**Universidad Nacional de Costa Rica.**

**Escuela de Informática, Lagunilla de Heredia.**

**Curso: EIF400 Paradigmas de Programación.**

**Tema: Proyecto de programación #3.**

**Profesor: Georges Alfaro Salazar.**

**Integrantes:**

**Esteban Montero Fonseca, 304830405.**

**Jefferson Moreno Zúñiga, 116270399.**

**Daniel Zamora García, 402310604.**

**II CICLO 2018.**



**Introducción.**

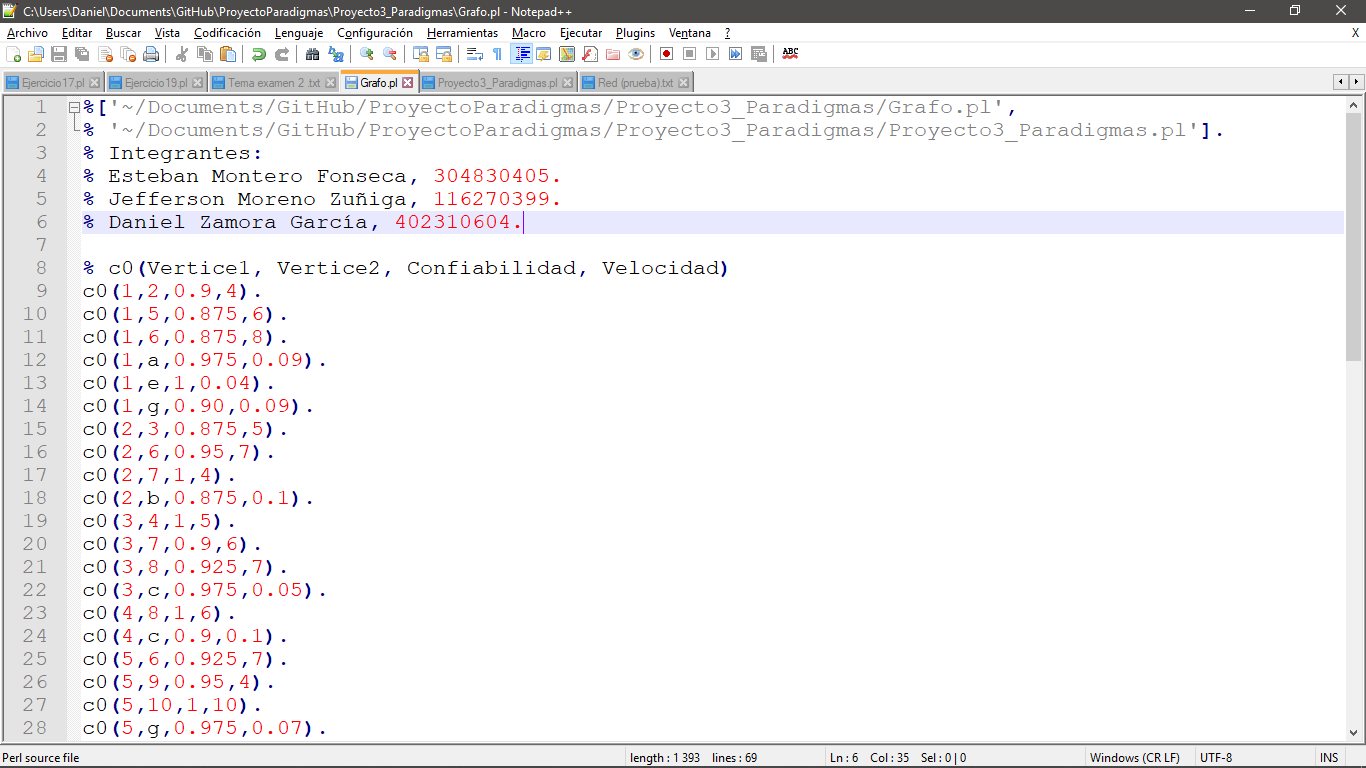
En este documento se describe en qué consiste el proyecto # 3 del curso de Paradigmas de programación. El proyecto se realiza en SWI Prolog, la idea es poner en práctica los conocimientos de programación lógica; adquiridos en clase. El proyecto consiste en la simulación de una red que se representa a modo de grafo no dirigido. Cada nodo representa un equipo conectado a la red y los arcos describen algunas características (velocidad, confiabilidad) de las conexiones correspondientes.

