

# LogiMap Template Documentación



## Introducción

LogiMap Template provee una forma de crear rutas y paradas partiendo de un gran numero y dispersion de paquetes a entregar en diferentes direcciones.

El template es muy técnico y necesita de una buena comprensión del NetMaps y sus clases. Como complemento puede acceder a la documentación del mismo <https://www.capesoft.com/docs/NetTalk11/NetTalkMaps.Htm>

LogiMap permite fácilmente:

---

- Cargar direcciones y tamaño/peso de los paquetes a entregar
  - Crear y memorizer areas de distribución
  - Modificar esas areas base según la distribución de domicilios
  - Detectar automáticamente los domicilios pertenecientes a cada Area
  - Detectar direcciones huérfanas, que quedan fuera de la Areas
  - Crear Ruta a vuelo de pájaro con una estimación rápida de tiempo y kilometros
  - Crear Rutas por carreteras y secuenciar automáticamente el mejor orden
  - Optimizar esas rutas según el estado del transito
  - Controlar volumenes y pesos máximos a cargar
  - Y muchas funciones mas para ayudarlo a programar su sistema de logística
- 

## Instalación

Se provee un instalador, correrlo eligiendo el directorio de Clarion donde instalarlo. Templates necesarios para NetMaps:

- ☑ Activate CapeSoft jFiles - Version:3.03
- ☑ Activate CapeSoft NetTalk - Version:12.53
- ☑ Activate CapeSoft Reflection - Version:1.23
- ☑ Activate CapeSoft StringTheory - Version:3.61
- ☑ Activate CapeSoft WinEvent - Version:5.36
- ☑ Activate Clarion Freemage for this Application

Ingresa a Clarion y agregue al Registry el LogiMaps template.

Se proveen ejemplos en ClarionXX/accosory/Armi/Logimaps para Clarion10 y 11 ABC.

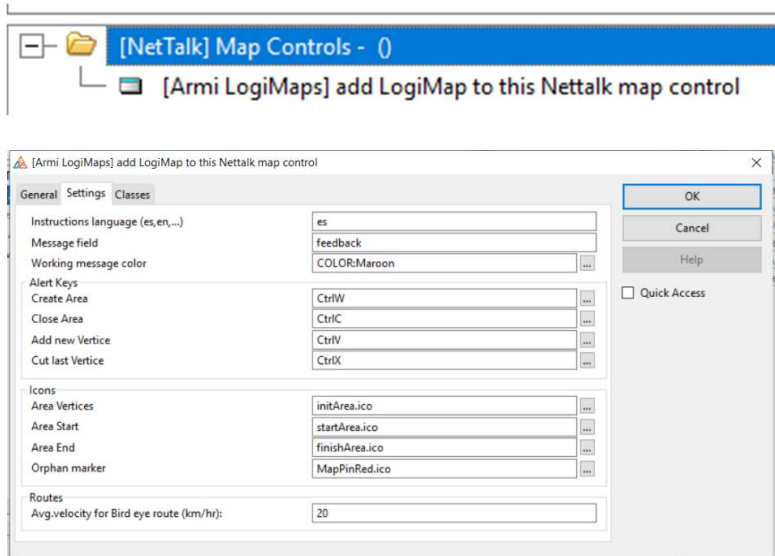
El idioma se define mediante equates, los mismos están definidos en el archivo ArmiLogiMaps.trn que se encuentra en ClarionXX/LibSrc/win, si necesita inglés comente las líneas en español y descomente las líneas en ingles.

Compile, si todo esta correcto, de click en el botón "Markers" y ya va a ver la primer aproximación a LogiMap viendo unas Areas ya dibujadas sobre el mapa.

