

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
MODELACIÓN Y SIMULACIÓN 1



PROYECTO ÚNICO

PAREJA NO. 21

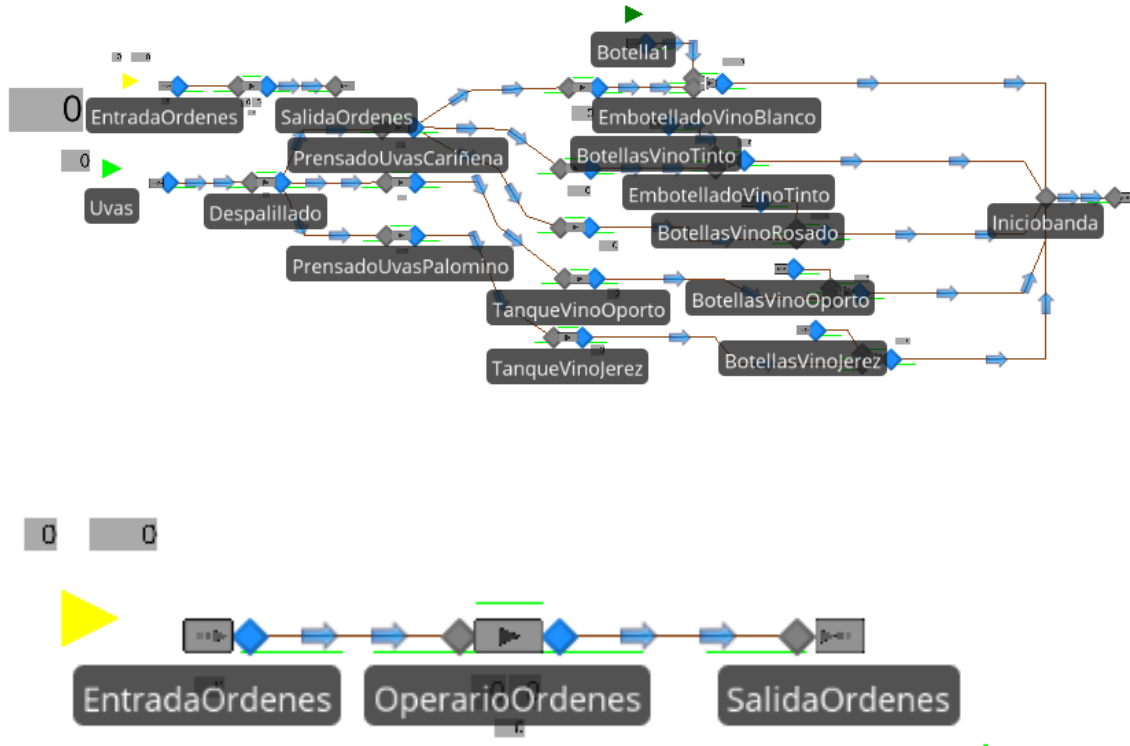
SEGUNDO SEMESTRE 2020

Nombre	Carne
Aldair Estrada García	201503855
Edgar Rolando Herrera Rivas	201520498

09/11/2020

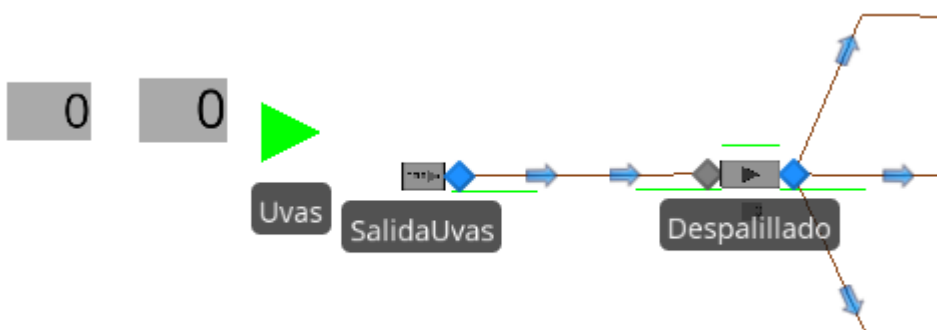
Elaboración de Vinos

Justificación del Modelo



Esta parte del modelo es donde se leerá el archivo Excel, se tomaran las ordenes para después ejecutar todo el proceso de elaboración de vino.

