



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

**Manual de Usuario
Chatbot**

Programado en Java

Gabriel Gaete L.

Santiago de Chile

1 - 2018

TABLA DE CONTENIDO

TABLA DE FIGURAS	3
CAPÍTULO 1. INTRODUCCIÓN AL CHATBOT	4
CAPÍTULO 2. COMPILACIÓN Y EJECUCIÓN.....	5
CAPÍTULO 3. FUNCIONALIDADES Y MODOS DE USO	6
3.1 !beginDialog seed	6
3.2 !saveLog	7
3.3 !rate notaChatbot notaUsuario	8
3.4 !endDialog.....	9
CAPÍTULO 4. EN CASO DE FALLOS	10

TABLA DE FIGURAS

Figura 1 Relación usuario/chatbot.....	4
Figura 2 Flujo de conversación esperado por el chatbot.....	5
Figura 3 Inicio del programa.....	6
Figura 4 Ejemplo de uso beginDialog.	6
Figura 5 Indicando el nombre dentro del chat.....	7
Figura 6 Ingreso de una frase con capital válida	7
Figura 7 Uso de la instrucción !saveLog	8
Figura 8 Archivo generado por !saveLog	8
Figura 9 Uso de !rate	9
Figura 10 Uso !endDialog	9
Figura 11 Uso de !exit.....	9
Figura 12 Error producido en la entrada del usuario	10

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 manual 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 desarrollado en Java, a través de ejemplos del programa funcionando en un sistema operativo MacOSX, aunque también es posible de compilar y posteriormente ejecutar en otro entorno, ya sea Windows o Linux.

