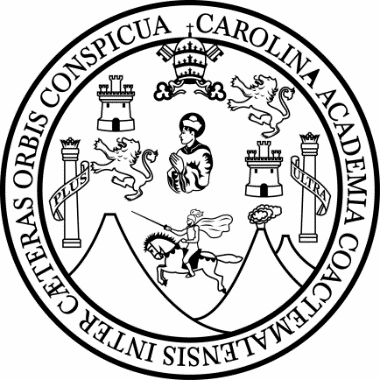
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

MODELACIÓN Y SIMULACIÓN 1

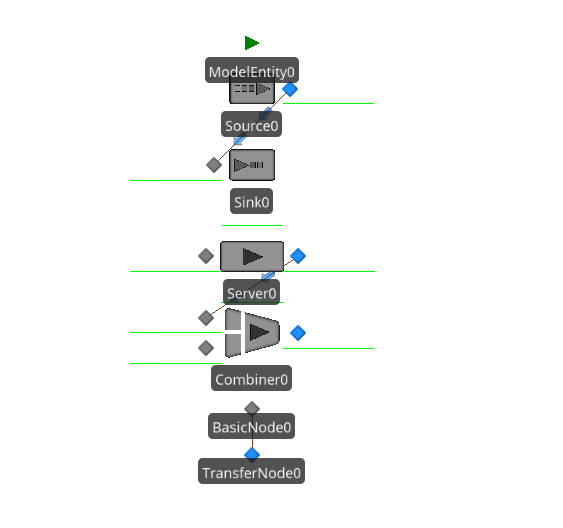
PRACTICA 3

PAREJA NO. 21

|  |  |
| --- | --- |
| Nombre | Carne |
| Aldair Estrada García | 201503855 |
| Edgar Rolando Herrera Rivas | 201520498 |

GUATEMALA 18 DE OCTUBRE DE 2020

# Justificación del Modelo Base



Para el modelo base es necesario colocar el cada uno de los componentes que se utilizaran en al API para el desarrollo del modelo final.

Componentes para utilizar

* ModelEntity : Este componente representa la entidad del modelo.
* Source: Es la salida de las entidades en un determinado tiempo
* Sink: Es el componente que destruye las entidades cuando ya cumplen su función en el modelo
* Server: Este componente se utilizará para demostrar las estaciones de información para cada región
* BasicNode y TransferNode: Estos componentes representaran algunas conexiones del mapa y los camiones de cada turista en diferentes regiones del país.
* Path y Conveyor: Representa las conexiones entre diferentes puntos del país,

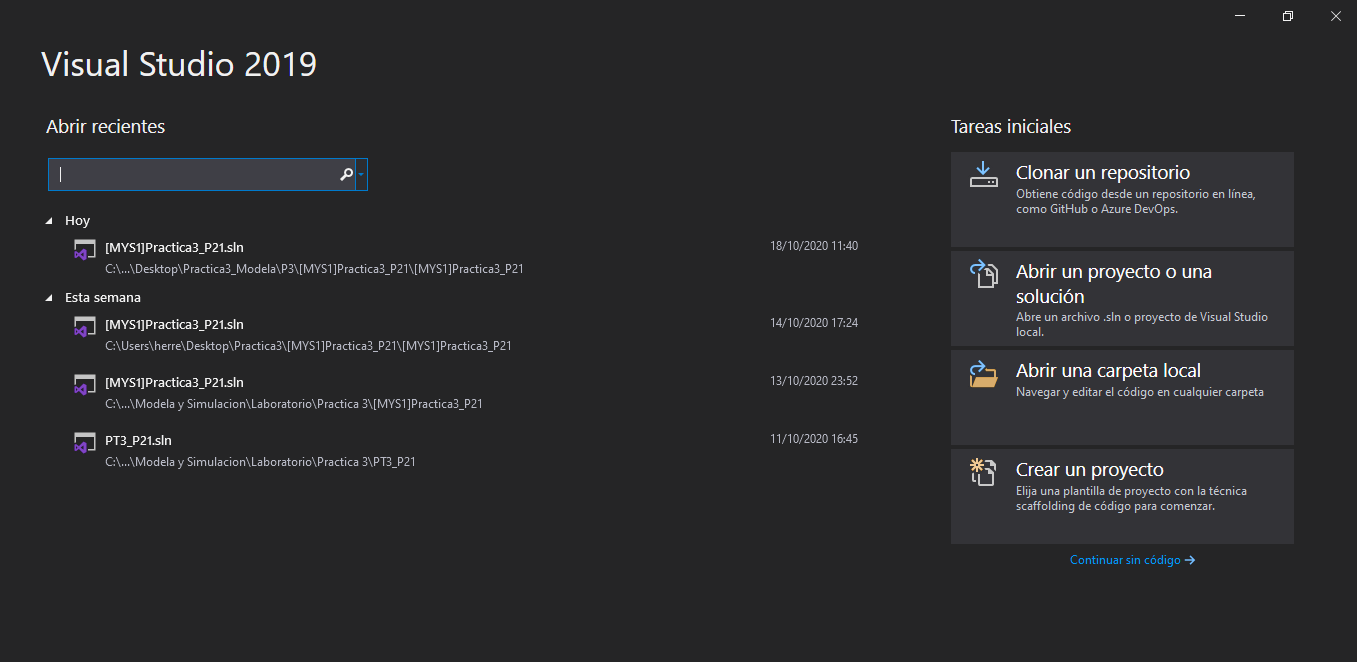
# Pasos para crear una Api en visual studio (C#) con simio.

