



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

**LABORATORIO 1  
PARADIGMAS DE PROGRAMACIÓN**

Catalina Morales Rojas

Profesor: Roberto González  
Fecha de Entrega: 23 de abril de 2018

Santiago de Chile

1 - 2018

# **TABLA DE CONTENIDOS**

<b>CAPÍTULO 1. INTRODUCCIÓN .....</b>	<b>4</b>
<b>CAPÍTULO 2. MARCO TEÓRICO .....</b>	<b>5</b>
<b>2.1. PARADIGMA FUNCIONAL .....</b>	<b>5</b>
<b>2.2. FUNCIÓN.....</b>	<b>5</b>
<b>2.3. FUNCIÓN DE ORDEN SUPERIOR .....</b>	<b>5</b>
<b>2.4. FUNCIÓN ANÓNIMA.....</b>	<b>5</b>
<b>2.5. NOTACIÓN PREFIJA .....</b>	<b>5</b>
<b>2.6. EVALUACIÓN PEREZOSA .....</b>	<b>5</b>
<b>CAPÍTULO 3. DESCRIPCIÓN DEL PROBLEMA .....</b>	<b>6</b>
<b>3.1 ANÁLISIS .....</b>	<b>6</b>
<b>CAPÍTULO 4. DESCRIPCIÓN DE LA SOLUCIÓN .....</b>	<b>7</b>
<b>CAPÍTULO 5. RESULTADOS .....</b>	<b>9</b>
<b>CAPÍTULO 6. CONCLUSIONES .....</b>	<b>10</b>
<b>BIBLIOGRAFÍA.....</b>	<b>11</b>

## TABLA DE FIGURAS

Figura 1.- Representación TDA chatbot.....	7
Figura 2.- Función endDialog con currificación de funciones .....	8
Figura 3.- Función sendMessage con composición de funciones.....	8
Figura 4.- Función displayLog con una conversación simulada .....	9

# CAPÍTULO 1. INTRODUCCIÓN

Los chatbots corresponden a programas que permiten mantener una conversación con el usuario, incorporan en su implementación sistemas de inteligencia artificial, lo cual les permite aprender a lo largo de la conversación gustos y preferencias. Las conversaciones que se establecen pueden ser a través de voz o texto. Chatbots conocidos son Siri o cortana, que cuentan principalmente con funciones de búsqueda. El área de servicios es la que cuenta con una mayor implementación de chatbots en la actualidad debido a que ahorran tiempo a los usuarios e incluso adhieren publicidad en su uso.

El objetivo del informe es presentar la manera en que fue planteado, pensado, diseñado y aplicado el laboratorio 1 del curso de Paradigmas de programación, el cual se encuentra implementado en el lenguaje Scheme, bajo el paradigma funcional.

El paradigma funcional se basa en el cálculo Lambda, que se caracteriza principalmente por poseer una notación simple para la definición de funciones matemáticas, utilizar lambda para definir funciones sin nombre y en donde todo se considera una función. Otros de los atributos de este paradigma son funciones anónimas, de orden superior, uso de la recursión, evaluación perezosa, uso de listas y notación prefija.

En el laboratorio se busca crear un chatbot orientado a un área en específica que permita establecer una conversación con el usuario y logré recabar información sobre lo hablado. El chatbot también debe ser capaz de generar respuestas empáticas e historias para que el programa no se convierta en un formulario. La implementación debe ser en español y escrito.

Los objetivos del laboratorio son:

- Aplicar los conceptos aprendidos en cátedra y en la investigación personal en la creación del chatbot, haciendo uso de buenas prácticas de programación.
- Realizar una correcta implementación del paradigma funcional, sin faltar a sus estándares.
- Comprender la estructura de las funciones anónimas, de orden superior, los tipos de datos, entre otros.
- Apreciar las ventajas y desventajas del paradigma funcional para el desarrollo del laboratorio.

## **CAPÍTULO 2. MARCO TEÓRICO**

### **2.1. PARADIGMA FUNCIONAL**

En la programación funcional se basa en el uso de funciones, para las cuales se utilizan constantes, parámetros y valores, pero no variables, por lo que tampoco los ciclos. Dentro de sus ventajas se puede apreciar el manejo de memoria dinámica y funciones de orden superior, por otra parte, tiene un alto costo de ejecución debido a los algoritmos recursivos.