**Objetivo Principal.**

Escribir los predicados necesarios para representar un grafo no dirigido de cualquier orden, además de las funciones solicitadas.

**Representación del grafo.**

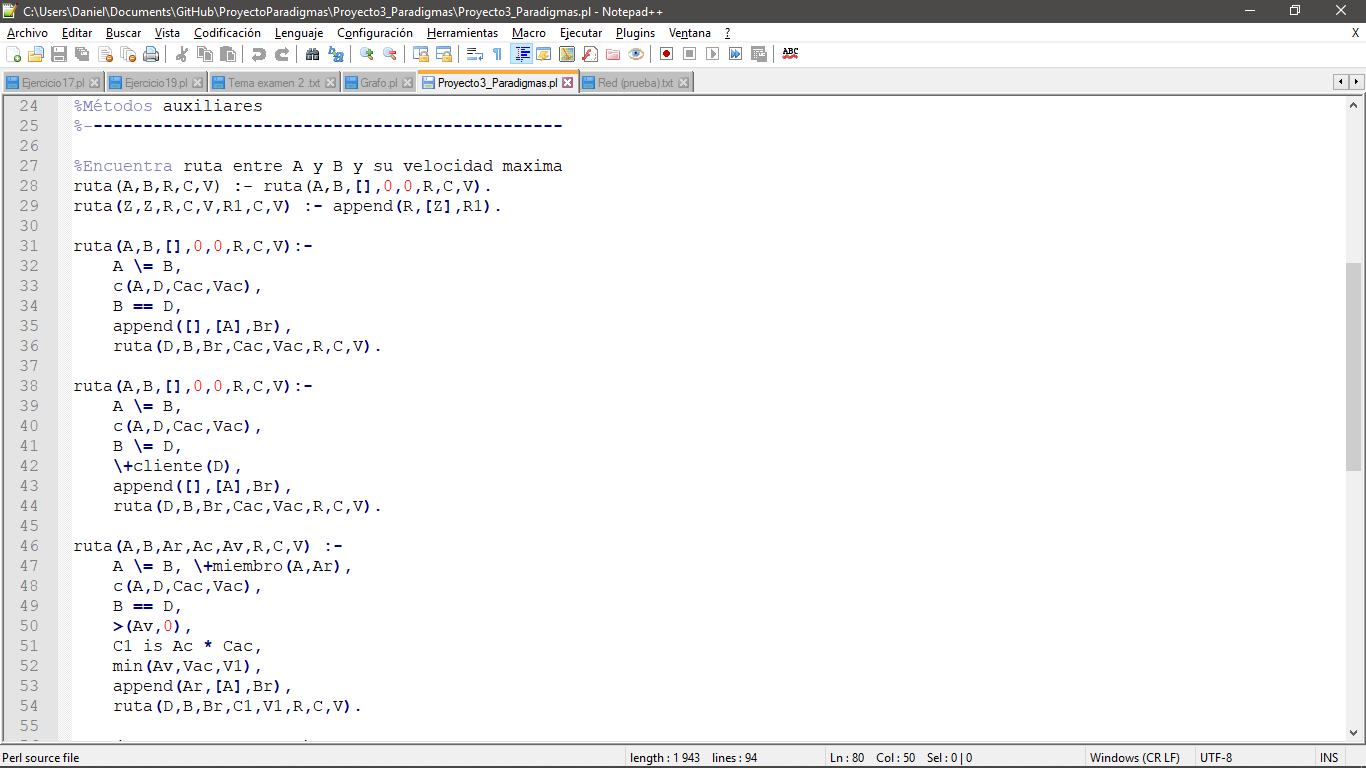
El grafo se construye en un .pl aparte el cual es cargado junto con los predicados que contienen las funciones principales. La forma en que se construye es declarando un conjunto, el cual está formado por el vértice de salida, el vértice de llegada, la confiabilidad y por último la velocidad. Se parte del ejemplo solicitado en el enunciado del proyecto.

