

# Encontrar la mejor ruta\*

Faustino Aguilar  
2-732-727

Kevin Vásquez  
9-736-2350

Profesor  
José Peralta

13 de julio de 2016

## Resumen

En este trabajo hemos realizado un programa que permita saber al usuario, cual es la mejor ruta que puede tomar en un momento dado, dependiendo de la parada en que él esta y la parada donde quiere ir, de modo se calcula todas las formas de llegar de la parada A a la parada B, y luego se considera cual es la que tiene menor valor posible.

**Palabras clave:** *rutas, prolog, transporte*

## Índice

<b>1. Descripción del Problema</b>	<b>1</b>
<b>2. Recolección de Datos</b>	<b>1</b>
<b>3. Rutas Utilizadas</b>	<b>1</b>
3.1. Ruta de Canto del LLano - Mercado .	2
3.2. Ruta de Forestal - Mercado . . . . .	2
<b>4. Ejemplo y Código Fuente</b>	<b>3</b>
<b>Referencias</b>	<b>4</b>
<b>5. Diagrama</b>	<b>5</b>

## 1. Descripción del Problema

El problema consistirá en averiguar la ruta entre dos paradas de autobuses de las rutas urbanas de transporte de Santiago de Veraguas según alguno de los siguientes criterios:

1. Mínimo tiempo de viaje
2. Mínimo número de transbordos

Además debemos dar información sobre las paradas(ruta a la que pertenece, ubicación, Servicios de Interés) incluidas en la ruta.

## 2. Recolección de Datos

Los datos para realizar el programa fueron tomados de la medición de datos como el tiempo y la ubicación donde utilizamos herramientas como GPS y Google Maps.

Parada	Ruta	Información
pa1	Canto del Llano-Mercado	Centro Salud
pa2	Canto del Llano-Mercado	CRUV Puente
pa3	Canto del Llano-Mercado	Universidad Oteima
pc1	Canto del Llano-Mercado	IPT Veraguas
pc2	Canto del Llano-Mercado	Terminal
pc3	Canto del Llano-Mercado	Policlínica
pc4	Canto del Llano-Mercado	Escuela Normal
pa4	Canto del Llano-Mercado	Mercado
pa5	Canto del Llano-Mercado	Escuela Normal
pb1	Forestal-Mercado	INADEH
pb2	Forestal-Mercado	Calle 18 Norte
pb3	Forestal-Mercado	CRUV Forestal
pb4	Forestal-Mercado	Escuela San Martín
pb5	Forestal-Mercado	Cancha San Martín
pb6	Forestal-Mercado	Parada Delicias
pc1	Forestal-Mercado	IPT Veraguas
pc2	Forestal-Mercado	Terminal
pc3	Forestal-Mercado	Policlínica
pc4	Forestal-Mercado	Escuela Normal
pb7	Forestal-Mercado	Mercado
pb8	Forestal-Mercado	Escuela Normal

Cuadro 1: Tabla de paradas recolectadas

## 3. Rutas Utilizadas

Nosotros en este Proyecto hemos utilizado 2 rutas:

1. Ruta de Canto del LLano - Mercado
2. Ruta de Forestal - Mercado

\*Proyecto Final del Curso de Programación 4 - CRUVFIEC

### 3.1. Ruta de Canto del LLano - Mercado

Esta ruta no fue tomada de su inicio a fin, pero se tomo la sección que consideramos más importante, luego determinamos que un buen comienzo de la misma era el Centro de Salud de Canto del LLano, entonces este fue nuestro punto de partida para la primera parada.

### 3.2. Ruta de Forestal - Mercado

Esta ruta tiene dos tipos, la que pasa por el Machetazo y la que solo va al mercado y regresa. Nosotros tomamos esta última, trazando un inicio en el INADEH. Cada ruta tiene sus respectivas interconexiones y rutas comunes.

En la siguiente sección podemos ver la data acerca de las paradas y los tiempos que hay entre cada una.

