



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**PROCESAMIENTO GEODÉSICO PARALELO UTILIZANDO UN SISTEMA  
EMPOTRADO**

PROYECTO FINAL  
QUE PARA OPTAR POR EL GRADO DE  
**ESPECIALISTA EN CÓMPUTO DE ALTO RENDIMIENTO**

PRESENTA:  
**ARMANDO RODRIGUEZ ARGUIJO**

DIRECTOR DE TESIS  
**DR. HÉCTOR BENÍTEZ PÉREZ**  
INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS  
Y EN SISTEMAS, UNAM

CIUDAD UNIVERSITARIA, CIUDAD DE MÉXICO. SEPTIEMBRE, 2019.



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## **Agradecimientos**

A la Universidad Nacional Autónoma de México por darme la oportunidad de seguir preparándome.

A mi asesor el Dr. Héctor Benítez Pérez, por sus enseñanzas, su paciencia y gran apoyo para realizar este trabajo.

A los miembros de mi jurado el Dr. Fabián García Nocetti y el Dr. Ernesto Rubio Acosta por su gran apoyo.

Al Ing. Adrián Durán Chavesti por su apoyo y consejos.

Al Posgrado en Ciencia e Ingeniería de la Computación, profesores, administrativos y compañeros de la especialidad.

A mi esposa Xiomara y a mi hijo Armando por su paciencia y apoyo para cursar y concluir la especialidad.

A mis padres Clara y Mario por sus consejos y apoyo durante toda mi educación.

A mis hermanos, Mario, Gaby y Vero.

# Índice

1. Introducción.....	5
1.1 Objetivos.....	6
1.2 Meta.....	6
1.3 Hipótesis.....	6
1.4 Caso de estudio.....	6
2. Problema Directo.....	10
2.1.1 Diagrama de dependencia entre variables.....	12
3. Problema inverso.....	13
3.1 Diagrama de dependencia entre variables.....	14
4.- Pruebas de la solución al problema directo e inverso.....	15
4.1 Resultados problema directo.....	15
4.2 Resultados problema inverso.....	16
5.- Solución propuesta.....	18
5.1 Programa de recepción.....	18
5.2 Programa de envío.....	19
5.3 Programa de lectura del GPS.....	19
6.- Resultados.....	21
7.- Conclusiones.....	26
8.- Bibliografía.....	28
Anexo 1 Instalación del lenguaje X10 en Raspbian.....	29
Anexo 2 Código Fuente directo.x10.....	35
Anexo 3 Código fuente inverso.x10.....	47
Anexo 4 Código fuente azimuth.x10.....	54
Anexo 5 Código fuente de la solución propuesta.....	56

## Índice de ilustraciones

Ilustración 1: Caso de estudio.....	6
Ilustración 2: Conexión Raspberry Pi GY-NEO6MV2.....	8
Ilustración 3: Conexión Raspberry Pi GY-NEO6MV2.....	8
Ilustración 4: Conexión Raspberry Pi GY-NEO6MV2.....	8
Ilustración 5: Dependencia entre variables problema directo.....	12
Ilustración 6: Dependencia entre variables problema indirecto.....	14
Ilustración 7: Comunicación entre dispositivos.....	18
Ilustración 8: Petición al programa.....	19
Ilustración 9: Resultado Lugares:1 Threads:1.....	21
Ilustración 10: Resultado Lugares:4 Threads:1.....	21
Ilustración 11: Resultado Lugares:1 Threads:4.....	22
Ilustración 12: Resultado Lugares:4 Threads:4.....	22
Ilustración 13: Resultado RaspberryPi1 1 Lugar y 1 Thread vs 1 Lugar y 4 Threads .....	23
Ilustración 14: Resultado RaspberryPi2 1 Lugar y 1 Thread vs 1 Lugar y 4 Threads .....	24
Ilustración 15: Resultado RaspberryPi3 1 Lugar y 1 Thread vs 1 Lugar y 4 Threads .....	24

# 1. Introducción

El camino más corto entre 2 puntos, tomando la tierra como un elipsoide de revolución, se le puede llamar geodésico. Existen 2 problemas en la geodésica: el problema directo que consiste en encontrar el punto final dado el punto inicial, acimut y longitud ; y el problema inverso que consiste en encontrar la trayectoria más corta entre 2 puntos dados [1].

La resolución de dichos problemas se basan en los trabajos de Legendre (1806), Oriani (1806,1808,1810), Bessel (1825) y Helmert (1880). Basado en dichos trabajos Vicenty (1975) desarrolló algoritmos para la solución del problema directo e inverso para las primeras calculadoras programables. Dichos algoritmos son ampliamente utilizados hoy en día [1].

Charles F. F. Karney (2013) en su trabajo “Algorithms for geodesics” adapta los métodos de Helmert (1880) a las computadoras modernas. Con lo cual asegura 3 mejoras a los algoritmos de Vicenty [1]:

- 1) La precisión de los cálculos mejora con el uso de las nuevas computadoras
- 2) La solución al problema inverso converge para todos los puntos
- 3) Las propiedades diferenciales e integrales son calculadas.

Las propiedades diferenciales permiten determinar el comportamiento del geodésico, con lo cual se puede escalar, sin necesidad de volver a realizar los cálculos. Las propiedades integrales, proveen un método para encontrar el área de un polígono geodésico [1].

Por otro lado, el dispositivo Raspberry Pi es una computadora totalmente funcional en un tamaño pequeño y a bajo costo. Se puede utilizar para navegar por internet, jugar videojuegos, programar, desarrollar circuitos y dispositivos físicos, etc.. Existe un libro publicado por la “Raspberry Pi Press” titulado “The official Raspberry Pi Projects Book” el cual muestra diferentes proyectos que se han realizado con el dispositivo [3].

El dispositivo Raspberry Pi fue desarrollado para alentar un mayor uso de la computación en la educación alrededor del mundo. Sus creadores crearon la fundación sin fines de lucro Raspberry Pi. En el 2012 construyeron algunas miles de ellas y fueron rápidamente vendidas, desde ese año se han vendido millones [2].

El modelo Raspberry Pi 3, cuenta con un procesador ARM de 4 núcleos a 1.2Ghz, 1GB en RAM, Wi-Fi y Bluetooth integrados. Como sistema operativo se puede instalar GNU/Linux [2] y Windows 10 IoT [7]; por lo cual podemos ejecutar programas paralelos desarrollados en X10.

Por lo cual al implementar los algoritmos de Karney [1] con el uso del dispositivo Raspberry Pi y la programación paralela, se espera obtener una sistema distribuido con una mejora en el tiempo de ejecución de los cálculos del problema directo e inverso.

## 1.1 Objetivos

1.- Programar la solución al problema directo e inverso geodésico, y paralelizar la implementación en un dispositivo empotrado.

2.- Demostrar que se pueden usar técnicas paralelas para resolver problemas de una manera sencilla, utilizando el lenguaje de programación X10; y con el uso de sistemas empotrados reducir el costo de su implementación.

## 1.2 Meta

El fin principal del presente trabajo es programar la solución al problema directo e inverso geodésico, y paralelizar la implementación en un dispositivo empotrado. Al final se comparará el desempeño, esperando una reducción al tiempo de ejecución. También se demostrará que se pueden usar técnicas paralelas para resolver problemas de una manera sencilla, utilizando el lenguaje de programación X10; y con el uso de sistemas empotrados reducir el costo de su implementación, el costo eléctrico y por su tamaño se puede transportar fácilmente. Al utilizar el lenguaje X10, el código generado puede ser ejecutado en diferentes arquitecturas de cómputo, incluyendo equipo de Alto Rendimiento.

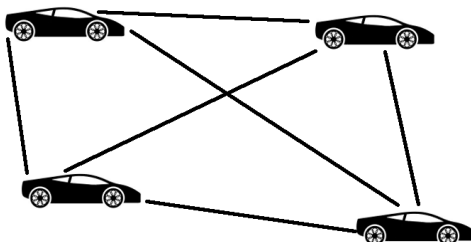
## 1.3 Hipótesis

El tiempo de ejecución del algoritmo para la solución al problema directo e inverso geodésico, se reducirá utilizando paralelismo mediante la implementación en el lenguaje X10.

Se esperan resultados exactos, aunque estos pueden variar de acuerdo a la precisión de la lectura de las coordenadas y a las fórmulas utilizadas.

## 1.4 Caso de estudio

Para la ejecución utilizaremos 3 dispositivos que se encuentran en movimiento, comparten sus coordenadas por WiFi y posteriormente cada dispositivo resuelve del problema directo e inverso. Un ejemplo del uso de dicho sistema distribuido es la coordinación entre automóviles autónomos, donde cada auto actúa como cliente y servidor.



*Ilustración 1: Caso de estudio*

Para realizar este trabajo, se utilizará como sistema empotrado el dispositivo raspberry pi 3 Modelo B. A continuación, se muestran sus especificaciones técnicas [9].

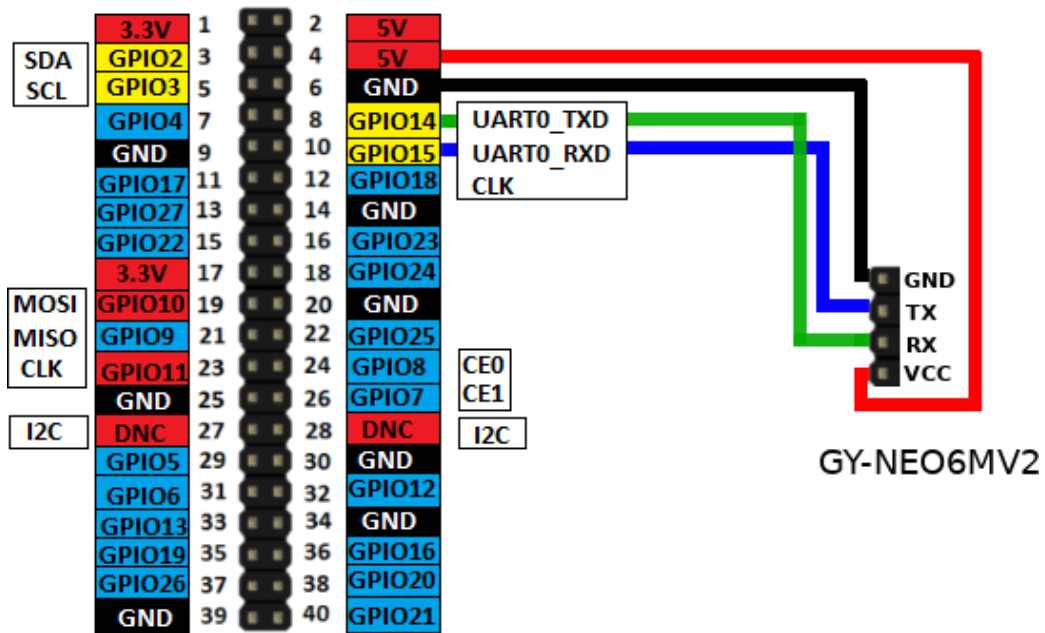
- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN
- Bluetooth Low Energy (BLE)
- 100 Base Ethernet
- Puerto 40-pin GPIO
- HDMI
- 4 puertos USB 2.0
- Puerto CSI para cámara
- Puerto DSI para pantalla Touch
- Salida de audio
- Puerto Micro SD
- Entrada 5V 2.5A DC

Se utilizará el módulo GPS GY-NEO6MV2 . A continuación, se muestran sus especificaciones técnicas [6].

- Voltaje de alimentación: 3.3 V a 5 V
- Corriente de funcionamiento: 45 mA
- Comunicación: UART
- Velocidad de comunicación: 9600 bps
- Memoria: EEPROM
- Interface: TTL serial
- Indicador de señal: LED
- Pines de módulo:
  - VCC
  - RX
  - TX
  - ND
- Sistema de coordenadas: WGS-84
- Sensibilidad de captura -148dBm
- Sensibilidad de rastreo: -161 dBm
- Máxima altura medible: 18000
- Máxima velocidad: 515 m/s
- Exactitud: 1 micro segundo
- Frecuencia receptora: L1 (1575.42 Mhz)
- Código C/A: 1.023 Mhz
- Tiempo de inicio por primera vez: 38 s en promedio
- Tiempo de inicio : 35 s en promedio
- Dimensiones antena: 25 mm X 25 mm
- Dimensiones módulo: 36 mm X 25.9 mm

Para la conexión entre el dispositivo raspberry pi y el módulo GPS se utilizará el siguiente diagrama de conexión.





### Raspberry Pi 3 Model B

Ilustración 2: Conexión Raspberry Pi GY-NEO6MV2

A continuación se muestran 2 fotografías de un dispositivo raspberry pi con un modulo GPS.



Ilustración 3: Conexión Raspberry Pi GY-NEO6MV2

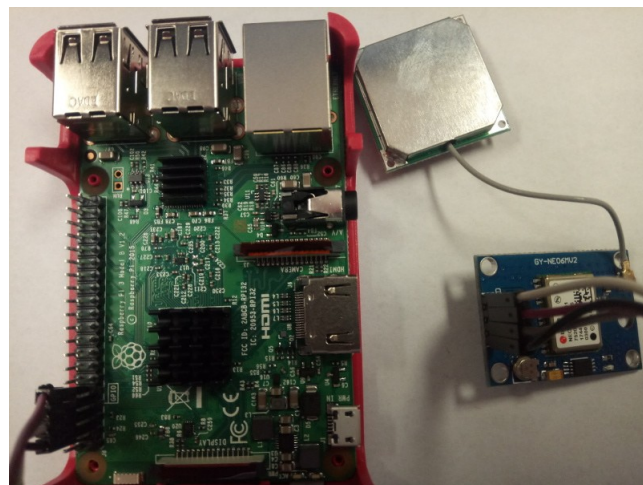


Ilustración 4: Conexión Raspberry Pi GY-NEO6MV2

Como lenguaje de programación se utilizará X10, al ser una variante de

Java con implementación del modelo APGAS [4] permite utilizar diferentes bibliotecas, con lo cual se reduce el tiempo de desarrollo y simplifica el código fuente.

En el lenguaje X10 ,en caso de requerir paralelizar alguna sección basta con utilizar la palabra reservada `async` [4].

```
async
{
    //bloque en ejecución paralela
}
```

Si se requiere poner una barrera, sería de la siguiente forma.

```
finish{
    async{
        //bloque en ejecución paralela
    }
}
```

En el lenguaje X10 se utilizan las variables globales `X10_NTHREADS` y `X10_NPLACES`. Las cuales indican el número de threads y el número de lugares de ejecución. En el caso de los “Lugares” estos pueden ser procesos en el mismo equipo, o pueden migrarse [4]. Para el presente trabajo, se utilizarán lugares estáticos. La ejecución paralela se realizará en el mismo dispositivo en sus 4 núcleos.

Para la conexión entre los dispositivos, se utilizará una conexión WiFi mediante el uso de un AP y el protocolo TCP/IP.

Como ejemplo de procesamiento paralelo, se realizará la solución al problema directo e inverso.

En el desarrollo del presente trabajo se utilizará como base los algoritmos presentados por Charles F. F. Karney en el trabajo titulado "Algorithms for Geodesics", publicado en "*Journal of Geodesy*, vol. 87, no. 1" en el año 2013 [1].

En la primera parte del trabajo se presentará problema directo e inverso. Posteriormente se presentará una propuesta donde cada dispositivo enviará sus coordenadas a todos los demás. Al momento de recibir las coordenadas de otro dispositivo, se calculará el problema directo e inverso. Es decir, cada dispositivo funcionará como maestro y esclavo a la vez.

## 2. Problema Directo

El problema directo consiste en dadas las coordenadas geodésicas del punto A  $\phi_1$ , el acimut al vértice B  $\alpha_1$  y la distancia entre A y B  $s_{12}$ , se pretende obtener la longitud  $\lambda_{12}$ , latitud del punto B  $\phi_2$  y acimut al vértice A  $\alpha_2$ .

A continuación se presentan las ecuaciones principales utilizadas, de acuerdo al trabajo presentado por Karney [1]. También se muestran las variables que se usan en el presente trabajo, donde x es el número de uno de los puntos geodésicos. Algunas ecuaciones se utilizan varias veces durante el cálculo del problema directo.

Ecuación: 1 Variables: f , a , b , e

$$f = (a - b) / a = 1 - \sqrt{1 - e^2}$$

Ecuación: 2 Variables: n , a , b , f

$$n = (a - b) / (a + b) = f / (2 - f)$$

Ecuación: 3 Variables: e , a , b , f

$$e^2 = (a^2 - b^2) / a^2 = f(2 - f)$$

Ecuación: 4 Variables: e , a , b

$$e'^2 = (a^2 - b^2) / b^2 = e^2 / (1 - e^2)$$

Ecuación: 5 Variables: alpha\_0 , alpha\_1 , alpha\_2 , beta\_1 , beta\_2

$$\sin \alpha_0 = \sin \alpha_1 \cos \beta_1 = \sin \alpha_2 \cos \beta_2$$

Ecuación: 6 Variables: beta\_x , phi\_x

$$\tan \beta = (1 - f) \tan \phi$$

Ecuación: 7 Variables: s , b , k , l1

$$\frac{s}{b} = \int_0^{\sigma} \sqrt{1 + k^2 \sin^2 \sigma'} d \sigma' = I_1(\sigma)$$

Ecuación: 8 Variables: lamda\_x , omega\_x , l3

$$\lambda = \omega - f \sin \alpha_0 I_3(\sigma)$$

Ecuación: 9 Variable: k

$$k = e' \cos \alpha_0$$

Ecuación: 10 Variable: alpha\_0

$$\alpha_0 = \text{ph}(|\cos \alpha + i \sin \alpha \sin \beta| + i \sin \alpha \cos \beta)$$

Ecuación: 11 Variable: sigma\_x

$$\sigma = ph(\cos \alpha \cos \beta + i \sin \beta)$$

Ecuación: 12 Variable: omega\_x

$$\omega = ph(\cos \sigma + i \sin \alpha_0 \sin \sigma)$$

Ecuación: 13 Variable: beta\_x

$$\beta = ph(|\cos \alpha_0 \cos \sigma + i \sin \alpha_0| + i \cos \alpha_0 \sin \sigma)$$

Ecuación: 14 Variable: alpha\_x

$$\alpha = ph(\cos \alpha_0 \cos \sigma + i \sin \alpha_0)$$

Ecuación: 15 Variable: I1, A1, sigma\_x, C11, l

$$I_1(\sigma) = A_1 \left( \sigma + \sum_{l=1}^{\infty} C_{1l} \sin 2l\sigma \right)$$

Ecuación: 16 Variable: epsilon

$$\epsilon = \frac{\sqrt{1+k^2} - 1}{\sqrt{1+k^2} + 1}$$

Ecuación: 20 Variables: sigma\_x, tau\_x, C11

$$\sigma = \tau + \sum_{l=1}^{\infty} C'_{1l} \sin 2l\tau$$

Ecuación: 23 Variables: I3, A3, sigma\_x

$$I_3(\sigma) = A_3 \left( \sigma + \sum_{l=1}^{\infty} C_{3l} \sin 2l\sigma \right)$$

Para la solución de las series de Fourier, ecuaciones 15 y 23, se utilizaron los coeficientes definidos en la biblioteca de Karney. Dichos coeficientes de acuerdo a su publicación, los obtuvo con el programa Maxima [10].

