



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

**MANUAL DE USUARIO**

**CHATBOT**

**Programado en Prolog**

Alberto Rodríguez Z.

Santiago de Chile

1- 2018

## Tabla de Contenidos

<b>1. Introducción al Chatbot.....</b>	<b>3</b>
<b>2. Compilación y ejecución.....</b>	<b>4</b>
<b>3. Funcionalidades y modo de uso.....</b>	<b>4</b>
3.1 Iniciar Conversación.....	4
3.2 Enviar Mensaje.....	5
3.3 Finalizar conversación.....	5
3.4 Representación.....	6
3.5 Mostrar conversación.....	6
3.6 Posibles respuestas.....	7
<b>4. En caso de fallos.....</b>	<b>7</b>

# 1. Introducción al Chatbot

El presente manual pretende explicar cómo 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 y un vocabulario (relacionado al contexto).

Al ser creado el chatbot bajo el paradigma lógico su creación se constituye de una unión de predicados, por lo que para poder comunicarse con el chatbot es necesario llamar a los predicados con los que se puede realizar la comunicación.

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

-////Papás Fritas////: -Chicas
                        -Mediana
                        -Grande

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

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

Figura 1.- Menú del restaurant

Ya definidos los predicados con los 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 SWI-Prolog que es un entorno de desarrollo integrado, con el que se podrá utilizar el programa del chatbot.

Ya descargado el IDE se tiene que buscar el código que contiene el chatbot, para eso debemos presionar el botón File que se encuentra en la esquina superior izquierda de la pantalla, al presionarlo saldrán opciones se debe presionar la opción Edit, con este se debe buscar el código del chatbot en el equipo. Realizado esto se abrirá un editor que contiene el código buscado, para la compilación de este debe escoger la opción Compile que se encuentra en la parte superior de la pantalla del editor y escoger la opción Compile buffer. Ya presionado el botón si no arroja ningún error y en la pantalla de ejecución sale una línea verde es porque se ha compilado con éxito. Ya compilado se puede ejecutar el programa con normalidad llamando cualquier predicado incorporado en el sistema.