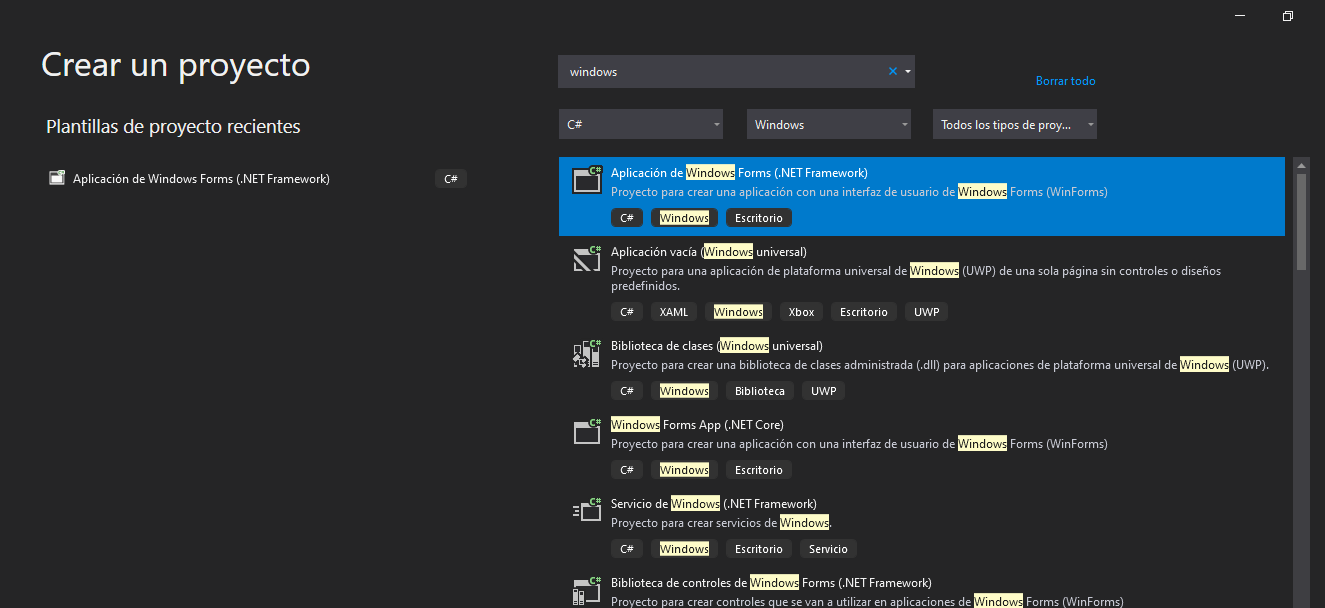
## Paso 1.

En este paso es necesario tener instalado visual estudio de preferencia versión 2019 ya que funciona mejor el programa de simio en este editor.