Listing 1: Datos de conexiones entre paradas

```
% Ruta A
conexion(pa1, pa2, 3).
conexion(pa2, pa3, 0.5).
conexion(pa3, pc1, 0.67).
conexion(pc1, pc2, 3).
conexion(pc2, pc3, 4).
conexion(pc3, pc4, 0.8).
conexion(pc4, pa4, 1.5).
conexion(pa4, pa5, 2).
conexion(pa5, pc2, 2).

% Ruta B
conexion(pb1, pb2, 0.5).
conexion(pb2, pb3, 1).
conexion(pb3, pb4, 2).
conexion(pb4, pb5, 0.5).
conexion(pb5, pb6, 0.67).
conexion(pb6, pc1, 0.5).
conexion(pc1, pc2, 3).
conexion(pc2, pc3, 4).
conexion(pc3, pc4, 0.8).
conexion(pc4, pb7, 1).
conexion(pb7, pb8, 2).
conexion(pb8, pc4, 0.5).

% Interconexiones
conexion(pa1, pi1, 3).
conexion(pb2, pi1, 1).
conexion(pi1, pb4, 2).
conexion(pc4, pi2, 1.25).
conexion(pi3, pi2, 2).
conexion(pi3, pc4, 1).
conexion(pi3, pc2, 2.5).
```



Figura 1: Las rutas utilizadas y sus datos.

## 4. Ejemplo y Código Fuente

El programa esta adjunto a este documento, su código está documentado, Un ejemplo a probar en el mismo sería:

```
?- mejor_ruta(pa1, pb3, R, T).
```

Parada	Ruta	Información
pa1	Canto-Mercado	Centro Salud
pa2	Canto-Mercado	CRUV Puente
pa3	Canto-Mercado	Universidad Oteima
pc1	Canto-Mercado	IPT Veraguas
pb6	Forestal-Mercado	Parada Delicias
pb5	Forestal-Mercado	Cancha San Martin
pb4	Forestal-Mercado	Escuela San Martin
pb3	Forestal-Mercado	CRUV Forestal

```
(pa1, pa2) = 3 s
(pa2, pa3) = 0.5 s
(pa3, pc1) = 0.67 s
(pc1, pb6) = 0.5 s
(pb6, pb5) = 0.67 s
(pb5, pb4) = 0.5 s
(pb4, pb3) = 2 s
(pxx, pyy) = 7.84s <- Total
```

Parada	Ruta	Información
pa1	Canto-Mercado	Centro Salud
pa2	Canto-Mercado	CRUV Puente
pa3	Canto-Mercado	Universidad Oteima
pc1	Canto-Mercado	IPT Veraguas
pb6	Forestal-Mercado	Parada Delicias
pb5	Forestal-Mercado	Cancha San Martin
pb4	Forestal-Mercado	Escuela San Martin
pi1	Interconexión A-B	CRUV
pb2	Forestal-Mercado	Calle 18 Norte
pb3	Forestal-Mercado	CRUV Forestal

```
(pa1, pa2) = 3 s
(pa2, pa3) = 0.5 s
(pa3, pc1) = 0.67 s
(pc1, pb6) = 0.5 s
(pb6, pb5) = 0.67 s
(pb5, pb4) = 0.5 s
(pb4, pi1) = 2 s
(pi1, pb2) = 1 s
(pb2, pb3) = 1 s
(pxx, pyy) = 9.84s <- Total
```

Parada	Ruta	Información
pa1	Canto-Mercado	Centro Salud
pi1	Interconexión A-B	CRUV
pb2	Forestal-Mercado	Calle 18 Norte
pb3	Forestal-Mercado	CRUV Forestal

```
(pa1, pi1) = 3 s
(pi1, pb2) = 1 s
(pb2, pb3) = 1 s
(pxx, pyy) = 5s <- Total
```

Parada	Ruta	Información
pa1	Canto-Mercado	Centro Salud
pi1	Interconexión A-B	CRUV
pb4	Forestal-Mercado	Escuela San Martin
pb3	Forestal-Mercado	CRUV Forestal

```
(pa1, pi1) = 3 s
(pi1, pb4) = 2 s
(pb4, pb3) = 2 s
(pxx, pyy) = 7s <- Total
```

===== Mejor Ruta =====

Parada	Ruta	Información
pa1	Canto-Mercado	Centro Salud
pi1	Interconexión A-B	CRUV
pb2	Forestal-Mercado	Calle 18 Norte
pb3	Forestal-Mercado	CRUV Forestal

Tiempos:

```
(pa1, pi1) = 3 s
(pi1, pb2) = 1 s
(pb2, pb3) = 1 s
R = [pa1, pi1, pb2, pb3],
T = 5
```

Esto nos devuelve todas las rutas y cual es la mejor ruta.