---

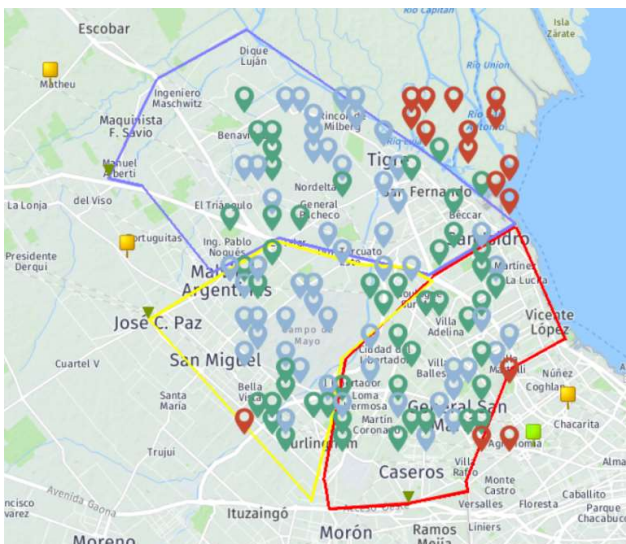
## Seteos del template

En el demo esto esta todo hecho pero si necesita agregar el template a una app nueva, agregue la extensión global a la app y luego, en la ventana en la que tiene el objeto NetMaps, señalelo y oprima Insert para colgar el LogiMap template del mismo eligiéndolo de la lista como cualquier otro template de Clarion. Quedará algo así:

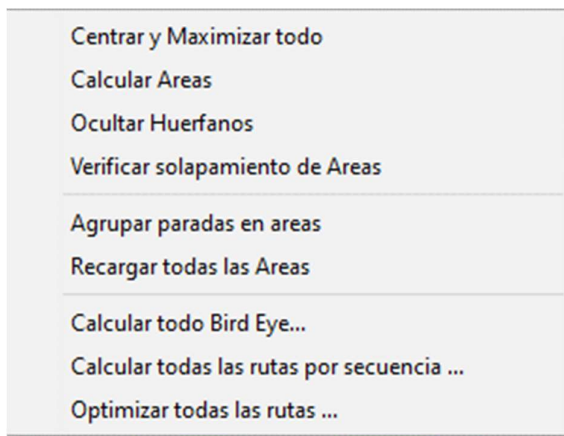


Puede modificar las teclas de alerta, los iconos, etc. Los iconos sugeridos se proveen en el directorio imágenes del ejemplo, simplemente cópielos al directorio de su app. Para crear un Area, pulse "Ctrl-W", el cursor se convertirá en una cruz, mueva la cruz hacia donde desea comenzar el área y pulse "Ctrl-V" para crear en punto de inicio, siga moviendo el cursor a los demás vértices y creandolos con "Ctrl-V", para finalizar pulse "Ctrl-C" y automáticamente se cerrara el área. Si se equivoca con alguno puede pulsar "Ctrl-X" para borrarlo. Pulse ahora el botón "Random Markers" que se encuentra arriba a la derecha, este botón simula en forma aleatoria la creación de sus putos de entrega. Estudiando su código tendrá una idea de como agregar sus entregas al mapa.

Suponiendo la siguiente distribución aleatoria de puntos:



Presionando ahora botón derecho del mouse sobre el mapa obtendrá las siguientes funciones:



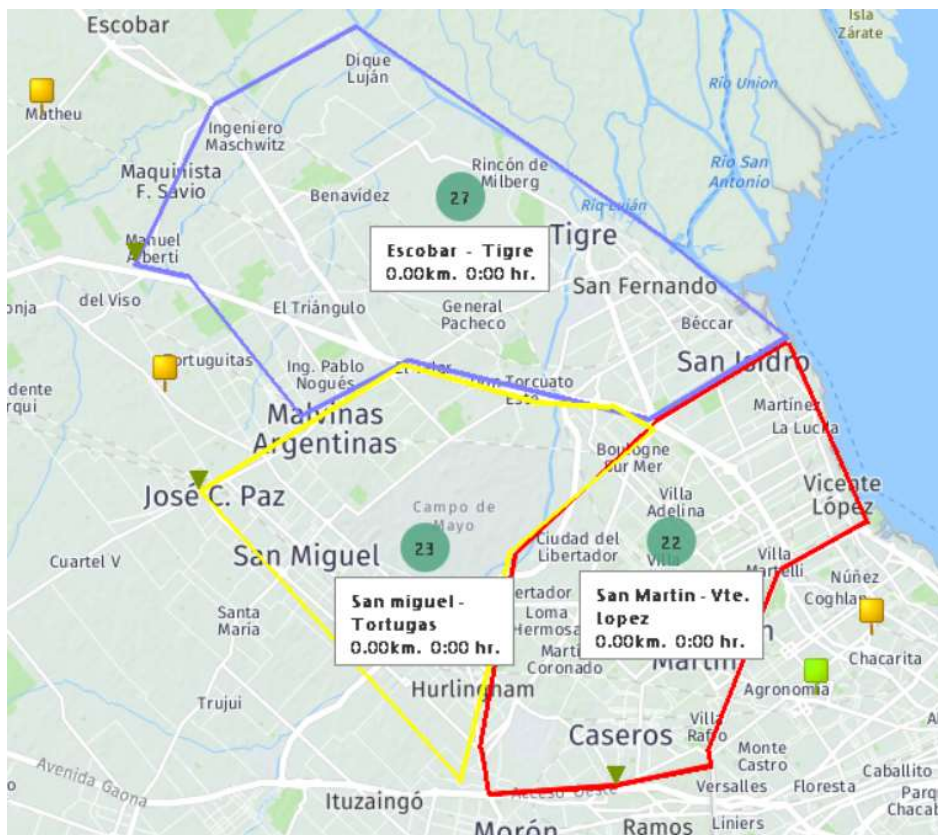
**Centrar y maximizar todo:** Centra el mapa recalculando el zoom para mostrar todos los objetos sobre el mismo.

**Calcular Areas:** Detecta que punto de entrega pertenece a cada área, así como los puntos huérfanos (que quedan fuera de área) y los pinta de rojo.

**Ocultar Huerfanos:** Oculta los puntos huérfanos

**Verificar solapamiento de Areas:** verifica que ningún área este dibujada ocupando el lugar de otra, e indica en que vertice se encuentra el problema para solucionarlo.

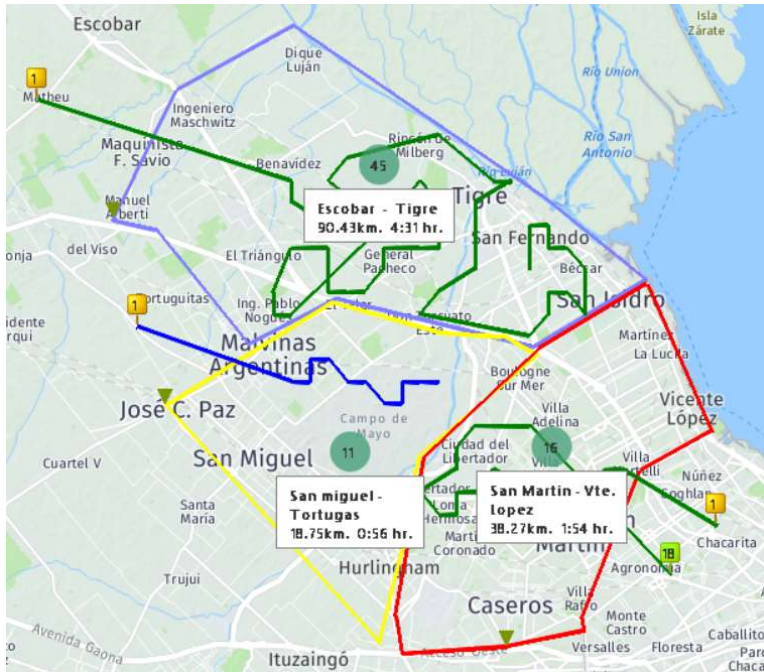
**Agrupar parada en Areas:** Borra los puntos e indica en un circulo central la cantidad de puntos dentro de cada área.



