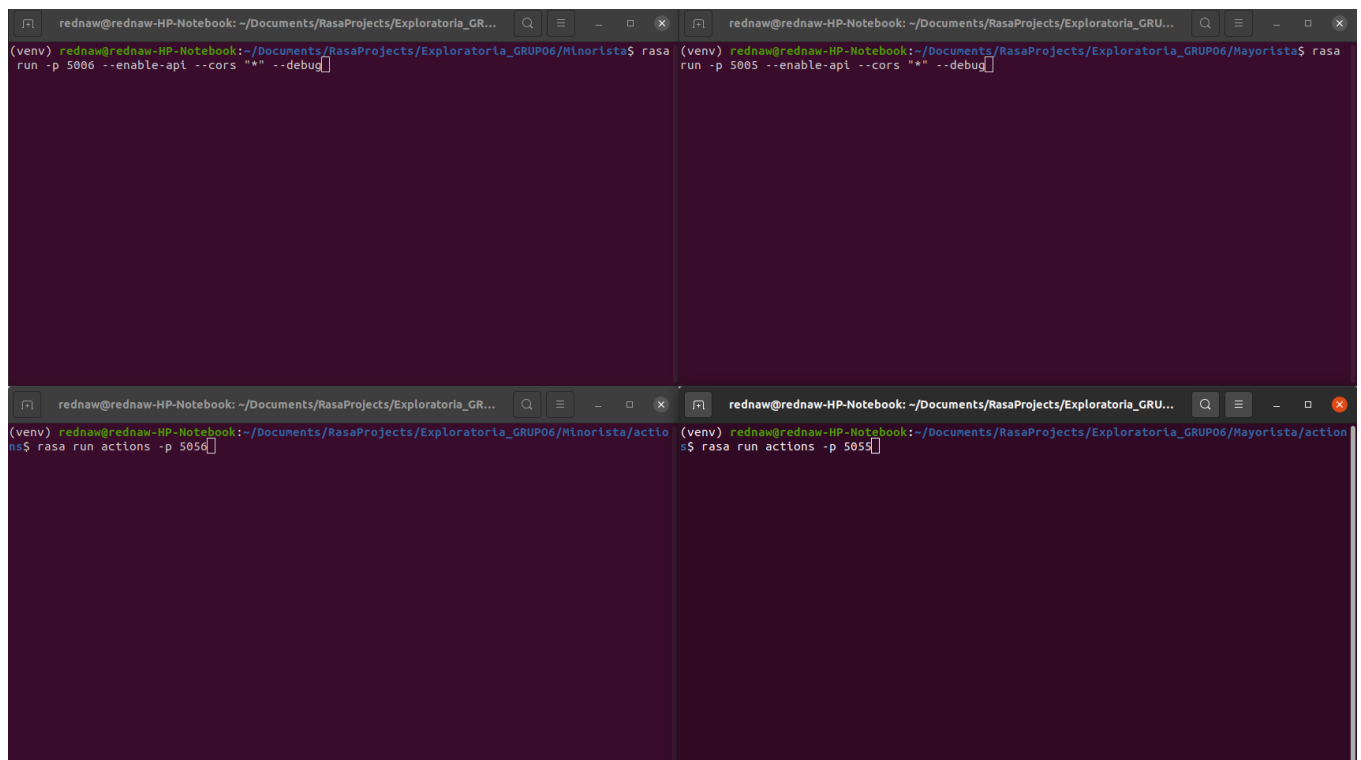
Los puertos que asignamos para poder correr los servidores de los bots son:

- Mayorista = puerto **5005**
- Mayorista acciones = **5055**
- Minorista = **5006**
- Minorista acciones = **5056**

Y los comandos utilizados para las terminales son los siguientes:

- `#rasa run actions -p [nroPuerto]`
- `#rasa run -p [nroPuerto] --enable-api --cors "*" --debug`

Para abrir los puertos del minorista, debemos ir a la carpeta donde esta el bot, abrir una terminal allí y colocar el comando ***rasa run -p 5006 --enable-api --cors "*" --debug***, y para abrir los puertos de sus respectivas acciones, debemos ir a la carpeta ***actions*** dentro de la carpeta donde se encuentra el bot minorista, abrir una terminal y colocar el comando ***rasa run actions -p 5056***



(En la imagen se muestran las terminales del Minorista, Mayorista, acciones del Minorista y acciones del mayorista respectivamente)

Una vez las 4 terminales estén corriendo sin problemas, procedemos a correr el archivo llamado *Shell.py* (en la carpeta RASA server).