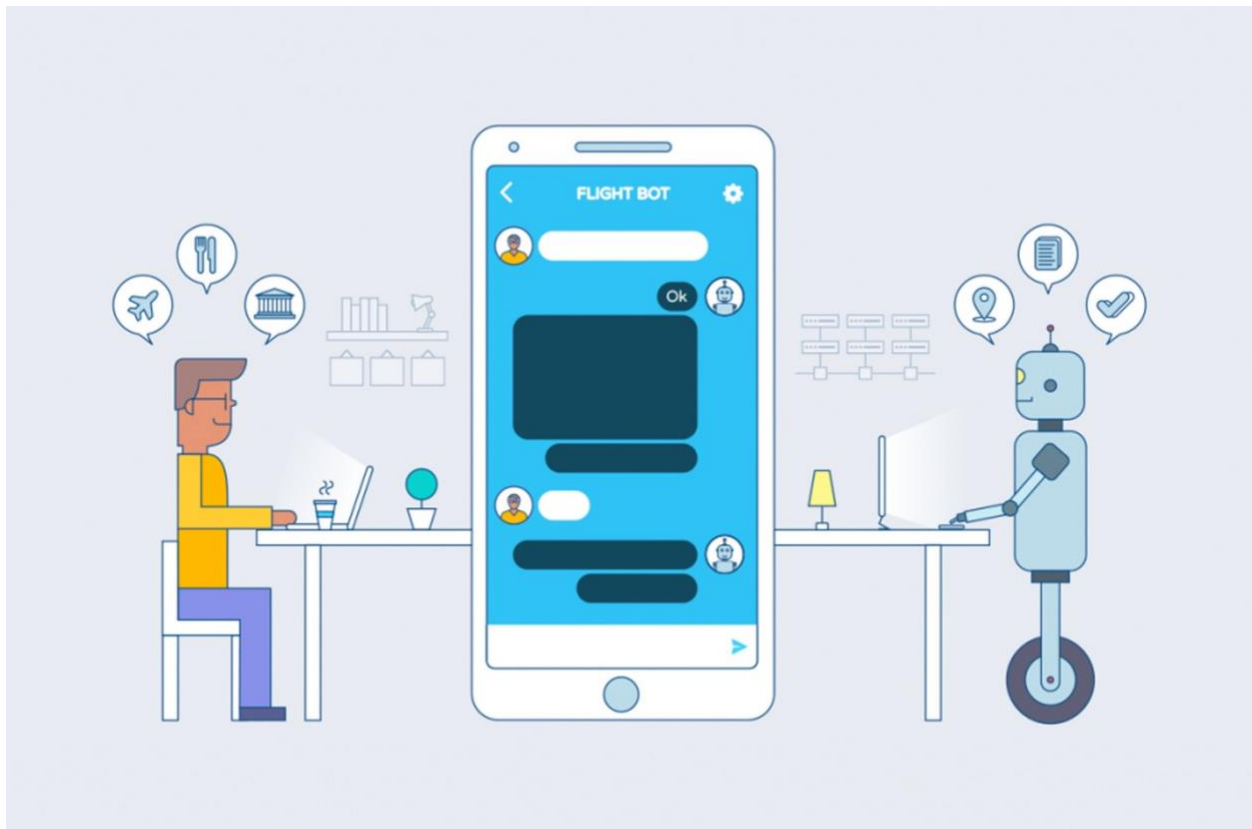


Figura 1 Relación usuario/chatbot (Fuente: <http://www.tci.net.pe/sera-2018-ano-los-chatbots/>)

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.

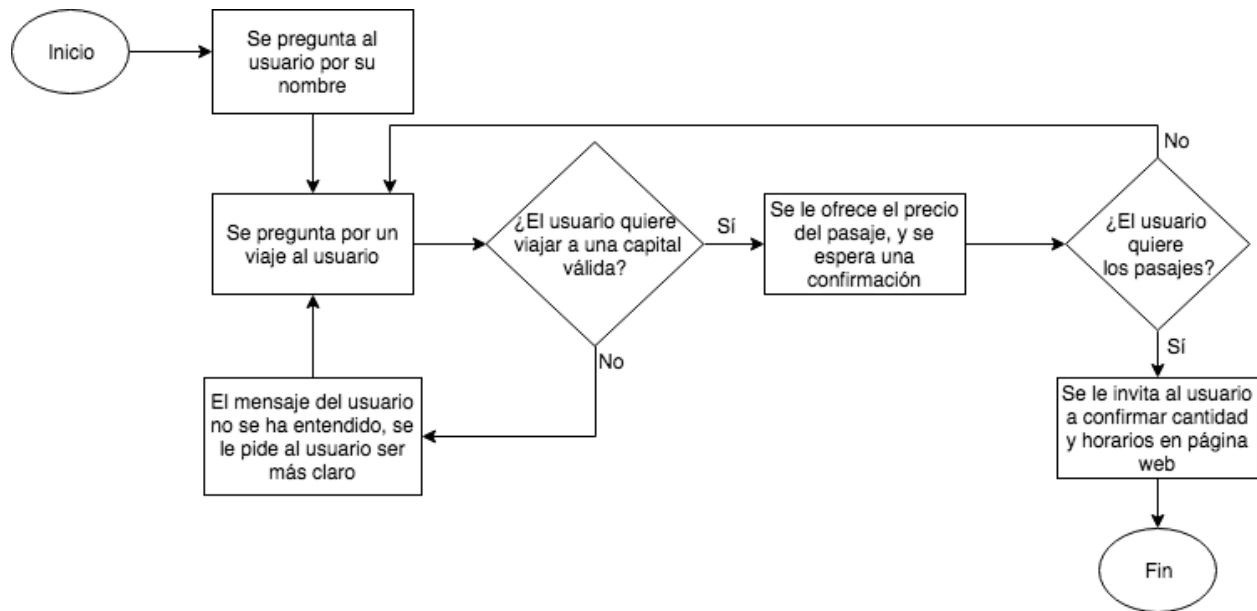


Figura 2 Flujo de conversación esperado por el chatbot

CAPÍTULO 2. COMPILACIÓN Y EJECUCIÓN

Para ejecutar, primero es necesario compilar todos los archivos de extensión *java* del programa. El comando a utilizar, tanto en Linux como en Windows es el siguiente:

```
X javac *.java
```

Este comando debe ingresarse a través de la terminal, estando ubicados en el mismo directorio que el código fuente, es decir, dentro de la carpeta “*chatbot*” que acompaña a este documento.