**Recargar todas la Areas:** vuelve a mostrar los puntos.



**Calcular todo Bird Eye** (a vuelo de pájaro): Calcula las rutas en cada área como si fuese volando sin tener en cuenta las calles. Si ahora presionada Agrupar paradas en áreas tendría un pantallazo general de paradas, kilómetros y tiempos en cada área.

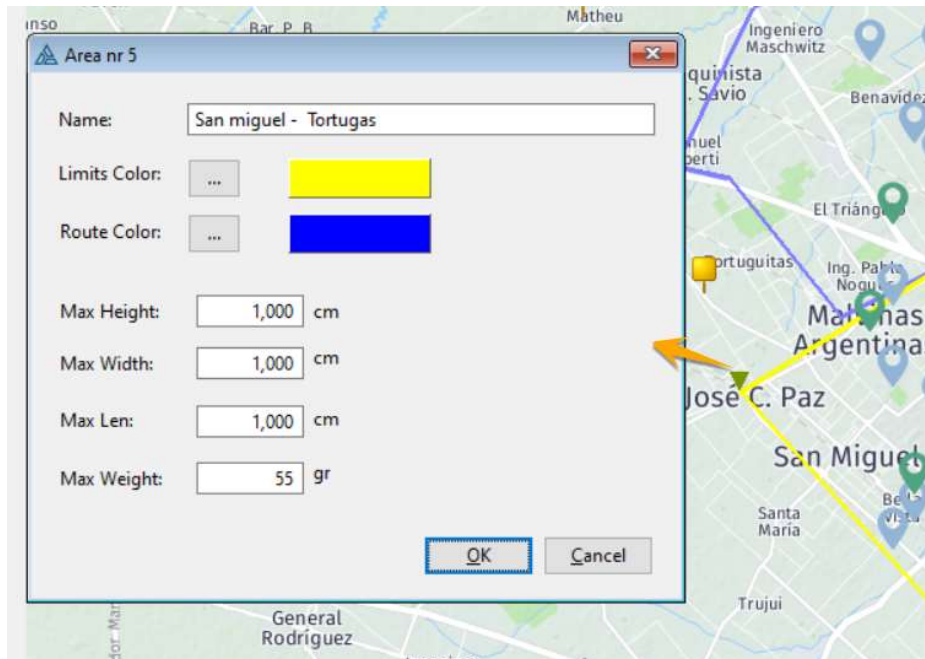


**Calcular todas las rutas por secuencia:** Calcula las rutas yendo por caminos utilizando la secuencia de paradas indicada por Ud al cargar las paradas.

**Optimizar todas las rutas:** Calcula las rutas por caminos pero optimizando el recorrido creando una nueva secuencia de paradas.