Este archivo permite comunicarse con el bot minorista una vez los servidores están abiertos, pero también a su vez, funciona como un *pipe* o conector entre el bot minorista y los bots mayorista.

Ya corrido el *Shell.py* se puede establecer una charla con el bot minorista.

Por ejemplo, se le puede consultar por la disponibilidad de los productos actualmente, lo cual devolverá el stock actual de cada producto en venta

(Para fines practicos, escribirle *testear* o *testeo* al bot, reducirá el stock de cada producto en venta por 500, esto ayuda a probar más fluidamente las opciones del bot)

```
/bin/python3 "/home/rednaw/Documents/RasaProjects/Exploratoria GRUP06/RASA Server/shell.py"
rednaw@rednaw-HP-Notebook:~$ /bin/python3 "/home/rednaw/Documents/RasaProjects/Exploratoria GRUP06/RASA Server/shell.py"
¿Mensaje?: que productos hay disponibles
Monarca: Este es el catalogo actualizado de nuestros productos:
Monarca: | Producto: sopa knorr
Monarca: | Precio individual: 28
Monarca: | Precio bulto: 624
Monarca: | Stock: 1000
Monarca: |-----|
Monarca: | Producto: mayonesa helmanns
Monarca: | Precio individual: 140
Monarca: | Precio bulto: 3160
Monarca: | Stock: 1000
Monarca: |-----|
Monarca: | Producto: fideos matarazzo
Monarca: | Precio individual: 39
Monarca: | Precio bulto: 1821
Monarca: | Stock: 1000
Monarca: |-----|
Monarca: | Producto: papel higienico scott
Monarca: | Precio individual: 80
Monarca: | Precio bulto: 1761
Monarca: | Stock: 1000
Monarca: |-----|
¿Mensaje?: testear
Monarca: Se redujo 500 de stock a los productos
¿Mensaje?:
```

El bot minorista cuenta con un “máximo de stock” (para los ejemplos, el máximo de stock = 1000) que puede almacenar la empresa, cuando el stock de algún producto es menor a ese “máximo”, el usuario puede pedirle al bot que rellene el stock: Esto hará que el bot busque que productos estan bajos de stocks y se comunice con el proveedor del cual se compran (actualmente solo están en venta productos comprados del mayorista Jaguar, por lo que solo se comunicara con este) y posteriormente rellene los valores de almacenamiento.

Por otro lado, el usuario puede pedirle al minorista que se comunice con un bot Mayorista para consultar cosas como precios, horarios de atención o para ver el historial de compras realizadas a este proveedor.

Para el siguiente ejemplo, le pido al bot que rellene el stock de los productos. El minorista detecta que ciertos productos están bajos de stock, por lo que se conecta con el Mayorista que los provee y compra las cantidades necesarias.

```
rednaw@rednaw-HP-Notebook:~$ /bin/python3 "/home/rednaw/Documents/RasaProjects/Exploratoria GRUP06/RASA Server/shell.py"
¿Mensaje?: rellena el stock
Monarca: conectandose con el bot Mayorista Jaguar...
Monarca: Hola
Jaguar: Hola, soy el asistente automatizado de Jaguar S.A
En que puedo ayudar?
Jaguar: Buen dia, ¿cómo estás? ¿Puedo ayudarte?
Monarca: me gustaria realizar un pedido
Jaguar: Que le gustaria comprar?
ingrese la cantidad de producto que necesita seguido del producto que desea comprar.
Monarca: quiero comprar 500 bultos de sopa knorr 500 bultos de mayonesa helmanns 500 bultos de fideos matarazzo 500 bultos de papel higienico scott
```