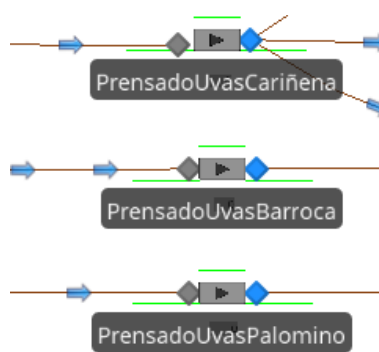
Parte del Depalillado de Uvas



En esta parte se crea un proceso en el cual conforme se va tomado las ordenes de cuantas uvas necesita el cliente para hacer la cantidad de vino necesaria.

Luego para por una despalillada que consiste en quitarle toda la semillas o polillos que tenga la uva para dejar solo la uva limpia y lista para empezar la selección de vinos.

Prensado de Uvas según tipo de vino



Proceso que consiste en deshacer la uva para sacarle el jugo.

Tanques de Vino

Estos servidores lo que realizan es el proceso de fermentación y depende del tipo de vino realiza una segunda acción para el caso de Vino Blanco (Filtración) y para vino Jerez (Encabezado)

Properties: TanqueVinoJerez (Server)

Process Logic	
Capacity Type	Fixed
Initial Capacity	3333
Ranking Rule	First In First Out
Dynamic Selecti...	None
Transfer-In ...	0.0
Process Type	Task Sequence
Processing T...	2 Rows
Loopback Br...	0 Rows
Task Preced...	Sequence Number Met...
Task Resour...	
Task Material...	
Task State A...	
Off Shift Rule	Suspend Processing
+ Other Processing Options	
Buffer Logic	

Processing Tasks - Repeating Property Editor

Items:

10, Fragmentacion, Always, Process Name, Fragmentacion, All Immediate Pre...
20, Enbellezido, Always, Specific Time, Random Uniform(5 , 10), Minimum, A...

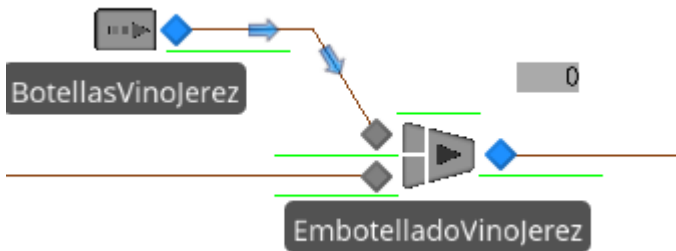
Process Properties:

Task Information	
Sequence Number	10
Name	Fragmentacion
Branch Type	Always
Process Type	Process Name
Process Name	Fragmentacion
Auto Cancel Trigger	All Immediate Predecessors Cancelled
Resource Requirements	
Material Requirements	
State Assignments	
Add-On Process Triggers	

Task Information
Task Information

Add Delete Close

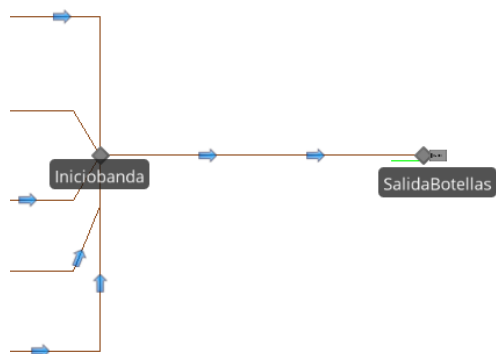
Embotellado de Vinos



Este proceso es cuando ya pasa el proceso de fermentación y esta lista para llenar las botellas, Existen 4 tipos de Botellas (Estándar, Magnum, Doble Magnum, Imperial).