### **2.2. FUNCIÓN**

Toma como argumento la entrada al programa y genera una salida como resultado.

### **2.3. FUNCIÓN DE ORDEN SUPERIOR**

Pueden tomar funciones como parámetro, entregarlas como resultado o realizar variaciones de ambas.

### **2.4. FUNCIÓN ANÓNIMA**

Corresponden a funciones que no poseen nombres, se utilizan principalmente cuando la función solo es ocupada una vez a lo largo del programa.

### **2.5. NOTACIÓN PREFIJA**

También se conoce como notación polaca, se caracteriza por colocar la operación antes que los operandos.

### **2.6. EVALUACIÓN PEREZOSA**

Es una estrategia que retrasa el cálculo de una operación hasta que se ve forzada a realizarlo. Permite reducir el tiempo de ejecución de algunas funciones.

### **2.7. RECURSIÓN NATURAL**

Posee estados pendientes (se encuentran fuera del llamado recursivo), depende mucho de la memoria.

### **2.8. RECURSIÓN DE COLA**

No requiere estados pendientes, por lo que no depende de los llamados anteriores, el resultado se entrega en el último llamado, ocupa menos memoria que la recursión natural.

### **2.9. TDA**

El tipo de dato abstracto puede ser definido como un modelo de cualquier cosa que puede ser operado en una máquina, presenta seis niveles, los cuales son: representación, constructor, funciones de pertenencia, selectores, modificadores y otras funciones.

## CAPÍTULO 3. DESCRIPCIÓN DEL PROBLEMA

Se solicita crear un chatbot en el lenguaje programación Scheme (Racket 6.12), bajo el paradigma funcional y los estándares R6RS, el cual se encontrará orientado a un contexto o temática. El programa debe contar con una serie de parámetros:

1. **Chatbot:** Estructura con diversos parámetros que refleja la personalidad del chatbot y las respuestas que este debe generar.
2. **Log:** Es el registro histórico de las conversaciones que ha tenido el chatbot.
3. **Seed:** Semilla que norma el comportamiento de funciones pseudoalatorias, para lograr la variabilidad de respuestas, tomando en cuenta que no se puede producir un número “random”, ya que el paradigma funcional es biyectivo.

Además, el programa debe contar con las siguientes funciones: **beginDialog**, **sendMessage**, **endDialog**, **Rate** y **Test**.

Las entradas y las salidas de funciones se encuentran definidas en el enunciado del laboratorio, por lo cual en el desarrollo se cuenta con una serie de funciones extras que permiten su correcta implementación.

### 3.1 ANÁLISIS

Para implementar la solución, es necesario tener claro cuál es la representación del problema y definir el TDA. Para este caso, son dos los elementos que presentan mayor importancia y que se utilizan durante todo el programa, los cuales son el chatbot, en donde su representación corresponde a una lista de listas, en la que el primer elemento de cada sublista es un string que funciona como identificador, y el resto corresponde a posibles respuestas a las interrogantes del usuario. El segundo es la representación del log, que corresponde a un string vacío en un inicio, y que se concatena con los nuevos diálogos a medida que se desarrolla la conversación.

Cada función solicitada en el enunciado del laboratorio puede descomponerse en sub problemas y se pueden apoyar en las etapas anteriores del TDA.

Además, para generar una correcta implementación se debe contar con funciones de pertenencia, que indican si los datos ingresados por el usuario son válidos, también con funciones selectoras, las cuales permiten obtener datos del TDA chatbot y agregarlos al string log.

Las funciones modificadoras y otras funciones tienen una implementación sencilla si las funciones anteriormente nombrada se logran implementar de manera correcta, además de generar un TDA robusto.