Para este ejemplo, se le pide al minorista que consulte por los precios y los horarios del mayorista Jaguar, para los casos en los que no se necesite rellenar el stock, se debe especificar a qué Mayorista se desea hacer la consulta

```
Monarca: ¿Me ayuda con el stock de los productos
¿Mensaje?: consulta por los precios y los horarios
Monarca: No se a quien debo consultarle eso
¿Mensaje?: perdon, consulta por los precios y los horarios de Jaguar
Monarca: conectandose con el bot Mayorista Jaguar...
Monarca: Hola
Jaguar: Hola, soy el asistente automatizado de Jaguar S.A
En que puedo ayudar?
Monarca: necesito saber el precio de sus productos
Jaguar: Este es el catalogo actualizado de nuestros productos:
Jaguar: |      Producto: sopa knorr
Jaguar: |      Precio individual: 28
Jaguar: |      Precio bulto: 624
Jaguar: |      Stock: 1498
Jaguar: -----
Jaguar: |      Producto: mayonesa helmanns
Jaguar: |      Precio individual: 140
Jaguar: |      Precio bulto: 3160
Jaguar: |      Stock: 1492
Jaguar: -----
Jaguar: |      Producto: fideos matarazzo
Jaguar: |      Precio individual: 39
Jaguar: |      Precio bulto: 1821
Jaguar: |      Stock: 7990
Jaguar: -----
Jaguar: |      Producto: papel higienico scott
Jaguar: |      Precio individual: 80
Jaguar: |      Precio bulto: 1761
Jaguar: |      Stock: 8000
Jaguar: -----
Jaguar: Siempre buscamos brindar los mejores precios
Monarca: Gracias
Jaguar: Si necesitas algo mas, aqui estare para ayudarte.
Monarca: quiero saber cuales son los horarios de atencion al cliente
Jaguar: Los horarios de atencion al cliente son de 8:00 a 17:00
Si quieres contactarte con nosotros, nuestro numero de telefono es: 0-800-rasahelp
Monarca: Gracias
Jaguar: Si necesitas algo mas, aqui estare para ayudarte.
Monarca: Eso seria todo, muchas gracias
¿Mensaje?: █
```

Si se le pide al bot que realice varias acciones antes de terminar la interacción, este intentará completarlas a todas.

Por ejemplo, se le puede decir al minorista:

- **consulta por el historial de Jaguar**
- **consulta por los precios de Jaguar**

pero también se puede pedir ambas al mismo tiempo, haciendo que el bot intente realizarlas a ambas

- **consulta por el historial y los precios de Jaguar**
- **consulta por el historial, los horarios y los precios de Jaguar**
- **rellena el stock y consulta por el historial, los horarios y los precios**

Las acciones que realiza el bot se van guardando en un archivo llamado *LogMinorista.txt* (este archivo se re-escribe cada vez que el bot vuelve a ser iniciado).

Si el usuario deseara ver el log o las decisiones tomadas por el bot, puede hacerlo escribiéndole, por ejemplo:

- **mostrame el log**
- **mostra el log**
- **mostra las decisiones**

Esto escribirá en pantalla paso a paso las acciones y decisiones que fue tomando el bot durante la sesión actual.

Mejores aspectos:

Los mejores aspectos son la utilización de un sistema de registro de ingreso ya que permite tener un control de usuarios más eficiente y utilizar esta información para el desarrollo de las charlas. También la emisión de tickets de entrega permite más flexibilidad a la hora del reconocimiento de entregas y perfiles. A su vez también destaca el listado de consultas y reconocimiento de productos para dar más fluidez a la hora de estas misma.

