**INFORME DE LABORATORIO 2: SIMULACIÓN DE UN PROGRAMA PARA REALIZAR UN TRATAMIENTO DE IMÁGENES SIMPLIFICADO**



Nombre: Aracely Castro V.

Profesor: Roberto Gonzales I.

Asignatura: Paradigmas de Programación (2/2022)

Índice

[1. INTRODUCCIÓN 3](#_Toc115040750)

[**1.1** **DESCRIPCIÓN DEL PROBLEMA** 3](#_Toc115040751)

[**1.2** **DESCRIPCIÓN DEL PARADIGMA** 3](#_Toc115040752)

[2. DESARROLLO 4](#_Toc115040753)

[**2.1** **ANALISIS DEL PROBLEMA** 4](#_Toc115040754)

[**2.2** **DISEÑO DE LA SOLUCIÓN** 5](#_Toc115040755)

[**2.3** **ASPECTOS DE IMPLEMENTACIÓN** 6](#_Toc115040756)

[**2.3.1 EJEMPLOS DE USO** 6](#_Toc115040757)

[**2.3.2 RESULTADOS ESPERADOS** 6](#_Toc115040758)

[**2.3.3 POSIBLES ERRORES** 6](#_Toc115040759)

[**2.4** **Resultados y autoevaluación** 6](#_Toc115040760)

[**2.4.1** **RESULTADOS** 6](#_Toc115040761)

[**2.4.2 AUTOEVALUACIÓN** 6](#_Toc115040762)

[3. CONCLUSIÓN 7](#_Toc115040763)

[4. BIBLIOGRAFÍA Y REFERENCIAS 7](#_Toc115040764)

[5. ANEXOS 7](#_Toc115040765)

# **INTRODUCCIÓN**

Con el objetivo de aplicar conceptos del paradigma de programación lógico usando el lenguaje de programación Prolog en la resolución de un tratamiento de imágenes simple, a continuación, se dará detalle sobre el proceso de desarrollo de la solución al problema a través del compilador SWI-Prolog versión 8.4.3. El informe constará de una breve introducción de cómo surgió el problema, una descripción del paradigma utilizado, el análisis del problema y como fue el diseño de las soluciones para algunas funciones, sus aspectos de implementación, las instrucciones necesarias para compilar el archivo con el script de pruebas junto con ejemplos de algunas funciones, los resultados y la autoevaluación. Finalmente, se dará una conclusión respecto a todo lo anterior.

## **DESCRIPCIÓN DEL PROBLEMA**

Al igual que en el laboratorio uno se busca desarrollar la simulación de un programa para el tratamiento de imágenes de forma simplificada como GIMP y Adobe Photoshop. Este simulador permitiría crear imágenes con distintos formatos de pixeles sobre los cuales se pueden aplicar distintas operaciones como, por ejemplo, recortar una imagen, invertir una imagen, obtener histograma de la imagen, entre otros. Para implementar esto, se deben tener en cuenta que:

Una imagen Pixmap-d es una imagen donde cada uno de sus pixeles contiene información de espacio de colores y profundidad (R)ed, (G)reen, (B)lue y (D)epth. Cada color cubre valores entre 0 y 255 formando en conjunto un color del espectro RGB. La profundidad en cambio ofrece información más detallada de la imagen como espacio tridimensional.

Una imagen Bitmaps-d es una imagen donde, en este caso, cada uno de sus pixeles tiene asociado solo un color, negro (1) o blanco (0) representando si hay tinta o no, cada píxel también tendría una profundidad asociada.

Una imagen Hexmap-d es similar a una imagen Pixmap-d con la diferencia que los valores RGB son representados en hexadecimal.

## **DESCRIPCIÓN DEL PARADIGMA**

El paradigma lógico forma parte de la familia de los paradigmas declarativos, su ventaja esta en facilitar pensar en la solución al problema en vez de los detalles procedurales que llevan a su solución. Prolog funciona declarando una base de conocimientos con hechos y reglas en donde se pueden hacer consultas sobre estas. El paradigma lógico cuenta con tres mecanismos básicos que son:

Unificación:

Backtracking automático:

Estructuras de datos basadas en árboles:

Además de contar con otros conceptos como son:

Átomo:

Predicado:

Clausuras:

# **DESARROLLO**

## **ANALISIS DEL PROBLEMA**

## **DISEÑO DE LA SOLUCIÓN**

## **ASPECTOS DE IMPLEMENTACIÓN**

### **2.3.1 EJEMPLOS DE USO**

### **2.3.2 RESULTADOS ESPERADOS**

Se espera que cada función no cause errores y compile correctamente.

### **2.3.3 POSIBLES ERRORES**

## **Resultados y autoevaluación**

### **RESULTADOS**

### **2.4.2 AUTOEVALUACIÓN**

La Autoevaluación se realiza de la siguiente forma: 0: No realizado – 0.25: Funciona 25% de las veces – 0.5: Funciona 50% de las veces 0.75: Funciona 75% de las veces – 1: Funciona 100% de las veces. Para ver la tabla de Autoevaluación, ver la Tabla XX del anexo XX

Tras mejorar la implementación de los requerimientos funcionales no se encontró errores, por lo que se considera de que funcionan el 100% de las veces

# **CONCLUSIÓN**

# **BIBLIOGRAFÍA Y REFERENCIAS**

# **ANEXOS**