## CAPÍTULO 4. DESCRIPCIÓN DE LA SOLUCIÓN

Como se mencionó en análisis, la representación de chatbot corresponde a una lista de listas, en donde el primer elemento es un identificador, y el resto la respuesta que se entrega al usuario. El chatbot implementado en este caso corresponde a un guía para un estudiante nuevo en la escuela ficticia Hogwarts de magia y hechicería, al cuál se le podrán realizar una serie de preguntas.

```
;TDA CHATBOT
;Representación: Lista de listas con las respuestas que puede generar el chatbot,
;en el inicio de cada sublista se encuentra un ID que permite contextualizar lo ingresado por el usuario
;con la respuesta del chatbot. Este chatbot corresponde a un guía por Hogwarts.
(define chatbot
  (list (list "Saludo" " HOGWARTSBOT: Buenos Días, bienvenido a la Hogwarts, hoy seré tu guía en el castillo, ¿En qué puedo ayudarte "
    (list "casas" " HOGWARTSBOT: Hay cuatro casas en Hogwarts, ¿cuál es la tuya? " " HOGWARTSBOT: ¿A que casa de Hogwarts pertenec
    (list "Ravenclaw" " HOGWARTSBOT: Ravenclaw es una de las cuatro casas que componen el Colegio Hogwarts de Magia y Hechicería, f
    (list "historia" " HOGWARTSBOT: El Colegio Hogwarts de Magia y Hechicería es un internado mágico ubicado en Escocia[1]. El cast
    (list "asignaturas" " HOGWARTSBOT: Algunas materias son: Adivinación, defensa contra las artes oscuras y cuidado de criaturas m
    (list "deportes" " HOGWARTSBOT: El quidditch es el mejor deporte del mundo mágico,¿juegas en alguna posición ?" " HOGWARTSBOT:
    (list "fin" " HOGWARTSBOT: Fue un gusto ayudarte, distruta tu estadía en Hogwarts" " HOGWARTSBOT: Hogwarts es la mejor escuela
    (list "nula" " HOGWARTSBOT: no logré entender tu pregunta porque es demasiado muggle, quizás quisiste decir " "HOGWARTSBOT: No
    (list "hechizos" " HOGWARTSBOT: Usa alohomora cuando olvides tus llaves " " HOGWARTSBOT: Los tres maleficios imperdonables son
    (list "profesores" " HOGWARTSBOT: Los profesores son Minerva McGonagall de transformaciones, Rubius Hagrid de cuidado de criatu
    (list "ravenclaw" " asdldsklas" "sadnjaskdkasjl " "asdnasdkmasl ")
    (list "puntos" " HOGWARTSBOT: no se cuantos puntos tienen las casas ahora, te lo digo más tarde" " HOGWARTSBOT: no se cuantos p
    (list "ahora" " HOGWARTSBOT: Gryffindor 1000 Hufflepuf 500 Ravenclaw 400 slytherin 100 " " HOGWARTSBOT: Gryffindor 1000 Huffle
    (list "palabrasClave" "ravenclaw" "harry" "potter" "snake" "dumbledore" )
    (list 0 2 0 0 0)
    (list "rate")
  ))
```

Figura 1.- Representación TDA chatbot

Las funciones beginDialog, sendMessage y endDialog, utilizan selectores para obtener los datos solicitados por el usuario del TDA chatbot, y el retorno de cada una de ellas corresponde a un log modificado. El selector (función getRespuesta) para el caso de sendMessage genera una lista con la frase ingresada por el usuario y compara cada uno de los strings con los identificadores presentes en chatbot, si los datos son iguales, mediante la función “random” se obtiene una de las respuestas de chatbot. Para el caso de beginDialog y endDialog, estos siempre obtendrán una respuesta presente en la lista “Saludo” o “fin” respectivamente.

El chatbot presenta memoria a corto, mediano y largo plazo.

- La memoria a corto plazo corresponde a la respuesta que da el chatbot a la solicitud ingresada por el usuario.
- La memoria a mediano plazo se realiza en endDialog, en donde se procede a revisar el log de conversaciones y se obtiene la preferencia de casa del usuario para así dar un mensaje personalizado de despedida al usuario.
- La memoria a largo plazo se realiza con la función rate, en donde en primera instancia se modifica el chatbot agregando la calificación dada por el usuario, la entregada por la métrica del programa y el identificador de la conversación. Luego, se procede a leer el log de la conversación para encontrar la frecuencia que presentan las palabras más usadas de una lista definida en chatbot como “palabrasClave”. En futuras conversaciones, cuando el usuario ingrese una frase que el chatbot no entiende, se le sugiere buscar la palabra con mayor frecuencia.

