



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

MANUAL DE USUARIO

CHATBOT

Programado en Scheme

Alberto Rodríguez Z.

Santiago de Chile

1- 2018

Tabla de Contenidos

1. Introducción al Chatbot.....	3
2. Compilación y ejecución.....	4
3. Funcionalidades y modo de uso.....	4
3.1 Iniciar Conversación.....	4
3.2 Enviar Mensaje.....	5
3.3 Finalizar conversación.....	6
3.4 Mostrar conversación.....	6
3.5 Evaluación del Chatbot.....	7
4. En caso de fallos.....	8

1. Introducción al Chatbot

El presente manual pretende explicar como utilizar efectivamente el programa para poder comunicarnos correctamente con el chatbot, a través de ejemplos que emulan conversaciones, estas conversaciones se adaptan al contexto para que exista una respuesta coherente del parte del chatbot.

El chatbot se rige bajo el contexto en específico, el cual está ambientado en un negocio de comida rápida llamado Mr. Ham. El negocio tiene un cierto menú que es el que está incluido dentro del vocabulario del chatbot, es decir solo responderá y se registrará por ese menú .

En cuanto a la estructura básica del chatbot, consta de una personalidad, un vocabulario (relacionado al contexto) y evaluaciones de dialogo.

Al ser creado el chatbot bajo el paradigma funcional su creación se constituye de una interacción de funciones, por lo que para poder comunicarse con el chatbot es necesario llamar a las funciones con las que se puede realizar la comunicación.

```
////////////////////////////////////////MENÚ////////////////////////////////////////
-Tipos de Hamburguesa: -Clasica
                        -Vegetariana
                        -Gigante

-////Papas Fritas////: -Chicas
                        -Mediana
                        -Grande

-////////Bebidas////////: -Pepsi
                           -Sprite
                           -Fanta

////////////////////////////////////////
```

Figura 1.- Menú del restaurant

Ya definidas las funciones con las que nos comunicaremos con el chatbot es posible establecer una conversación relacionada al contexto, con los que se obtendrán las distintas posibilidades de conversación

2. Compilación y ejecución

Primero para poder usar el programa es necesario el programa Dr.Racket que es un entorno de desarrollo integrado, con el que se podrá utilizar el programa del chatbot.

Ya descargado el IDE se compila el programa presionado la flecha verde Run ubicada en la esquina superior derecha de la pantalla. También verificar que etiqueta esta escrita en el programa, porque no todas las funciones son posibles en ciertas etiquetas. Este programa está regido por `#! scheme`, implementado bien esto no debería de haber problemas en la compilación. Ya presionado el botón Run si no arroja algún error es porque ha compilado con éxito. Ya compilado se puede ejecutar el programa con normalidad llamando cualquier función incorporada en el sistema.

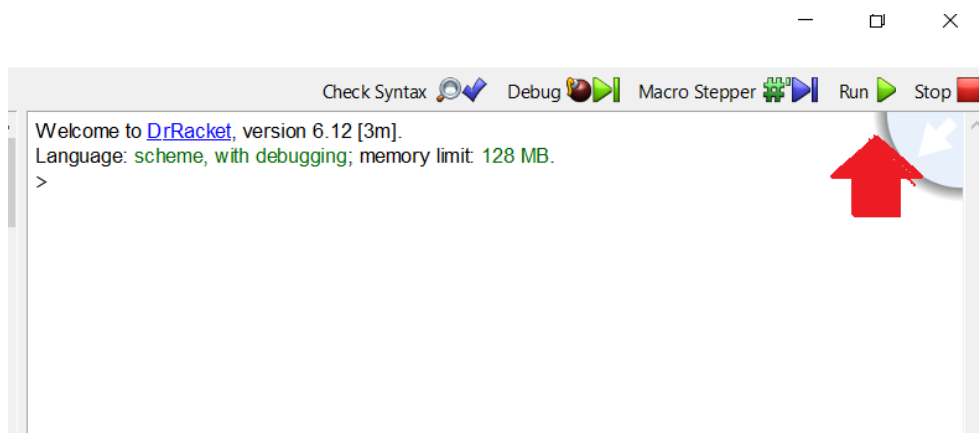


Figura 2.- Forma de compilar en Dr.Racket

3. Funcionalidades y modo de uso

Iniciando el programa se tendrá la pantalla de ejecución de Dr.Racket en blanco lista para iniciar la conversación con chatbot.

3.1 Iniciar conversación

Para poder iniciar una conversación con el chatbot se debe definir un log. El log será donde se guardará la conversación que se llevará a cabo con el chatbot. Se define el log y dentro de los parámetros llamamos a la función `beginDialog`, esta función contiene como parámetros 3 elementos: la estructura chatbot, el log que comienza como una lista vacía y el seed que es un numero pseudoaleatorio que decidirá la personalidad del chatbot.