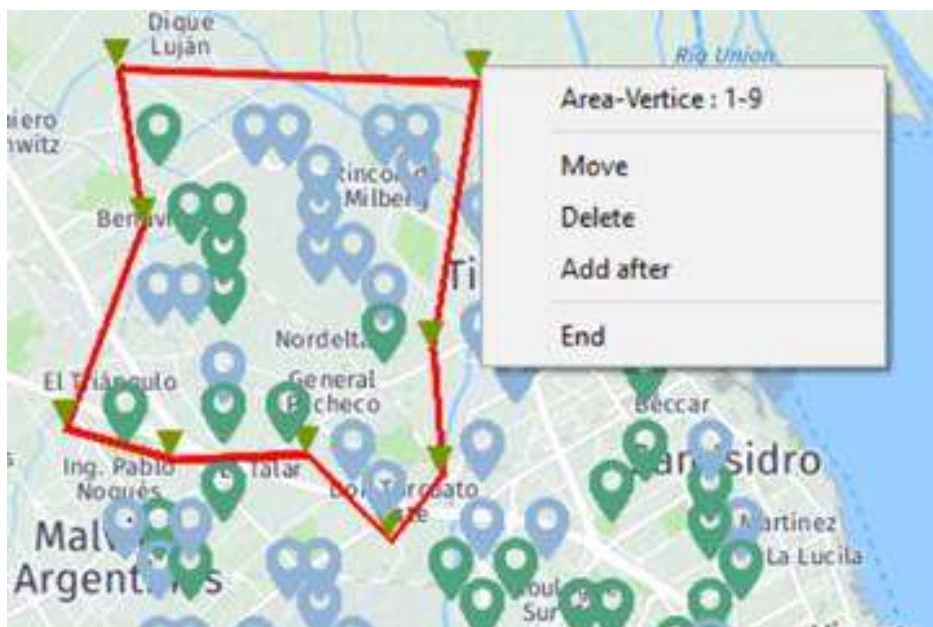


Cada Area posee los datos máximos de capacidad del móvil que se asignará a la misma para realizar la entrega, pudiendo calcular los máximos según dimensiones y peso de la suma de los paquetes a entregar, pulsando en el icono de cada area:

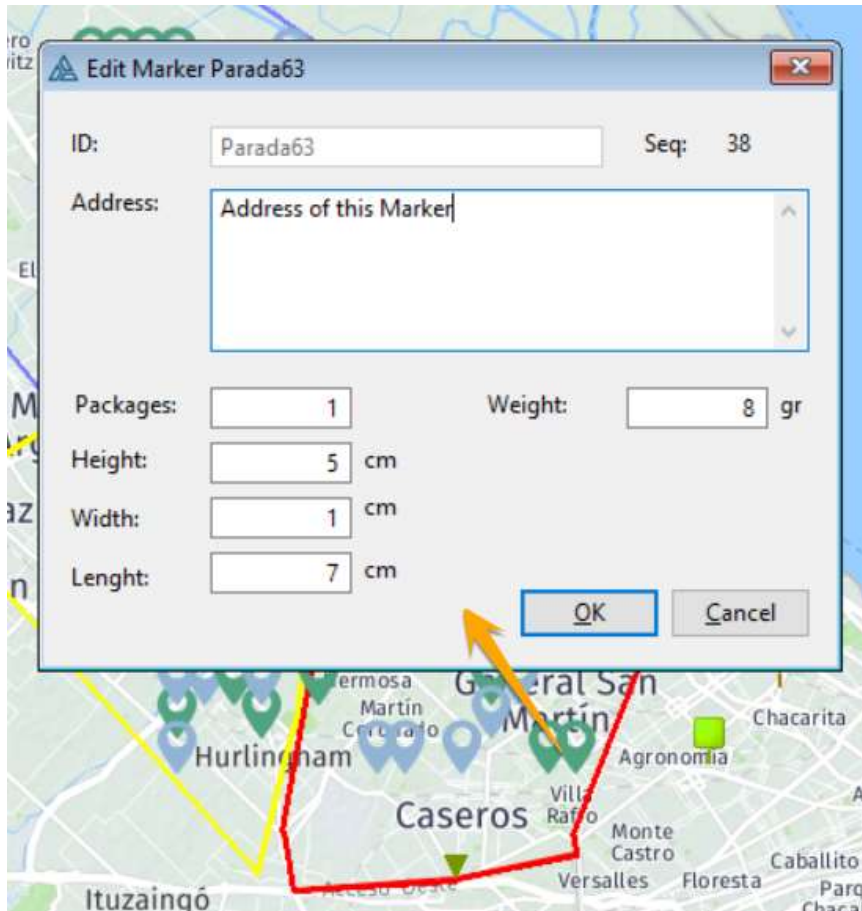


Podra modificar y adaptar cada área para incluir o no huérfanos, pudiendo mover o borrar cualquiera de sus vértices.

Para modificar el área de botón derecho sobre el triángulo verde que indica el inicio del área, eso hará aparecer el mismo icono en cada vertice del área y podrá, usando también el botón derecho sobre cada vertice, moverlo, borrarlo, etc



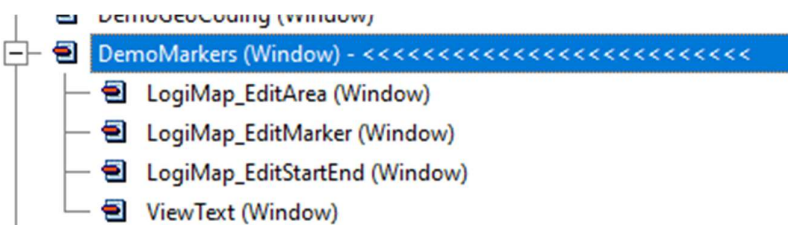
Cada direccion puede ser investigada y modificada:



Finalmente toda la información se maneja en colas de memoria de fácil acceso para ser grabadas o trabajadas en su sistema de logística.