En la mayor parte de las funciones implementadas se utiliza la recursividad, esto se debe a que en el paradigma funcional no existen las variables, por lo tanto, para recorrer el chatbot o el log se debe avanzar llamando nuevamente a la función, pero con el siguiente elemento.

La recursión de cola es la que se encuentra implementada en la mayor parte de las funciones, ya que en gran parte de los casos el algoritmo debe parar cuando encuentre una palabra en

específico y, de esta manera se entrega inmediatamente el resultado, generando un mejor uso de la memoria debido a que no se presentan estados pendientes.

También se presenta el uso de la currificación en las funciones que se encuentran con una cantidad definida de parámetros de entrada, la currificación en estos casos permite entregar parámetros extras a la función para así obtener todos los resultados solicitados. Un ejemplo de esto es la función endDialog, en donde se agrega a endDialog2 el tiempo en que se desarrolla la conversación.

```
(define (endDialog chatbot log seed)
  (define (endDialog2 chatbot log seed ID time)
    (string-append log "<" (number->string ID) "> " "<" time "> " "*****endDialog *****" (resumenUsuario log))
  )

  (endDialog2 chatbot log seed (ID seed) (time seed))
)
```

Figura 2.- Función endDialog con currificación de funciones.

Finalmente, en la mayor parte de las funciones solicitadas se realizó una descomposición de las funciones para lograr retornar el resultado esperado, el principal ejemplo es la función sendMessage, que hace uso de las funciones sendMessage2 que se encarga de llamar a getRespuesta (selectora) para obtener la frase del chatbot, la función semilla que permite generar el número “random” del elemento que es tomado de chatbot, la función agregar Hora que entrega la hora en que se desarrolla la conversación y cadena que genera una lista con la frase ingresada por el usuario. Por lo cual un gran problema es abordado usando el método de división y conquista y se convierte en pequeños problemas que presentan una menor dificultad.

```
;SendMessage
;Entrada: mensaje del usuario, TDA chatbot, el log anterior y la semilla
;Salida: log actualizado con el mensaje ingresado por el usuario y la respuesta del chatbot, además de la hora
;No presenta recursión, genera un string con el log anterior, el tiempo, el mensaje del usuario y la respuesta del usuario
; que se obtiene llamando a la función sendMessage2
(define (sendMessage msg chatbot log seed)
  (define (agregarHora msg chatbot log seed time)
    (string-append log "<" time "> " "USUARIO: " msg "\n"
      (sendMessage2 (cadena msg) chatbot log seed)
      "\n")
  )
  (agregarHora msg chatbot log seed (time seed))
)
```

Figura 3.- Función sendMessage con composición de funciones.



## CAPÍTULO 5. RESULTADOS

El resultado del laboratorio corresponde a un chatbot que puede generar una conversación con el usuario, cuenta con memoria a corto, mediano y largo plazo, es posible simular una conversación, dar una nota y fueron aplicados tanto la recursión natural como la recursión de cola.

```
> (displayLog chatbot (test user1 chatbot log 1))
<4935> <23:23> *****BeginDialog *****HOGWARTSBOT: Buenas noches, bienvenido a Hogwarts, hoy seré tú guía en el castillo, ¿En qué puedo ayudarte?
<23:23>USUARIO: cuantos puntos tienen las casas ?
HOGWARTSBOT: no se cuantos puntos tienen las casas ahora, te lo digo más tarde
<23:23>USUARIO: yo soy ravenclaw
HOGWARTSBOT: Ravenclaw es una de las cuatro casas que componen el Colegio Hogwarts de Magia y Hechicería, fundada por Rowena Ravenclaw, quién tuvo una hija, cuyo fantasma es la Dama Gris. ¿Qué más quieres saber?
<23:23>USUARIO: quien es harry ?
HOGWARTSBOT: no logré entender tu pregunta porque es demasiado muggle, quizás quisiste decir ravenclaw
<23:23>USUARIO: qué es un hipogrifo ?
HOGWARTSBOT: no logré entender tu pregunta porque es demasiado muggle, quizás quisiste decir ravenclaw
<23:23>USUARIO: me gustaría conocer algunos hechizos
HOGWARTSBOT: Usa alohomora cuando olvides tus llaves
<23:23>USUARIO: cuales son las asignaturas ?
HOGWARTSBOT: Algunas materias son: Adivinación, defensa contra las artes oscuras y cuidado de criaturas mágicas, ¿Quieres saber de alguna?
<23:23>USUARIO: ahora cuantos puntos tienen las casas
HOGWARTSBOT: Gryffindor 1000 Hufflepuf 500 Ravenclaw 400 slytherin 100
<23:23>USUARIO: quienes son los profesores ?
HOGWARTSBOT: Los profesores son Minerva McGonagall de transformaciones, Rubius Hagrid de cuidado de criaturas mágicas, pero ten cuidado en pociones con Snape
<4935> <23:23> *****endDialog *****Espero que ganes muchos puntos para ravenclaw
```

