|  |  |
| --- | --- |
| **UNIVERSIDAD DE SANTIAGO DE CHILE**  **FACULTAD DE INGENIERÍA**  **DEPARTAMENTO DE INGENIERÍA EN INFORMÁTICA** |  |



**Manual de Usuario**

**Chatbot**

**Programado en Scheme**

Gabriel Gaete L.

Santiago de Chile

1 - 2018

# TABLA DE CONTENIDO

[TABLA DE CONTENIDO 2](#_Toc512142538)

[TABLA DE FIGURAS 3](#_Toc512142539)

[CAPÍTULO 1. INTRODUCCIÓN AL CHATBOT 4](#_Toc512142540)

[CAPÍTULO 2. COMPILACIÓN Y EJECUCIÓN 5](#_Toc512142541)

[CAPÍTULO 3. FUNCIONALIDADES Y MODOS DE USO 6](#_Toc512142542)

[3.1 beginDialog 6](#_Toc512142543)

[3.2 sendMessage 6](#_Toc512142544)

[3.3 endDialog 7](#_Toc512142545)

[3.4 rate 7](#_Toc512142546)

[3.5 test 8](#_Toc512142547)

[CAPÍTULO 4. EN CASO DE FALLOS 8](#_Toc512142548)

# TABLA DE FIGURAS

[Figura 1 Relación usuario/chatbot 4](#_Toc512142599)

[Figura 2 Flujo de conversación esperado por el chatbot 5](#_Toc512142600)

[Figura 3 Ejemplo de uso beginDialog. 6](#_Toc512142601)

[Figura 4 Composición de funciones para producir una conversación 6](#_Toc512142602)

[Figura 5 Uso de endDialog 7](#_Toc512142603)

[Figura 6 Ejemplo función rate 7](#_Toc512142604)

[Figura 7 Ejemplo de conversación simulada mediante la función test 8](#_Toc512142605)

# CAPÍTULO 1. INTRODUCCIÓN AL CHATBOT

Un chatbot es un programa que permite simular una conversación con una persona, entregando respuestas automáticas a entradas hechas por un usuario. Estos son utilizados por diferentes marcas y compañías para obtener información, reservar algo, o comprar un producto, entre muchas otras aplicaciones. El chatbot del presente informe permite hacer una compra de pasajes hacia capitales regionales de Chile, asumiendo que el usuario se encuentra en Santiago.

El presente manual pretende explicar cómo utilizar de manera efectiva el chatbot, a través de ejemplos de llamadas a las funciones del programa.