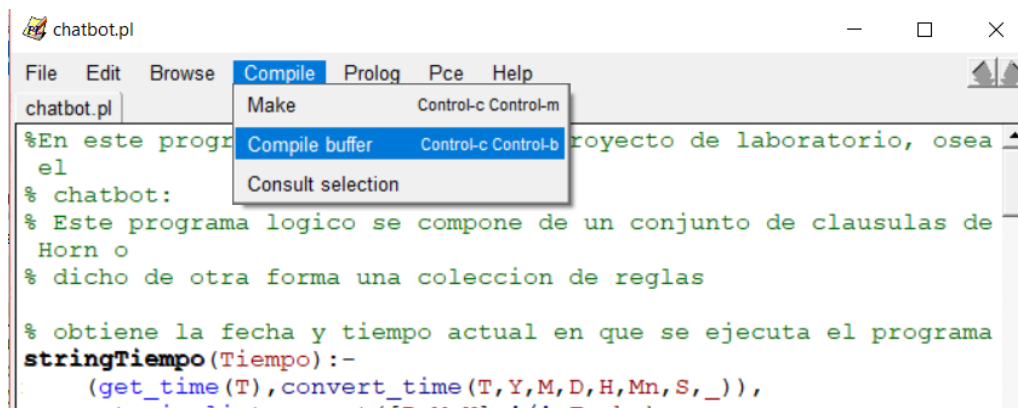


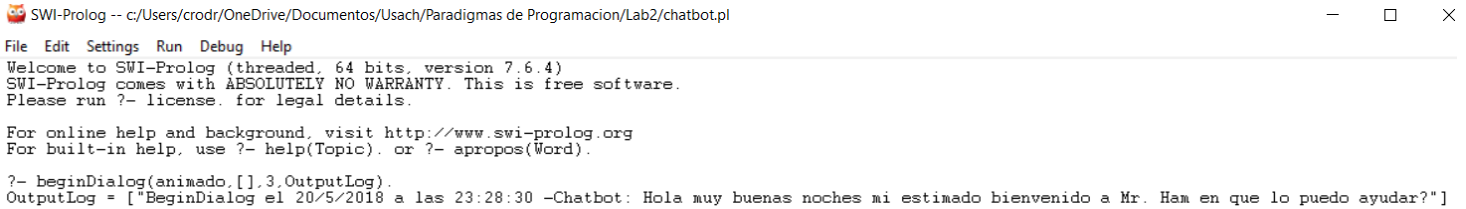
Figura 2.- Forma de compilar en SWI-Prolog

## 3. Funcionalidades y modo de uso

Iniciando el programa se tendrá la pantalla de ejecución de SWI-Prolog en blanco lista para iniciar la conversación con el chatbot.

### 3.1 Iniciar conversación

Para poder iniciar la conversación con el chatbot se debe llama al predicado "beginDialog", este predicado tiene como parámetros 4 elementos: la estructura chatbot donde se debe ingresar con que personalidad se quiere que actúa el chatbot, InputLog que inicialmente debe ser una lista vacía, el seed que es el numero pseudoaleatorio y OutputLog que es el una lista donde va agregado el InputLog y el mensaje de saludo del chatbot.



```

SWI-Prolog -- c:/Users/crodr/OneDrive/Documentos/Usach/Paradigmas de Programacion/Lab2/chatbot.pl
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?- beginDialog(animado,[],3,OutputLog).
OutputLog = ["BeginDialog el 20/5/2018 a las 23:28:30 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?"]

```

Figura 3.- Ejecución y llamado del predicado “beginDialog”, la salida es la lista OutputLog con el mensaje.

### 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 las palabras que deben ir incluidas en una oración para que el chatbot responda coherentemente, estas palabras 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 deben llamar al predicado “beginDialog” pero en vez de llamarlo de nuevo colocando un punto al final se escribe una coma y se escribe el otro predicado que en este caso será el predicado “sendMessage”, este nuevo predicado tiene cuatro parámetros, los tres anteriores y el mensaje del usuario al chatbot que debe ser escrito como string, además el InputLog de este predicado ya no será una lista vacía sino que será el OutputLog del predicado anterior (el seed también puede variar), por lo que la variable salida de este debe tener otro nombre ejemplo OutputLog2 en el cual será la lista con toda la conversación.

```
?- beginDialog(animado,[],3,OutputLog),sendMessage("hola como estas",animado,OutputLog,8,OutputLog2).
```

Figura 4.- Forma correcta de definir los predicados

```

?- beginDialog(animado,[],3,OutputLog),sendMessage("hola como estas",animado,OutputLog,8,OutputLog2).
OutputLog = ["BeginDialog el 20/5/2018 a las 23:54:6 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?"].
OutputLog2 = ["BeginDialog el 20/5/2018 a las 23:54:6 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?",
"20/5/2018 a las 23:54:6 -Usuario: hola como estas", "20/5/2018 a las 23:54:6 -Chatbot: Estoy excelente, gracias por preguntarlo!"] .

```

Figura 5.- Ejecución y salida de las dos listas Output, la ultima con el mensaje del usuario y la respuesta del chatbot.