## Paso 2.

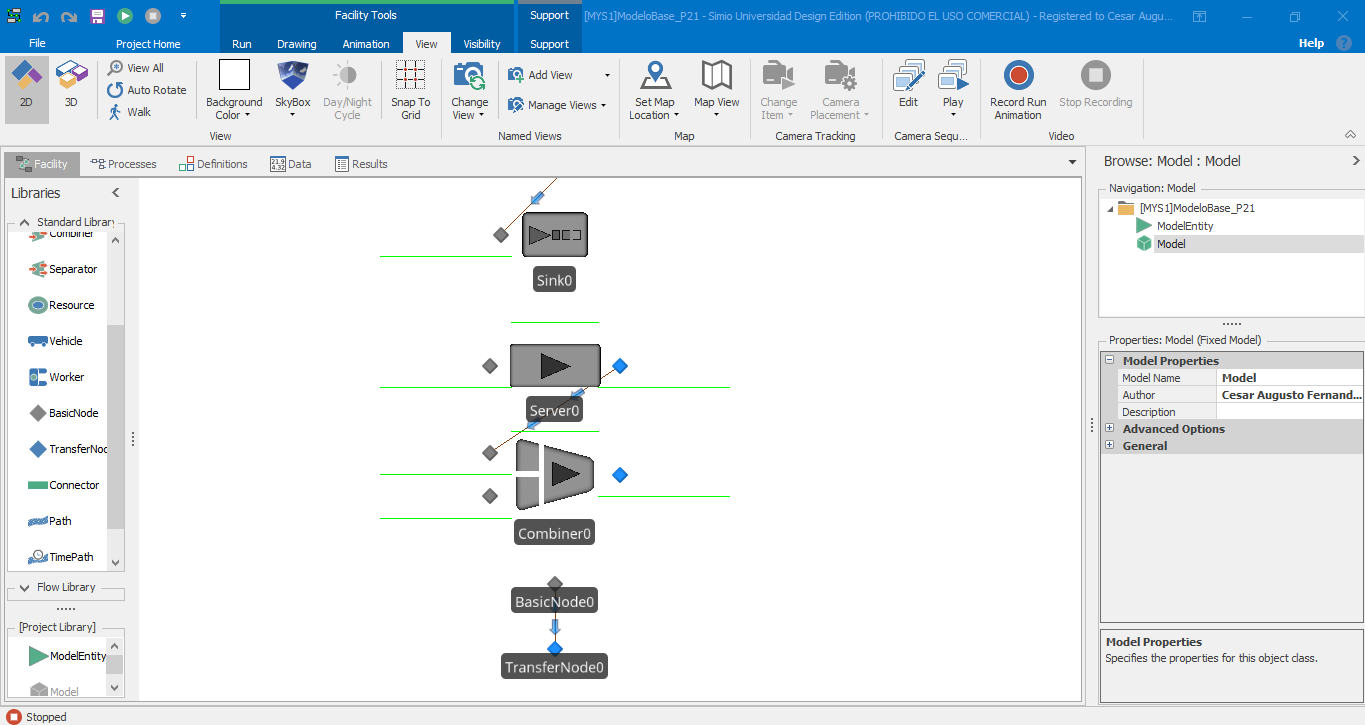
Se abre visual studio 2019 y se crea un nuevo programa y se elige la opción Aplicación de Windows Form.





## Paso 3

Se crea un modelo base que se necesitara para crear sobre el el modelo final. En este modelo se colocará todos los componentes que se utilizaran para llegar al modelo final.



## Paso 4

Se incorporan a visual studio 2019 los componentes que necesita para poder genera una api que pueda crear un documento con extensión de simio el cual tendrá nuestro modelo final. Estas librerías están en la carpeta donde esta instalado simio. Luego las colocamos el la carpeta debug de nuestro proyecto de visual