Procesos para todos los tipos de vino

Cola de botellas y Salida de Botellas

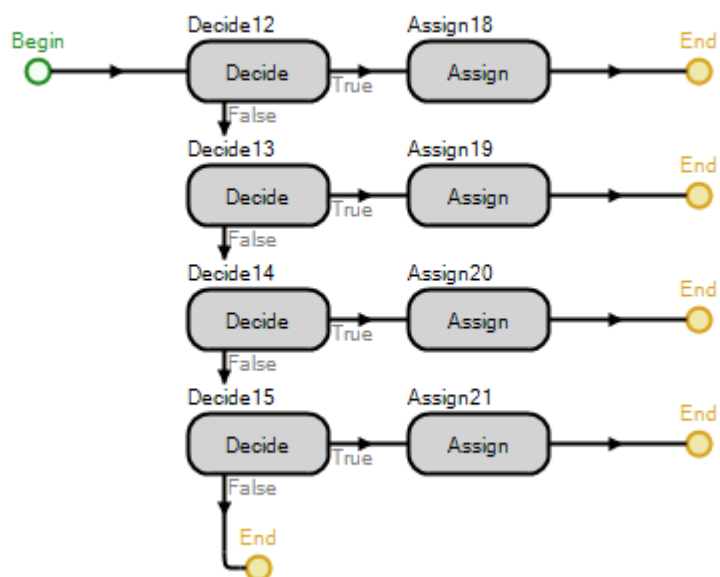


Procesos

Cambio de botella

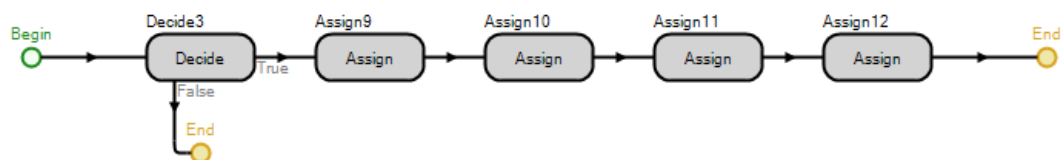
Proceso que realiza el cambio del tamaño de botellas

CambiosBotella



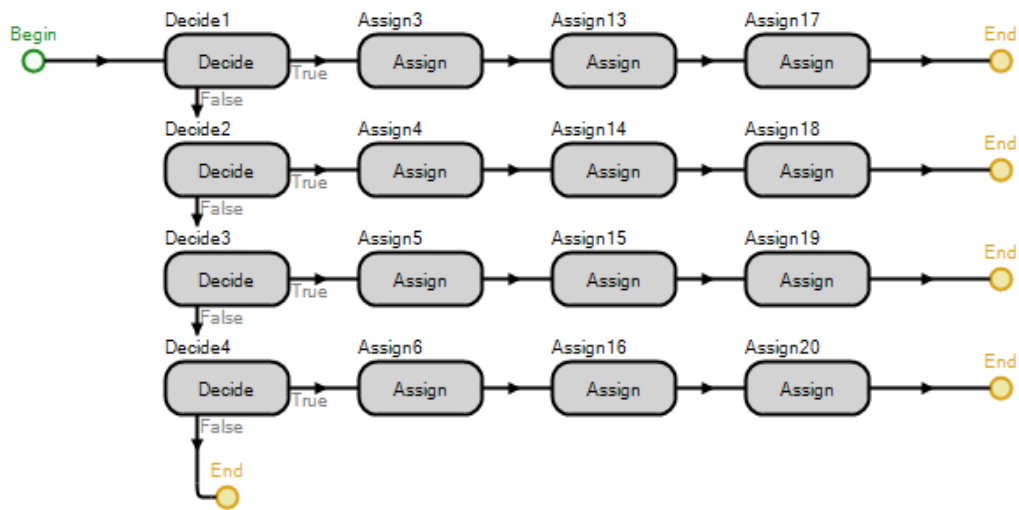
Contador de Entidades de Uva para cada tipo de botella