Figura 4.- Función displaylog con una conversación simulada.

A pesar del uso de recursión en la mayor parte del laboratorio, no se presenta un retardo en el desarrollo, esto es debido a que no se maneja una gran cantidad de información, por lo que en otros casos, la implementación de este proyecto en el paradigma funcional no es la mejor solución.

Dentro de las limitaciones del laboratorio, no fue posible abordar una gran cantidad de respuestas a las preguntas del usuario, esto se debe a que se utiliza solo un identificador para relacionar la frase que ingresa el usuario con la respuesta. Por lo cual, si se busca algo en un tiempo verbal distintos al que conoce el chatbot u otra conjugación, a pesar de poseer la información el chatbot no logrará responder.

Finalmente, la evaluación perezosa sólo fue simulada, debido a que no fue posible generar un algoritmo que permitiera su realización.

## CAPÍTULO 6. CONCLUSIONES

Entre las dificultades que presenta el paradigma funcional, específicamente en este laboratorio se encuentra el no uso de variables y ciclos, lo que hace imposible entregar resultados de funciones que requieren recorrer elementos sin usar la recursión, y en casos donde la recursión a utilizar no fue la correcta, se puede generar un uso excesivo de memoria y tiempo de ejecución.

Que el paradigma sea biyectivo, implica tener un especial cuidado en el retorno de las funciones, las cuales para cada elemento en el dominio deben presentar un único elemento en el recorrido, lo que impide el uso de funciones random, por lo que es necesario crear una función que genere un “número aleatorio”.

El uso de recursión de cola fue una buena técnica, ya que permite entregar el resultado de manera inmediata cuando se llega a la condición base. Generar listas con diversos tipos de datos fue una ventaja en el uso del paradigma, ya que permitió almacenar variados tipos de información en solo un elemento (debido a que corresponde a un lenguaje no tipificado). Además, es un paradigma que puede ser denominado como “simple”, ya que las funciones son pequeñas, claras y concisas.

Se logró generar una implementación robusta, ya que las funciones de orden superior en el TDA se apoyan en los niveles más bajos, tales como los constructores, funciones de pertenencia y selectores.

Con respecto a los objetivos planteados al inicio, se logró cumplir con la aplicación de los conceptos aprendidos en cátedra y en la investigación personal para la creación del chatbot, haciendo uso de buenas prácticas de programación, comprender la estructura de las funciones anónimas, de orden superior, los tipos de datos, entre otros y apreciar las ventajas y desventajas del paradigma funcional para el desarrollo del laboratorio.

## BIBLIOGRAFÍA

- Anónimo, 30/06/2017 “Programación funcional” (Recuperado 16/04/2018) <http://programacionfuncional.es/2017/06/30/programacion-funcional-ventajas-desventajas/>
- Departamento de Ciencia de la Computación e Inteligencia Artificial, 2010-2011 “Lenguajes y Paradigmas de Programación” <http://www.dccia.ua.es/dccia/inf/asignaturas/LPP/2010-2011/teoria/tema2.html>