**Instituto Tecnológico de Costa Rica**

**Área de Ingeniería en Computadores**

CE-3104: Lenguajes, Compiladores e Intérpretes

**Profesor:**

Marco Rivera Meneses

**Tarea 1: WazeLog**

Paradigma lógico

**Integrantes:**

Carlos Adrian Araya Ramirez

Michael Shakime Richards Sparks

José Andrés Solano Mora

**Abril, 2021**

**Tabla de Contenidos**

[1. Breve descripción del proyecto 3](#_Toc67018133)

[1.1. Descripción detallada de los algoritmos de solución desarrollados 3](#_Toc67018134)

[1.2. Descripción de las funciones implementadas 6](#_Toc67018135)

[1.3. Descripción de la ejemplificación de las estructuras de datos desarrolladas 8](#_Toc67018136)

[1.4. Problemas sin solución 8](#_Toc67018137)

[1.5. Problemas encontrados 8](#_Toc67018138)

[1.5.1. Concatenar y comparar un entero con un string. 9](#_Toc67018144)

[1.5.2. Inconvenientes en la estructura de datos del juego. 9](#_Toc67018145)

[1.6. Plan de Actividades realizadas por estudiante 10](#_Toc67018146)

[1.7. Conclusiones 12](#_Toc67018147)

[1.8. Recomendaciones 12](#_Toc67018148)

[1.9. Bibliografía consultada en todo el proyecto 12](#_Toc67018149)

[2. Bitácora en digital, donde se describen las actividades realizadas. 13](#_Toc67018150)

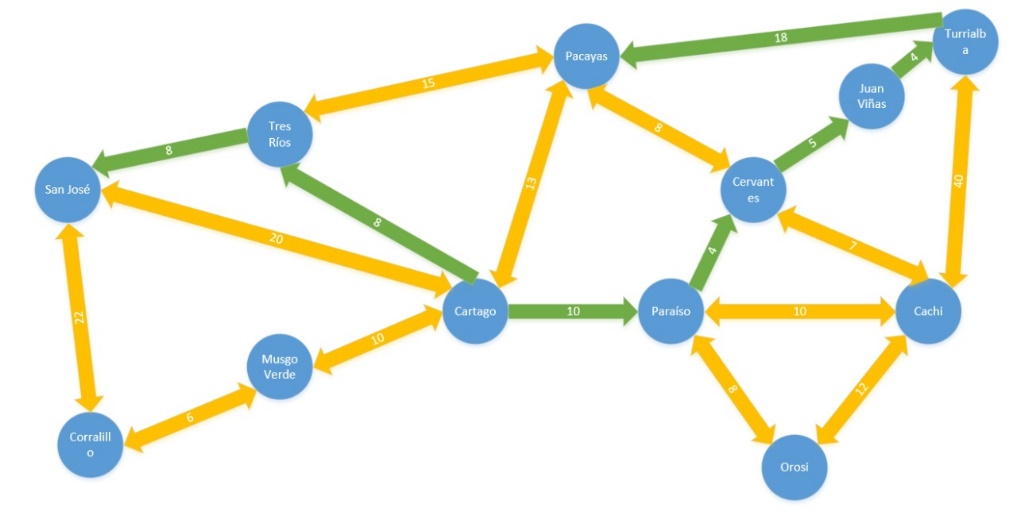
# Breve descripción del proyecto

WazeLog consiste en un sistema experto que ofrece el servicio de ubicación y guía de tránsito, este sistema funciona a partir grafos y gramáticas libres de contexto para brindar al usuario una interfaz amigable.

## Descripción de los hechos y reglas implementadas.

*unions(LugarA,LugarB,Peso).*

Para la estructuración del grafo se utilizó el hecho “unions” que definirá el nombre del nodo origen “Lugar A”, así como el nodo destino “Lugar B”, incluyendo a su vez un “Peso” específico de un nodo a otro, cabe recalcar que estas uniones son unidireccionales, o sea que si se desea especificar una unión que funcione en ambos sentidos se deberá crear una regla más con el “Lugar A” y “Lugar B” alternados.



## Descripción de las estructuras de datos desarrolladas.

* **Para el grafo:**

Reglas:

***findapath:***

Encuentra un camino entre dos nodos especificados.

Parámetros:

findapath(Inicio,Final,Peso,Camino,Lista)

***findminpath:***

Encuentra el camino más corto entre dos nodos especificados.

Parámetros:

findminpath(Inicio,Final,Peso,Camino)

***findminpath\_t:***

Encuentra el camino más corto entre dos nodos especificados y además, se añade el tiempo de viajes que es equivalente al doble del peso total.

Parámetros:

findminpath\_t(Inicio,Final,Peso,Tiempo,Camino)

findminpath\_t devuelve como “Peso” un número, como “Tiempo” el doble del “Peso” y como “Camino” devuelve una **lista** con la ruta a seguir. Ejemplo:

Peso = 41.

Tiempo = 82.

Camino = ['Turrialba', 'Pacayas', 'Cartago', 'Musgo Verde'].

