

Paradigmas de la Programación

Informe Laboratorio nro. 2

Paradigma Lógico

*Profesor: Gonzalo Martínez.*

*Alumna: Karina Bustamante H.*

4 de junio de 2023

Tabla de contenido

[Introducción 2](#_Toc134312314)

[Descripción del problema 3](#_Toc134312315)

[Descripción del paradigma 3](#_Toc134312316)

[Análisis del problema 4](#_Toc134312317)

[Diseño de la solución 4](#_Toc134312318)

[Consideraciones de implementación 4](#_Toc134312319)

[Instrucciones de uso 5](#_Toc134312320)

[Resultados y evaluación 5](#_Toc134312321)

[Conclusiones 6](#_Toc134312322)

[Referencias 6](#_Toc134312323)

# Introducción

Para el desarrollo del laboratorio Nº2, de la asignatura de Paradigmas de la Programación, se nos solicita realizar una implementación bajo el paradigma lógico, el cual se basa en la lógica matemática y en la resolución de problemas mediante la deducción lógica. Los algoritmos a implementar deberán resolver el problema que corresponde al siguiente enunciado:

***“Crear un simulador de sistema operativo centrado específicamente en un sistema de archivos simplificado y los comandos que*** ***permiten operar sobre éste.”***

Para el desarrollo de esta implementación usaremos el lenguaje de programación, Prolog, con su editor virtual, (https://swish.swi-prolog.org/) y a través de la creación de una base de conocimientos que crearemos con predicados, reglas y clausulas, se buscará dar solución a los requerimientos funcionales solicitados, de manera tal, que la solución propuesta permita al usuario aplicar sobre este sistema de archivos ficticio, acciones como crear, buscar, listar, etc.

# 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

Un sistema de archivos es una parte fundamental de un sistema operativo, corresponde a un conjunto de procesos, métodos y reglas que utiliza un sistema operativo para administrar el almacenamiento de datos en la memoria de una computadora. Este sistema permite la organización, seguridad y localización de archivos, además de otras operaciones.

A lo largo de este informe se desarrollaran los requerimientos funcionales que permitan simular un sistema de archivos, donde un usuario pueda realizar acciones tales como crear y agregar contenido, así como también crear, modificar y eliminar archivos o carpetas, según permisos asignados, generando rutas de acceso a los archivos y carpetas, cambiarse de ruta, crear usuarios y drivers, cambiarse de unidad, iniciar y cerrar sesión.

# Análisis del problema

# Diseño de la solución

El diseño de la solución está basado en el lenguaje de programación Prolog, que debe simular los comandos básicos de operación de un sistema de archivos a través de consola de comandos.

# 

# Consideraciones de implementación

# Para la implementación de este laboratorio se recomienda la utilización del entorno virtual de prolog <https://swish.swi-prolog.org/> para realizar las pruebas de funcionamiento del código entregado.

# Instrucciones de uso

Para la ejecución de los algoritmos propuestos, debe abrir el archivo lab2\_13452929\_Bustamante.pl, seleccionar todo el contenido, copiar y pegar en el editor virtual de prolog, <https://swish.swi-prolog.org/>

Para hacer las pruebas de funcionamiento debe ingresar en el apartado para las consultas del intérprete de prolog, lo que se encuentra en el contenido del archivo, pruebas\_13452929\_Bustamante.pl

# Resultados y evaluación

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RF Nº | Requerimiento funcional | Grado de alcance | Tipo o cantidad de pruebas | Cantidad de éxitos | Cantidad de fracasos | Razones de fallo |
| 1 | TDAs |  |  |  |  |  |
| 2 | system - constructor | 1 |  |  |  |  |
| 3 | system - addDrive | 1 |  |  |  |  |
| 4 | system - register | 1 |  |  |  |  |
| 5 | system - login | 1 |  |  |  |  |
| 6 | system - logout | 1 |  |  |  |  |
| 7 | system - switch-drive | 1 |  |  |  |  |
| 8 | system - mkdir | 1 |  |  |  |  |
| 9 | system- cd |  |  |  |  |  |
| 10 | system - add-file | 0,5 |  |  |  |  |
| 11 | system - del |  |  |  |  |  |
| 12 | system - copy |  |  |  |  |  |
| 13 | system - move |  |  |  |  |  |
| 14 | system - ren |  |  |  |  |  |

**Tabla 1.-** Muestra de

# Conclusiones

Para concluir este informe puedo decir que el objetivo del laboratorio 2 de la asignatura de paradigmas de la programación, correspondiente a la aplicación del paradigma lógico esta parcialmente resuelto, ya que la implementación se desarrolló en base a lo solicitado y los resultados obtenidos fueron los esperados.

# Referencias

Moodle del curso de paradigmas:

https://uvirtual.usach.cl/moodle/course/view.php?id=10036&section=17

Presentaciones del curso de paradigmas:

Ayuda de prolog:

<https://www.swi-prolog.org/pldoc/doc_for?object=atomic_list_concat/3>

Manual de prolog

<https://swish.swi-prolog.org/p/Tutorial%20de%20prolog.swinb>

Uso de módulos en prolog

https://www.swi-prolog.org/pldoc/man?section=modules