Figura 3.- Forma correcta definir el log

Ya definido el log1 ejecutamos la función, luego llamamos a log1 y obtendremos el saludo inicial del chatbot.

```
Welcome to DrRacket, version 6.12 [3m].
Language: scheme, with debugging; memory limit: 128 MB.
> (define log1(beginDialog chatbot '() 5))
> log1
(("BeginDialog a las [23/04/2018 19:33:37]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"))
>
```

Figura 4.- Ejecución y llamado de la función beginDialog

3.2 Enviar mensaje

Después de iniciar una conversación con el chatbot podremos enviarle mensajes, estos mensajes deben registrarse con el contexto del menú y también registrarse con el orden de palabras que se incluyen en el archivo Palabras.txt donde especifica como escribirle al chatbot.

Ya conociendo las condiciones de la forma de enviarle mensajes al chatbot se debe definir de nuevo un log que puede ser el mismo que el anterior (en este ejemplo log1) o distinto, pero ahora dentro como parámetro será la función sendMessage, que contiene como parámetros 4 elementos: los mismos de la función anterior pero con algunas modificaciones, como la función log que ya no es una lista vacía sino el log de la función anterior (en este caso log1), en cambio al seed si se quiere que el chatbot mantenga la misma personalidad se debe introducir el mismo seed. El otro parámetro es el mensaje que debe ser escrito como un string, el cual es el mensaje que el usuario le envía al chatbot.

```

Welcome to DrRacket, version 6.12 [3m].
Language: scheme, with debugging; memory limit: 128 MB.
> (define log1(beginDialog chatbot '() 5))
> log1
(("BeginDialog a las [23/04/2018 19:33:37]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"))
> (define log1(sendMessage "hola, me gustaria una hamburguesa" chatbot log1 5))

```

Figura 5.- forma correcta de definir el segundo log

```

Welcome to DrRacket, version 6.12 [3m].
Language: scheme, with debugging; memory limit: 128 MB.
> (define log1(beginDialog chatbot '() 5))
> log1
(("BeginDialog a las [23/04/2018 19:33:37]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"))
> (define log1(sendMessage "hola, me gustaria una hamburguesa" chatbot log1 5))
> log1
(("BeginDialog a las [23/04/2018 19:33:37]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
  "[23/04/2018 20:03:25]-Usuario: hola, me gustaria una hamburguesa"
  "[23/04/2018 20:03:25]-Chatbot: Cual hamburguesa especificamente?"))

```

Figura 6.- Ejecución del log1 y respuesta del chatbot respecto a la pregunta

3.3 Finalizar conversación

Para poder finalizar la conversación se define un nuevo log (puede ser de nuevo log1) y como parámetro se ingresa la función endDialog que contiene los mismos parámetros que la función beginDialog con la diferencia de que el parámetro log no es una lista vacía sino el log definido anteriormente. Al finalizar la conversación el chatbot realiza una autoevaluación que servirá en el punto 3.5

```

Welcome to DrRacket, version 6.12 [3m].
Language: scheme, with debugging; memory limit: 128 MB.
> (define log1(beginDialog chatbot '() 5))
> log1
(("BeginDialog a las [23/04/2018 19:33:37]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"))
> (define log1(sendMessage "hola, me gustaria una hamburguesa" chatbot log1 5))
> log1
(("BeginDialog a las [23/04/2018 19:33:37]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
  "[23/04/2018 20:03:25]-Usuario: hola, me gustaria una hamburguesa"
  "[23/04/2018 20:03:25]-Chatbot: Cual hamburguesa especificamente?"))
> (define log1(endDialog chatbot log1 5))
> log1
(("BeginDialog a las [23/04/2018 19:33:37]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
  "[23/04/2018 20:03:25]-Usuario: hola, me gustaria una hamburguesa"
  "[23/04/2018 20:03:25]-Chatbot: Cual hamburguesa especificamente?"
  "EndDialog a las [23/04/2018 20:10:48]-Chatbot: Hasta luego gracias por su pedido"))

```

Figura 7.-Definición y ejecución del log1 y la finalización de la conversación

3.4 Mostrar conversación

Para poder simular el desarrollo de una conversación completa se utiliza la función test que contiene 4 parámetros: el chatbot, el log que es una lista vacía, el seed que es un número y se añade un usuario que es una lista de string que contiene el conjunto de mensajes predefinidos. Si uno quiere crear una conversación debe crear un usuario (user) con la conversación que quiere que realice con el chatbot.

```

Welcome to DrRacket, version 6.12 [3m].
Language: scheme, with debugging; memory limit: 128 MB.
> (test user1 chatbot '() 3)
(("BeginDialog a las [23/04/2018 20:15:48]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
 "[23/04/2018 20:15:48]-Usuario: hola, me gustaria una hamburguesa"
 "[23/04/2018 20:15:48]-Chatbot: Cual hamburguesa especificamente?"
 "[23/04/2018 20:15:48]-Usuario: quiero una hamburguesa clasica"
 "[23/04/2018 20:15:48]-Chatbot: Bueno quiere algo mas?"
 "[23/04/2018 20:15:48]-Usuario: quiero unas papas medianas"
 "[23/04/2018 20:15:48]-Chatbot: Bueno quiere algo mas?"
 "[23/04/2018 20:15:48]-Usuario: que bebidas tiene"
 "[23/04/2018 20:15:48]-Chatbot: Tenemos pepsi,fanta y sprite"
 "[23/04/2018 20:15:48]-Usuario: quiero una bebida fanta"
 "[23/04/2018 20:15:48]-Chatbot: bueno, quiere algo mas?"
 "[23/04/2018 20:15:48]-Usuario: no gracias"
 "[23/04/2018 20:15:48]-Chatbot: Bueno, anotado todo"
 "EndDialog a las [23/04/2018 20:15:48]-Chatbot: Adios que le vaya bien"))