* **Para el sistema experto:**

## Descripción detallada de los algoritmos desarrollados.

La regla findminpath(Inicio,Final,Peso,Camino) se comporta como una variación del algoritmo de **Dijkstra** utilizado para la búsqueda del camino más corto entre dos nodos dentro de un grafo, utilizando findapath(Inicio,Final,Peso,Camino,Lista) se realiza una búsqueda de todos los caminos posibles de un nodo a otro, luego estos son almacenados temporalmente en solution(Peso,Camino). A diferencia del algoritmo **Dijkstra**, findminpath evalúa cada una de las posibles rutas, sacrificando eficiencia, sin embargo, fue necesario realizarlo de esta manera para asegurarse que la ruta calculada sea en definitiva la más corta, pues findapath no muestra las rutas ordenadas y así como la ruta más corta podría ser la primera solución, también podría tratarse de la última.

## Problemas sin solución

De manera general, se logró implementar la totalidad del proyecto y fue posible solucionar la mayoría de los problemas que fueron detectados.



## Plan de Actividades realizadas por estudiante

|  |  |  |  |
| --- | --- | --- | --- |
| **Tarea** | **Tiempo estimado** | **Responsable** | **Fecha de entrega** |
| Iniciar el plan de trabajo y la documentación | 1h | Todos |  |
| Encargarse de las bitácoras | 3h | José |  |
| Investigar los grafos dirigidos y mixtos en PROLOG | 2h | José y Adrián |  |
| Implementar el grafo | 2 días | Todos |  |
| Investigar los sistemas expertos en PROLOG | 2h | Shakime y José |  |
| Investigar como obtener y analizar el input del usuario | 2h | Adrián |  |
| Implementar la interfaz de usuario con gramáticas libres de contexto | 2 días | Todos |  |
| Realizar todas las pruebas necesarias de la interfaz | 1 día | Adrián |  |
| Hacer el manual de usuario | 2h | Shakime |  |

## Problemas solucionados.

## Conclusiones

1. Se concluye que desarrollar una aplicación con el comportamiento experto utilizando Prolog, es una opción factible y favorable, debido a las propiedades y ventajas que brinda un paradigma lógico.
2. Se aplicaron los conceptos y habilidades aprendidas durante el estudio del paradigma de programación lógico, como la declaración de hechos y reglas para satisfacer el objetivo.
3. Se logró modelar el problema mediante la implementación de diversos hechos y reglas para manipular las listas con las que fue modelada la base de datos de los restaurantes. Entre ellas, la función miembro, la obtención de las cabezas de las listas, entre otras.
4. Se logró implementar un sistema experto que se encarga de recomendar al usuario restaurantes a partir de las preferencias brindadas, analizando su gramática y realizando deducciones a partir de la base de datos.

## Recomendaciones

1. Prolog brinda características favorables para el desarrollo de sistemas expertos utilizando gramáticas libres de contexto, por lo que se recomienda la implementación de aplicaciones que involucren este tipo de sistemas utilizando un paradigma lógico.
2. Para la utilización de la aplicación, se recomienda leer el manual de usuario antes de utilizar la aplicación, esto con el fin de comprender las gramáticas válidas y las distintas partes que componen el programa.
3. Recordar utilizar comillas a la hora de escribir el input al programa, ya que de no utilizarlas el programa no funciona correctamente.
4. Se recomienda el uso de oraciones claras, que contengan estructuras con verbos y predicados definidos, esto con el fin de facilitar la interacción con el Sistema Experto .

## Bibliografía consultada en todo el proyecto

Soler, F (2020). Programación de gramáticas clausales en Prolog. Recuperado el 4 de

Octubre de 2020, de https://personal.us.es/fsoler/papers/05capgram.pdf

Valverde, J. (2020). Gramáticas de cláusulas definidas - JARV's blog. Recuperado el 4 de

Octubre de 2020, de

http://jariaza.es/blog/post/prolog/gramaticas-de-clausulas-definidas.md

SWI-Prolog -- split\_string/4. (2014). Converting a user input string to list. Recuperado el 5

de Octubre de 2020, de

https://www.swi-prolog.org/pldoc/man?predicate=split\_string/4

SWI-Prolog -- atom\_string/2. (2020). Recuperado el 13 de Octubre de 2020, de

https://www.swi-prolog.org/pldoc/man?predicate=atom\_string/2

# Bitácora en digital, donde se describen las actividades realizadas.

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Estudiante(s)** | **Actividad** |
|  | Todos | Primera reunión para comprender, compartir ideas y  organización: División del trabajo correspondiente a  cada uno de los integrantes de acuerdo con la  dificultad estimada de este. |
|  | José y Adrián | Investigación e inicio de implementación inicial de los hechos y reglas para el grafo. |
|  | José y Shakime | Se termina la implementación del grafo. |
|  | Shakime y Adrián | Definición de hechos y reglas para el BNF. Para iniciar este desarrollo, fue necesario consultar el punto uno y dos de la bibliografía, para reforzar los conceptos y comprender el funcionamiento de las gramáticas libres de contexto. |
|  | Shakime y José | Implementación del análisis del input del usuario.  Actualización del BNF. Para este análisis se  necesitó investigar en la página web del punto tres  de la bibliografía. |
|  | Todos | Reunión de organización y actualización del trabajo realizado hasta el momento. |
|  | Todos | Reunión para terminar de definir la búsqueda, reglas y hechos para el comportamiento experto de la aplicación. |
|  | Shakime y Adrián | Implementación del análisis gramático y obtención de palabras claves para oraciones relacionadas con ubicaciones. |