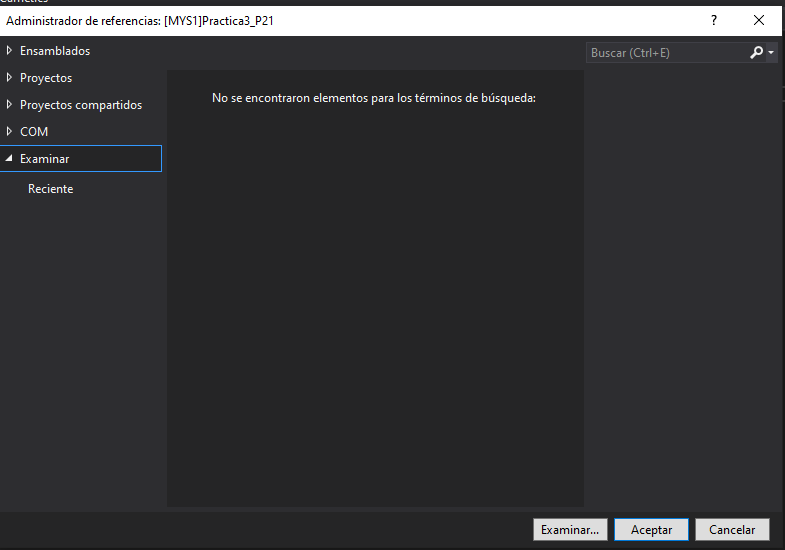


## Paso 5

En visual studio se agregan los componentes y librerías que se acaban de colocar en la carpeta debug de nuestro proyecto.

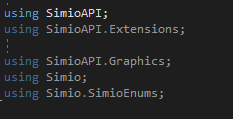
### Paso 5.1

Se agrega una nueva referencia y seleccionamos los componentes y librerías que están en nuestra carpeta debug



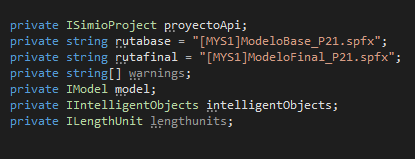
## Paso 6

Se agregan las librerías a nuestra clase de C#



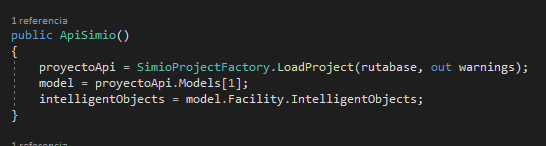
## Paso 7

Se manda a llamar las librerías a utilizar y las que nos ayudaran a genera nuestro modelo final.



## Paso 8

Acá es donde creamos un método para poder llamarlo en cualquier parte de nuestro código



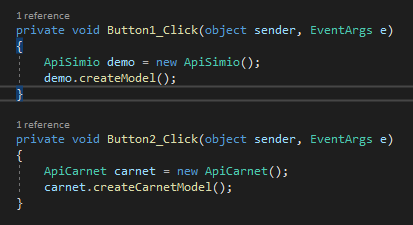
## Paso 9

Ya se puede programar, ya que todo lo que se necesita para crear la API ya esta completamente instalado y configurado para poderlo utilizar en el código.

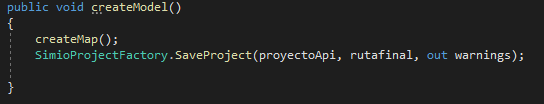
# Descripción de métodos o funciones mas importantes utilizados en la implementación de la API

## Button1\_Click y Button2\_Click

Estos métodos son los que ejecuta el botón para la creación del modelo final y la creación del modelo de carnet respectivamente.

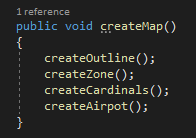


## CreateModel

Este método nos sirve para mandar a llamar el método de createMap, además nos sirve para guardar la versión final del proyecto.

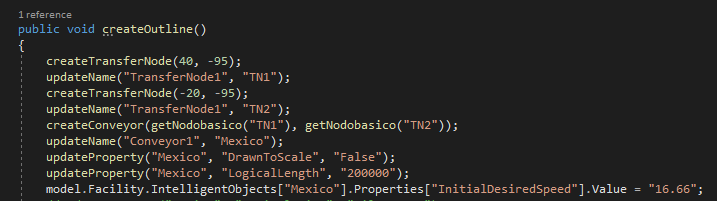
## createMap

Este método nos sirve para mandar a llamar el método de creación de contorno, zonas, puntos cardinales y rutas.