CantidadDeBotellasParaCadaTipo



Contador de Botellas según tipo de vino

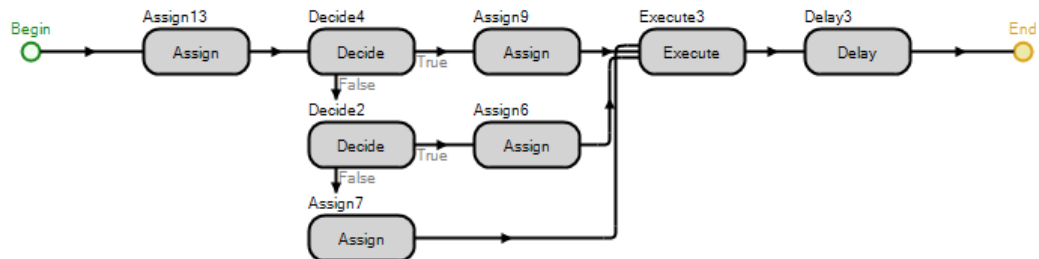
ContadorBotellasVinoBlanco



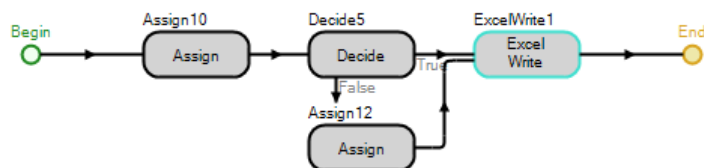
Proceso realizado para los 5 tipos de vino