El código fuente en X10 del problema directo se encuentra en el Anexo 2.

## 2.1.1 Diagrama de dependencia entre variables

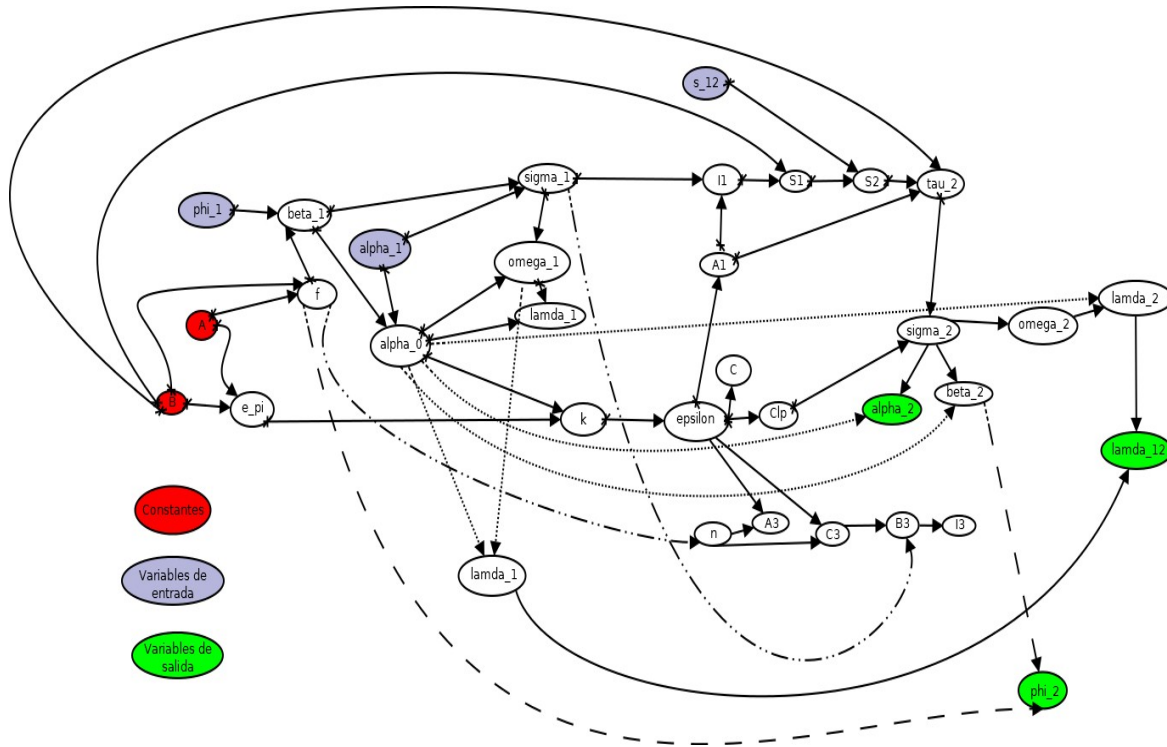


Ilustración 5: Dependencia entre variables problema directo

Al desarrollar un diagrama de dependencia entre variables, se puede observar que existe una gran dependencia de datos en el problema; por lo cual la parte paralela sería mínima y también existen una granularidad muy fina [8].

### 3. Problema inverso

El problema inverso consiste en: conocidas las coordenadas geodésicas de dos puntos latitud A  $\phi_1$  , latitud B  $\phi_2$  y su longitud  $\lambda_{12}$ , se obtiene la distancia entre ambos  $s_{12}$  y los acimut  $\alpha_1$  y  $\alpha_2$ .

A continuación se presentan las ecuaciones principales utilizadas, de acuerdo al trabajo presentado por Karney [1]. También se muestran las variables que se usan en el presente trabajo, donde x es el número de uno de los puntos geodésicos. Algunas ecuaciones se utilizan varias veces durante el cálculo del problema inverso.

Ecuación: 5 Variables: alpha\_x , beta\_x

$$\sin \alpha_0 = \sin \alpha_1 \cos \beta_1 = \sin \alpha_2 \cos \beta_2$$

Ecuación: 6 Variables: alpha\_x , phi\_x

$$\tan \beta = (1 - f) \tan \phi$$

Ecuación: 7 Variables: s , b , sigma\_x , I1

$$\frac{s}{b} = \int_0^{\sigma} \sqrt{1 + k^2 \sin^2 \sigma'} d\sigma' = I_1(\sigma)$$

Ecuación: 8 Variables: lamda\_x , omega\_x , alpha\_0 , I3

$$\lambda = \omega - f \sin \alpha_0 I_3(\sigma)$$

Ecuación: 10 Variable: alpha\_0

$$\alpha_0 = ph \left( |\cos \alpha + i \sin \alpha \sin \beta| + i \sin \alpha \cos \beta \right)$$

Ecuación: 11 Variable: sigma\_x

$$\sigma = ph \left( \cos \alpha \cos \beta + i \sin \beta \right)$$

Ecuación: 12 Variable: omega\_x

$$\omega = ph \left( \cos \sigma + i \sin \alpha_0 \sin \sigma \right)$$

Ecuación: 45 Variable: alpha\_2

$$\cos \alpha_2 = \frac{+\sqrt{\left(\cos^2 \alpha_1 \cos^2 \beta_1 + \left(\cos^2 \beta_2 - \cos^2 \beta_1\right)\right)}}{\cos \beta_2}$$

Ecuación: 53 Variables: lamda\_12 , beta\_2

$$\lambda_{12} = \pi + \frac{fa \pi \cos^2 \beta_1}{a \cos \beta_1} x$$

$$\beta_2 = -\beta_1 + \frac{fa \pi \cos^2 \beta_1}{a} y$$

Ecuación: 55 Variables: x , y , u

$$\mu^4 + 2\mu^3 + (1 - x^2 - y^2)\mu^2 - 2y^2\mu - y^2 = 0$$

Ecuación: 56 Variable: alpha\_1

$$\alpha_1 = ph(y/\mu - ix/(1+\mu))$$

Para la solución de la ecuación 55, se utilizó el método Newton-Raphson con un máximo de 100 iteraciones.

El código fuente en X10 del problema inverso se encuentra en el Anexo 3.

### 3.1 Diagrama de dependencia entre variables

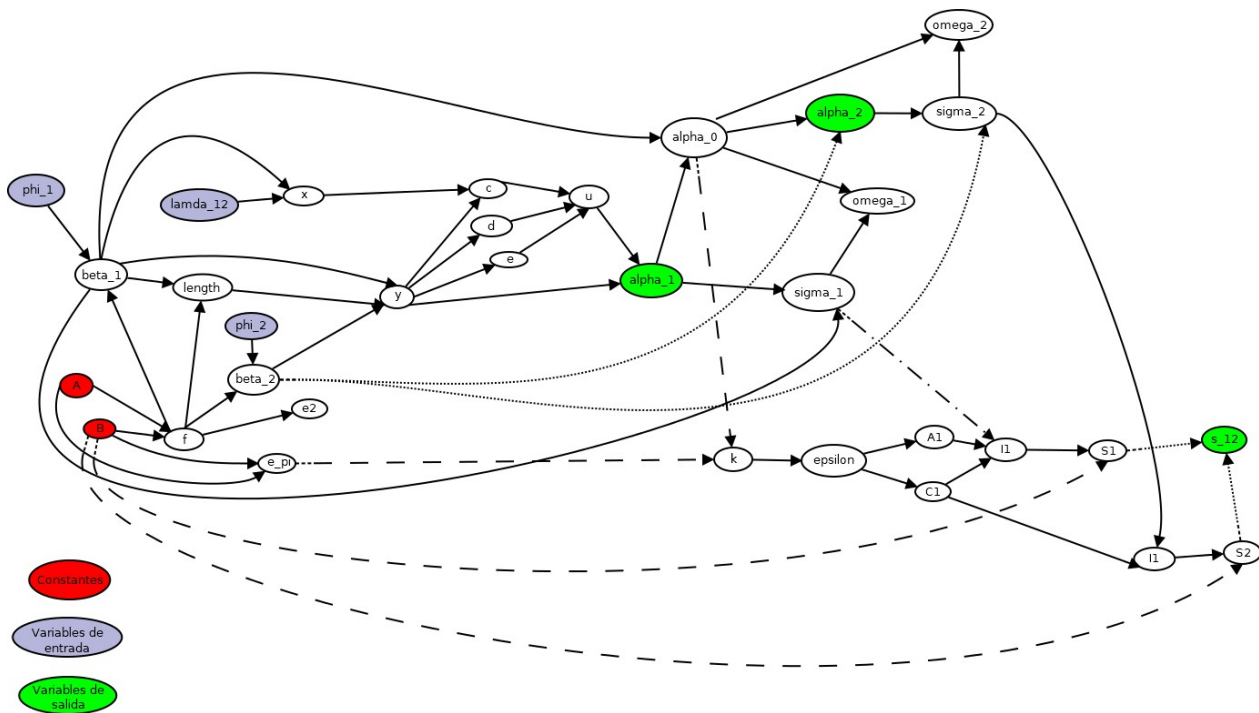


Ilustración 6: Dependencia entre variables problema indirecto

Al desarrollar un diagrama de dependencia entre variables, se puede observar que existe una gran dependencia de datos en el problema; por lo cual la parte paralela sería mínima y también existen una granularidad muy fina [8].

## 4.- Pruebas de la solución al problema directo e inverso

El siguiente programa se realizó para verificar el funcionamiento de las 2 clases (Anexo 2 y 3), correspondientes al cálculo del problema directo e inverso. Obteniendo resultados aproximados a los descritos por Karney en su publicación [\[1\]](#).

```
import x10.io.Console;
import x10.lang.Math;

public class Prueba{

    public static def main(Rail[String]) {

        val phi_1:Double=40;
        val alpha_1:Double=30;
        val s_12:Double=10000000; //en metros
        val phi_11:Double=-30;
        val phi_2:Double=29.9;
        val lamda_12:Double=179.8;

        async
        {
            val prueba=new directo(phi_1,alpha_1,s_12);
            prueba.calculo();

            val prueba3=new inverso(phi_11,phi_2,lamda_12);
            prueba3.calculo();
        }
    }
}
```

### 4.1 Resultados problema directo

En esta sección se muestra una comparación entre los resultados obtenidos por Karney [\[1\]](#) y los generados con el código directo.x10

	Resultados por Karney	Ecuación	Resultados obtenidos
$\phi_1$	40		40
$\alpha_1$	30		30
$s_{12}$	10000000		10000000
$\beta_1$	39.90527714601	6	beta_1 39.90527714601267
$\alpha_0$	22.55394020262	10	alpha_0 22.553940202618076
$\sigma_1$	43.9991534500	11	sigma1 43.999153644996106
$\omega_1$	20.32371827837	12	omega_1 20.323718278370354



$k^2$	0.005748 02962857	9	k2 0.0057480296286200674
$\varepsilon$	0.001432 89220416	16	epsilon 0.0014328922041679198
$A_1$	1.001435 46236207	17	A1 1.0
$I_1(\sigma_1)$	0.768315 38886412	15	I1 0.7679300991960463
$s_1$	4883990.626232	7	s1 4881541.435242859
$s_2$	14883990.626232	$s_1 + s_2$	s2 1.4881541435242858E7
$\tau_2$	133.96266050208	$s_2 / (bA_1)$	tau_2 134.13288338728418
$\sigma_2$	133.92164083038	20	sigma_2 134.13288338728418
$\alpha_2$	149.09016931807	14	alpha_2 149.18640035637526
$\beta_2$	41.69771809250	13	beta_2 41.51638570316971
$\omega_2$	158.28412147112	12	omega_2 158.429043526726
$A_3$	0.99928424306	24	A3 1.0
$I_3(\sigma_1)$	0.76773786069	23	I3 0.7679300991960463
$I_3(\sigma_2)$	2.33534322170	23	I3 2.3410604503017134
$\lambda_1$	20.26715038016	8	lamda_1 20.322730735080633
$\lambda_2$	158.11205042393	8	lamda_2 158.4260329681596
$\lambda_{12}$	137.84490004377	$\lambda_2 - \lambda_1$	lamda_12 138.10330223307898
<b>SOLUCIÓN</b>			
$\phi_2$	41.79331020506	6	phi_2 41.61190685205572
$\lambda_{12}$	137.84490004377	$\lambda_2 - \lambda_1$	lamda_12 138.10330223307898
$\alpha_2$	149.09016931807	14	alpha_2 149.18640035637526

## 4.2 Resultados problema inverso

En esta sección se muestra una comparación entre los resultados obtenidos por Karney [1] y los generados con el código inverso.x10

	Resultados por Karney	Ecuación	Resultados obtenidos
$\phi_1$	-30		
$\phi_2$	29.9		
$\lambda_{12}$	179.8		
x	-0.382344	53	x -0.33139691506719077
y	-0.220189	53	y -0.19084940128345773

$\mu$	0.231633	55	u 0.1985910441193725
$\alpha_1$	161.914	56	alpha_1 163.9492476133637
$\beta_1$	-29.916 747 713 24	6	Beta_1 -29.91674771323539
$\alpha_0$	15.629 479 665 37	10	alpha_0 13.86570008900294
$\sigma_1$	-148.809 136 917 76	11	sigma_1 -149.0886927711873
$\omega_1$	-170.736 343 780 66	12	omega_1 -171.83436917662448
$\beta_2$	29.816 916 421 89	6	Beta_2 29.81691642188545
$\alpha_2$	18.090 737 245 74	5,45	alpha_2 16.03426761122078
$\sigma_2$	31.085 834 470 40	11	sigma_2 30.80743066789771
$\omega_2$	9.226 028 621 10	12	omega_2 8.13252793531086
$s_1$	-16 539 979.064 227	7	s1 -1.6540832516025273E7
$s_2$	3 449 853.763 383	7	s2 3417969.139409058
$s_{12}$	19 989 832.827 610	$s_2 - s_1$	s_12 1.9958801655434333E7
SOLUCIÓN			
$\alpha_1$	161.914	56	alpha_1 163.9492476133637
$\alpha_2$	18.090 737 245 74	5,45	alpha_2 16.03426761122078
$s_{12}$	19 989 832.827 610	$s_2 - s_1$	s_12 1.9958801655434333E7

## 5.- Solución propuesta

La solución propuesta consta de 3 programas que se ejecutarán de forma paralela en el dispositivo. Por lo cual cada dispositivo será maestro y esclavo al mismo tiempo. La solución presenta paralelismo al procesar el problema directo e inverso utilizando la misma información obtenida por el GPS. También se utiliza paralelismo con descomposición funcional [8] al poder recibir y enviar peticiones a otros dispositivos, y leer el modulo GPS.

A continuación se muestra la comunicación que se realiza entre los dispositivos y los programas.

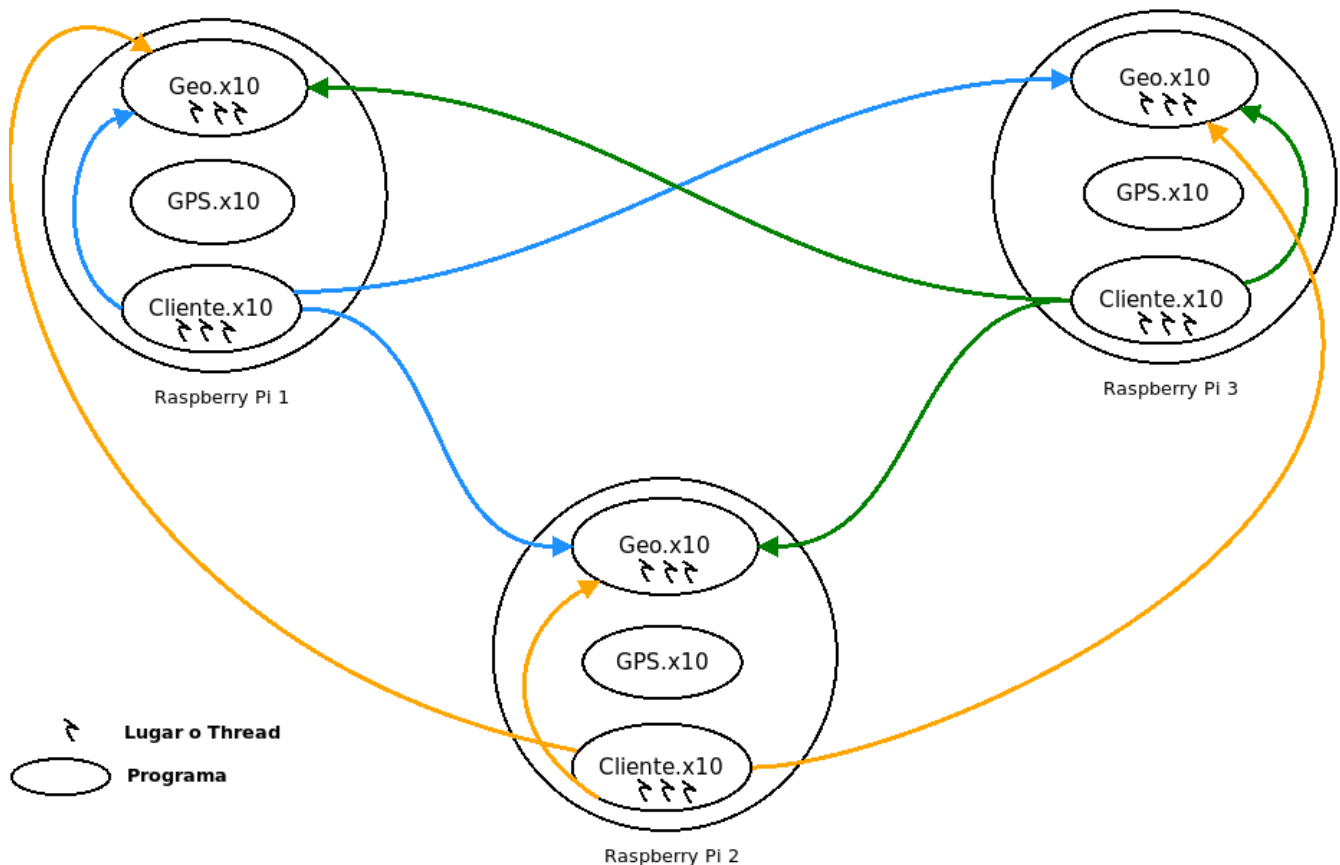


Ilustración 7: Comunicación entre dispositivos

El código fuente se encuentra en el Anexo 5. A continuación, se describe brevemente su funcionamiento.