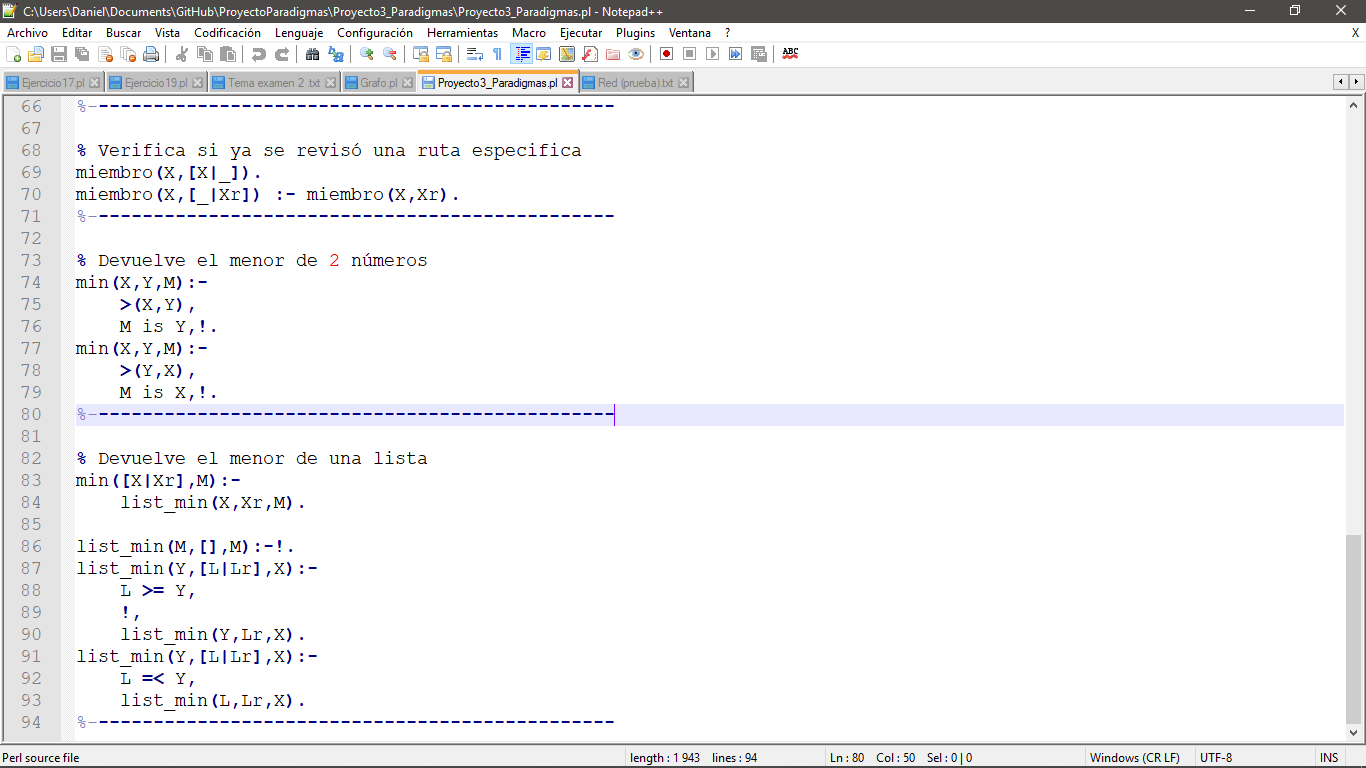


**Predicados auxiliares.**

Se construyen predicados auxiliares que son necesarios para realizar los principales, algunas de ellos tomadas de los ejemplos brindados por el profesor, en los materiales de apoyo, los demás son elaboración propia. De los predicados auxiliares sobresalen las siguientes:

1. **ruta**: existen varios casos, en algunos verifica si existe una ruta entre dos vértices dados, en otro caso retorna la ruta que hay entre dos vértices sin pasar por nodos clientes que es la restricción dada en el enunciado y además muestra la velocidad máxima. Este predicado es utilizado por el predicado principal llamado conexion.
2. **miembro**: verifica si ya se revisó una ruta especifica.
3. **min**: devuelve el menos de dos números.
4. **list\_min:** el cual se utiliza en min también, retorna el valor menor de una lista dada.