Determinar el tipo de Uvas y Escritura de archivo Excel con el reporte

DeterminarUvas

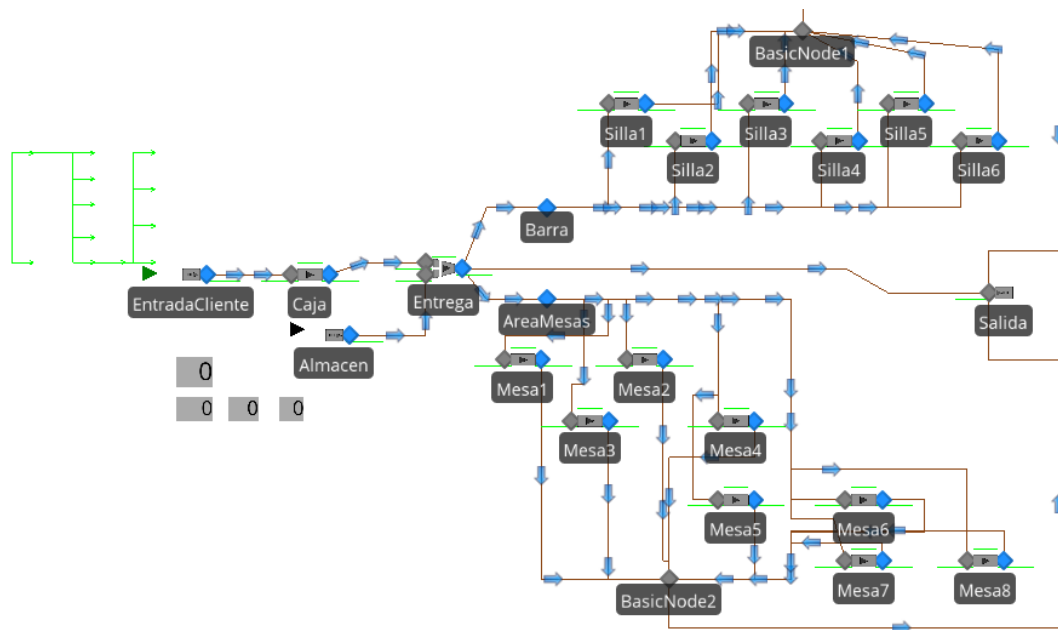


EscribirArchivoBotellas

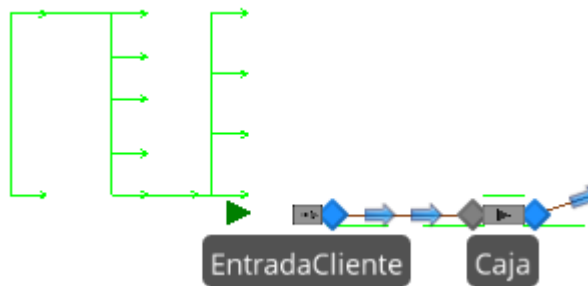


Tienda 5

Justificación del Modelo

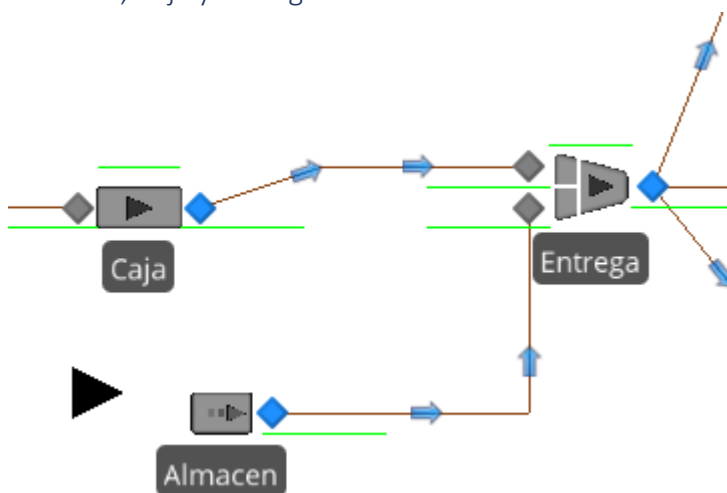


Entrada del Cliente y caja



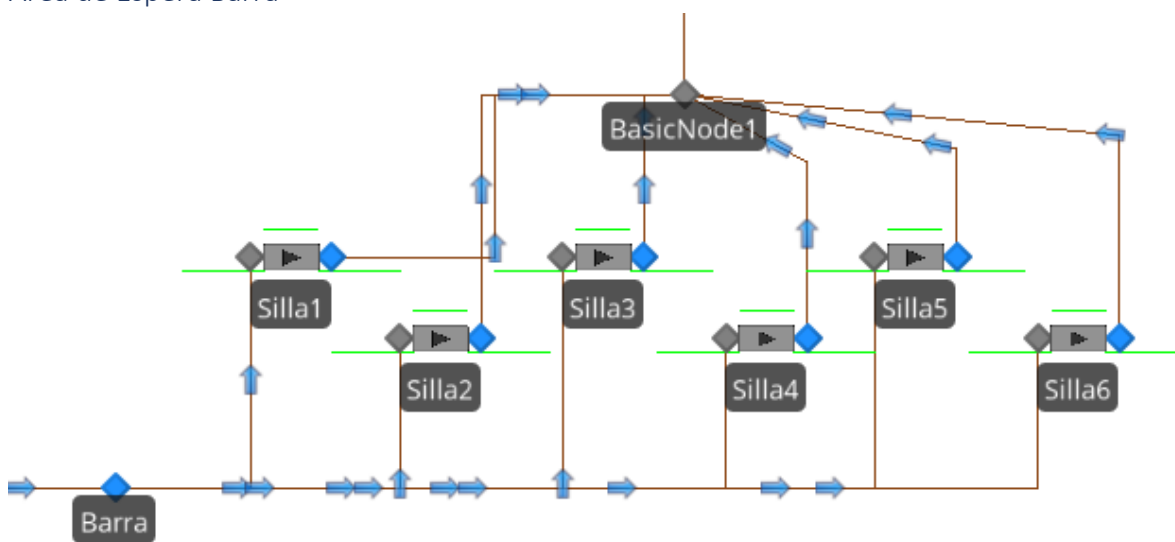
Se crea un Source y un servidor en el cual realiza la lectura del archivo de Excel