Conclusión:

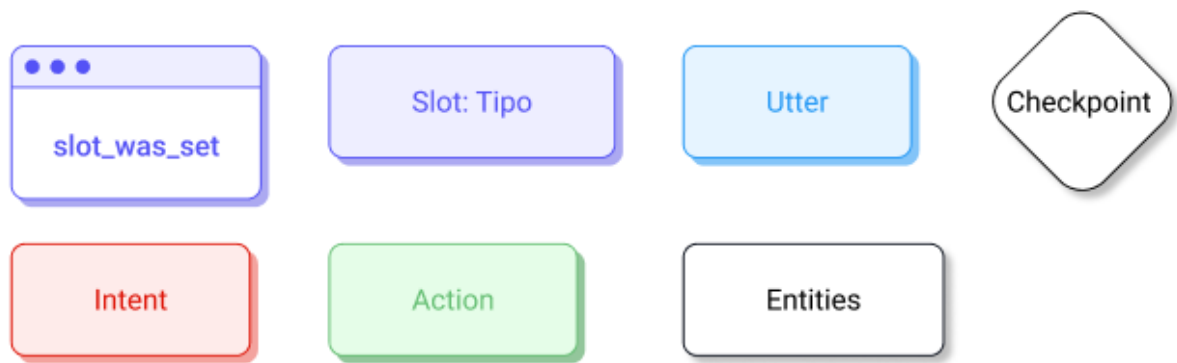
A lo largo de este trabajo abarcamos distintos aspectos de los chatbots, ya sea la utilización de multiagentes, modelar perfiles e identificar compartimentos humanos. A su vez también se utiliza Python para permitir relacionar chatbots entre ellos y ver su comportamiento. Como medio de aprendizaje resultó muy productivo. También pudimos concluir como equipo que los chatbots son el futuro cercano para la asistencia personal en el día a día y que sus posibilidades son muy amplias con las nuevas tecnologías que vendrán en el futuro.

Anexo:

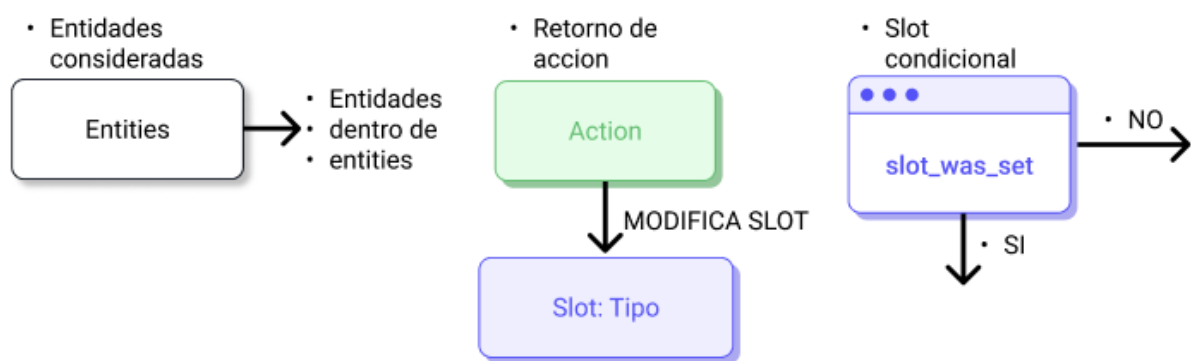
En esta sección definimos el significado y función de los diagramas, estos se utilizan para poder entender pequeñas porciones del funcionamiento del bot, con la intención de generar un conocimiento más abstracto y rápido de la totalidad de su diseño.

Los diagramas, simulan el camino de un diálogo con el usuario (implementado en las Stories), el [diagrama general](#), unifica todos los posibles caminos de diálogo en un solo diagrama.

Elementos del diagrama:



Simplificación de cómo influye el elemento en el diálogo actual:



Bases de datos: Las bases de datos que controla el bot, son archivos .json, los cuales son modificados en el transcurso del diálogo, mediante las **Actions**.

Diagrama general:

En el siguiente diagrama, se visualiza el diseño que tiene el chatbot en su totalidad, representando todos los posibles caminos de diálogo y la interacción entre estos.

Para mayor comodidad visual está cargado en el siguiente [Link: Diagrama general](#)