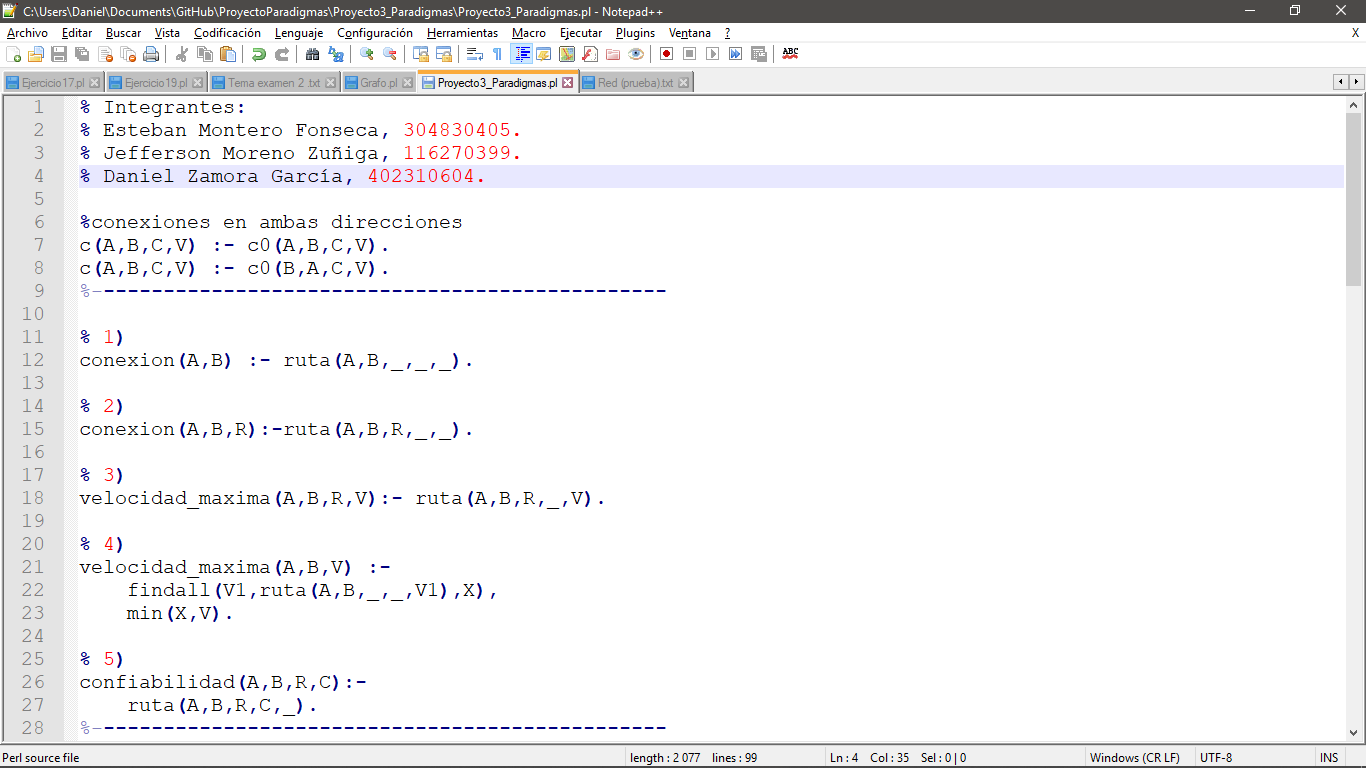




**Predicados principales.**

Los predicados principales, son los solicitados en el enunciado el proyecto, los cuales son:

1. **conexion (A, B):** consiste en verificar si dos nodos cualesquiera se encuentran conectados o no.
2. **conexion (A, B, R):** como en el predicado anterior verifica si existe una ruta entre los nodos dados y además la muestra, mostrando a la vez su velocidad máxima.
3. **velocidad\_maxima (A, B, R, V):** verifica si V es la mayor velocidad de transmisión posible de los nodos A y B a través de la ruta R.
4. **velocidad\_maxima (A, B, R, V):** de todas las rutas posibles entre los nodos dados verifica que V sea la velocidad máxima.
5. **Confiabilidad (A, B, R, C):** verifica que C sea la confiabilidad que existe entre los nodos A y B a través de la ruta R.



**Conclusiones.**

Con la elaboración del proyecto #3 de paradigmas de programación se logra fortalecer los conceptos y conocimientos adquiridos durante las lecciones en cuanto a programación lógica. Se logra mejorar aspectos como manejo y adaptación de SWI Prolog. Se adquiere una mayor fluidez a la hora de pensar y generar los predicados ya que se maneja mejor la sintaxis y la lógica de los ejercicios.

La programación lógica nos permite encontrar soluciones a problemas de forma automática, por medio de la declaración de este problema y una afirmación o condición. Esta declaración se hace por medio de recursividad como estructura básica de control. Prolog nos permite realizar backtracking es decir las instrucciones no se ejecutan de forma secuencial. Además permite unificación, el programa se ejecuta a partir de un punto de elección y continua ejecutando hasta determinar si el resultado es verdadero o falso.