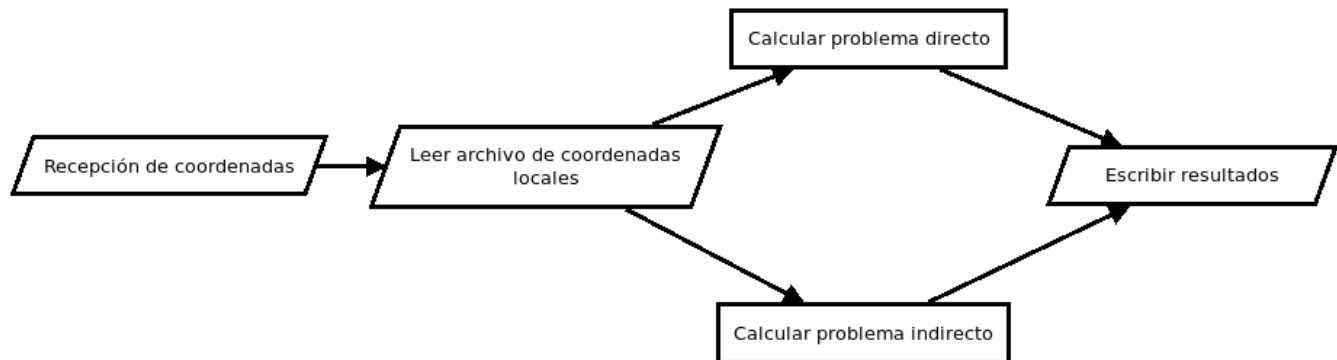
### 5.1 Programa de recepción

En general el programa realiza las siguientes actividades.

1. Recibir coordenadas de otro dispositivo

2. Leer archivo de coordenadas del dispositivo
3. Calcular problema directo e inverso de forma paralela

A continuación se presenta un diagrama de flujo que ilustra una petición al programa. El cálculo del problema directo e inverso se ejecutan de forma paralela.



*Ilustración 8: Petición al programa*

El programa de recepción se encarga de procesar las peticiones en el puerto TCP 4022. La información recibida debe tener la siguiente estructura, de lo contrario no se procesará.

Origen:Latitud:Longitud

El campo origen corresponde a la IP del dispositivo. El campo Latitud y Longitud acepta valores en grados.

## 5.2 Programa de envío

El programa realiza las siguientes actividades.

1. Leer archivo de coordenadas del dispositivo
2. Leer archivo de direcciones IP
3. Enviar coordenadas a los demás dispositivos

El envío se realiza a las ips que se encuentran en un archivo de texto plano (una ip por línea), de manera paralela.

## 5.3 Programa de lectura del GPS

El programa realiza las siguientes actividades.

1. Leer el puerto serial utilizando la biblioteca jSerialComm
2. Obtener la cadena de texto GPGLL
3. Convertir a grados decimales
4. Escribir latitud y longitud separados por : en un archivo de texto

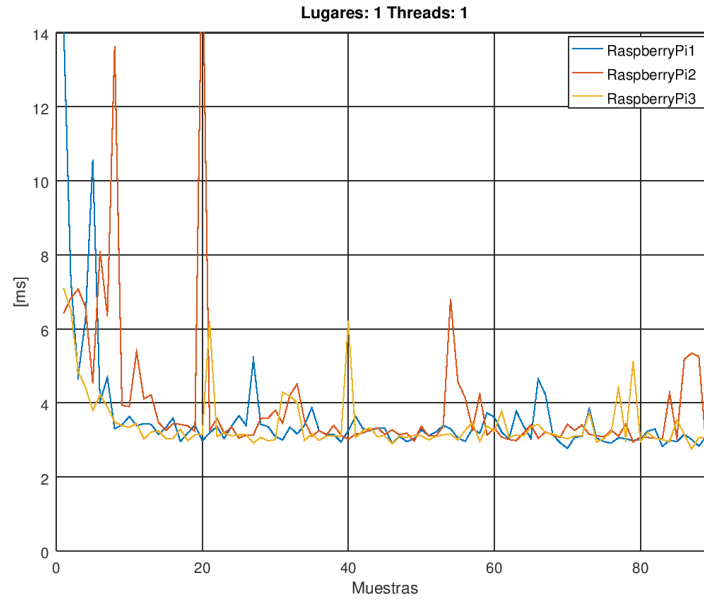
Respecto a la cadena NEMA GPGLL (Geographic Latitude and Longitude), contiene la siguiente estructura:

\$GPGLL,5106.7198674,N,11402.3587526,W,220152.50,A

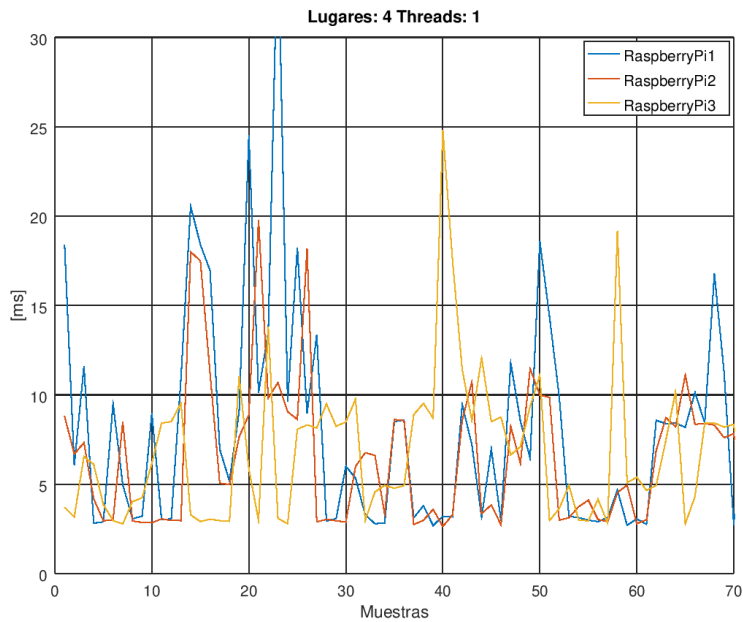
#	Estructura	Descripción	Ejemplo
1	\$GPGLL	Cabecera	\$GPGLL
2	Lat	Latitud (DDmm.mm)	5106.7198674
3	Latdir	Dirección latitud (N = Norte, S = Sur)	N
4	Lon	Longitud (DDDmm.mm)	11402.3587526
5	Londir	Dirección longitud (E = Este, W = Oeste)	W
6	UTC	Tiempo de estatus de la posición en UTC (hours/minutes/seconds/decimal seconds)	220152.50
7	Estatus	Estado: A = Datos validos, V = Datos invalidos	A

## 6.- Resultados

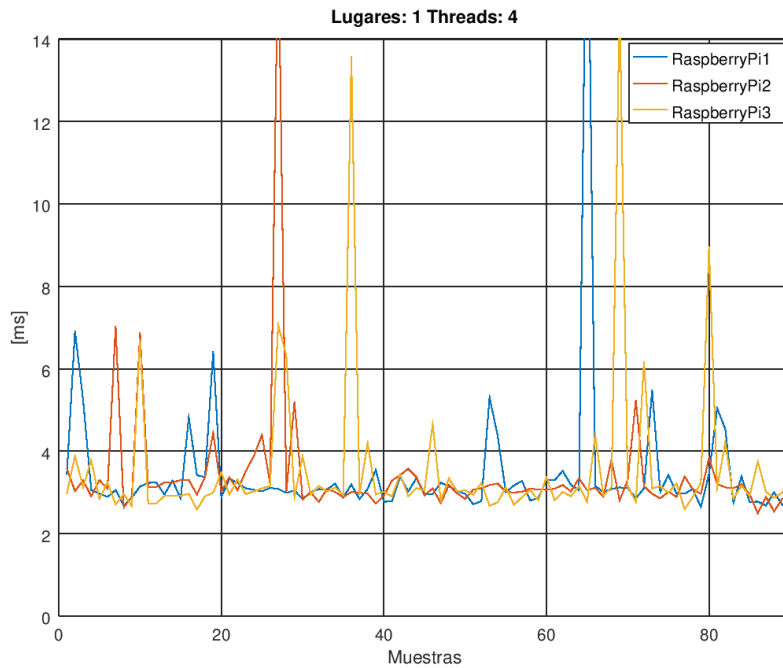
Al ejecutar los 3 programas en los 3 dispositivos durante un tiempo aproximado de 2 minutos, se obtuvieron los siguientes tiempos de ejecución en 1 dispositivo.



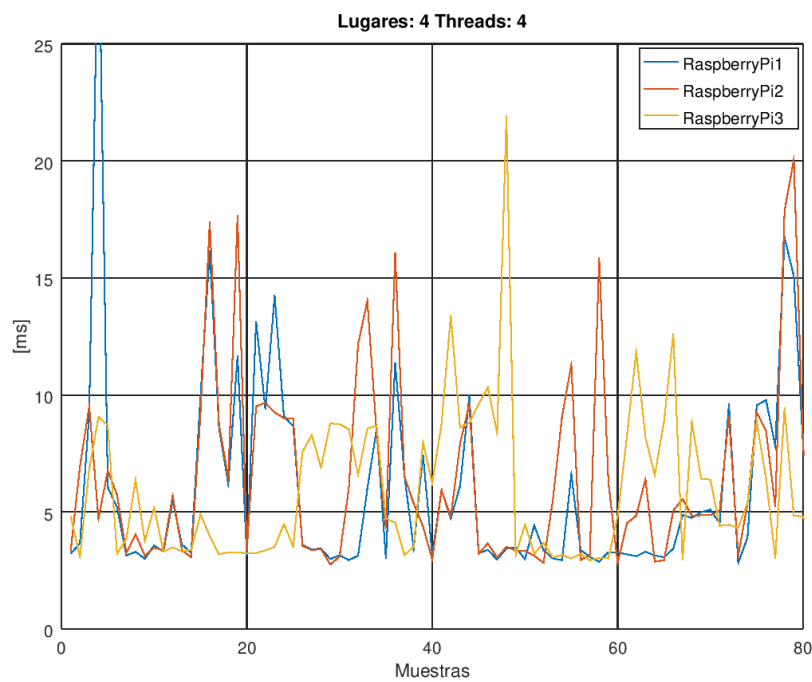
*Ilustración 9: Resultado Lugares:1 Threads:1*



*Ilustración 10: Resultado Lugares:4 Threads:1*



*Ilustración 11: Resultado Lugares:1 Threads:4*



*Ilustración 12: Resultado Lugares:4 Threads:4*

Al analizar las figuras se observa que los mejores tiempos de ejecución se obtienen con 4 Threads y 1 Lugar , y de forma serial con 1 Thread y 1 Lugar. Por lo cual a continuación se

comparan las peticiones de cada uno de los dispositivos de forma serial, 1 Thread y 1 Lugar, y con 4 Threads y 1 Lugar.

### 1 Lugar y 1 Thread vs 1 Lugar y 4 Threads

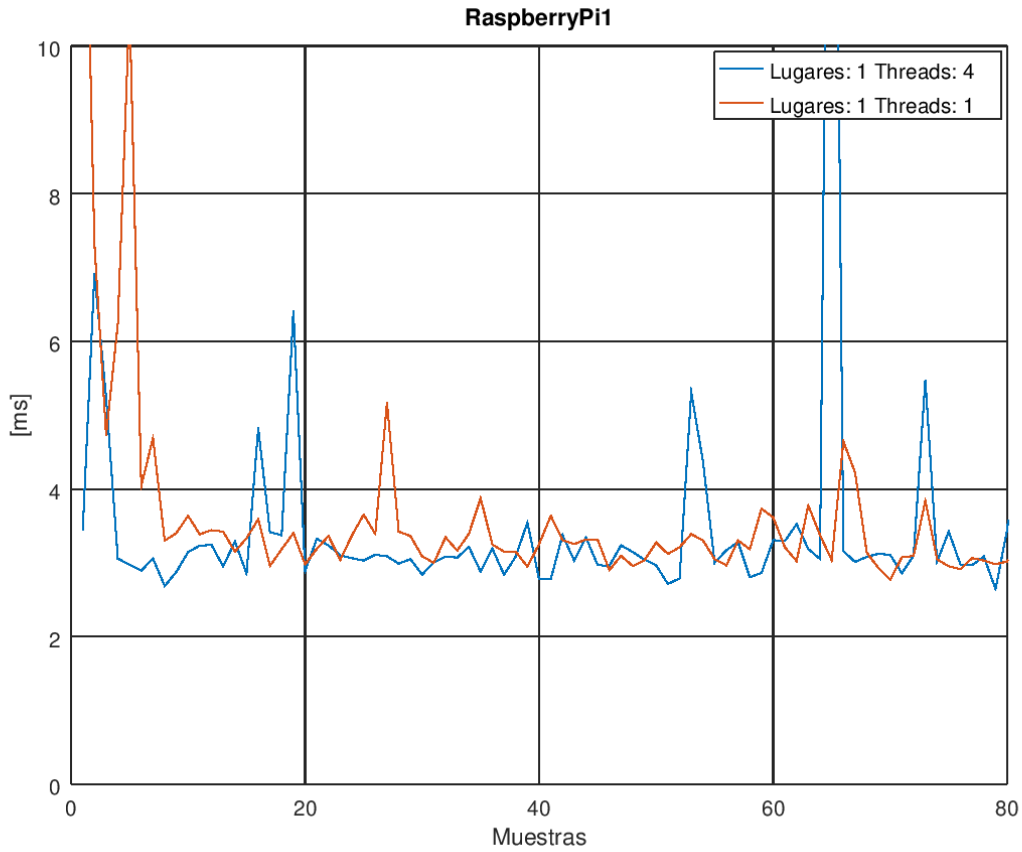
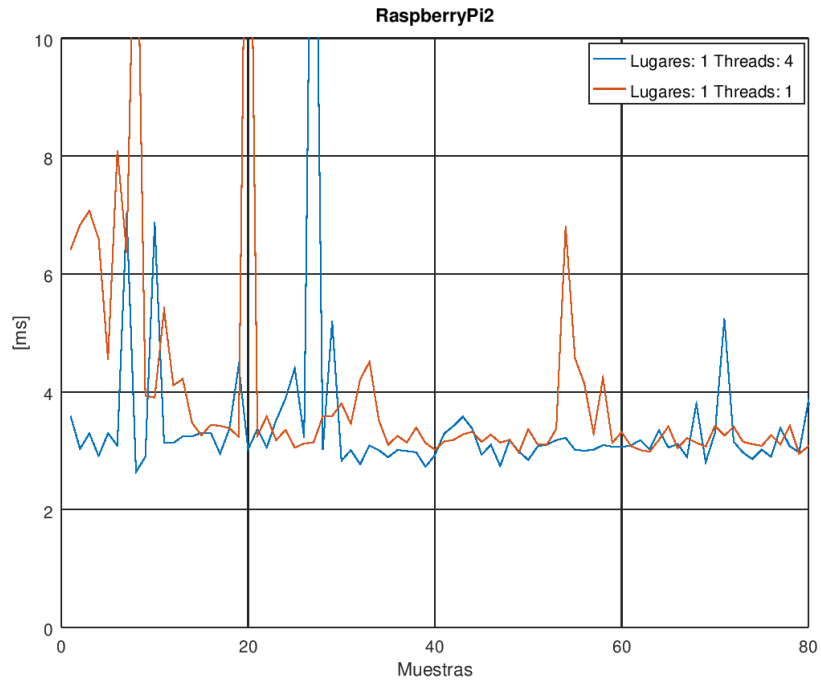
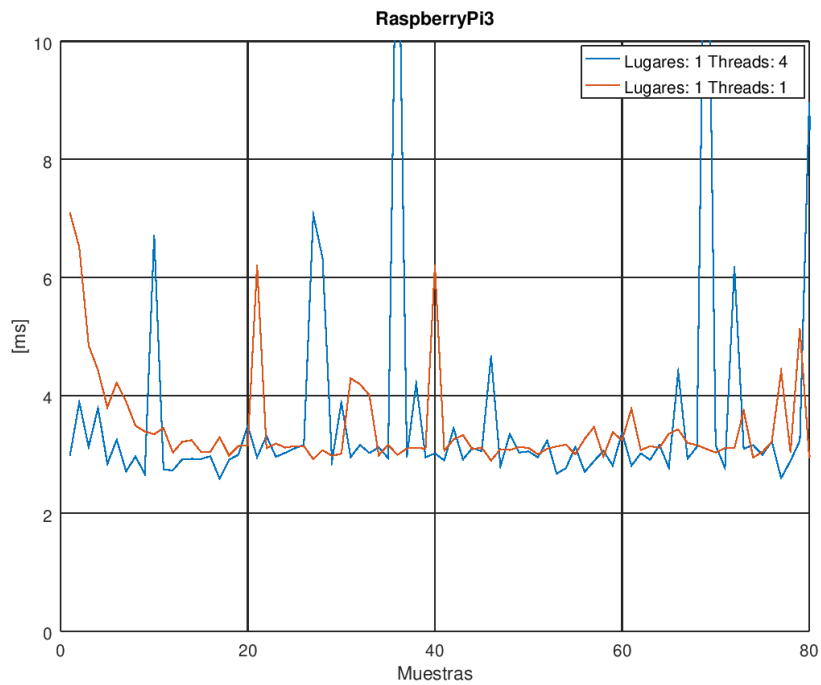