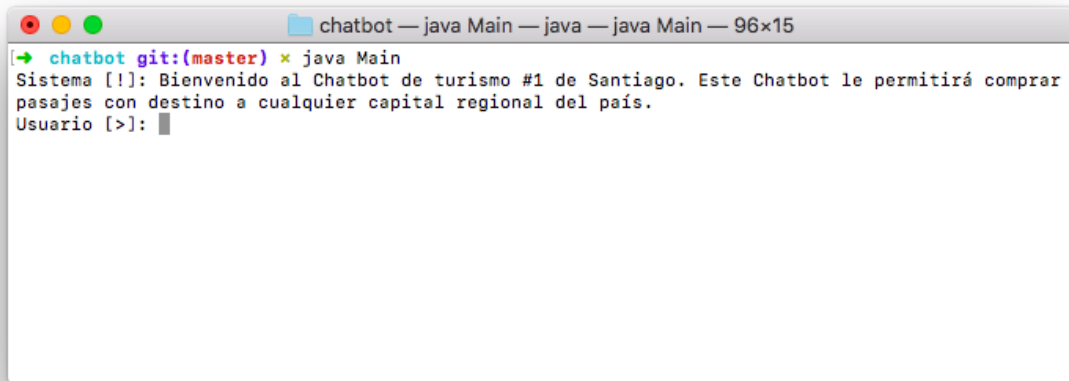
Una vez utilizado este comando, el código queda completamente compilado y listo para ser utilizado. Ahora, para ejecutar el programa, se debe utilizar el siguiente comando:

```
X java Main
```

Realizado esto, se dará inicio al chat que permite hacer la compra de pasajes a capitales regionales.

CAPÍTULO 3. FUNCIONALIDADES Y MODOS DE USO

Ejecutado el programa, se da la bienvenida a través de un mensaje del sistema, recordando que el uso de este chatbot es para la compra de pasajes a capitales regionales del país.



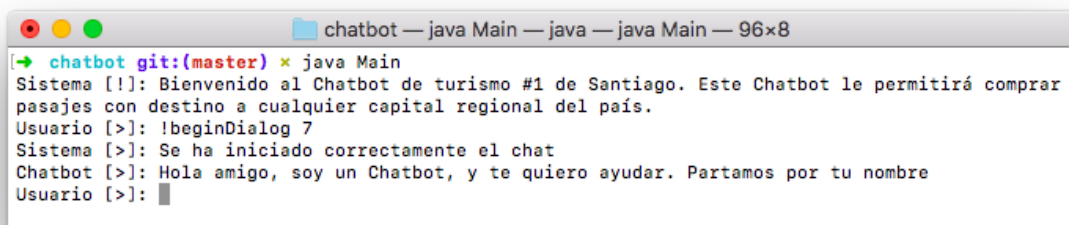
```
chatbot — java Main — java — java Main — 96x15
[→ chatbot git:(master) * java Main
Sistema [!]: Bienvenido al Chatbot de turismo #1 de Santiago. Este Chatbot le permitirá comprar
pasajes con destino a cualquier capital regional del país.
Usuario [>]:
```

Figura 3 Inicio del programa

Como muestra la figura 3, una vez dado el mensaje de bienvenida, el sistema queda esperando por una interacción del usuario. Lo que se debe hacer aquí para inicializar la conversación con el chatbot, es utilizar la siguiente instrucción