Presionando el botón "Inspect", arriba a la derecha podrá ver cada queue generado por el template de donde obtener información para llevar a su sistema de logística o gestión.

En la instalación, el template agrega unas ventanas con las que se hace la edición.



El template se provee con código fuente.

# Clase, métodos y parámetros

Todo se entrega con código fuente así que Ud dispone de .inc y .clw para investigar la clase, mostramos ahora lo mas sobresaliente.

## Queues, definición:

Areas, soporta hasta 64 vertices cada una.

```
AreasQueueType      Queue, TYPE
Nr                   long
Name                 string(101)
Color                long
GUID                 string(16)
Vertices             long, dim(64)
VerticesCount        long
Stops                long
Distance             LONG
Time                 LONG
VelocityAvg          LONG
RouteColor           LONG
HasStart             LONG
HasEnd               LONG
                    End
```

Vértices:

```
VerticesQueueType   QUEUE, TYPE
AreaNr               long
Nr                   long
Latitud              REAL
Longitud              REAL
                    END
```

Puntos de Inicio y final de cada Area:

```
StartEndQueueType   QUEUE, TYPE
AreaNr               long
Type                 string(10)
Name                 string(255)
Latitud              REAL
Longitud              REAL
                    END
```

Puntos de la ruta del area:

```
StopsQueueType      QUEUE, TYPE
AreaNr               long
MarkerID             string(255)
Latitud              REAL
Longitud              REAL
Sequence             long
Distance2Start        long
Distance2previous     LONG
DistanceRunning        long
Time2next              long
TimeRunning           long
                    END
```



Instrucciones de la ruta:

```
InstructionsQueueType    QUEUE,TYPE
AreaNr                  LONG
LegNumber               LONG
Time                   LONG
Distance               LONG
InstructionText         STRING(1024)
END
```

AreasQueue	Definición de cada Area
StartEndQueue	Cada punto de Start y End de cada Area
StopsQueue	Paradas de cada ruta de cada area
OrphansQueue	Puntos huérfanos
RouteQueue	(NetTalk) Rutas calculadas
WaypointsQueue	(NetTalk) Cada punto de entrega
ManeuverQueue	(NetTalk) Puntos intermedios de cada ruta
InstructionsQueue	(NetTalk) Instrucciones coloquiales de que hacer

## Principales Métodos y parámetros:

**DeleteAllAreas**      **procedure**(long pRefresh=1)  
Borra todas la Areas, sus vértices y la Ruta, ejemplo:  
**DeleteAllAreas**(0) borra sin refrescar la ventana

**SetArea**      **procedure**(long pNr, string pName, <long pColor>, <long pRouteColor>, <string pGuid>)  
Crea un Area, con su nombre y color, GUID puede usarse para indicar el GUID o el ID de su sistema de gestión, ejemplo:  
**SetArea**(1, 'North West', 16744448, 32896)

**SetAreaVertice**      **procedure**(long a, long v, real pLatitude, real pLongitude)  
Crea un vertice de un Area, ejemplo:  
**SetAreaVertice**(1, 1, -33.8851739726444, 18.5310778623994)

**SetStartEnd**      **procedure**(long pAreaNr, string pType, real pLatitude, real pLongitude)  
Crea un punto de comienzo o final de ruta de un Area, el pType puede ser solo 'Start' o 'End' , los iconos de start y end se definen con los siguientes parámetro:  
**StartIcon**      **string**(255)  
**EndIcon**      **string**(255)  
ejemplo:  
**SetStartEnd**(1, 'Start', -33.8638843, 18.5103867)