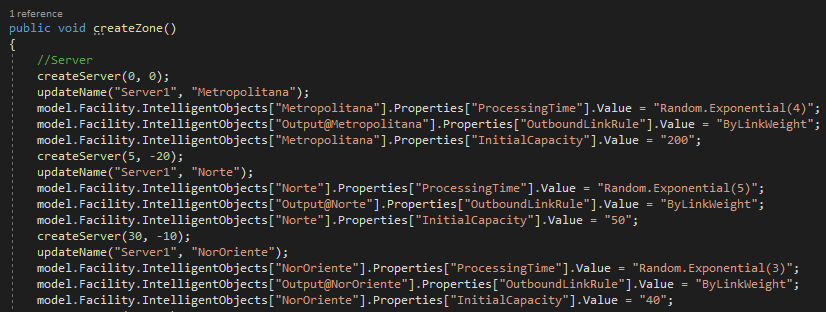
## createOutline

Es método nos sirve para crear el contorno de mapa o fronteras, utilizando transfernode, conveyor y la modificación de propiedades y características de los elementos.

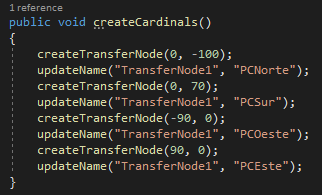


## createZone

Este método se utiliza para crear los servers de las regiones descritas en el enunciado modificandoles también su tasa de servicio.

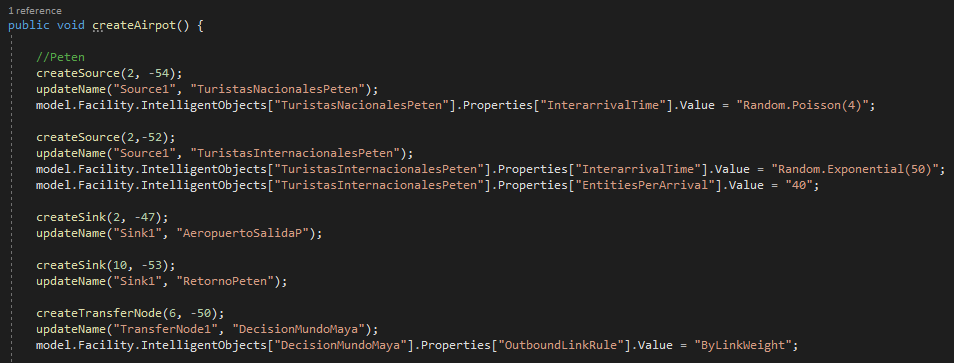


## createCardinals

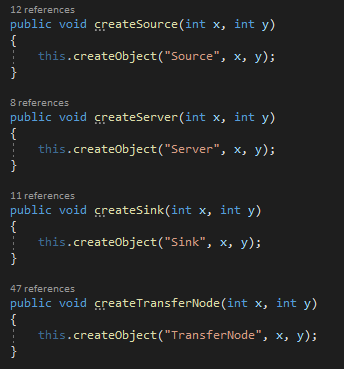
Este método sirve para crear los puntos cardinales norte, sur, este y oeste.

## createAirpot

Este método es el utilizado para crear toda la parte interna del mapa, los source, los sink, las rutas utilizando transfernode y conveyor. Aquí también se realiza la modificación de sus propiedades.

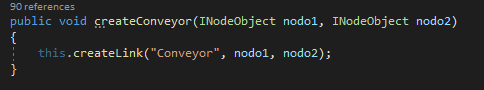


## createSource, createServer, createSink y createTransferNode

Estos métodos nos sirven para llamar el método para crear objetos, enviándole como parámetros el tipo de objeto, su posición en X y su posicion en Y.

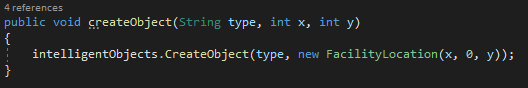
## createConveyor

Este método nos sirve para crear los tipos de enlace entre los diferentes componentes del modelo. Llama al método de crear enlace mandándole los parámetros de tipo de enlace, nodo1 y nodo2.

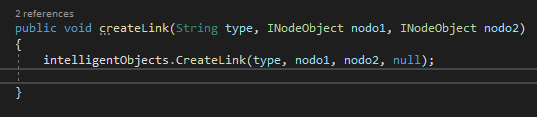


## createObject

Este método recibe la llamada de los métodos descritos anteriormente con sus parámetros y utiliza la función CreateObject de intelligentObjects para crear el componente solicitado.