Ilustración 13: Resultado RaspberryPi1 1 Lugar y 1 Thread vs 1 Lugar y 4 Threads





*Ilustración 14: Resultado RaspberryPi2 1 Lugar y 1 Thread vs 1 Lugar y 4 Threads*



*Ilustración 15: Resultado RaspberryPi3 1 Lugar y 1 Thread vs 1 Lugar y 4 Threads*

Respecto a la información geodésica, a continuación se muestran los datos obtenidos entre 2 dispositivos.

en phi 1	en alpha 1	en s 12	sd phi 2	sd alpha 2	sd lamda 12	en phi 1	en phi 2	en lamda 12	si alpha 1	si alpha 2	si s 12
19.4161834	43.3235483988616	3.59593267816034	19.4162070408746	43.3235562109128	2.35691035612717E-05	19.4161834	19.4162069	-7.39000000038459E-05	128.526346086702	51.4736642547963	-13441731.6786356
19.4161829	0	0	19.4161829	0	0	19.4161829	19.4161829	0	44.6877996132526	44.6877996132526	0
19.4161829	43.32354822298474	3.74895109061824	19.4162075468693	43.3235563743262	2.45720441434116E-05	19.4161829	19.4162074	-7.37999999955718E-05	48.7767907701321	48.7768005677039	4.11395982420072
19.4161829	0	0	19.4161829	0	0	19.4161829	19.4161829	0	44.6877996132526	44.6877996132526	0
19.4161829	43.3235487348444	3.44291427664652	19.4162055348799	43.3235562144675	2.25661629826845E-05	19.4161829	19.4162054	-7.37999999955718E-05	-102.800703421237	-77.1993312757141	-7153748.18238534
19.4161815	0	0	19.4161815	0	0	19.4161815	19.4161815	0	139.443260685318	40.5567393146821	-14464196.6341027
19.4161815	43.3235483221681	3.84076214945215	19.4162067504661	43.323556661032	2.51738084848796E-05	19.4161815	19.4162066	-7.45999999907099E-05	-99.3318518127713	-80.6682017047732	-5497937.13912872
19.4161815	0	0	19.4161815	0	0	19.4161815	19.4161815	0	139.443260685318	40.5567393146821	-14464196.6341027
19.4161784	43.3235478017202	4.48343949456803	19.4162078756437	43.3235575418511	2.93861589142352E-05	19.4161784	19.4162077	-8.35999999964088E-05	147.608774368561	32.3912321441897	-14958153.2817034
19.4161784	0	0	19.4161784	0	0	19.4161784	19.4161784	0	13.5538128765626	13.5538128765626	0
19.4161783	43.3235477405859	4.52934501822074	19.4162080774421	43.3235575804451	2.96870410885219E-05	19.4161783	19.4162079	-8.37999999845351E-05	14.0248636524283	14.0248662429995	3.37605982786044
19.4161783	0	0	19.4161783	0	0	19.4161783	19.4161783	0	164.033844024037	15.9661559759634	-15516586.7669913
19.4161784	43.3235481841279	4.25391188421149	19.4162063666517	43.3235574256169	2.78817480428017E-05	19.4161784	19.4162062	-8.56000000055701E-05	45.5637892289725	45.5637991629417	4.39390958566219
19.4161775	0	0	19.4161775	0	0	19.4161775	19.4161775	0	61.2996575816084	61.2996575816084	0
19.4161775	43.3235480076346	4.45283582507781	19.4162067744447	43.3235576812795	2.91855707921229E-05	19.4161775	19.4162066	-8.78000000028578E-05	132.57837181044	47.4216392858705	-13881937.2604004
19.4161769	0	0	19.4161769	0	0	19.4161769	19.4161769	0	23.1235170537797	23.1235170537797	0
19.4161764	43.3235478200743	4.6823634491113	19.4162071834367	43.3235579923605	3.06899816635564E-05	19.4161764	19.416207	-9.40999999841097E-05	31.35245663159223	31.35245663159223	3.9650422851555
19.4161764	0	0	19.4161764	0	0	19.4161764	19.4161764	0	139.97682293632	40.0231770636799	-14502334.5305703
19.4161769	43.3235477567565	4.66706160201983	19.4162075828373	43.3235578958001	3.05898687611382E-05	19.4161769	19.4162074	-9.64000000038823E-05	115.915119681068	64.0849023118867	-11362608.4135981
19.4161762	0	0	19.4161762	0	0	19.4161762	19.4161762	0	38.8528074399092	38.8528074399092	4.65661287307739E-10
19.4161762	43.3235477024823	4.77417449621808	19.4162075870335	43.3235580742251	3.12917460121298E-05	19.4161762	19.4162074	-9.57999999968706E-05	135.392219436712	44.6077913462735	-14142686.7858918
19.4161765	0	0	19.4161765	0	0	19.4161765	19.4161765	0	55.2369918122753	55.2369918122753	9.31322574615479E-10
19.4161784	43.3235475979652	4.60585421926203	19.4162086804394	43.3235576040385	3.01885113742628E-05	19.4161784	19.4162085	-9.40999999983205E-05	93.269297882402	86.7308867490801	-2046017.76046648
19.4161784	0	0	19.4161784	0	0	19.4161784	19.4161784	0	13.5538128765626	13.5538128765626	0
19.4161784	43.3235476478273	4.57525053872347	19.4162084792405	43.323557587415	2.99879232592559E-05	19.4161784	19.4162083	-9.45999999970582E-05	-92.6980687656765	-87.3021535489642	-1693442.4794832
19.4161784	0	0	19.4161784	0	0	19.4161784	19.4161784	0	13.5538128765626	13.5538128765626	0
19.4161744	43.3235473308783	5.18732421935742	19.4162085032191	43.3235586001756	3.39996855771574E-05	19.4161744	19.4162083	-9.20000000093069E-05	94.7318669709236	85.2682765244629	-2932304.59670405
19.4161744	0	0	19.4161744	0	0	19.4161744	19.4161744	0	176.392524775771	3.60747522422939	-15667500.3837319
19.4161743	43.3235481374229	4.71296715891426	19.4162052846357	43.3235583761936	3.0890569782116E-05	19.4161743	19.4162051	-9.07000000012204E-05	161.188842398478	18.111612775887	-15452463.2185554
19.4161743	0	0	19.4161743	0	0	19.4161743	19.4161743	0	34.4207045401855	34.4207045401854	-4.65661287307739E-10
19.4161733	43.3235482917981	4.72826901350737	19.4162043852351	43.3235585638111	3.09908638378431E-05	19.4161733	19.4162042	-8.940000000215556E-05	128.906590109193	51.0934233051064	-13486673.2715013
19.4161735	0	0	19.4161735	0	0	19.4161735	19.4161735	0	94.9412546491863	85.0587453508137	-3056959.60852238
19.4161748	43.3235484067225	4.4987413817684	19.4162043762431	43.3235581800945	2.94864529699623E-05	19.4161748	19.4162042	-8.68999999852349E-05	174.004096785668	5.99590429626194	-15653533.2415992
19.4161748	0	0	19.4161748	0	0	19.4161748	19.4161748	0	161.026064873939	18.9739351260607	-15448433.162481
19.4161741	43.3235485077528	4.51404323185917	19.4162037768426	43.3235583143672	2.95867470399003E-05	19.4161741	19.4162036	-8.56999999996333E-05	21.5622870099866	21.5622910944305	3.50999015988782
19.4161741	0	0	19.4161741	0	0	19.4161741	19.4161741	0	173.57325341711	6.42674658289046	-15650268.1034467
19.4161759	43.3235486501347	4.23861007404349	19.4162037660522	43.3235578583798	2.77814539941801E-05	19.4161759	19.4162036	-8.40000000010832E-05	63.1731076380393	63.1731268289433	6.79193492806282
19.4161762	0	0	19.4161762	0	0	19.4161762	19.4161762	0	38.8528074399092	38.8528074399092	4.65661287307739E-10
19.4161762	43.3235486695976	4.19270454844827	19.4162037642538	43.3235577781145	2.74805718163407E-05	19.4161762	19.4162036	-8.25999999989335E-05	41.6004167449821	41.6004252685788	4.0545676490292
19.4161771	0	0	19.4161771	0	0	19.4161771	19.4161771	0	153.82477259772	26.17522740228	-15226927.1056591
19.4161771	43.3235489206514	3.94787508501262	19.4162030546623	43.3235574972841	2.58758668927328E-05	19.4161771	19.4162029	-8.32999999857975E-05	4.14854017639544	4.14854083206153	2.86242832802236
19.4161771	0	0	19.4161771	0	0	19.4161771	19.4161771	0	153.82477259772	26.17522740228	-15226927.1056591
19.4161771	43.3235489685085	3.91727140300437	19.4162028534634	43.3235574786556	2.5675278777259E-05	19.4161771	19.4162027	-8.40999999809355E-05	14.8030486484662	14.8030510188329	2.93004595767707
19.4161793	0	0	19.4161793	0	0	19.4161793	19.4161793	0	68.6741358021652	68.6741358021652	-2.79396772384644E-09
19.4161792	43.3235493380973	3.47351799563508	19.4162020360789	43.3235568842046	2.27667511083496E-05	19.4161792	19.4162019	-8.01999999850978E-05	-96.3827473914382	-83.6173237089511	-3896522.08868927
19.4161792	0	0	19.4161792	0	0	19.4161792	19.4161792	0	14.5759799366432	14.5759799366432	0
19.4161785	43.3235492318529	3.61123457022565	19.4162022414741	43.3235570771449	2.3669397627657E-05	19.4161785	19.4162021	-7.94000000041706E-05	177.281293570382	2.71870682227089	-15670909.0110538
19.4161785	0	0	19.4161785	0	0	19.4161785	19.4161785	0	93.3443905364313	86.6556094635688	-2092186.02349113
19.4161781	-136.67644462915	0.061207367456832	19.4161776976021	-136.676444762121	-4.01176208697507E-07	19.4161781	19.4161777	1.89999998667645E-06	138.393792023438	41.6062078521046	-14386460.530744
19.4161781	43.3235491274444	3.71834746069706	19.4162025456703	43.3235572054358	2.4371456024852E-05	19.4161781	19.4162024	-7.5600000002396E-05	-96.957182898446	-83.0428868743797	-4221380.46255787
19.4161774	0	0	19.4161774	0	0	19.4161774	19.4161774	0	74.747577618998	74.747577618998	0
19.4161774	43.3235490921368	3.81015851349901	19.4162024492671	43.3235573695845	2.49732203734254E-05	19.4161774	19.4162023	-7.54000000142696E-05	24.445254041858	24.4452580076805	3.02665611542761
19.4161772	0	0	19.4161772	0	0	19.4161772	19.4161772	0	10.174621431779	10.174621431779	0
19.4161772	43.3235486764228	4.08559165070044	19.4162040600575	43.3235575522406	2.67785134120402E-05	19.4161772	19.4162039	-7.46000000049207E-05	10.0556656020338	10.0556672609379	3.0006125443615
19.4161789	-136.676444585824	0.122414735187638	19.4161780952043	-136.676444851765	-8.02352445816723E-07	19.4161789	19.4161781	2.1999999972877E-06	91.7495392325927	88.2504515907832	-1102155.29343042
19.4161789	43.3235484242445	4.0549879493039	19.416205588586	43.3235572335775	2.65779252934806E-05	19.4161789	19.4162054	-7.3099999994497E-05	160.890945364153	19.109057852669	-15445050.1992199

## 7.- Conclusiones

De acuerdo a los datos obtenidos respecto al problema directo e inverso, se puede observar que el algoritmo del problema directo muestra resultados esperados

en_phi_1	en_alpha_1	en_s_12	sd_phi_2	sd_alpha_2	sd_lamda_12
19.4161834	43.3235483988616	3.59593267816034	19.4162070408746	43.3235562109128	2.35691035612717E-05

Una longitud  $\lambda_{12}$  de 2.35691035612717E-05, latitud del punto B  $\phi_2$  19.4162070408746 y acimut al vértice A  $\alpha_2$  43.3235562109128.

Respecto al problema inverso, en la mayoría de las muestras se obtuvieron datos incorrectos. Por ejemplo, en la siguiente muestra, se obtuvo una distancia entre los 2 puntos de -13441731.6786356 [m] la cual es incorrecta.

en_phi_1	en_phi_2	en_lamda_12	si_alpha_1	si_alpha_2	si_s_12
19.4161834	19.4162069	-7.39000000038459E-05	128.526346086702	51.4736642547963	-13441731.6786356

Sin embargo, se presentaron algunas mediciones que si muestran correctamente la distancia entre los 2 puntos. Por ejemplo, en la siguiente muestra la distancia es de 3.37605982786044 [m] que es consistente con la medición del problema directo.

en_phi_1	en_phi_2	en_lamda_12	si_alpha_1	si_alpha_2	si_s_12
19.4161783	19.4162079	-8.37999999845351E-05	14.0248636524283	14.0248662429995	3.37605982786044

Una distancia  $s_{12}$  de 3.37605982786044 y los acimut  $\alpha_1$  14.0248636524283 y  $\alpha_2$  14.0248662429995.

Al ejecutar los algoritmos con los mismo datos de entrada, es decir, cuando el dispositivo hace una petición a si mismo, se obtuvo lo siguiente.

En el caso del problema directo, una vez más se obtuvieron datos correctos.

Una longitud  $\lambda_{12}$  de 0, latitud del punto B  $\phi_2$  19.4161783 y acimut al vértice A  $\alpha_2$  0.

en_phi_1	en_alpha_1	en_s_12	sd_phi_2	sd_alpha_2	sd_lamda_12
19.4161783	0	0	19.4161783	0	0

En el caso del problema inverso, en la mayoría de las muestras, los datos son incorrectos.

Una distancia  $s_{12}$  de -15516586.7669913 y los acimut  $\alpha_1$  164.033844024037 y  $\alpha_2$  15.9661559759634.

en_phi_1	en_phi_2	en_lamda_12	si_alpha_1	si_alpha_2	si_s_12
19.4161783	19.4161783	0	164.033844024037	15.9661559759634	-15516586.7669913

Al obtener las mediciones respecto al tiempo, se observa que existe una mejoría de alrededor de 1ms al utilizar 4 Threads, respecto a la parte serial. En teoría dicha brecha entre ambas mediciones debería de aumentar al incrementar el número de dispositivos, ya que aunque se generan 4 Threads el uso del procesador fue de un 34%.

También se observa que el tiempo de ejecución no es constante, esto se debe a procesos que se encuentran en ejecución por parte del sistema operativo, como lectura/escritura de la memoria SD, interrupciones del puerto serial, el colector de basura, swap, etc..

En general existe una mejora al utilizar paralelismo, aunque el tiempo de ejecución disminuye alrededor de 1ms, se puede considerar que es un tiempo razonable, ya que se utiliza el puerto serial, acceso a la memoria SD y en este caso los 3 ejecutables de X10 corren sobre la máquina virtual de Java.