Para comprender mejor el algoritmo que vamos a desarrollar tenemos que entender como encontrar las posibles rutas desde el punto *A* al punto *B*. En el siguiente código podemos ver n ejemplo:

Listing 2: Ejemplo básico de algoritmo de rutas

```
m(a,b).
m(b,c).
m(c,d).
m(b,d).

con(X,Y) :-
    m(Y, X).
con(X,Y) :-
    m(X, Y).

rutas(X, Y, L) :-
    mov(X, Y, [X], M),
    reverse(M, L).

mov(X, Y, W, [Y|W]) :-
    con(X, Y).
mov(X, Y, V, M) :-
    con(X, Z),
    Z \== Y,
    not(member(Z, V)),
    mov(Z, Y, [Z|V], M).
```

El siguiente código fue el utilizado en el programa donde se obtiene todas las rutas posibles y además se va acumulando el tiempo de cada parada.

Listing 3: Código utilizado en el programa

```

opciones(A, B, P, [B|P], T) :-
    conectados(A, B, T).
opciones(A, B, Marcados, Ruta, T) :-
    conectados(A, C, D),
    C \== B,
    not(member(C, Marcados)),
    opciones(C, B, [C|Marcados], Ruta, T1),
    T is D + T1.

```

Luego utilizamos la siguiente serie de reglas para calcular el minimo tiempo dada las rutas y sus respectivos tiempos.

Listing 4: Calculo del minimo

```

minimo([F|R], M) :-
    min(R, F, M).
% minimo recorrido
min([], M, M).
min([P,L|R], [_,M], Min) :-
    L < M,
    !,
    min(R, [P,L], Min).
min([_|R], M, Min) :-
    min(R, M, Min).

```

Como podemos ver los codigos anteriores, estos fueron programados para buscar la relación entre las paradas de manera recursiva, ya que bien podríamos definir el tiempo que hay entre cada parada (cada iteración); pero consideramos que sería ineficiente para desarrollar rutas más complejas y se podría llevar a errores al actualizar los datos. Por esto hemos experimentado con nuestro código en Prolog.

También consideramos que esto nos permite tomar en cuenta las interconexiones ya que bien podemos estar el El Centro de Salud de Canto de LLano y queremos llegar a la Calle 18 Norte cerca del INADEH, esto podríamos acerlo más rápido gracias a la interconexión *pi1*, ya que la parada común *pc1* estaría más lejos.

Así como esto podemos analizar el mapa al final de este documento y establecer donde estamos y hacia donde queremos ir.

Como conclusión podemos recalcar que este proyecto puede ser un antecesor a alguna APP movil que permita a los usuarios de la ciudad Santiago encontrar la mejor ruta. Se nos ocurre utilizar el poder de Prolog para obtener resultados que luego serán enviados a un servidor, que sea la fuente de información para actualizar datos de paradas y sus tiempos, siendo más útil para el usuario.

## Referencias

- [Adamson and Tick, 1991] Adamson, P. and Tick, E. (1991). Greedy partitioned algorithms for the shortest-path problem. *International Journal of Parallel Programming*, 20(4):271–298.
- [Korf, 1993] Korf, R. E. (1993). Linear-space best-first search. *Artificial Intelligence*, 62(1):41–78.
- [López Villalba, 2011] López Villalba, N. V. (2011). Modelo para gestión de entregas por rutas óptimas, mediante gps y conexión celular.
- [Shoham, 2014] Shoham, Y. (2014). *Artificial intelligence techniques in Prolog*. Morgan Kaufmann.

5. Diagrama

En este documento podemos ver un mapa donde se ven dibujadas las paradas y sus rutas, así como las paradas cercanas y las comunas, también se detalla el tiempo entre cada parada y el nombre de las mismas.

RUTAS INTERNAS DE LA CIUDAD DE SANTIAGO