3.1 !beginDialog seed

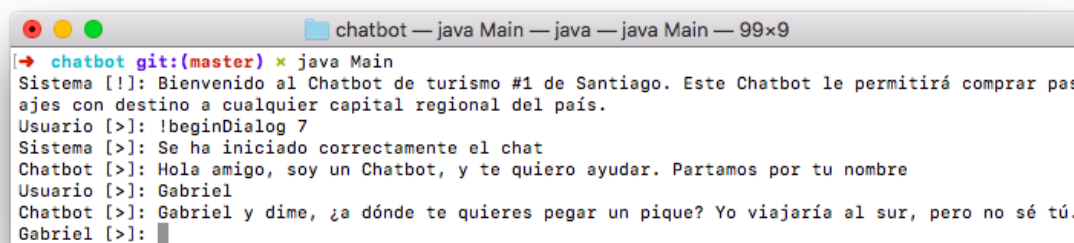
La instrucción **!beginDialog seed** corresponde a la primera instrucción que el sistema debe recibir, con el fin de inicializar la conversación con un chatbot determinado. La utilización de una semilla o **seed** no es obligatoria, sin embargo, el utilizarla le agrega cierta aleatoriedad a las respuestas que el chatbot podría entregar. La siguiente figura muestra la forma en que debe ser utilizada esta instrucción:



```
chatbot — java Main — java — java Main — 96x8
[→ chatbot git:(master) * java Main
Sistema [!]: Bienvenido al Chatbot de turismo #1 de Santiago. Este Chatbot le permitirá comprar
pasajes con destino a cualquier capital regional del país.
Usuario [>]: !beginDialog 7
Sistema [>]: Se ha iniciado correctamente el chat
Chatbot [>]: Hola amigo, soy un Chatbot, y te quiero ayudar. Partamos por tu nombre
Usuario [>]:
```

Figura 4 Ejemplo de uso beginDialog.

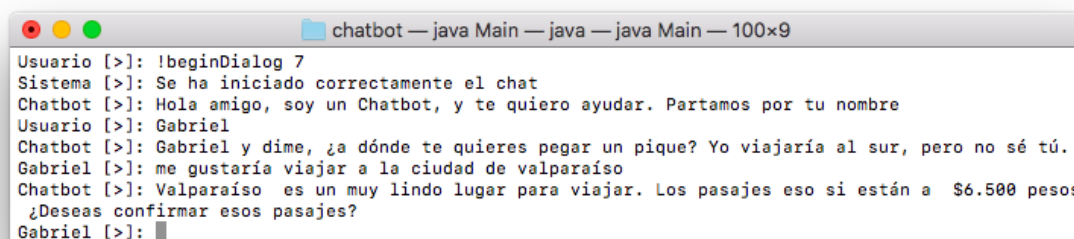
Una vez iniciado el chatbot como en la figura 4, lo primero que se le pregunta al usuario es por el nombre. Una cosa importante a tener en cuenta es que el texto ingresado en este campo, corresponderá al nombre, por lo que si se le escribe al chatbot “mi nombre es (nombre)”, para el chatbot, su nombre será “mi nombre es (nombre)”. La siguiente figura muestra la forma correcta de ingresar un nombre al chat:



```
chatbot git:(master) * java Main
Sistema [!]: Bienvenido al Chatbot de turismo #1 de Santiago. Este Chatbot le permitirá comprar pasajes con destino a cualquier capital regional del país.
Usuario [>]: !beginDialog 7
Sistema [>]: Se ha iniciado correctamente el chat
Chatbot [>]: Hola amigo, soy un Chatbot, y te quiero ayudar. Partamos por tu nombre
Usuario [>]: Gabriel
Chatbot [>]: Gabriel y dime, ¿a dónde te quieres pegar un pique? Yo viajaría al sur, pero no sé tú.
Gabriel [>]:
```

Figura 5 Indicando el nombre dentro del chat

Una vez dado el nombre, se debe ingresar una frase que contenga una capital regional a la cual el usuario desee viajar. No es necesario que se respete el uso de mayúsculas, sin embargo, si se debe respetar la ortografía de la ciudad (por ejemplo, Valparaíso, que lleva tilde en una de sus letras). La siguiente imagen muestra un ejemplo de frase con una capital válida:



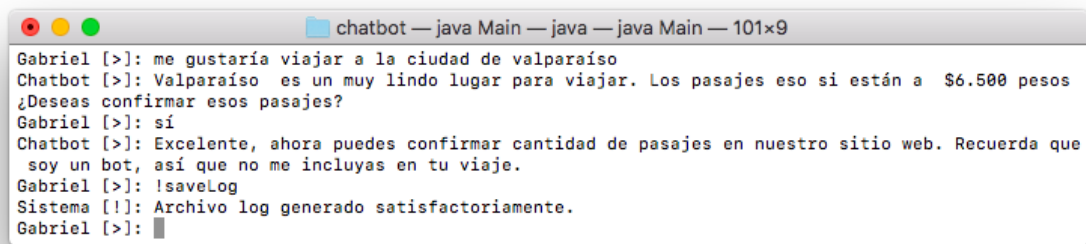
```
chatbot — java Main — java — java Main — 100x9
Usuario [>]: !beginDialog 7
Sistema [>]: Se ha iniciado correctamente el chat
Chatbot [>]: Hola amigo, soy un Chatbot, y te quiero ayudar. Partamos por tu nombre
Usuario [>]: Gabriel
Chatbot [>]: Gabriel y dime, ¿a dónde te quieres pegar un pique? Yo viajaría al sur, pero no sé tú.
Gabriel [>]: me gustaría viajar a la ciudad de valparaíso
Chatbot [>]: Valparaíso es un muy lindo lugar para viajar. Los pasajes eso si están a $6.500 pesos
¿Deseas confirmar esos pasajes?
Gabriel [>]:
```

Figura 6 Ingreso de una frase con capital válida

Además del intercambio de mensajes que se ha mostrado en el presente documento, también es posible guardar un registro de la conversación tenida con el chat, mediante el uso de la siguiente instrucción.

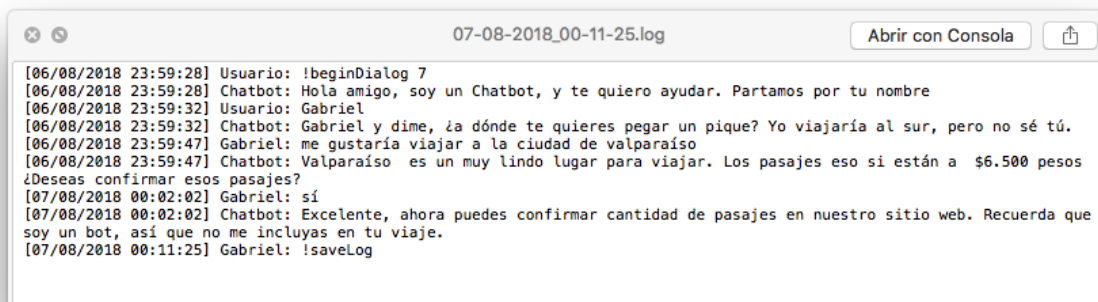
3.2 !saveLog

La instrucción **!saveLog** permite guardar dentro de un archivo un registro de la conversación tenida con el chatbot hasta el momento del uso de esta instrucción. Siguiendo con la conversación mantenida en la figura 6, la siguiente figura retrata el uso de esta instrucción, y el resultado que esta entrega.



```
chatbot — java Main — java — java Main — 101x9
Gabriel [>]: me gustaría viajar a la ciudad de valparaíso
Chatbot [>]: Valparaíso es un muy lindo lugar para viajar. Los pasajes eso si están a $6.500 pesos
¿Deseas confirmar esos pasajes?
Gabriel [>]: sí
Chatbot [>]: Excelente, ahora puedes confirmar cantidad de pasajes en nuestro sitio web. Recuerda que
soy un bot, así que no me incluyas en tu viaje.
Gabriel [>]: !saveLog
Sistema [!]: Archivo log generado satisfactoriamente.
Gabriel [>]:
```

Figura 7 Uso de la instrucción !saveLog



```
07-08-2018_00-11-25.log
[06/08/2018 23:59:28] Usuario: !beginDialog 7
[06/08/2018 23:59:28] Chatbot: Hola amigo, soy un Chatbot, y te quiero ayudar. Partamos por tu nombre
[06/08/2018 23:59:32] Usuario: Gabriel
[06/08/2018 23:59:32] Chatbot: Gabriel y dime, ¿a dónde te quieres pegar un pique? Yo viajaría al sur, pero no sé tú.
[06/08/2018 23:59:47] Gabriel: me gustaría viajar a la ciudad de valparaíso
[06/08/2018 23:59:47] Chatbot: Valparaíso es un muy lindo lugar para viajar. Los pasajes eso si están a $6.500 pesos
¿Deseas confirmar esos pasajes?
[07/08/2018 00:02:02] Gabriel: sí
[07/08/2018 00:02:02] Chatbot: Excelente, ahora puedes confirmar cantidad de pasajes en nuestro sitio web. Recuerda que
soy un bot, así que no me incluyas en tu viaje.
[07/08/2018 00:11:25] Gabriel: !saveLog
```