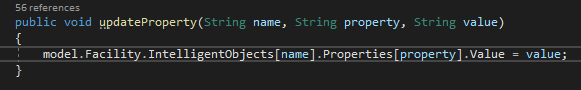


## createLink

Este método recibe la llamada de los métodos descritos anteriormente con sus parámetros y utiliza la función CreateLink de intelligentObjects para crear el enlace solicitado.

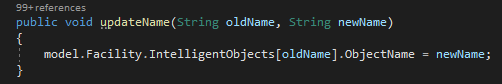
## updateProperty

Este método nos sirve para modificar las propiedades de los diferentes componentes utilizados en el modelo. Recibe como parámetros el nombre del componente o enlace, la propiedad a modificar y el nuevo valor a setear.

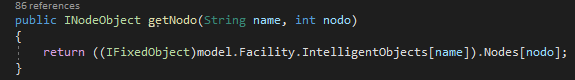


## updateName

Este método es parecido al anterior solo que este únicamente sirve para modificar el nombre del componente o enlace a cambiar. Recibe los parámetros de nombre antiguo y nombre nuevo.

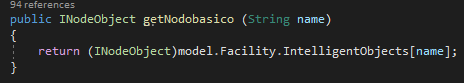


## getNodo

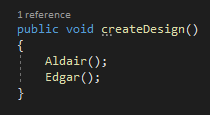
Este método sirve para obtener el nodo de un componente que servirá de enlace hacia otro componente o transfernode.

## getNodobasico

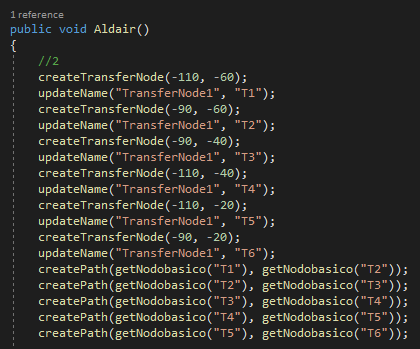
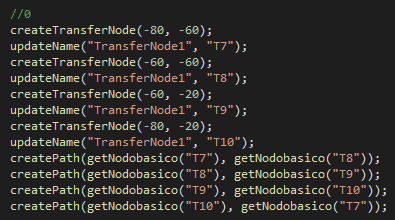
Este método sirve para obtener el nodo de un transfernode que servirá de enlace con otro componente o transfernode.



## createDesign

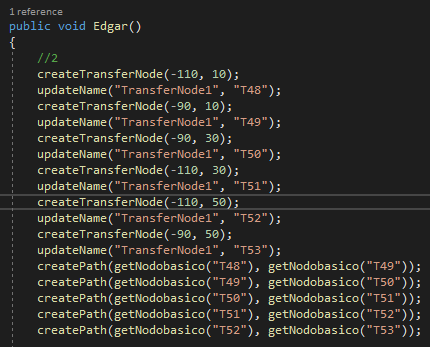
Este método nos sirve para mandar a llamar los métodos que dibujan los carnet en el modelo.

## Aldair

Este método sirve para crear el carné de Aldair.

## Edgar

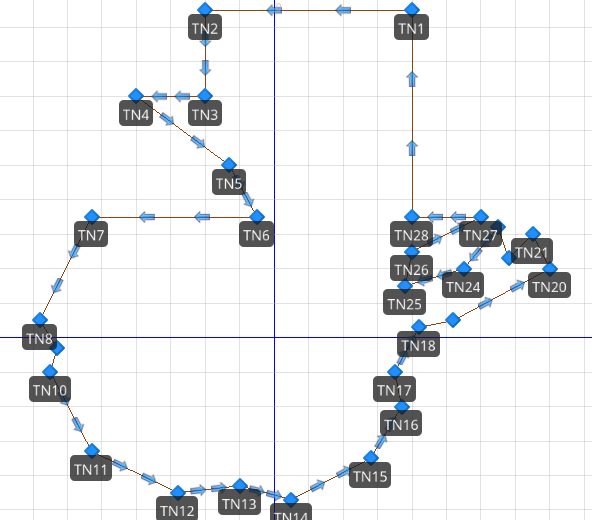
Este método sirve para crear el carné de Edgar.