**DrawAllAreas**      **procedure**(long pRefresh=1)  
Redibuja todas las areas, ejemplo:  
**DrawAllAreas**(0)

**SetDelivery**      **Procedure**(String pId, Real pLatitude, Real pLongitude, String pIcon, long pSequence, long pPackages=0, long pHeight=0, long pWidth=0, long pLength=0, long pWeight=0, Long pIconIndex=1, Long pPointX=16, Long pPointY = 32, Long pHide=false )  
Crea un punto de entrega, este metodo basicamente hace lo mismo que el **SetMarker()** del **NetTalk** pero puede ademas mantener datos adicionales del punto que no estan previstos en **NetTalk**, ejemplo:  
**SetDelivery**('Parada'&m#, 'Dirección de la parada'&m#, Lat\$, Lon\$,  
**choose**(int(m#/2)=m#/2, 'MapPinLightBlue.ico', 'MapPinGreen.ico'), 0, 1, **random**(1, 9), **random**(1, 9), **random**(1, 9), **random**(1, 9))



## Ejemplo de creado de Areas por código:

```
lm.DeleteAllAreas (0)
lm.HideOrphans = 0
lm.ShowOnlyTotalMarkers = 0

lm.SetArea (1, 'North West', 16744448, 32896)
lm.SetAreaVertice (1, 1, -33.8851739726444, 18.5310778623994)
lm.SetAreaVertice (1, 2, -33.8868842364711, 18.524205641905)
lm.SetAreaVertice (1, 3, -33.9139634137285, 18.4949987048042)
lm.SetAreaVertice (1, 4, -33.9179540293243, 18.4781617645932)
lm.SetAreaVertice (1, 5, -33.9213745569779, 18.4383028857262)
lm.SetAreaVertice (1, 6, -33.9236549087469, 18.4321178872813)
lm.SetAreaVertice (1, 7, -33.9310660519963, 18.4369284416273)
lm.SetAreaVertice (1, 8, -33.9484537342352, 18.4654481566787)
lm.SetAreaVertice (1, 9, -33.9447481626106, 18.4840031520133)
lm.SetAreaVertice (1, 10, -33.9527293938022, 18.4960295378784)
lm.SetAreaVertice (1, 11, -33.9507340860043, 18.5163025883366)
lm.SetAreaVertice (1, 12, -33.9561499214558, 18.5403553600667)
lm.SetAreaVertice (1, 13, -33.9267903924294, 18.5455095254374)
lm.SetAreaVertice (1, 14, -33.9125381938729, 18.5410425821161)
lm.SetStartEnd (1, 'Start', -33.8638843, 18.5103867, 'startArea.ico')

lm.DeleteAllRoutes ()
lm.DrawAllAreas (0)
lm.MapMaximizeAndCenter (0)
lm.FindAllMarkers ()
lm.IdentifyOrphanMarkers (1)
```

## Ejemplo de creación de puntos de entrega random:

```
Latitude = -34.5
Longitude = -58.6
Margen = 10
loop m#=1 to random(50,50)
  Lat$ = Latitude+random(-10,10)/100
  Lon$ = Longitude+random(-Margen,Margen)/100
  net.MarkerQueue.Latitude = Lat$
  net.MarkerQueue.Longitude = Lon$

  get(net.MarkerQueue,net.MarkerQueue.Latitude,net.MarkerQueue.Longitude)
  if error ()
    lm.SetDelivery ('Parada' & m#, 'Dirección de la parada
' & m#, Lat$, Lon$, choose (int (m#/2)=m#/2, 'MapPinLightBlue.ico', 'Map-
PinGreen.ico'), 0, 1, random (1, 9), random (1, 9), random (1, 9), random (1, 9))
    !(String pId, String pDescription, Real pLatitude, Real pLongitude, String
pIcon, long pSequence, long pPackages=0, long pHeight=0, long pWidth=0, long
pLength=0, long pWeight=0, Long pIconIndex=1, Long pPointX=16, Long pPointY = 32,
Long pHide=false )
  END
END
```

Fin del documento

---