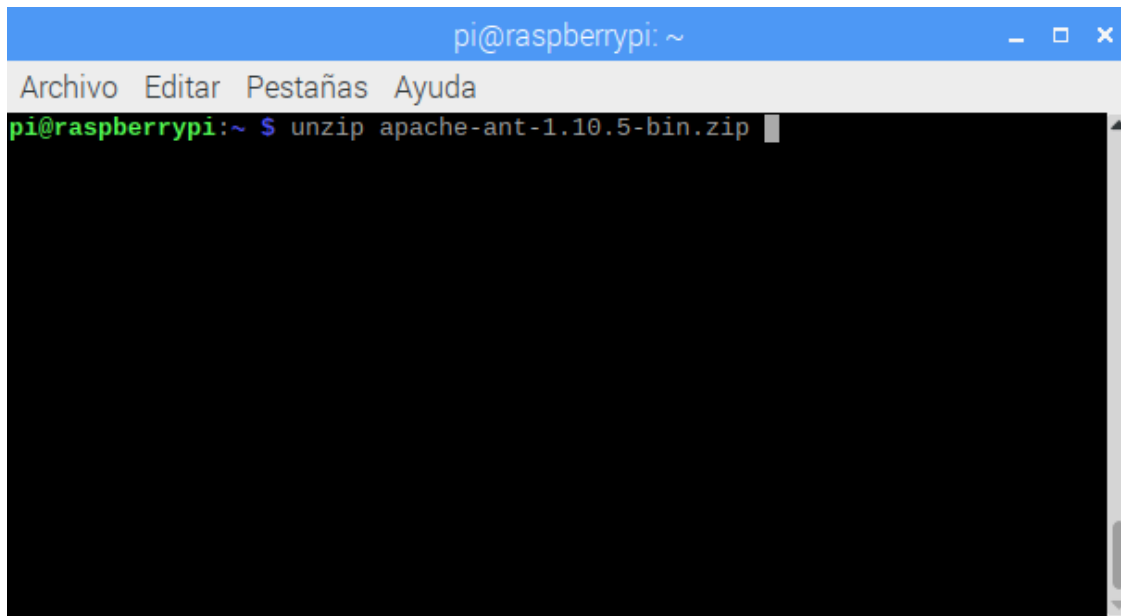
## 8.- Bibliografía

- [1] Charles F. F. Karney , "**Algorithms for Geodesics**", *Journal of Geodesy*, vol. 87, no. 1, pp. 43-55, 2013.
- [2] Raspberry Pi Foundation. "**The Official Raspberry Pi Beginner's Guide: How to Use Your New Computer**". Raspberry Pi Press, 2018. 241p. ISBN: 9781912047680
- [3] Barnes, Russell. "**Official Raspberry Pi Projects Book**". Raspberry Pi Press, 2015. 202p. ISBN: 9781908256690
- [4] Olivier Tardieu. "**APGAS Programming in X10**". Hartree Centre Summer School 2013 "Programming for Petascale".
- [5] <https://www.omnicalculator.com/other/azimuth>
- [6] <https://www.carrod.mx/products/modulo-gps-gy-neo6mv2>
- [7] <https://docs.microsoft.com/es-es/windows/iot-core/windows-iot>
- [8] [https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/)
- [9] <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [10] <https://geographiclib.sourceforge.io/html/geodesic.html#geodseries>

## Anexo 1 Instalación del lenguaje X10 en Raspbian

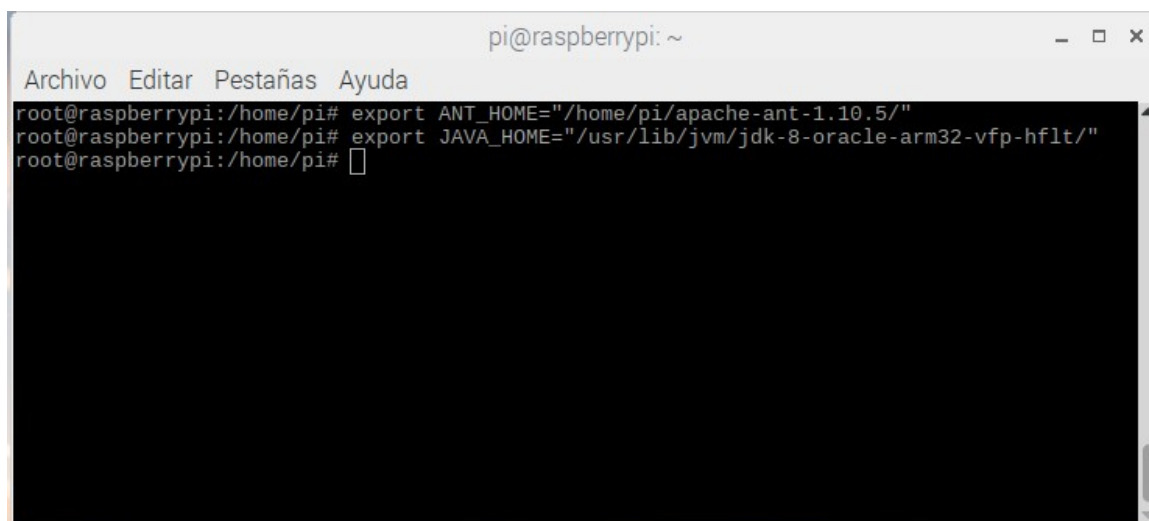
Para el presente trabajo se utilizó la distribución GNU/Linux raspbian stretch del 13 de Noviembre de 2018.

La distribución raspbian con el nombre de imagen 2018-11-13-raspbian-stretch-full.zip ya contiene instalado Oracle Java SDK 8. Por lo cual solamente se necesita instalar una versión más actualizada de ant. Una vez descargado el archivo apache-ant-1.10.5-bin.zip, se descomprime con el siguiente comando:



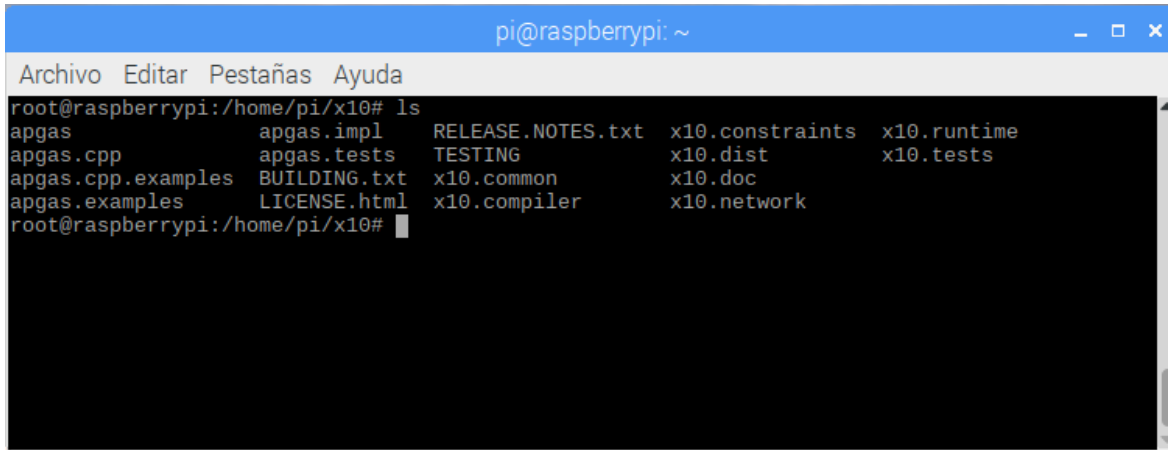
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~ $ unzip apache-ant-1.10.5-bin.zip
```

Después se tiene que definir 2 variables de entorno ANT\_HOME y JAVA\_HOME. La primera corresponde al directorio que se obtuvo en el paso anterior. En el caso de la variable JAVA\_HOME, el valor será el directorio de instalación por defecto del paquete sdk de raspbian.



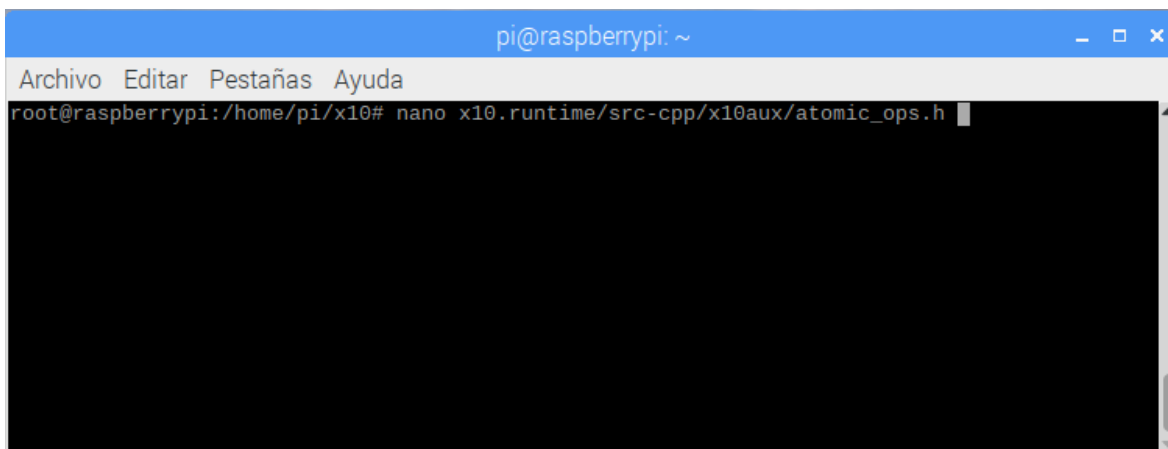
```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
root@raspberrypi:/home/pi# export ANT_HOME="/home/pi/apache-ant-1.10.5/"  
root@raspberrypi:/home/pi# export JAVA_HOME="/usr/lib/jvm/jdk-8-oracle-arm32-vfp-hflt/"  
root@raspberrypi:/home/pi#
```

Después se procede a descargar el código fuente del lenguaje X10 (<http://x10-lang.org/releases/x10-release-262.html>) y descomprimir su contenido. Se obtendrá la siguiente estructura de directorio.



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/x10# ls
appgas          appgas.impl    RELEASE.NOTES.txt  x10.constraints  x10.runtime
appgas.cpp      appgas.tests   TESTING            x10.dist         x10.tests
appgas.cpp.examples BUILDING.txt   x10.common         x10.doc
appgas.examples LICENSE.html   x10.compiler       x10.network
root@raspberrypi:/home/pi/x10#
```

Al utilizar el dispositivo raspberry una arquitectura armv6, se necesita cambiar el siguiente archivo fuente en C. Ya que la arquitectura no está soportada oficialmente por el proyecto X10.



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/x10# nano x10.runtime/src-cpp/x10aux/atomic_ops.h
```

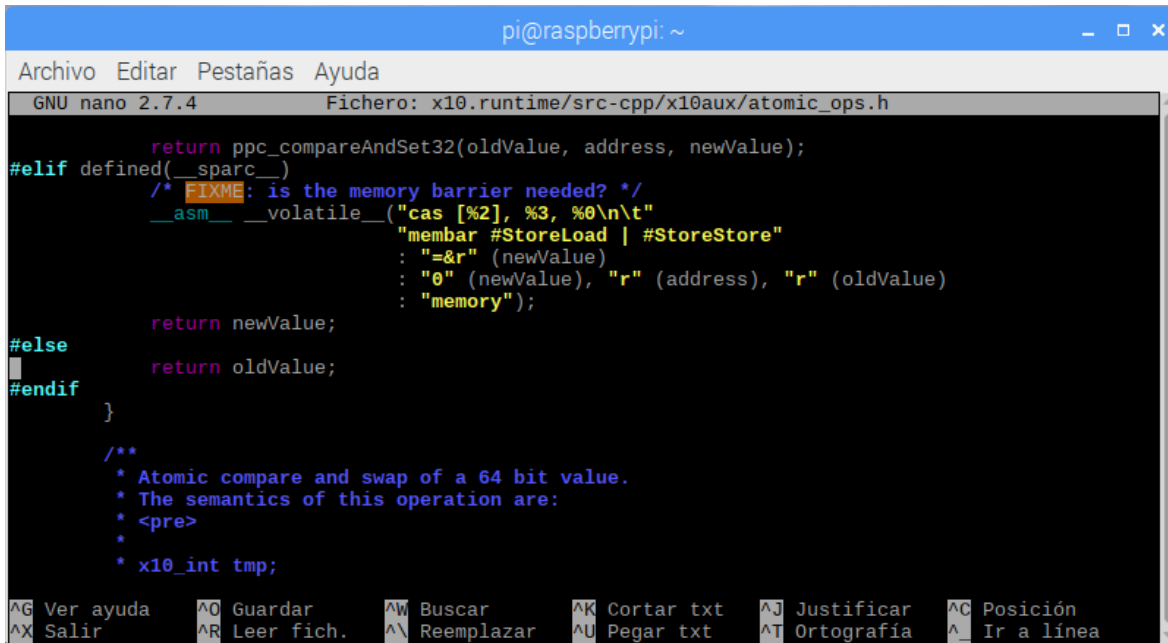
Al abrir el archivo se tiene que cambiar el siguiente código (líneas 179 y 227)

```
#else
# error "Unknown architecture"
#endif
```

por:

```
#else
    return oldValue;
#endif
```

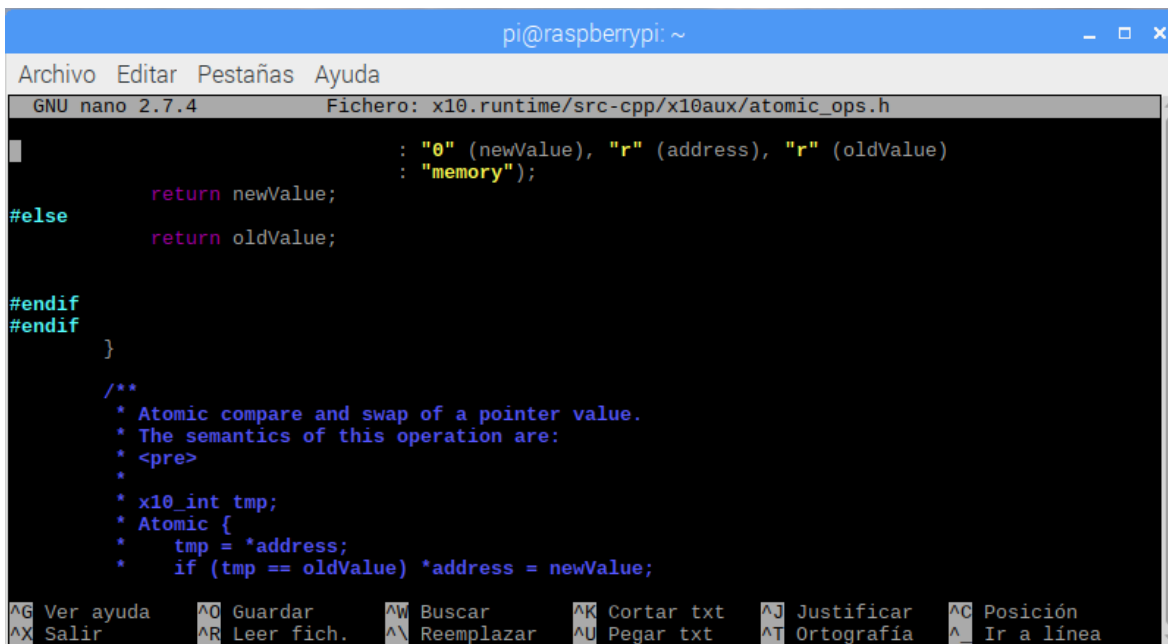
A continuación se muestra la edición con el programa nano.



The screenshot shows the nano editor interface on a Raspberry Pi. The title bar reads "pi@raspberrypi: ~". The menu bar includes "Archivo", "Editar", "Pestañas", and "Ayuda". The status bar shows "GNU nano 2.7.4" and "Fichero: x10.runtime/src-cpp/x10aux/atomic\_ops.h". The code being edited is as follows:

```
        return ppc_compareAndSet32(oldValue, address, newValue);
#elif defined(__sparc__)
    /* FIXME: is the memory barrier needed? */
    __asm__ __volatile__ ("cas [%2], %3, %0\n\t"
        "membar #StoreLoad | #StoreStore"
        : "=&r" (newValue)
        : "0" (newValue), "r" (address), "r" (oldValue)
        : "memory");
#else
    return newValue;
#endif
return oldValue;
#endif
}

/**
 * Atomic compare and swap of a 64 bit value.
 * The semantics of this operation are:
 * <pre>
 *
 * x10_int tmp;
```



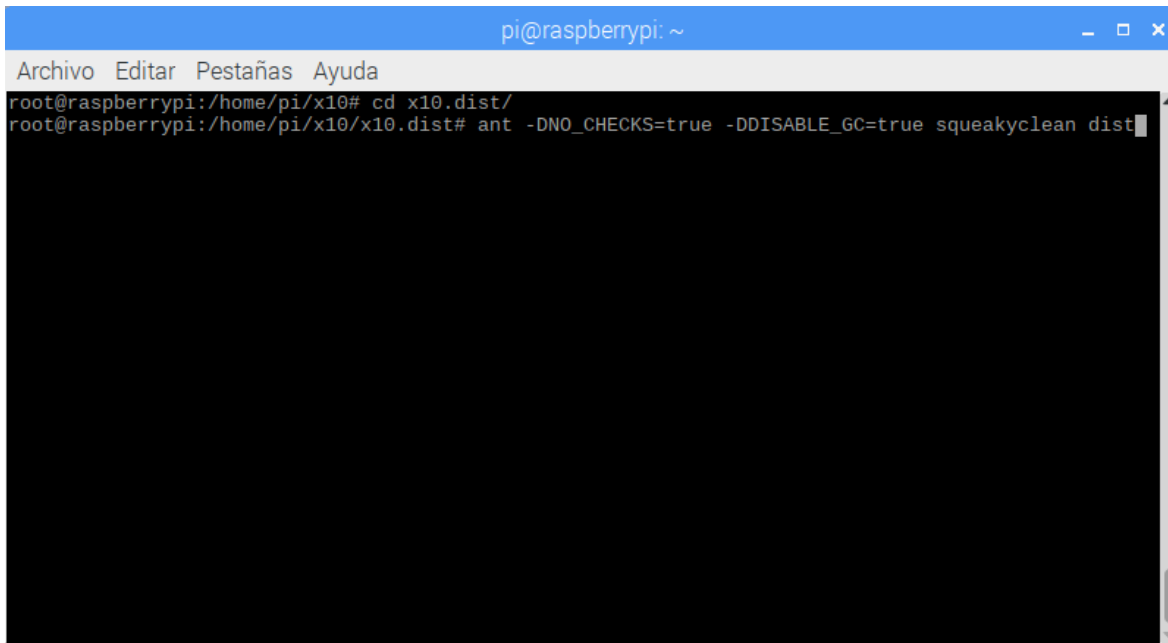
The screenshot shows the nano editor interface on a Raspberry Pi, displaying the modified code. The title bar and menu bar are the same as in the previous screenshot. The code being edited is as follows:

```
        : "0" (newValue), "r" (address), "r" (oldValue)
        : "memory");
#else
    return newValue;
#endif
return oldValue;
#endif
}

