

Paradigmas de la Programación

Informe Laboratorio nro. 2

Paradigma Lógico

*Profesor: Gonzalo Martínez.*

*Alumna: Karina Bustamante H.*

noviembre de 2023

Tabla de contenido

[Introducción 2](#_Toc134312314)

[Descripción del problema 2](#_Toc134312315)

[Descripción del paradigma 3](#_Toc134312316)

[Análisis del problema 3](#_Toc134312317)

[Diseño de la solución 6](#_Toc134312318)

[Consideraciones de implementación 6](#_Toc134312319)

[Instrucciones de uso 7](#_Toc134312320)

[Resultados y evaluación 8](#_Toc134312321)

[Conclusiones 9](#_Toc134312322)

[Referencias 10](#_Toc134312323)

# Introducción

La asignatura Paradigmas de la Programación busca enseñar los diferentes paradigmas que existen frente a la forma de programar. El presente informe, correspondiente al laboratorio Nº 2 de la asignatura, en donde se solicita realizar una implementación basándonos en los principios del paradigma lógico, de un algoritmo para resolver el problema que corresponde al siguiente enunciado:

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

Esta solución permitirá a un usuario realizar distintas operaciones como: crear, vincular e interactuar entre otras cosas en un sistema centrado en la operación con chatbots.

# Descripción del paradigma

El paradigma lógico corresponde al paradigma de programación declarativo, puede usarse como la especificación de un problema en lugar de establecer los pasos necesarios para llegar a una solución. Se basa en la lógica booleana y no existen funciones ni retornos normales. En este paradigma, se describe el problema en términos de relaciones lógicas entre objetos y se utiliza la inferencia lógica para resolver el problema. Se declara una base de conocimiento que es un conjunto de hechos y reglas que describen las relaciones entre los objetos en un dominio de problema. El programador describe el objetivo o la pregunta que se quiere responder en términos de estas relaciones lógicas. Luego, el sistema de programación utiliza la lógica para deducir la respuesta al objetivo a partir de los hechos y reglas definidos en la base de conocimiento. Los hechos, son un tipo de cláusula que describe una relación entre uno o más término, estos hechos se asumen siempre verdaderos.

Los mecanismos básicos de este paradigma son:

* ***Unificación***: se refiere a la técnica de encontrar una asignación de valores a las variables en una expresión lógica que hace que la expresión sea verdadera. Si se encuentra una asignación de valores que haga que las expresiones sean iguales, la consulta se considera resuelta y se devuelve el resultado
* ***Backtraking automático***: Sucede cuando no se logra la unificación, el proceso da un paso atrás para probar otros caminos para lograr la unificación, si se encuentra una asignación de valores que haga que las expresiones sean iguales, la consulta se considera resuelta y se devuelve el resultado, True.
* ***Inferencia***: corresponde a la aplicación de reglas lógicas para derivar nuevas proposiciones a partir de proposiciones previas.

Si comparamos los paradigmas que ya conocemos, Funcional v/s Lógico, tenemos que el paradigma funcional se basa en la evaluación de expresiones matemáticas y la teoría de funciones puras y se debe lograr la inmutabilidad evitando así los cambios de estados, estos últimos, solo se consiguen generando nuevas estructuras de datos en lugar de modificar las existentes. Por otro lado, el paradigma de programación lógico que se basa en la lógica matemática y la teoría de conjuntos, el cual se compone de un conjunto de hechos y reglas lógicas los que se usan para responder consultas mediante la inferencia y lo que se conoce como proceso de unificación. A través de ello, se buscan todas las respuestas posibles mediante la lógica de inferencia utilizando una programación declarativa.

De lo anterior podemos decir que, la programación lógica se basa en la lógica y la inferencia, mientras que la programación funcional se basa en la evaluación de expresiones matemáticas y la composición de funciones.

# Descripción del problema

# Análisis del problema

.

# Diseño de la solución

# Consideraciones de implementación

# Instrucciones de uso

# 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 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |

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

# 

# Conclusiones

# 

# Referencias

* Swi-prolog, Reference manual. https://www.swi-prolog.org/pldoc/doc\_for?object=manual
* Campus Virtual, (2023). Paradigmas de Programación. https://uvirtual.usach.cl/moodle/