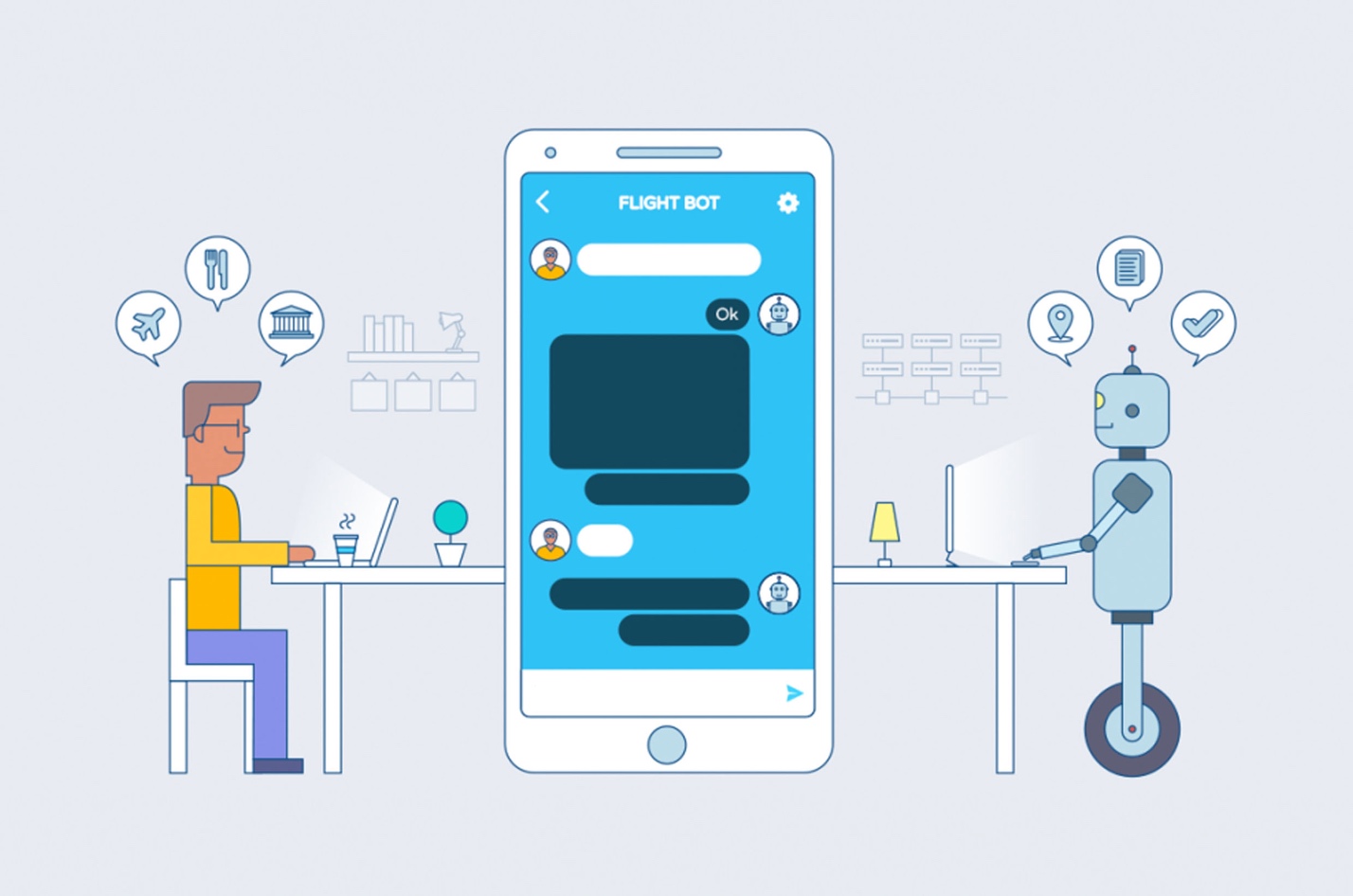


Figura 1 Relación usuario/chatbot (Fuente: http://www.tci.net.pe/sera-2018-ano-los-chatbots/)

Para la correcta utilización de este programa, se debe conocer cómo es el flujo conversacional esperado por el bot frente al usuario, por lo que se recomienda guiarse por el siguiente diagrama para evitar posibles errores en el flujo de la conversación.

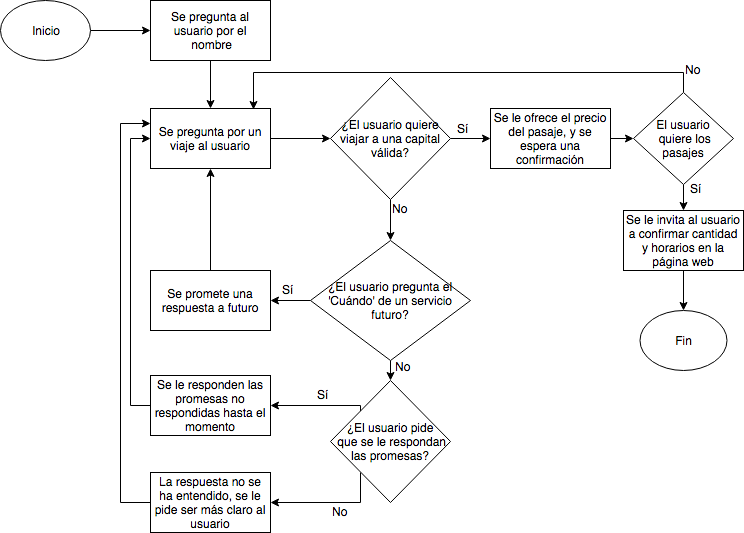


Figura 2 Flujo de conversación esperado por el chatbot

# CAPÍTULO 2. COMPILACIÓN Y EJECUCIÓN

Para ejecutar el programa, solo basta con abrir el archivo “chatbot\_19753546\_GaeteLucero.rkt” con DrRacket, y dar click al botón “Ejecutar”. Una vez realizado este proceso, es posible acceder a las funciones del código.

# CAPÍTULO 3. FUNCIONALIDADES Y MODOS DE USO

Una vez abierto y ejecutado el programa, se pueden utilizar las siguentes funciones para mantener una conversación con el chatbot.

## 3.1 beginDialog

La función beginDialog permite iniciar un nuevo diálogo con el chatbot, donde siempre el mensaje iniciales ofrecido por el bot, consultando de alguna forma, el nombre del usuario. Esta función, retorna un log, el cual debe ser utilizado como argumento de la siguiente función. La siguiente figura, muestra un ejemplo de su retorno.

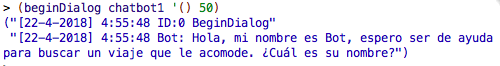


Figura 3 Ejemplo de uso beginDialog.