/**
 * Atomic compare and swap of a pointer value.
 * The semantics of this operation are:
 * <pre>
 *
 * x10_int tmp;
 * Atomic {
 *     tmp = *address;
 *     if (tmp == oldValue) *address = newValue;
```

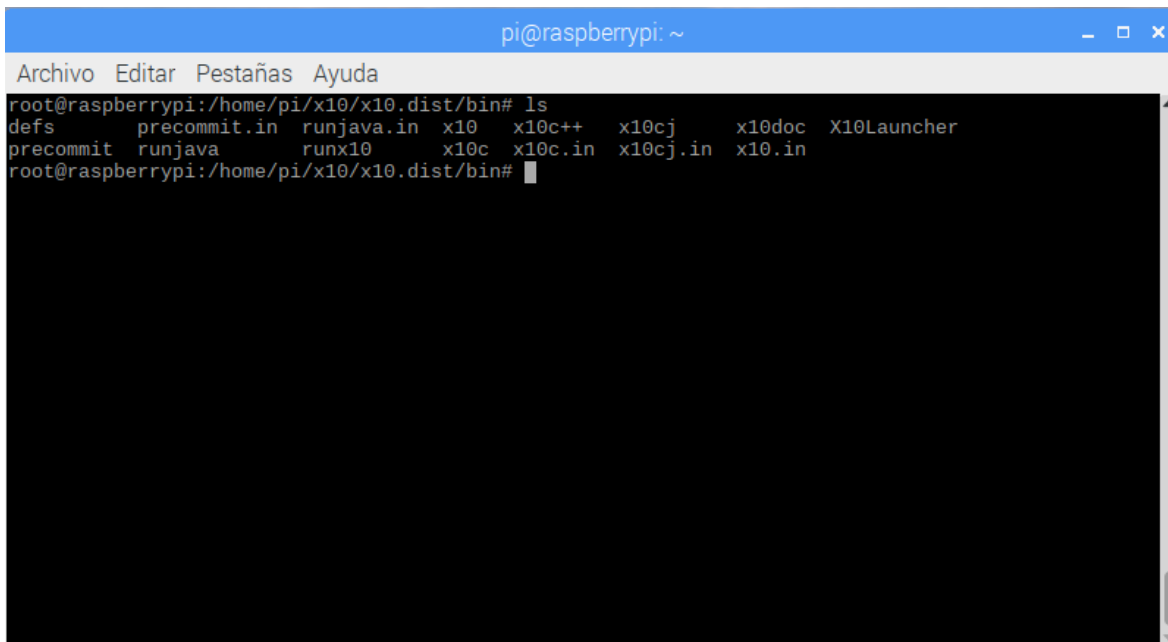


Después se procede a la compilación.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
root@raspberrypi:/home/pi/x10# cd x10.dist/  
root@raspberrypi:/home/pi/x10/x10.dist# ant -DNO_CHECKS=true -DDISABLE_GC=true squeakyclean dist
```

Al finalizar el proceso, los ejecutables se encontrarán en la carpeta bin, a continuación se muestra su contenido.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
root@raspberrypi:/home/pi/x10/x10.dist/bin# ls  
defs      precommit.in  runjava.in  x10      x10c++    x10cj      x10doc  X10Launcher  
precommit  runjava      runx10      x10c     x10c.in  x10cj.in  x10.in  
root@raspberrypi:/home/pi/x10/x10.dist/bin#
```

Para verificar el funcionamiento podemos ejecutar el comando x10c que es compilador de X10 con backend java.

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
defs      precommit.in  runjava.in  x10      x10c++   x10cj      x10doc   X10Launcher
precommit runjava      runx10      x10c     x10c.in  x10cj.in  x10.in
root@raspberrypi:/home/pi/x10/x10.dist/bin# ./x10c
Usage: x10c [options] <source-file>.x10 ...
where [options] includes:
  -sourcepath <path>          source path
  -classpath <path>          search path for class files
  -x10lib <lib.properties>   use the precompiled x10 library described by <lib.properties>
  -buildx10lib <dir>         generate property file in <dir>
  -config <conf>             read configuration <conf> from etc/<conf>.cfg
  -version                   print version info
  -x10rt <impl>              select x10rt impl (sockets, standalone, pami, mpi)
  -t -time                   display execution time
  -v -verbose --verbose      print verbose debugging information
  -h -help --help           print this message
  -disable <pass>           disable compiler pass <pass>
                             valid passes are: async-elimination
  -O -optimize              generate optimized code
  -O0                       generate unoptimized code
  -O1 -O2 -O3 -O4 -O5      generate optimized code and
                             invoke post compiler at specified level
  -g                         generate debug information

Use "x10c -- -help" to get more detailed help on compiler options
root@raspberrypi:/home/pi/x10/x10.dist/bin#
```

Después se compilará un ejemplo de la carpeta “samples”.

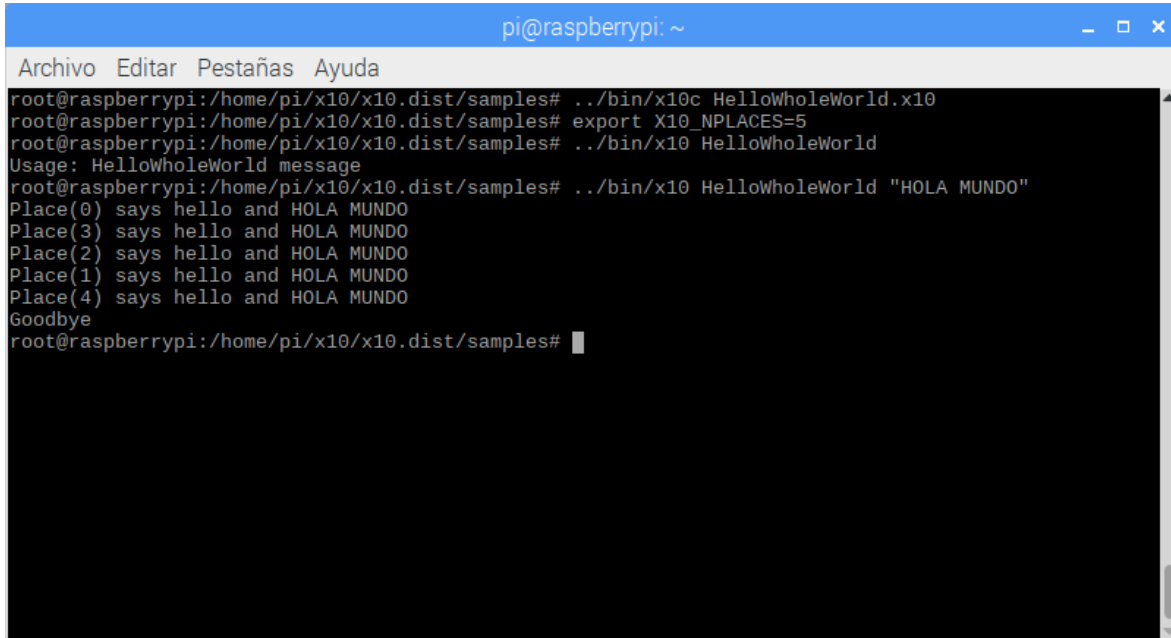
```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
root@raspberrypi:/home/pi/x10/x10.dist/samples# ../bin/x10c HelloWorld.x10
root@raspberrypi:/home/pi/x10/x10.dist/samples#
```

Dicho ejemplo es el programa “Hola Mundo” desde diferentes lugares (procesos).

En X10 para definir el número de procesos/lugares basta con definir la variable global X10\_NPLACES , y si se desea definir uno o más hilos (threads) se utiliza la variable X10\_NTHREADS

Al compilarlo con x10c, con backed java, se necesita ejecutar con la ayuda de la máquina virtual. Por lo cual se utiliza el comando

x10 nombreclase



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
root@raspberrypi:/home/pi/x10/x10.dist/samples# ./bin/x10c HelloWholeWorld.x10  
root@raspberrypi:/home/pi/x10/x10.dist/samples# export X10_NPLACES=5  
root@raspberrypi:/home/pi/x10/x10.dist/samples# ./bin/x10 HelloWholeWorld  
Usage: HelloWholeWorld message  
root@raspberrypi:/home/pi/x10/x10.dist/samples# ./bin/x10 HelloWholeWorld "HOLA MUNDO"  
Place(0) says hello and HOLA MUNDO  
Place(3) says hello and HOLA MUNDO  
Place(2) says hello and HOLA MUNDO  
Place(1) says hello and HOLA MUNDO  
Place(4) says hello and HOLA MUNDO  
Goodbye  
root@raspberrypi:/home/pi/x10/x10.dist/samples#
```

## Anexo 2 Código Fuente directo.x10

```
import x10.io.Console;
import x10.lang.Math;
import x10.array.*;

public class directo {
    val PI:Double=3.14159265358979323846264338327950288;
    val A:Double=6378137.000000;
    val B:Double=6356752.314245;

    var phi_1:Double;
    var alpha_1:Double;
    var s_12:Double;

    public def degToRad(grados:Double){
        return ((grados * PI) / 180.00000);
    }

    public def radToDeg(grados:Double){
        return ((grados) * 180.00000 / PI);
    }

    public def ph(real:Double,img:Double)
    {
        return Math.atan2(img,real);
    }

    public def this( phi_11:Double,alpha_11:Double,s_121:Double) {

        phi_1=phi_11;
        alpha_1=alpha_11;
        s_12=s_121;

    }

    public def calculo()
    {
        /*

        * funcion directa
```

```

*

* entrada

* phi_1 //latitud 1

* alpha_1 //azimut 1

* s_12 distancia 1 a 2

*

* regreso

* phi_2 //latitud 2

* alpha_2 //azimut 2

* lamda_12 //longitud de 2 respecto a 1

*/

/* ec6 */

//flattening

var f:Double=(A-B)/A;
Console.OUT.println("F " + f);
phi_1=degToRad(phi_1);
var beta_1:Double=Math.atan((1-f)*Math.tan(phi_1));
beta_1=radToDeg(beta_1);
Console.OUT.print("Beta_1 "+beta_1);
Console.OUT.print("\n");

/* ec10 */
alpha_1=degToRad(alpha_1);
beta_1=degToRad(beta_1);
var real_tmp:Double=Math.sqrt( (Math.cos(alpha_1)*Math.cos(alpha_1)) +
((Math.sin(alpha_1)*Math.sin(beta_1))*(Math.sin(alpha_1)*Math.sin(beta_1))) );
var img_tmp:Double=Math.sin(alpha_1)*Math.cos(beta_1);
var alpha_0:Double=ph(real_tmp,img_tmp);
alpha_1=radToDeg(alpha_1);
beta_1=radToDeg(beta_1);
alpha_0=radToDeg(alpha_0);
Console.OUT.print("A0 "+alpha_0);
Console.OUT.print("\n");

/* ec11 */
alpha_1=degToRad(alpha_1);

```

```

beta_1=degToRad(beta_1);
real_tmp=Math.cos(alpha_1)*Math.cos(beta_1);
img_tmp=Math.sin(beta_1);
var sigma_1:Double=ph(real_tmp,img_tmp);
alpha_1=radToDeg(alpha_1);
beta_1=radToDeg(beta_1);
sigma_1=radToDeg(sigma_1);
Console.OUT.print("sigma1 "+sigma_1);
Console.OUT.print("\n");

/* ec12 */
alpha_0=degToRad(alpha_0);
sigma_1=degToRad(sigma_1);
real_tmp=Math.cos(sigma_1);
img_tmp=Math.sin(alpha_0)*Math.sin(sigma_1);
var omega_1:Double=ph(real_tmp,img_tmp);
alpha_0=radToDeg(alpha_0);
sigma_1=radToDeg(sigma_1);
omega_1=radToDeg(omega_1);
Console.OUT.print("omega_1 "+omega_1);
Console.OUT.print("\n");

/* ec9 */
alpha_0=degToRad(alpha_0);
var e_pi:Double=Math.sqrt( (A*A)-(B*B))/(B*B) );
var k:Double=e_pi*Math.cos(alpha_0);
alpha_0=radToDeg(alpha_0);
k=k*k;
Console.OUT.print("k2 "+k);
Console.OUT.print("\n");

/* ec16 */
var epsilon:Double= (Math.sqrt(1+k) -1) / (Math.sqrt(1+k) +1) ;
Console.OUT.print("epsilon "+epsilon);
Console.OUT.print("\n");

/* ec17 */
var A1:Double = (1 + 1/4 * (epsilon*epsilon)
+ 1/64 * (epsilon*epsilon*epsilon*epsilon)
+ 1/256 * (epsilon*epsilon*epsilon*epsilon*epsilon*epsilon)
+ 25/16384 * (epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon)
+ 49/65536 * (epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon) / (1 -
epsilon));

Console.OUT.print("A1 "+A1);
Console.OUT.print("\n");

/* ec15 */

val C1=new Array_1[Double](11);
C1(0)=0;

```

$$\begin{aligned}
C1(1) &= - 1/2 * \text{epsilon} \\
&+ 3/16 * \text{epsilon} \\
&- 1/32 * \text{Math.pow}(\text{epsilon},5) \\
&+ 19/2048 * \text{Math.pow}(\text{epsilon},7) \\
&- 3/4096 * \text{Math.pow}(\text{epsilon},9); \\
C1(2) &= - 1/16 * \text{Math.pow}(\text{epsilon},2) \\
&+ 1/32 * \text{Math.pow}(\text{epsilon},4) \\
&- 9/2048 * \text{Math.pow}(\text{epsilon},6) \\
&+ 7/4096 * \text{Math.pow}(\text{epsilon},8) \\
&+ 1/65536 * \text{Math.pow}(\text{epsilon},10); \\
C1(3) &= - 1/48 * \text{Math.pow}(\text{epsilon},3) \\
&+ 3/256 * \text{Math.pow}(\text{epsilon},5) \\
&- 3/2048 * \text{Math.pow}(\text{epsilon},7) \\
&+ 17/24576 * \text{Math.pow}(\text{epsilon},9); \\
C1(4) &= - 5/512 * \text{Math.pow}(\text{epsilon},4) \\
&+ 3/512 * \text{Math.pow}(\text{epsilon},6) \\
&- 11/16384 * \text{Math.pow}(\text{epsilon},8) \\
&+ 3/8192 * \text{Math.pow}(\text{epsilon},10); \\
C1(5) &= - 7/1280 * \text{Math.pow}(\text{epsilon},5) \\
&+ 7/2048 * \text{Math.pow}(\text{epsilon},7) \\
&- 3/8192 * \text{Math.pow}(\text{epsilon},9); \\
C1(6) &= - 7/2048 * \text{Math.pow}(\text{epsilon},6) \\
&+ 9/4096 * \text{Math.pow}(\text{epsilon},8) \\
&- 117/524288 * \text{Math.pow}(\text{epsilon},10); \\
C1(7) &= - 33/14336 * \text{Math.pow}(\text{epsilon},7) \\
&+ 99/65536 * \text{Math.pow}(\text{epsilon},9);
\end{aligned}$$

```

C1(8) = - 429/262144 * Math.pow(epsilon,8)

+ 143/131072 * Math.pow(epsilon,10);

C1(9) = - 715/589824 * Math.pow(epsilon,9);

C1(10) = - 2431/2621440 * Math.pow(epsilon,10);

var tmp:Double=0;
sigma_1=degToRad(sigma_1);

for(var i:Long=1;i<11;i++)
{
    tmp=tmp + C1(i)*Math.sin(2*i*sigma_1);
}

var l1:Double=A1*(sigma_1+tmp);
sigma_1=radToDeg(sigma_1);
Console.OUT.print("l1 "+l1);
Console.OUT.print("\n");

/* ec7 */

var s1:Double=l1*B;
Console.OUT.print("s1 " +s1);
Console.OUT.print("\n");

/* s1 +s_12 */
var s2:Double= s1+s_12;
Console.OUT.print("s2 "+s2);
Console.OUT.print("\n");

/* s2/(BA1) */
var tau_2:Double= s2/(B*A1);
tau_2=radToDeg(tau_2);
Console.OUT.print("tau_2 "+tau_2);
Console.OUT.print("\n");
tau_2=degToRad(tau_2);

```



```

/* ec20 */
val C1p=new Array_1[Double](11);

C1p(0)=0;

C1p(1) = + (1/2) * epsilon
- (9/32) * Math.pow(epsilon,3)
+ (205/1536) * Math.pow(epsilon,5)
- (4879/73728) * Math.pow(epsilon,7)
+ (9039/327680) * Math.pow(epsilon,9);

C1p(2) = + 5/16 * Math.pow(epsilon,2)
- 37/96 * Math.pow(epsilon,4)
+ 1335/4096 * Math.pow(epsilon,6)
- 86171/368640 * Math.pow(epsilon,8)
+ 4119073/28311552 * Math.pow(epsilon,10);

C1p(3) = + 29/96 * Math.pow(epsilon,3)
- 75/128 * Math.pow(epsilon,5)
+ 2901/4096 * Math.pow(epsilon,7)
- 443327/655360 * Math.pow(epsilon,9);

C1p(4) = + 539/1536 * Math.pow(epsilon,4)
- 2391/2560 * Math.pow(epsilon,6)
+ 1082857/737280 * Math.pow(epsilon,8)
- 2722891/1548288 * Math.pow(epsilon,10);

C1p(5) = + 3467/7680 * Math.pow(epsilon,5)
- 28223/18432 * Math.pow(epsilon,7)
+ 1361343/458752 * Math.pow(epsilon,9);

```

```

C1p(6) = + 38081/61440 * Math.pow(epsilon,6)
- 733437/286720 * Math.pow(epsilon,8)
+ 10820079/1835008 * Math.pow(epsilon,10);
C1p(7) = + 459485/516096 * Math.pow(epsilon,7)
- 709743/163840 * Math.pow(epsilon,9);
C1p(8) = + 109167851/82575360 * Math.pow(epsilon,8)
- 550835669/74317824 * Math.pow(epsilon,10);
C1p(9) = + 83141299/41287680 * Math.pow(epsilon,9);
C1p(10) = + 9303339907/2972712960 * Math.pow(epsilon,10);

```

```

var tmp2:Double=0;
for(var i:Long=1;i<11;i++)
{
    tmp2=tmp2+ (C1p(i)*Math.sin(2*i*tau_2));
}

```

```

var sigma_2:Double=tau_2+tmp2;
sigma_2=radToDeg(sigma_2);
Console.OUT.print("sigma_2 "+sigma_2);
Console.OUT.print("\n");

```

```

/* ec14 */

```

```

var alpha_2:Double;
alpha_0=degToRad(alpha_0);
sigma_2=degToRad(sigma_2);

```

```

alpha_2=ph( Math.cos(alpha_0)*Math.cos(sigma_2) , Math.sin(alpha_0) );
alpha_2=radToDeg(alpha_2);
Console.OUT.print("alpha_2 "+alpha_2);
Console.OUT.print("\n");
alpha_0=radToDeg(alpha_0);
sigma_2=radToDeg(sigma_2);

```

```

/* ec13 */

```

```

var beta_2:Double;
alpha_0=degToRad(alpha_0);
sigma_2=degToRad(sigma_2);

```

```

    beta_2=ph(Math.sqrt ( (Math.cos(alpha_0)*Math.cos(sigma_2))*(Math.cos(alpha_0)*Math.cos(sigma_2))
+ ( Math.sin(alpha_0)*Math.sin(alpha_0) ) ),( Math.cos(alpha_0)*Math.sin(sigma_2) ) );
    beta_2=radToDeg(beta_2);
    Console.OUT.print("beta_2 "+beta_2);
    Console.OUT.print("\n");
    sigma_2=radToDeg(sigma_2);
    alpha_0=radToDeg(alpha_0);

