

Paradigmas de la Programación

Informe Laboratorio nro. 1

Paradigma Funcional

*Profesor: Gonzalo Martínez.*

*Alumna: Karina Bustamante H.*

27 diciembre de 2023

[Introducción 2](#_Toc134312314)

[Descripción del problema 2](#_Toc134312315)

[Descripción del paradigma 3](#_Toc134312316)

[Análisis del problema 3](#_Toc134312317)

[Diseño de la solución 3](#_Toc134312318)

[Aspectos de la implementación 4](#_Toc134312319)

[Instrucciones de uso 5](#_Toc134312320)

[Resultados y evaluación](#_Toc134312321) 6

[Conclusiones 7](#_Toc134312322)

[Referencias](#_Toc134312323) 7

# 

# Introducción

La asignatura Paradigmas de la Programación busca enseñar los diferentes paradigmas que existen frente a forma de programar. El presente informe, correspondiente al laboratorio Nº1 de la asignatura, en donde se solicita realizar una implementación basándonos en la programación declarativa – funcional, de un algoritmo para resolver el problema que corresponde al siguiente enunciado:

***“Sistema para la creación, despliegue y administración de chatbots simplificado.”***

En base a lo solicitado y aplicando conceptos del paradigma de programación funcional, usaremos el lenguaje de programación Scheme en la resolución de este problema acotado.

# Descripción del paradigma

La programación funcional corresponde al paradigma donde la programación se asemeja al álgebra ya que podemos reemplazar las ecuaciones por sus resultados, es decir, si llamamos a una función con los mismos argumentos siempre obtendremos los mismos resultados, esto se conoce como ***Referential*** ***Transparency***. La programación funcional se centra en el uso de funciones y la composición de funciones para resolver problemas evitando los cambios de estados. Otros conceptos importantes de la programación funcional son:

* ***Inmutabilidad***: Los datos son inmutables, una vez creados, no se pueden modificar, se crean nuevas estructuras de datos cuando se realizan operaciones.
* ***Funciones Puras***: Una función crea otro estado basado en la estructura que recibe como parámetro, no existen estados intermedios, las salidas solo dependen delas entradas.
* ***Recursividad***: Se fomenta el uso de recursión
* ***Cálculo Lambda***: Es un símbolo que toma argumentos y que retorna una salida, donde estas salidas solo dependen de las entradas.

# Descripción del problema

El propósito fundamental de este proyecto consiste en crear una simulación de un sistema de chatbot empleando el enfoque de la programación funcional y el lenguaje de programación Scheme. Este chatbot pertenece a la categoría de los ITR, lo que significa que está diseñado para ofrecer respuestas estructuradas a través de acciones específicas. Las opciones de interacción pueden variar desde el uso de palabras clave con sinónimos hasta una alternativa más simplificada mediante números o letras. Esto permite a los usuarios seleccionar sus preguntas de una lista predefinida.

# Análisis del problema

El problema que se intenta resolver corresponde a la creación de un sistema de chatbot del tipo ITR que ofrezca respuestas estructuradas mediante acciones específicas, que permita a los usuarios interactuar de manera estructurada, utilizando palabras clave, sinónimos, números o letras. Esta solución se limitará a una lista de preguntas que se podrá generar a través del mismo sistema.

# Diseño de la solución

Para el diseño de la solución se debe identificar los objetivos del chatbot, que tipo de preguntas responderá y cuales son las acciones especificas que debe realizar. Dentro de las acciones clave estará:

* ***Implementación en Scheme***: para implementar la lógica del chatbot. Se definirán las funciones y estructuras de datos necesarias para procesar las preguntas y proporcionar respuestas adecuadas.
* ***Manejo de Preguntas y Respuestas***: Se deberá desarrollar un mecanismo para que el chatbot entienda y procese las preguntas, creando estructuras de datos para mapear preguntas a respuestas o acciones específicas.
* ***Gestión palabras claves***: deberá realizar el reconocimiento exacto de palabras claves o números de opciones para realizar operaciones.

En cuanto al registro de la hora y fecha de creación, se ha utilizado el Unix Timestamp, que cuenta los segundos que han transcurridos desde el 1 de enero de 1970 desde las 00.00 hrs. que corresponde al nacimiento de Unix, este será el formato que utilizaremos para representar la fecha (current-seconds) en la solución.

# Aspectos de la implementación

Para la implementación del sistema simplificado de chatbot, diseñado bajo el paradigma funcional y codificado en el lenguaje de programación Scheme, resaltaremos a continuación algunos aspectos importantes.