# Capturas de pantalla con la evolución del modelo final, describiendo cada una de las evoluciones

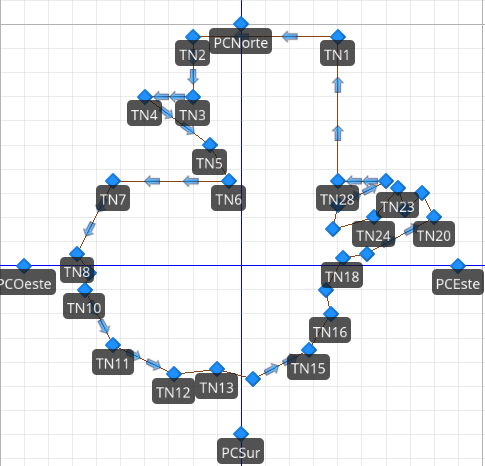
## Contorno

Esta parte se creó el contorno de mapa, lo que son las fronteras, este contorno fue hecho con 28 transfernode y 27 conveyor.



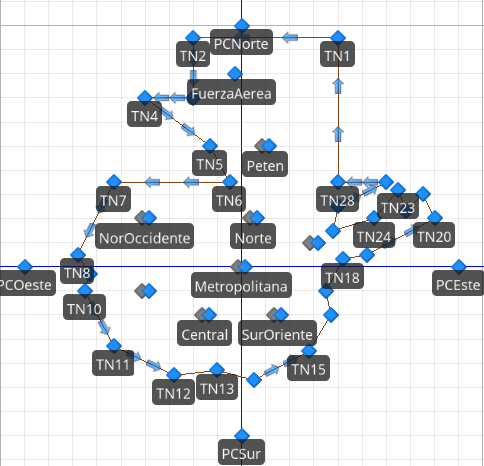
## Puntos Cardinales

Se realizaron los respectivos puntos cardinales norte, sur, este y oeste utilizando transfernode.

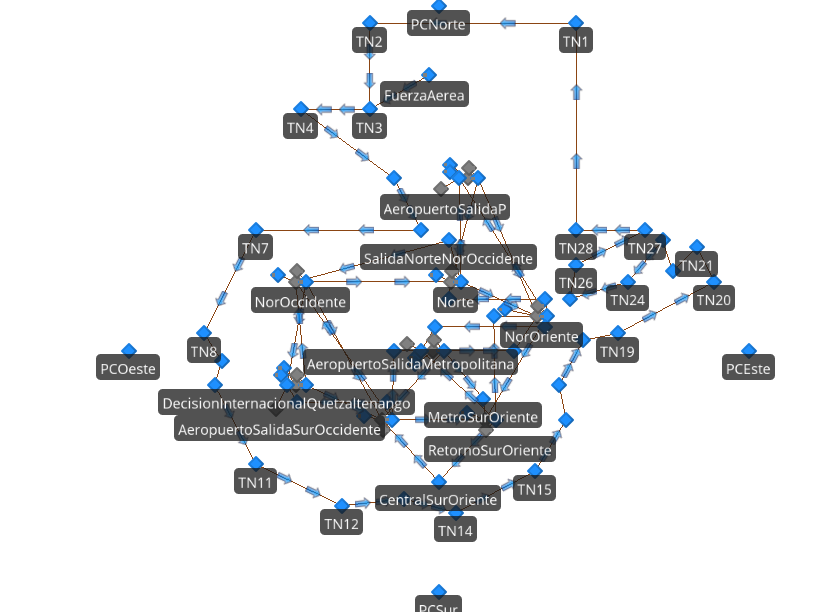


## Creación de Regiones

Aquí se crean las distintas regiones que componen el mapa de Guatemala, se utilizaron servers para representar sus puntos exactos. También se creo la fuerza armada en Peten.



## Modelo Final

En esta parte ya se terminan de agregar los diferentes source de las regiones, los aeropuertos, los sinks y las respectivas rutas que comunican a las diferentes regiones.

## Modelo Final Carné

En este modelo se muestran directamente los carnés de los integrantes.