/* ec12 */
var omega_2:Double;
alpha_0=degToRad(alpha_0);
sigma_2=degToRad(sigma_2);
    omega_2=ph( Math.cos(sigma_2), ( Math.sin(alpha_0)*Math.sin(sigma_2) ) );
omega_2=radToDeg(omega_2);
Console.OUT.print("omega_2 "+omega_2);
Console.OUT.print("\n");
    sigma_2=radToDeg(sigma_2);
alpha_0=radToDeg(alpha_0);

/* ec2 */
var n:Double=f/(2-f);
Console.OUT.print("n "+n);
Console.OUT.print("\n");

/* ec24 */
var A3:Double = 1 - (1/2 - (1/2)*n) * epsilon

- (1/4 + (1/8)*n - (3/8)*Math.pow(n,2)) * Math.pow(epsilon,2)

- (1/16 + (3/16)*n + (1/16)*Math.pow(n,2) - (5/16)*Math.pow(n,3)) * Math.pow(epsilon,3)

- (3/64 + (1/32)*n + (5/32)*Math.pow(n,2) + (5/128)*Math.pow(n,3) - (35/128)*Math.pow(n,4)) *
Math.pow(epsilon,4)

- (3/128 + (5/128)*n + (5/256)*Math.pow(n,2) + (35/256)*Math.pow(n,3) + (7/256)*Math.pow(n,4)) *
Math.pow(epsilon,5)

- (5/256 + (15/1024)*n + (35/1024)*Math.pow(n,2) + (7/512)*Math.pow(n,3)) * Math.pow(epsilon,6)

- (25/2048 + (35/2048)*n + (21/2048)*Math.pow(n,2) ) * Math.pow(epsilon,7)

- (175/16384 + (35/4096)*n) * Math.pow(epsilon,8)

- 245/32768 * Math.pow(epsilon,9);

Console.OUT.print("A3 "+A3);
Console.OUT.print("\n");

```

```

/* ec23 */
val C3=new Array_1[Double](10);
C3(0)=0;

C3(1) = + ((1/4) - (1/4)*n) * epsilon
+ ((1/8) - (1/8)*Math.pow(n,2)) * Math.pow(epsilon,2)
+ ((3/64) + (3/64)*n - (1/64)*Math.pow(n,2) - (5/64)*Math.pow(n,3)) * Math.pow(epsilon,3)
+ ((5/128) + (1/64)*n + (1/64)*Math.pow(n,2) - (1/64)*Math.pow(n,3) - (7/128)*Math.pow(n,4)) *
Math.pow(epsilon,4)
+ ((3/128) + (11/512)*n + (3/512)*Math.pow(n,2) + (1/256)*Math.pow(n,3) - (7/512)*Math.pow(n,4)) *
Math.pow(epsilon,5)
+ ((21/1024) + (5/512)*n + (13/1024)*Math.pow(n,2) + (1/512)*Math.pow(n,3)) * Math.pow(epsilon,6)
+ ((243/16384) + (189/16384)*n + (83/16384)*Math.pow(n,2)) * Math.pow(epsilon,7)
+ ((435/32768) + (109/16384)*n) * Math.pow(epsilon,8)
+ (345/32768) * Math.pow(epsilon,9);

C3(2) = + ((1/16) - (3/32)*n + (1/32)*Math.pow(n,2)) * Math.pow(epsilon,2)
+ ((3/64) - (1/32)*n - (3/64)*Math.pow(n,2) + (1/32)*Math.pow(n,3)) * Math.pow(epsilon,3)
+ ((3/128) + (1/128)*n - (9/256)*Math.pow(n,2) - (3/128)*Math.pow(n,3) + (7/256)*Math.pow(n,4)) *
Math.pow(epsilon,4)
+ ((5/256) + (1/256)*n - (1/128)*Math.pow(n,2) - (7/256)*Math.pow(n,3) - (3/256)*Math.pow(n,4)) *
Math.pow(epsilon,5)
+ ((27/2048) + (69/8192)*n - (39/8192)*Math.pow(n,2) - (47/4096)*Math.pow(n,3)) * Math.pow(epsilon,6)
+ ((187/16384) + (39/8192)*n + (31/16384)*Math.pow(n,2)) * Math.pow(epsilon,7)
+ ((287/32768) + (47/8192)*n) * Math.pow(epsilon,8)
+ (255/32768) * Math.pow(epsilon,9);

C3(3) = + ((5/192) - (3/64)*n + (5/192)*Math.pow(n,2) - (1/192)*Math.pow(n,3)) * Math.pow(epsilon,3)
+ ((3/128) - (5/192)*n - (1/64)*Math.pow(n,2) + (5/192)*Math.pow(n,3) - (1/128)*Math.pow(n,4)) *
Math.pow(epsilon,4)

```

$$\begin{aligned}
& + ((7/512) - (1/384)*n - (77/3072)*\text{Math.pow}(n,2) + (5/3072)*\text{Math.pow}(n,3) + (65/3072)*\text{Math.pow}(n,4)) * \\
& \text{Math.pow}(\text{epsilon},5) \\
& + ((3/256) - (1/1024)*n - (71/6144)*\text{Math.pow}(n,2) - (47/3072)*\text{Math.pow}(n,3)) * \text{Math.pow}(\text{epsilon},6) \\
& + ((139/16384) + (143/49152)*n - (383/49152)*\text{Math.pow}(n,2)) * \text{Math.pow}(\text{epsilon},7) \\
& + ((243/32768) + (95/49152)*n) * \text{Math.pow}(\text{epsilon},8) \\
& + (581/98304) * \text{Math.pow}(\text{epsilon},9); \\
& \text{C3}(4) = + ((7/512) - (7/256)*n + (5/256)*\text{Math.pow}(n,2) - (7/1024)*\text{Math.pow}(n,3) + \\
& (1/1024)*\text{Math.pow}(n,4)) * \text{Math.pow}(\text{epsilon},4) \\
& + ((7/512) - (5/256)*n - (7/2048)*\text{Math.pow}(n,2) + (9/512)*\text{Math.pow}(n,3) - (21/2048)*\text{Math.pow}(n,4)) * \\
& \text{Math.pow}(\text{epsilon},5) \\
& + ((9/1024) - (43/8192)*n - (129/8192)*\text{Math.pow}(n,2) + (39/4096)*\text{Math.pow}(n,3)) * \text{Math.pow}(\text{epsilon},6) \\
& + ((127/16384) - (23/8192)*n - (165/16384)*\text{Math.pow}(n,2)) * \text{Math.pow}(\text{epsilon},7) \\
& + ((193/32768) + (3/8192)*n) * \text{Math.pow}(\text{epsilon},8) \\
& + (171/32768) * \text{Math.pow}(\text{epsilon},9); \\
& \text{C3}(5) = + ((21/2560) - (9/512)*n + (15/1024)*\text{Math.pow}(n,2) - (7/1024)*\text{Math.pow}(n,3) + \\
& (9/5120)*\text{Math.pow}(n,4)) * \text{Math.pow}(\text{epsilon},5) \\
& + ((9/1024) - (15/1024)*n + (3/2048)*\text{Math.pow}(n,2) + (57/5120)*\text{Math.pow}(n,3)) * \text{Math.pow}(\text{epsilon},6) \\
& + ((99/16384) - (91/16384)*n - (781/81920)*\text{Math.pow}(n,2)) * \text{Math.pow}(\text{epsilon},7) \\
& + ((179/32768) - (55/16384)*n) * \text{Math.pow}(\text{epsilon},8) \\
& + (141/32768) * \text{Math.pow}(\text{epsilon},9); \\
& \text{C3}(6) = + ((11/2048) - (99/8192)*n + (275/24576)*\text{Math.pow}(n,2) - (77/12288)*\text{Math.pow}(n,3)) * \\
& \text{Math.pow}(\text{epsilon},6) \\
& + ((99/16384) - (275/24576)*n + (55/16384)*\text{Math.pow}(n,2)) * \text{Math.pow}(\text{epsilon},7) \\
& + ((143/32768) - (253/49152)*n) * \text{Math.pow}(\text{epsilon},8) \\
& + (33/8192) * \text{Math.pow}(\text{epsilon},9); \\
& \text{C3}(7) = + ((429/114688) - (143/16384)*n + (143/16384)*\text{Math.pow}(n,2)) * \text{Math.pow}(\text{epsilon},7) \\
& + ((143/32768) - (143/16384)*n) * \text{Math.pow}(\text{epsilon},8) \\
& + (429/131072) * \text{Math.pow}(\text{epsilon},9);
\end{aligned}$$

```

C3(8) = + ((715/262144) - (429/65536)*n) * Math.pow(epsilon,8)

+ (429/131072) * Math.pow(epsilon,9);

C3(9) = + (2431/1179648) * Math.pow(epsilon,9);

var B3:Double=0;
var I3:Double;

sigma_1=degToRad(sigma_1);

for(var i:Long=1;i<10;i++)
{
    B3=B3+ C3(i)*Math.sin(2*i*sigma_1);
}

I3=A3*(sigma_1+B3);
Console.OUT.print("I3 "+I3);
Console.OUT.print("\n");
    sigma12_1=radToDeg(sigma_1);

/* lamda_1 */
alpha_0=degToRad(alpha_0);
var lamda_1:Double=omega_1 - f*Math.sin(alpha_0)*I3;
Console.OUT.print("lamda_1 "+lamda_1);
Console.OUT.print("\n");
alpha_0=radToDeg(alpha_0);

B3=0;

sigma_2=degToRad(sigma_2);
for(var i:Long=1;i<10;i++)
{
    B3=B3+ C3(i)*Math.sin(2*i*sigma_2);
}

I3=A3*(sigma_2+B3);
Console.OUT.print("I3 "+I3);
Console.OUT.print("\n");

/* lamda_2 */
alpha_0=degToRad(alpha_0);
var lamda_2:Double=omega_2 - f*Math.sin(alpha_0)*I3;

```

```

Console.OUT.print("lamda_2 "+lamda_2);
Console.OUT.print("\n");
alpha_0=radToDeg(alpha_0);
sigma_2=radToDeg(sigma_2);

/* lamda_12 */
var lamda_12:Double=lamda_2-lamda_1;
Console.OUT.print("lamda_12 "+lamda_12);
Console.OUT.print("\n");

/* ec6 phi_2 */
var phi_2:Double;
beta_2=degToRad(beta_2);
phi_2= Math.atan( Math.tan(beta_2)/(1-f) );
phi_2=radToDeg(phi_2);
Console.OUT.print("phi_2 "+phi_2);
Console.OUT.print("\n");
beta_2=radToDeg(beta_2);

Console.OUT.print("====SOLUCION====\n");
Console.OUT.println("phi_2 "+phi_2);
Console.OUT.println("lamda_12 "+lamda_12);
Console.OUT.print("alpha_2 "+alpha_2);
Console.OUT.print("\n");
}
}

```

## Anexo 3 Código fuente inverso.x10

```
import x10.io.Console;
import x10.lang.Math;
import x10.array.*;

public class inverso {
    val PI:Double=3.14159265358979323846264338327950288;
    val A:Double=6378137.000000;
    val B:Double=6356752.314245;

    var phi_1:Double;
    var phi_2:Double;
    var lamda_12:Double;

    public def degToRad(grados:Double){
        return ((grados * PI) / 180.00000);
    }

    public def radToDeg(grados:Double){
        return ((grados) * 180.00000 / PI);
    }

    public def ph(real:Double,img:Double)
    {
        return Math.atan2(img,real);
    }

    private def ec55(c:Double, d:Double,e:Double)
    {
        /* ec55 */
        /* método newton raphson */
        /* max 100 iteraciones */

        var inicio:Double=1;
        var tmp:Double=0;
        var tmp2:Double=0;
        var x0:Double;

        var k:Double=0;
        x0=inicio;

        while(k<100)
        {
            tmp=Math.pow(x0,4) + (2*Math.pow(x0,3)) + (c*Math.pow(x0,2)) - (2*d*x0) - e;
            tmp2=(4*Math.pow(x0,3)) + (6* Math.pow(x0,2)) + (2*c*x0) + 2*d;

```



```

    x0=x0 - (tmp/tmp2);

    k++;
}

return x0;
}

public def this(phi_11:Double,phi_21:Double,lamda_121:Double) {
    phi_1=phi_11;
    phi_2=phi_21;
    lamda_12=lamda_121;
}

public def calculo()
{
    /* ec6 */

    //flattening
    var f:Double=(A-B)/A;
    Console.OUT.println("F "+f);
    phi_1=degToRad(phi_1);
    var beta_1:Double=Math.atan((1-f)*Math.tan(phi_1));
    beta_1=radToDeg(beta_1);
    Console.OUT.print("Beta_1 "+beta_1);
    Console.OUT.print("\n");
    phi_2=degToRad(phi_2);
    var beta_2:Double=Math.atan((1-f)*Math.tan(phi_2));
    beta_2=radToDeg(beta_2);
    Console.OUT.println("Beta_2 "+beta_2);
    var e2:Double=f*(2-f);
    Console.OUT.println("e2 "+e2);

    /* ec53 */
    beta_1=degToRad(beta_1);
    var length:Double=f*A*PI*Math.cos(beta_1);
    var x:Double;
    lamda_12=degToRad(lamda_12);
    x=((lamda_12-PI)*A*Math.cos(beta_1))/length ;
    Console.OUT.println("x "+x);
    lamda_12=radToDeg(lamda_12);
    beta_1=radToDeg(beta_1);

    /* ec53 */
    var y:Double;
    beta_2=degToRad(beta_2);
    beta_1=degToRad(beta_1);
    y=(beta_2+beta_1)*(A/length);
    Console.OUT.println("y "+y);
    beta_2=radToDeg(beta_2);
    beta_1=radToDeg(beta_1);
}

```

```

/* ec55 */
var c:Double=1-Math.pow(x,2)-Math.pow(y,2);
var d:Double=Math.pow(y,2);
var e:Double=Math.pow(y,2);
var u:Double=ec55(c,d,e);
Console.OUT.println("u "+u);

/* ec56 */
var alpha_1:Double;
alpha_1=ph( (y/u) , (-x/(1+u)) );
alpha_1=radToDeg(alpha_1);
Console.OUT.println("alpha_1 "+alpha_1);

/* ec10 */
var alpha_0:Double;
alpha_1=degToRad(alpha_1);
beta_1=degToRad(beta_1);
alpha_0=ph( Math.sqrt( (Math.cos(alpha_1)*Math.cos(alpha_1)) +
(Math.sin(alpha_1)*Math.sin(beta_1)*Math.sin(alpha_1)*Math.sin(beta_1))) ,
(Math.sin(alpha_1)*Math.cos(beta_1)) );
alpha_0=radToDeg(alpha_0);
Console.OUT.println("alpha_0 "+alpha_0);
alpha_1=radToDeg(alpha_1);
beta_1=radToDeg(beta_1);

/* ec11 */
var sigma_1:Double;
alpha_1=degToRad(alpha_1);
beta_1=degToRad(beta_1);
sigma_1=ph( Math.cos(alpha_1)*Math.cos(beta_1), Math.sin(beta_1) );
sigma_1=radToDeg(sigma_1);
Console.OUT.println("sigma_1 "+sigma_1);
alpha_1=radToDeg(alpha_1);
beta_1=radToDeg(beta_1);

/* ec12 */
var omega_1:Double;
sigma_1=degToRad(sigma_1);
alpha_0=degToRad(alpha_0);
omega_1=ph(Math.cos(sigma_1),Math.sin(alpha_0)*Math.sin(sigma_1) );
omega_1=radToDeg(omega_1);
Console.OUT.println("omega_1 "+omega_1);
sigma_1=radToDeg(sigma_1);
alpha_0=radToDeg(alpha_0);

/* ec5 */
var alpha_2:Double;
alpha_0=degToRad(alpha_0);
beta_2=degToRad(beta_2);
alpha_2= Math.asin(Math.sin(alpha_0)/Math.cos(beta_2)) ;
alpha_2=radToDeg(alpha_2);
Console.OUT.println("alpha_2 "+alpha_2);
alpha_0=radToDeg(alpha_0);
beta_2=radToDeg(beta_2);

```

```

/* ec11 */
var sigma_2:Double;
alpha_2=degToRad(alpha_2);
beta_2=degToRad(beta_2);
sigma_2=ph( Math.cos(alpha_2)*Math.cos(beta_2),Math.sin(beta_2) );
sigma_2=radToDeg(sigma_2);
Console.OUT.println("sigma_2 "+sigma_2);
alpha_2=radToDeg(alpha_2);
beta_2=radToDeg(beta_2);

```

```

/* ec 12 */
var omega_2:Double;
alpha_0=degToRad(alpha_0);
sigma_2=degToRad(sigma_2);
omega_2=ph(Math.cos(sigma_2), Math.sin(alpha_0)*Math.sin(sigma_2) );
omega_2=radToDeg(omega_2);
Console.OUT.println("omega_2 "+omega_2);
alpha_0=radToDeg(alpha_0);
sigma_2=radToDeg(sigma_2);

```

```

/* ec9 */
alpha_0=degToRad(alpha_0);
var e_pi:Double=Math.sqrt( (( A*A)-( B*B))/( B*B) );
var k:Double=e_pi*Math.cos(alpha_0);
alpha_0=radToDeg(alpha_0);
k=k*k;
Console.OUT.print("k2 "+k);
Console.OUT.print("\n");

```

```

/* ec16 */
var epsilon:Double= (Math.sqrt(1+k) -1) / (Math.sqrt(1+k) +1 ) ;
Console.OUT.print("epsilon "+epsilon);
Console.OUT.print("\n");

```

```

/* ec17 */
var A1:Double = (1 + 1/4 * (epsilon*epsilon)

    + 1/64 * (epsilon*epsilon*epsilon*epsilon)

    + 1/256 * (epsilon*epsilon*epsilon*epsilon*epsilon*epsilon)

    + 25/16384 * (epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon)