***Diseño de Funciones:***

* Chatbot: representa la función principal del chatbot.
* Interfaz Usuario: se realiza a través de la consola, maneja la interacción con el usuario, mostrando mensajes y recibiendo la entrada.
* BaseDeDatos: funciones que manipulan información almacenada de manera funcional.

***Implementación de Funciones:***

* Chatbot: Función para crear un chatbot
* Flow: función para crear un flujo para un chatbot
* Option: función crea opciones para un flujo de conversación.
* System: función que crea un sistema contenedor del sistema simplificado de chatbots
* System-talk-rec: función para comenzar la interacción con el usuario.

***Manejo de Datos***:

• Entrada/salida:

* Se utiliza la función getLine para obtener la entrada del usuario desde la consola.
* Las respuestas del chatbot se imprimen en la consola.
* Con este enfoque funcional, se busca modularizar las funciones y evitar el uso de variables globales, enfocándose en la manipulación de datos de manera más pura y declarativa.

# Instrucciones de uso

Para operar esta simulación del sistema simplificado de chatbots desarrollado en lenguaje Scheme, se provee de un archivo de pruebas\_13452929\_Bustamante.rkt, el que contiene el código de las líneas de pruebas para el sistema. En este archivo se crearon opciones, flujos y chatbots para operar el sistema. Se agregan opciones a flujos ya creados, se agregan flujos a chatbots creados y se agregan los chatbots creados al sistema.

# Resultados y evaluación

A continuación se muestra la autoevaluación de los requerimientos funcionales requeridos por el laboratorio, y el grado de logro alcanzado para cada uno de ellos.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RF Nº | Requerimiento funcional | Grado de alcance | Tipo o cantidad de pruebas | Éxitos | Fracasos | Razones de fallo |
| 1 | TDA’s | 1 |  |  |  |  |
| 2 | TDA Option-Constructor | 1 |  |  |  |  |
| 3 | TDA Flow-Constructor | 1 |  |  |  |  |
| 4 | TDA Flow-flowAddOption | 1 | 3 | 1 | 2 | Verificar duplicidad |
| 5 | TDA Chatbot-Constructor | 1 | 2 | 1 | 1 | verificar duplicidad |
| 6 | TDA Chatbot-chatbotAddFlow | 1 | 2 | 2 |  |  |
| 7 | TDA Usuario - Constructor | 1 | 1 | 1 |  |  |
| 8 | TDA System - Constructor | 1 | 3 | 1 | 2 | Error en orden de argumentos |
| 9 | TDA System - systemAddChatbot | 1 | 3 | 1 | 3 | Error en orden de argumentos |
| 10 | TDA System - systemAddUser | 1 | 2 | 1 | 1 | No mostraba usuario en sistema |
| 11 | TDA System - systemLogin | 1 | 2 | 1 | 1 | No mostraba usuario en sistema |
| 12 | TDA System - systemLogout | 1 | 1 | 1 |  |  |
| 13 | TDA System – system-talk-rec | 0,75 | 2 |  | 2 | Error tipo de argumento que ingresa por consola, espera algo y recibe otra cosa |

**Tabla 1.-** Muestra los requerimientos funcionales y el nivel de logro.

# Conclusiones

En el transcurso de este proyecto, exploramos la creación de un sistema de chatbot simplificado utilizando el lenguaje de programación Scheme y adoptando el Paradigma Funcional. El análisis detallado del problema proporcionó una comprensión clara de los objetivos, requisitos, alcance y limitaciones del proyecto, orientando la estrategia de implementación, sin embargo, durante el desarrollo del sistema nos enfrentamos a una limitación, específicamente en la implementación de la interacción con el chatbot, la cual no pudo ser implementada en toda su funcionalidad.

Al considerar el paradigma funcional, destacamos su enfoque en la manipulación de datos de manera pura y declarativa, evitando el uso de variables globales y fomentando la modularidad de funciones. En comparación con lenguajes como Python y C, conocidos por su versatilidad y eficiencia, el paradigma funcional ofrece una perspectiva diferente en términos de estructuración y gestión de la lógica del programa.

En resumen, este proyecto no solo nos permitió desarrollar un sistema de chatbot bajo el Paradigma Funcional, sino que también nos llevó a reflexionar sobre las diferencias notables con otros lenguajes de programación, como Python y C.

# Referencias

* Campus Virtual, (2023). Paradigmas de Programación. <https://uvirtual.usach.cl/moodle/>
* Clases Grabadas, (2023). Paradigmas de Programación. https://drive.google.com/drive/folders/1F5sP-ATm\_swyIhb5XK4qDwkApUOgguoA?usp=sharing