```

Figura 8.-Definicion y ejecuci3n de test con el resultado de una conversaci3n

3.5 Evaluaci3n del chatbot

Para poder evaluar el desempe1o de una conversaci3n se utiliza la funci3n `rate`, para que el chatbot pueda autoevaluarse se debe finalizar una conversaci3n (con la funci3n `endDialog`) y as3 poder realizar la funci3n `rate`, esta funci3n consta de 4 par3metros: el chatbot, el score que es la evaluaci3n que le da el usuario a la conversaci3n del chatbot, este valor debe estar entre 0 y 5 para que sea considerado muy malo o muy bueno. Tambi3n se ingresa el log que este pasara a la funci3n que lo autoevaluara y enviara un valor entre 0 y 5. El resultado que nos entregara es una lista con la evaluaci3n del usuario y la autoevaluaci3n del chatbot.

```

(("BeginDialog a las [23/04/2018 20:28:23]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
 "[23/04/2018 20:28:57]-Usuario: quiero unas papas fritas chicas"
 "[23/04/2018 20:28:57]-Chatbot: Bueno quiere algo mas?"
 "[23/04/2018 20:29:35]-Usuario: que bebidas tiene ?"
 "[23/04/2018 20:29:35]-Chatbot: Tenemos pepsi,fanta y sprite"))
> (define log1(sendMessage "quiero una bebida sprite" chatbot log1 15))
> log1
(("BeginDialog a las [23/04/2018 20:28:23]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
 "[23/04/2018 20:28:57]-Usuario: quiero unas papas fritas chicas"
 "[23/04/2018 20:28:57]-Chatbot: Bueno quiere algo mas?"
 "[23/04/2018 20:29:35]-Usuario: que bebidas tiene ?"
 "[23/04/2018 20:29:35]-Chatbot: Tenemos pepsi,fanta y sprite"
 "[23/04/2018 20:29:50]-Usuario: quiero una bebida sprite"
 "[23/04/2018 20:29:50]-Chatbot: bueno, quiere algo mas?"))
> (define log1(sendMessage "no gracias" chatbot log1 15))
> log1
(("BeginDialog a las [23/04/2018 20:28:23]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
 "[23/04/2018 20:28:57]-Usuario: quiero unas papas fritas chicas"
 "[23/04/2018 20:28:57]-Chatbot: Bueno quiere algo mas?"
 "[23/04/2018 20:29:35]-Usuario: que bebidas tiene ?"
 "[23/04/2018 20:29:35]-Chatbot: Tenemos pepsi,fanta y sprite"
 "[23/04/2018 20:29:50]-Usuario: quiero una bebida sprite"
 "[23/04/2018 20:29:50]-Chatbot: bueno, quiere algo mas?"
 "[23/04/2018 20:30:06]-Usuario: no gracias"
 "[23/04/2018 20:30:06]-Chatbot: Bueno, anotado todo"))
> (define log1(endDialog chatbot log1 15))
> log1
(("BeginDialog a las [23/04/2018 20:28:23]-Chatbot: Buenas tardes bienvenido a Mr. Ham que desea ordenar?"
 "[23/04/2018 20:28:57]-Usuario: quiero unas papas fritas chicas"
 "[23/04/2018 20:28:57]-Chatbot: Bueno quiere algo mas?"
 "[23/04/2018 20:29:35]-Usuario: que bebidas tiene ?"
 "[23/04/2018 20:29:35]-Chatbot: Tenemos pepsi,fanta y sprite"
 "[23/04/2018 20:29:50]-Usuario: quiero una bebida sprite"
 "[23/04/2018 20:29:50]-Chatbot: bueno, quiere algo mas?"
 "[23/04/2018 20:30:06]-Usuario: no gracias"
 "[23/04/2018 20:30:06]-Chatbot: Bueno, anotado todo"
 "EndDialog a las [23/04/2018 20:30:24]-Chatbot: Adios que le vaya bien"))
> (rate chatbot 4 log1 log1)
((4 1))

```

Figura 9.- Desarrollo de una conversaci3n y llamado de la funci3n `rate`

4. En caso de fallos

- En caso de ingresar mal un valor y devuelva un error o falso, debe compilar de nuevo el programa y empezar todo desde el principio
- Si define mal el log en algún momento de la conversación, debe definir de nuevo la función en donde esta, pero con el log anterior para no perder la conversación guardada
- Si el usuario ingresa un mensaje que no se encuentra en el vocabulario del chatbot este le devolverá un mensaje que no entendió el mensaje que lo pueda repetir y la conversación se guardara en el log.