### 3.3 Finalizar conversación

Para poder finalizar la conversación se debe realizar lo mismo explicado anteriormente, escribir una seguidilla de predicados y el ultimo de ellos debe ser el predicado “endDialog”, este predicado contiene 4 parámetros, los mismo del predicado “beginDialog”, pero el InputLog no es una lista vacía, sino es el OutputLog del predicado anterior a este.

```
?- beginDialog(animado,[],3,OutputLog).sendMessage("hola como estas",animado,OutputLog,8,OutputLog2).sendMessage("quiero una bebida sprite",animado,OutputLog2,12,OutputLog3).sendMessage("no gracias",animado,OutputLog3,10,OutputLog4).endDialog(animado,OutputLog4,3,OutputLog5).
OutputLog = ["BeginDialog el 21/5/2018 a las 0:8:21 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?"].
OutputLog2 = ["BeginDialog el 21/5/2018 a las 0:8:21 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?".
"21/5/2018 a las 0:8:21 -Usuario: hola como estas". "21/5/2018 a las 0:8:21 -Chatbot: Estoy excelente, gracias por preguntar!"].
OutputLog3 = ["BeginDialog el 21/5/2018 a las 0:8:21 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?".
"21/5/2018 a las 0:8:21 -Usuario: hola como estas". "21/5/2018 a las 0:8:21 -Chatbot: Estoy excelente, gracias por preguntar!". "21/5/2018 a las 0:8:21 -Usuario: quiero una bebida sprite". "21/5/2018 a las 0:8:21 -Chatbot: Buena bebida, algo mas!"].
OutputLog4 = ["BeginDialog el 21/5/2018 a las 0:8:21 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?".
"21/5/2018 a las 0:8:21 -Usuario: hola como estas". "21/5/2018 a las 0:8:21 -Chatbot: Estoy excelente, gracias por preguntar!". "21/5/2018 a las 0:8:21 -Usuario: quiero una bebida sprite". "21/5/2018 a las 0:8:21 -Chatbot: Buena bebida, algo mas!". "21/5/2018 a las 0:8:21 -Usuario: no gracias". "21/5/2018 a las 0:8:21 -Chatbot: oh bueno,espero que lo disfrute!"].
OutputLog5 = ["BeginDialog el 21/5/2018 a las 0:8:21 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?".
"21/5/2018 a las 0:8:21 -Usuario: hola como estas". "21/5/2018 a las 0:8:21 -Chatbot: Estoy excelente, gracias por preguntar!". "21/5/2018 a las 0:8:21 -Usuario: quiero una bebida sprite". "21/5/2018 a las 0:8:21 -Chatbot: Buena bebida, algo mas!". "21/5/2018 a las 0:8:21 -Usuario: no gracias". "21/5/2018 a las 0:8:21 -Chatbot: oh bueno,espero que lo disfrute". "EndDialog el 21/5/2018 a las 0:8:21 -Chatbot: Suerte en todo y gracias por su pedido"].
```

Figura 6.- Ejecución de una conversación, cerrando con el “endDialog” (marcado en rojo), y mostrando las salidas de cada predicado.

Después de ver la salida de los predicados que es una lista se hace confuso entender la conversación, por lo que en el punto 3.4 se explicará cómo se puede representar de mejor manera.

### 3.4 Representación

La manera de ver la conversación no es la mejor, por lo que para tener una mejor representación de esta se debe llamar al predicado “logToStr”, este predicado contiene 2 parámetros: Log que es la lista que contiene la conversación, para poder representar la conversación completa de debe colocar al predicado “logToStr” como ultimo predicado y en este parámetro la lista de salida del predicado anterior. El otro parámetro es la salida StrRep que será un string con la conversación, para poder representarlo mejor se realizara la función propia de Prolog “write”.

```
?- beginDialog(animado,[],3,OutputLog).sendMessage("hola como estas",animado,OutputLog,8,OutputLog2).sendMessage("quiero una bebida sprite",animado,OutputLog2,12,OutputLog3).sendMessage("no gracias",animado,OutputLog3,10,OutputLog4).endDialog(animado,OutputLog4,3,OutputLog5).logToStr(OutputLog5,StrRep).write(StrRep).
BeginDialog el 21/5/2018 a las 0:24:56 -Chatbot: Hola muy buenas noches mi estimado bienvenido a Mr. Ham en que lo puedo ayudar?
21/5/2018 a las 0:24:56 -Usuario: hola como estas
21/5/2018 a las 0:24:56 -Chatbot: Estoy excelente, gracias por preguntar!
21/5/2018 a las 0:24:56 -Usuario: quiero una bebida sprite
21/5/2018 a las 0:24:56 -Chatbot: Buena bebida, algo mas!
21/5/2018 a las 0:24:56 -Usuario: no gracias
21/5/2018 a las 0:24:56 -Chatbot: oh bueno,espero que lo disfrute
EndDialog el 21/5/2018 a las 0:24:56 -Chatbot: Suerte en todo y gracias por su pedido
```