Figura 8 Archivo generado por !saveLog

En la figura 8, se puede ver que además de guardar el registro de mensajes intercambiados hasta el momento, también se guarda un registro de la fecha y de la hora en que cada uno de los mensajes fue enviado, además del remitente de este mensaje.

Por último, se deja la posibilidad de que el usuario evalúe la calidad del chat, poniendo una nota al chatbot, y una nota a él mismo (a modo de autoevaluación), con el fin de mantener un registro de la calidad de la atención. Esto se permite a través de la instrucción siguiente.

3.3 !rate notaChatbot notaUsuario

La instrucción **!rate notaChatbot notaUsuario** permite evaluar la calidad de la atención, a través de una evaluación de cada participante del chat (tanto una evaluación directa al chatbot, como una autoevaluación por parte del usuario). Esta nota debe ser un número entero entre 0 y 5, ambos incluidos, siendo el 1 la nota más baja, un 5 la nota más alta, y un 0 si se requiere más información para entregar una evaluación. La siguiente figura muestra el modo de uso de la instrucción !rate.


```
chatbot — java Main — java — java Main — 101x8
¿Deseas confirmar esos pasajes?
Gabriel [>]: sí
Chatbot [>]: Excelente, ahora puedes confirmar cantidad de pasajes en nuestro sitio web. Recuerda que soy un bot, así que no me incluyas en tu viaje.
Gabriel [>]: !saveLog
Sistema [!]: Archivo log generado satisfactoriamente.
Gabriel [>]: !rate 5 4
Chatbot [>]:
```

Figura 9 Uso de !rate

Por último, para finalizar la conversación con el chatbot, se debe utilizar una instrucción que indique el término de la conversación.

3.4 !endDialog

La instrucción !endDialog permite finalizar la conversación mantenida con el chatbot actual, procediendo a una eventual despedida por parte del chatbot como respuesta, tal como muestra la siguiente figura:

```
chatbot — java Main — java — java Main — 99x8
ue soy un bot, así que no me incluyas en tu viaje.
Gabriel [>]: !saveLog
Sistema [!]: Archivo log generado satisfactoriamente.
Gabriel [>]: !rate 5 4
Gabriel [>]: !endDialog
Chatbot [>]: Adiós amigo, fue un gusto hablar contigo, espero que disfrutes tu viaje tanto como yo disfruté esta conversación.
Usuario [>]: !
```