En la figura 2, se utiliza beginDialog con una estructura chatbot predefinida, llamada chatbot1, un log inicial vacio (representado por ‘() ) y una semilla, la cual puede ser cualquier número entero. En este ejemplo, se utilizó el 50.

## 3.2 sendMessage

La función sendMessage permite el envío de un mensaje por parte del usuario al chatbot. Este mensaje parte del supuesto que se conoce el flujo de conversación que hay detrás, por lo que se recomienda tener en cuenta el diagrama presentado en la Figura 2. Esta función recibe los mismos argumentos que la anterior (**beginDialog**), salvo por el mensaje. La siguiente figura, muestra cómo sería una composición de dos mensajes, más la función anterior para iniciar el chat. Nótese cómo el log retornado por una función, es usado como argumento en la siguiente. Cabe destacar que si se ingresa la pregunta “¿Cuándo …”, el bot responderá con una promesa, la cual puede ser forzada a ser respondida mediante un mensaje que contenga las palabras “Respóndeme” o “responderme”. Considerar ortografía y capitalizado.

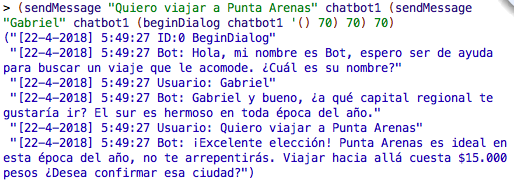


Figura 4 Composición de funciones para producir una conversación

## 3.3 endDialog

Al igual que la función **beginDialog**, se reciben los mismo tres argumentos de entrada. El retorno de esta función es un log con una despedida por parte del bot, y una etiqueta de cierre, como muestra la siguiente figura.

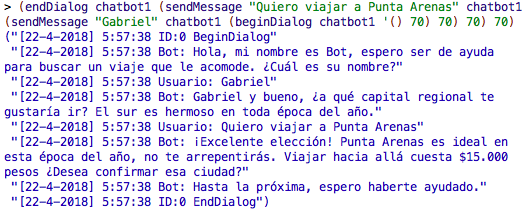


Figura 5 Uso de endDialog

Nótese cómo se le da un cierre adecuado a la conversación anterior mediante esta función.

## 3.4 rate

Esta función recibe como argumentos un chatbot, un puntaje, una función “f” y un log, con los cuales se le permite dar una calificación al bot tras haber finalizado una conversación. Tras esta función, se retorna un chatbot con una nueva calificación, y un nuevo ID. El puntaje del chatbot debe ser un número entero entre un rango de 0 a 5, incluyendo estos valores. El 0 corresponde a la nota en caso de que no se haya podido evaluar, mientras que del 1 al 5 se tiene la calificación más baja a la más alta. Por otro lado, la función f permite que el bot se autoevalúe en base a la última conversación escrita en el log. Esta función se llama “autoRate”. Para mayor claridad, se adjunta la siguiente figura.



Figura 6 Ejemplo función rate

Como se puede ver en la figura 6, la función rate retorna una estructura chatbot, la cual es diferente a chatbot1. Esta función solo puede ser aplicada cuando la conversación ha terminado, de lo contrario, el chatbot no se puede actualizar.

## 3.5 test

Esta función permite ilustrar/simular el desarrollo de una conversación entre un usuario “user” y un chatbot. El argumento user corresponde a una lista de strings, en las que en cada lista se encuentra el mínimo de mensajes para que la conversación tenga un inicio y un término adecuado. Para probar esta función, se brindan tres ejemplos de user dentro del código fuente, con el nombre user1, user2 y user3.

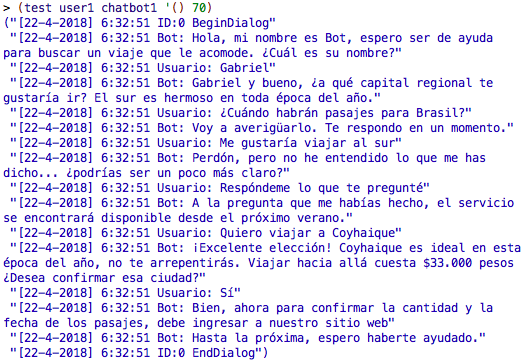


Figura 7 Ejemplo de conversación simulada mediante la función test

# CAPÍTULO 4. EN CASO DE FALLOS

En caso de cualquier tipo de fallos, se recomienda seguir los ejemplos presentados en este manual, puesto que, a pesar de usar funciones de pertenencia para verificar argumentos, no se está exento de incurrir en fallos más graves por entradas inválidas.