Figura 7.- Representación de la figura 6 aplicando el predicado “logToStr” al final y utilizando un write de este.

### 3.5 Mostrar conversación

Para poder simular el desarrollo de una conversación completa se debe utilizar el predicado “test” que contiene 5 parámetros: el chatbot, el InputLog que debe ser una lista vacía, el seed, la salida que es el OutputLog y el añade un usuario “user” que es una lista de strings 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

```
?- test(user2,agresivo,[],3,OutputLog).logToStr(OutputLog,StrRep).write(StrRep).
BeginDialog el 21/5/2018 a las 0:36:28 -Chatbot: Ya es de noche y pide algo, ya no importa que quiere
21/5/2018 a las 0:36:28 -Usuario: hola como estas
21/5/2018 a las 0:36:28 -Chatbot: Pide algo mejor que preguntar
21/5/2018 a las 0:36:28 -Usuario: que bueno, me gustaria una hamburguesa
21/5/2018 a las 0:36:28 -Chatbot: Pero que tipo quieres? especifica
21/5/2018 a las 0:36:28 -Usuario: que pesado
21/5/2018 a las 0:36:28 -Chatbot: Si que tanto problema?, ya mejor pide algo
21/5/2018 a las 0:36:28 -Usuario: ok..., quiero una bebida pepsi
21/5/2018 a las 0:36:28 -Chatbot: YAYÁ anotado, algo mas?
21/5/2018 a las 0:36:28 -Usuario: no
21/5/2018 a las 0:36:28 -Chatbot: Ya suficiente que bueno
EndDialog el 21/5/2018 a las 0:36:28 -Chatbot: Chao
```

Figura 8.- Ejecución de test con un user determinado, se realiza el predicado “logToStr” para mejor representación.

### 3.6 Posibles respuestas

Para poder saber que posibles respuestas tiene un chatbot según una determinada pregunta y un determinado estado de ánimo, se debe llamar al predicado “possibleResponses”. Este predicado tiene 4 parámetros: el mensaje enviado por el usuario escrito como string, el chatbot que es la personalidad de este, el InputLog debe ser una lista vacía y la salida que es una lista con las 3 posibles respuestas del chatbot, si el InputLog no es vacío el predicado devolverá false.

```
?- possibleResponses("quiero una bebida fanta",amable,[],Responses).
Responses = ["Tenemos una promocion asi que se lleva 2x1, algo mas?", "Se nos acabaron por la promocion, pero puede pedir otra", "Bueno esa se lleva, algo mas"] .
```

Figura 9.- Ejecución del predicado “possibleResponses” que contiene la lista con las 3 posibles respuestas

## 4. En caso de fallos

- En caso de ingresar mal un valor y devuelva un error o un falso, debe poner un punto para Prolog formule de nuevo la pregunta y ingresar los valores correctamente
- Asegurarse de definir bien los log de salida ya que si estos no se ingresan en orden puede que el log final no sea el esperado. En este caso solo es revisar bien los log y seguir las instrucciones definidas anteriormente.
- Si el usuario ingresa un valor que no se encuentra en el vocabulario del chatbot este devolverá un mensaje que no entendió el mensaje y la conversación igual será guardada en el log final.
- En caso de no saber que palabras si están en el vocabulario del chatbot leer el archivo palabras.txt adjunto a este manual con el cual se explica que palabras sirven en él.