Figura 10 Uso !endDialog

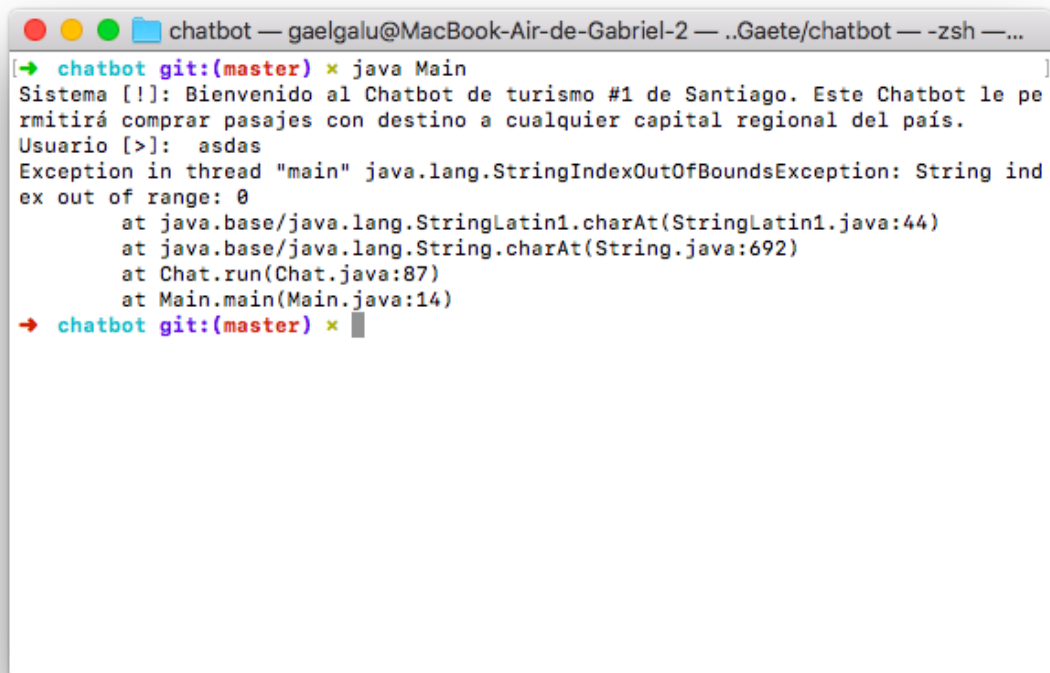
Como se puede apreciar en la figura 11, al utilizar la instrucción !endDialog, se vuelve al estado inicial del programa, esperando por una nueva instrucción. Aquí, el usuario tiene dos opciones, al igual que en el inicio; o iniciar un nuevo chat, mediante el uso de la instrucción adecuada, o el uso de la instrucción **!exit**, la cual permite el cese de la ejecución del programa, como muestra la siguiente figura.

```
chatbot — gaelgalu@MacBook-Air-de-Gabriel-2 — ..Gaete/chatbot — -zsh — 99x5
Chatbot [>]: Adiós amigo, fue un gusto hablar contigo, espero que disfrutes tu viaje tanto como yo disfruté esta conversación.
Usuario [>]: !exit
El programa ha finalizado con éxito
→ chatbot git:(master) x
```

Figura 11 Uso de !exit

CAPÍTULO 4. EN CASO DE FALLOS

Para la entrada del usuario, en caso de iniciar el mensaje con un espacio, el programa producirá el siguiente error:

A screenshot of a terminal window titled 'chatbot — gaelgalu@MacBook-Air-de-Gabriel-2 — ..Gaete/chatbot — -zsh —...'. The terminal shows the execution of 'java Main'. The system message is 'Sistema [!]: Bienvenido al Chatbot de turismo #1 de Santiago. Este Chatbot le permitirá comprar pasajes con destino a cualquier capital regional del país.' The user input is 'Usuario [>]: asdas'. An exception is thrown: 'Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 0'. The stack trace includes: 'at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:44)', 'at java.base/java.lang.String.charAt(String.java:692)', 'at Chat.run(Chat.java:87)', and 'at Main.main(Main.java:14)'. The prompt returns to 'chatbot git:(master) *'.

```
chatbot git:(master) * java Main
Sistema [!]: Bienvenido al Chatbot de turismo #1 de Santiago. Este Chatbot le pe
rmitirá comprar pasajes con destino a cualquier capital regional del país.
Usuario [>]: asdas
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String ind
ex out of range: 0
    at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:44)
    at java.base/java.lang.String.charAt(String.java:692)
    at Chat.run(Chat.java:87)
    at Main.main(Main.java:14)
chatbot git:(master) *
```

Figura 12 Error producido en la entrada del usuario

En caso de que esto ocurra, se recomienda volver a iniciar el programa, mediante el comando especificado en el capítulo 2.

En caso de un error generado por entrada del usuario, y que produzca un cierre anticipado del programa, se recomienda seguir el mismo paso, es decir, volver a ejecutar el programa mediante el comando del capítulo 2.

Por otro lado, en caso de ser un error generado por entrada del usuario, y que no produzca un cierre del programa, se deben seguir las instrucciones dadas por el sistema del chat.

Por último, se recomienda seguir las indicaciones dadas por el presente manual de usuario.