    + 49/65536 * (epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon*epsilon) / (1 -
epsilon));

```

```

Console.OUT.print("A1 "+A1);
Console.OUT.print("\n");

```

```

/* ec15 */
val C1=new Array_1[Double](11);

```

$$C1(0)=0;$$

$$C1(1) = - 1/2 * \text{epsilon}$$

$$+ 3/16 * \text{epsilon}$$

$$- 1/32 * \text{Math.pow}(\text{epsilon},5)$$

$$+ 19/2048 * \text{Math.pow}(\text{epsilon},7)$$

$$- 3/4096 * \text{Math.pow}(\text{epsilon},9);$$

$$C1(2) = - 1/16 * \text{Math.pow}(\text{epsilon},2)$$

$$+ 1/32 * \text{Math.pow}(\text{epsilon},4)$$

$$- 9/2048 * \text{Math.pow}(\text{epsilon},6)$$

$$+ 7/4096 * \text{Math.pow}(\text{epsilon},8)$$

$$+ 1/65536 * \text{Math.pow}(\text{epsilon},10);$$

$$C1(3) = - 1/48 * \text{Math.pow}(\text{epsilon},3)$$

$$+ 3/256 * \text{Math.pow}(\text{epsilon},5)$$

$$- 3/2048 * \text{Math.pow}(\text{epsilon},7)$$

$$+ 17/24576 * \text{Math.pow}(\text{epsilon},9);$$

$$C1(4) = - 5/512 * \text{Math.pow}(\text{epsilon},4)$$

$$+ 3/512 * \text{Math.pow}(\text{epsilon},6)$$

$$- 11/16384 * \text{Math.pow}(\text{epsilon},8)$$

$$+ 3/8192 * \text{Math.pow}(\text{epsilon},10);$$

$$C1(5) = - 7/1280 * \text{Math.pow}(\text{epsilon},5)$$

$$+ 7/2048 * \text{Math.pow}(\text{epsilon},7)$$

$$- 3/8192 * \text{Math.pow}(\text{epsilon},9);$$

$$C1(6) = - 7/2048 * \text{Math.pow}(\text{epsilon},6)$$

$$+ 9/4096 * \text{Math.pow}(\text{epsilon},8)$$

$$- 117/524288 * \text{Math.pow}(\text{epsilon},10);$$

$$C1(7) = - 33/14336 * \text{Math.pow}(\text{epsilon},7)$$

```

+ 99/65536 * Math.pow(epsilon,9);

C1(8) = - 429/262144 * Math.pow(epsilon,8)

+ 143/131072 * Math.pow(epsilon,10);

C1(9) = - 715/589824 * Math.pow(epsilon,9);

C1(10) = - 2431/2621440 * Math.pow(epsilon,10);

var tmp:Double=0;
sigma_1=degToRad(sigma_1);

for(var i:Long=1;i<11;i++)
{
    tmp=tmp + C1(i)*Math.sin(2*i*sigma_1);
}

var l1:Double=A1*(sigma_1+tmp);
sigma_1=radToDeg(sigma_1);
Console.OUT.print("l1 "+l1);
Console.OUT.print("\n");

/* ec7 */
var s1:Double=l1*B;
Console.OUT.print("s1 "+s1);
Console.OUT.print("\n");

/* ec7 s2 */
tmp=0;
sigma_2=degToRad(sigma_2);

for(var i:Long=1;i<11;i++)
{
    tmp=tmp + C1(i)*Math.sin(2*i*sigma_2);
}

l1=A1*(sigma_2+tmp);
sigma_2=radToDeg(sigma_2);
Console.OUT.print("l1 "+l1);
Console.OUT.print("\n");

/* ec7 */
var s2:Double=l1*B;
Console.OUT.print("s2 "+s2);
Console.OUT.print("\n");

/* s2 - s1 */
var s_12:Double;
s_12=s2-s1;

```

```
Console.OUT.print("s_12 "+s_12);  
Console.OUT.print("\n");
```

```
Console.OUT.println("====SOLUCION====\n");  
Console.OUT.println("alpha_1 "+alpha_1);  
Console.OUT.println("alpha_2 "+alpha_2);  
Console.OUT.println("s_12 "+s_12);  
Console.OUT.println("\n");
```

```
}
```

```
}
```

## Anexo 4 Código fuente azimuth.x10

```
import x10.io.Console;
import x10.lang.Math;
import x10.array.*;

public class azimuth {

    val PI:Double=3.14159265358979323846264338327950288;
    val A:Double=6378137.000000;
    val B:Double=6378136.6 ; //Radio tierra en metros

    var phi_1:Double; //latitud A
    var phi_2:Double; //latitud B
    var lamda_1:Double; //longitud A
    var lamda_2:Double; //longitud B
    var alpha_1:Double; //azimut
    var s_12:Double;      //distancia de A a B

    public def degToRad(grados:Double){
        return ((grados * PI) / 180.0000000);
    }

    public def radToDeg(grados:Double){
        return ((grados) * 180.0000000 / PI);
    }

    public def this( phi_11:Double,phi_12:Double,lamda_11:Double,lamda_12:Double) {
        phi_1=phi_11;
        phi_2=phi_12;
        lamda_1=lamda_11;
        lamda_2=lamda_12;
    }

    public def calculo()
    {
        Console.OUT.println("AZIMUT");

        phi_1=degToRad(phi_1);
        phi_2=degToRad(phi_2);
        lamda_1=degToRad(lamda_1);
        lamda_2=degToRad(lamda_2);
    }
}
```

```

    var a:Double;
    var c:Double;
    var d:Double;
    a=Math.pow(Math.sin((phi_2-phi_1)/2),2) + Math.cos(phi_1) * Math.cos(phi_2) *
Math.pow(Math.sin( (phi_2-phi_1)/2 ),2) ;
    Console.OUT.println("A:"+a);
    c= 2* Math.atan2(Math.sqrt(a), Math.sqrt(1-a) );
Console.OUT.println("C:"+c);
    d=B*c;

    s_12=d;
    alpha_1=Math.atan2( (Math.sin(phi_2-phi_1) * Math.cos(phi_2) ), (Math.cos(phi_1)*Math.sin(phi_2) -
Math.sin(phi_1)*Math.cos(phi_2)*Math.cos(phi_2-phi_1) ) );
    alpha_1=radToDeg(alpha_1);

    Console.OUT.println("alpha_1:"+alpha_1);
    Console.OUT.println("s_12:"+s_12);

    phi_1=radToDeg(phi_1);
    phi_2=radToDeg(phi_2);
    lamda_1=radToDeg(lamda_1);
    lamda_2=radToDeg(lamda_2);

}

}

```



# Anexo 5 Código fuente de la solución propuesta

## Código Fuente programa de recepción

```
import x10.io.Console;
import x10.lang.Math;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketException;
import java.io.IOException;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.lang.Exception;
import x10.lang.Int;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

import java.lang.Thread;

public class Geo{

public static def leer_coordenadas()
{
    var reader:BufferedReader;
    var coords:String="51.50:0";

    try {
        reader = new BufferedReader(new FileReader(
            "coords"));
        coords = reader.readLine();

        reader.close();
    } catch (exception:IOException){
        Console.OUT.print("Ex2 " + exception.toString() );
    }

    return coords;
}

public static def main(Rail[String]) {
    async{

var sd_phi_2:Double=0.0;
var sd_alpha_2:Double=0.0;
var sd_lamda_12:Double=0.0;
```

```

var si_alpha_1:Double=0.0;
var si_alpha_2:Double=0.0;
var si_s_12:Double=0.0;

var en_phi_2:Double=0.0;
var en_lamda_12:Double=0.0;
var en_phi_1:Double=0.0;
var en_alpha_1:Double=0.0;
var en_s_12:Double=0.0;

while(1==1)
{

val serverPort:Long;
    serverPort=4022;

    try{

        Console.OUT.println(" Server is Running ");
        var mysocket:ServerSocket = new ServerSocket(Int.operator_as(serverPort));

var x:Long=1;
        while(x==1)
        {
            var connectionSocket:Socket = mysocket.accept();

            var reader:BufferedReader =
                new BufferedReader(new InputStreamReader(connectionSocket.getInputStream()));
            var writer:BufferedWriter=
                new BufferedWriter(new OutputStreamWriter(connectionSocket.getOutputStream()));

            writer.write("*** SISTEMA DE NAVEGACIÓN : \n");
            writer.flush();
            var data1:String = reader.readLine().trim();

            Console.OUT.println("ENTRADA "+data1);

            if(data1.equals("1024"))
                x=2;

            // IP LAT LONG
            var partes:Rail[String] = data1.split(":");
            Console.OUT.println("ENTRADAS "+partes.size);

            connectionSocket.close();

            if(partes.size==3)

```

```

        {

                var inicio:Long = System.nanoTime();

                finish{

                async

                {

var origen:String=partes(0);
var latitud:String=partes(1);
var longitud:String=partes(2);

try{
    var phi_1:Double; //latitud A
    var phi_2:Double; //latitud B
    var lamda_1:Double; //longitud A
    var lamda_2:Double; //longitud B

    phi_2=Double.parse(latitud);
    lamda_2=Double.parse(longitud);

    //leer coordenadas locales
    var coords:Rail[String] = leer_coordenadas().split(":");
    phi_1=Double.parse(coords(0));
    lamda_1=Double.parse(coords(1));

    Console.OUT.println("ENTRADAS phi_1:"+phi_1+" lamda_1:"+lamda_1+ " phi_2:"+phi_2+"
lamda_2:"+lamda_2);

    val az=new azimut(phi_1,phi_2,lamda_1,lamda_2);
    finish
    {
    az.calculo();
    }

    Console.OUT.print("alpha_1 "+ az.alpha_1);
    Console.OUT.println("s_12 "+ az.s_12);

    var alpha_1:Double=30; //azimut
    var s_12:Double=10000000;

    alpha_1=az.alpha_1;
    s_12=az.s_12;

    var lamda_12:Double;
    lamda_12=lamda_2-lamda_1;

```

```

        val prueba=new directo(phi_1,alpha_1,s_12);
        val prueba3=new inverso(phi_1,phi_2,lamda_12);

        finish
        {
        prueba.calculo();
            prueba3.calculo();
        }

sd_phi_2=prueba.phi_2;
sd_alpha_2=prueba.alpha_2;
sd_lamda_12=prueba.lamda_12;

si_alpha_1=prueba3.alpha_1;
si_alpha_2=prueba3.alpha_2;
si_s_12=prueba3.s_12;

en_phi_2=phi_2;
en_lamda_12=lamda_12;
en_phi_1=phi_1;
en_alpha_1=alpha_1;
en_s_12=s_12;

    }
    catch(exception:NumberFormatException){
        Console.OUT.print("Ex3 "+ exception.toString());
    }

} //fin async

} //fin finish

var tiempo:Long = System.nanoTime() - inicio;

Console.OUT.println("Tiempo ms:"+tiempo/ 1000000.0);
var medicion:String=partes(0)+":"+ tiempo/1000000.0 +"\n";

        //abrir archivo para guardar mediciones
        var escritor:BufferedWriter;
        escritor = new BufferedWriter(new
FileWriter("mediciones",true));

        escritor.write(medicion);
        escritor.close();

var lecturas:String=en_phi_1+":"+ en_alpha_1 + ":" + en_s_12 + ":" +sd_phi_2 + ":" + sd_alpha_2 + ":" +
sd_lamda_12 + ":" + en_phi_1 + ":" + en_phi_2 + ":" + en_lamda_12 + ":" + si_alpha_1 + ":" + si_alpha_2 + ":" +
si_s_12 ;

        var escritor2:BufferedWriter;

```

```

        escritor2 = new BufferedWriter(new FileWriter("salida_coordenadas",true));
        escritor2.write(lecturas);
        escritor2.close();

    } //fin if

//fin try
}

}
catch(exception:SocketException){
    Console.OUT.print("Ex1 "+ exception.toString() );
}
catch(exception:IOException){
    Console.OUT.print("Ex2 " + exception.toString() );
}

catch(exception:NullPointerException){
    Console.OUT.print("Ex3 " + exception.toString() );
}

}

} //fin while

}

}
}

```

### Código Fuente programa de envío

```

import x10.io.Console;
import x10.lang.Math;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketException;
import java.io.IOException;
import java.io.BufferedReader;
import java.io.BufferedWriter;

```

```
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.lang.Exception;
import java.net.InetAddress;
import x10.lang.Int;
import java.io.FileReader;
import java.io.IOException;
import java.lang.StringBuilder;
import com.fazecast.jSerialComm.SerialPort;

public class Cliente {
    public static def leer_coordenadas()
    {
        var reader:BufferedReader;
        var coords:String="51.50:0";
        try {
            reader = new BufferedReader(new FileReader("coords"));
            coords = reader.readLine();
            reader.close();
        } catch (exception:IOException){
            Console.OUT.print("Ex2 " + exception.toString() );
        }
        return coords;
    }

    public static def leer_ips()
    {
```

```

var reader:BufferedReader;
var ips:String="127.0.0.1:";
var tmp2:String="";
try {
    reader = new BufferedReader(new FileReader("ips"));
while( (tmp2=reader.readLine()) != null )
    ips =ips+ tmp2+ ":";
    reader.close();
} catch (exception:IOException){
}
return ips;
}

public static def main(args: Rail[String]) {
    var ips:Rail[String] = leer_ips().split(":");
    var coords:Rail[String] = leer_coordenadas().split(":");
    var phi_1:Double; //latitud A
    var lamda_1:Double; //longitud A
    finish{
    async{
    for(var i:Long=0;i<ips.size;i++)
    {
        phi_1=Double.parse(coords(0));
        lamda_1=Double.parse(coords(1));
    val ip:String = InetAddress.getLocalHost().getHostName();
    val hostname:String = ips(i);
    val port:Long = 4022;
    try {
        var socket:Socket = new Socket(hostname, Int.operator_as(port));
        var reader:BufferedReader =
            new BufferedReader(new InputStreamReader(socket.getInputStream()));
        var data1:String = reader.readLine().trim();

```

```

        Console.OUT.println("ENTRADA "+data1);
        var writer:BufferedWriter=new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()));
        Console.OUT.println(ip+": "+phi_1+": "+lamda_1+"\n");
        writer.write(ip+": "+phi_1+": "+lamda_1+"\n");
        writer.flush();
    }
    catch(exception:SocketException){
        Console.OUT.print("Ex1 "+ exception.toString() );
    }
    catch(exception:IOException){
        Console.OUT.print("Ex2 " + exception.toString() );
    }
}
} //fin for
} // fin async
} //fin finish
}
}

```

## Código Fuente programa de lectura del GPS

```

import x10.io.Console;
import x10.lang.Math;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketException;
import java.io.IOException;
import java.io.BufferedWriter;
import java.io.InputStream;
import java.io.OutputStreamWriter;
import java.lang.Exception;
import x10.lang.Int;
import java.io.FileWriter;
import java.io.IOException;

```



```

import java.lang.StringBuilder;
import com.fazecast.jSerialComm.SerialPort;

public class GPS {

    public static def main(args: Rail[String]) {
        var coords:String="51.50:0";

        try
        {
            var comPort:SerialPort = SerialPort.getCommPort("/dev/ttyS0");

            comPort.setBaudRate(Int.operator_as(9600));
            comPort.setNumDataBits(Int.operator_as(8));
            comPort.setNumStopBits(SerialPort.ONE_STOP_BIT);
            comPort.setParity(SerialPort.NO_PARITY);

            comPort.openPort();
while(1==1)
{
            Console.OUT.println("Abierto:"+comPort.isOpen());
            comPort.setComPortTimeouts(SerialPort.TIMEOUT_READ_BLOCKING, Int.operator_as(0),
Int.operator_as(0));
            if(comPort.isOpen())
            {
                var inp:InputStream=comPort.getInputStream();
                var cadena:String="";

                //LEER GPGLL

                for(var i:Long=0;i<250;i++)
                {
                    cadena=cadena+Char.operator_as(Byte.operator_as(inp.read()));
                }

```

```

//Console.OUT.println("CADENA:" +cadena);

var lineas:Rail[String] = cadena.split("$");

for(var i:Long=0;i<lineas.size;i++)
{
    //Console.OUT.println("LINEA:"+lineas(i));
    var linea :Rail[String] =lineas(i).split(",");

    try{
        if(linea(0).equals("GPGLL"))
        {
            Console.OUT.println("Guardada!");
            if( linea(6).equals("A") )
            {
                //coords=linea(1)+":"+linea(3);
//dd + mm.mmmm/60 for latitude if S -
//ddd + mm.mmmm/60 for longitude if W -
//latitud conversion grados decimales

var tmp:Rail[String]=linea(1).split(".");
    var dd1:Double=Double.parse(tmp(0));
        if(dd1>=100)dd1=dd1/100;
var mm1:Double=Double.parse(linea(1))- (dd1 * 100);
var ftmp1:Double= dd1 + mm1/100;

if( linea(2).equals("S") )ftmp1=ftmp1*-1;

//longitud conversion grados decimales
var tmp2:Rail[String]=linea(3).split(".");
var dd2:Double=Double.parse(tmp2(0));

```

```

if(dd2>=100)dd2=dd2/100;
var mm2:Double=Double.parse(linea(3))- (dd2 * 100);
var ftmp2:Double= dd2 + mm2/100;

if( linea(4).equals("W") )ftmp2=ftmp2*-1;
        coords=ftmp1+"."+ftmp2;
}

        try {
                var escritor:BufferedWriter;
                escritor = new BufferedWriter(new FileWriter("coords"));
                escritor.write(coords);
                escritor.close();
        } catch (exception:IOException){
        Console.OUT.print("Error al escribir: " + exception.toString() );
        }

        }

    }

    catch (exception:ArrayIndexOutOfBoundsException){
        Console.OUT.print("Ex3 " + exception.toString() );
    }

    }//fin for
} //open port
} //fin loop
} //fin try
    catch (exception:IOException){
        Console.OUT.print("Ex3 " + exception.toString() );
    }
    }
    catch(exception:NullPointerException){
        Console.OUT.print("Ex Serial2 " + exception.toString() );
    }

```

```
}  
}  
}
```