Almacén, Caja y Entrega



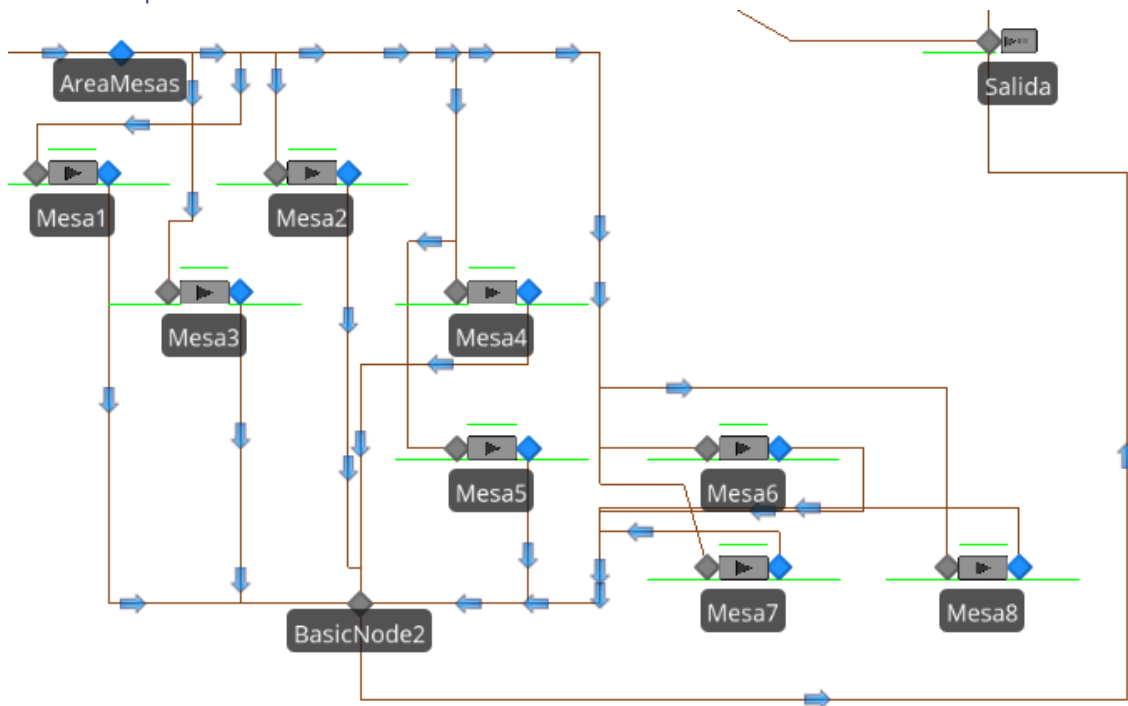
Aquí es donde el operario de la caja realiza el pedido al almacén para que le lleven la cantidad de vinos que solicita.

Área de Espera Barra



En esta parte del modelo el cliente puede ir a esperar a las sillas en lo que le llevan su pedido.

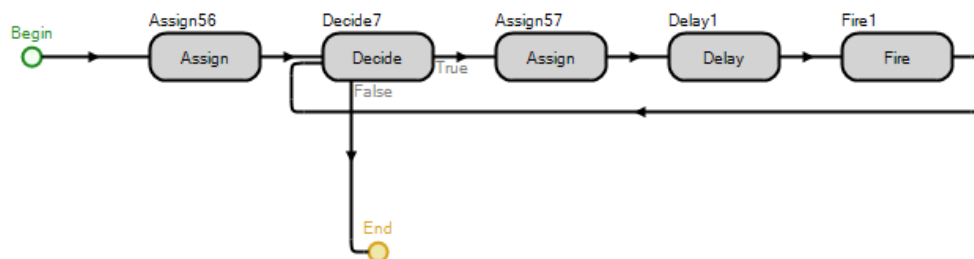
Área de Espera Mesa



El cliente puede ir a esperar a las mesas en lo que le llevan su pedido completo

Procesos

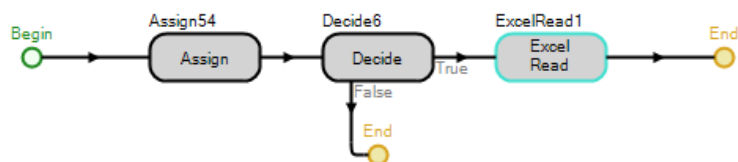
BuscarBotellas



ContadorBotella



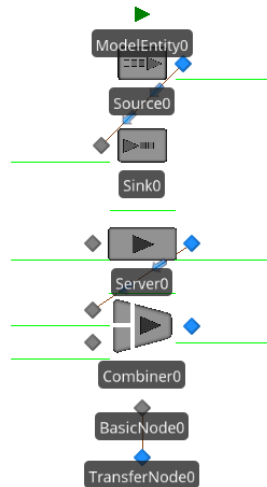
LecturaArchivo



Procesos en los cuales se realiza la lectura del archivo, contador de botellas y el buscador

API (Tienda 2)

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 la entidades que realizan una acción en concreto
- TransferNode: Estos componentes representaran puntos en los cuales se estará distribuyendo el modelo con el fin de estar un poco mas ordenado.
- Path y TimePath: Representa las conexiones entre entidades ya sea por una distancia en concreto o por un tiempo especifico

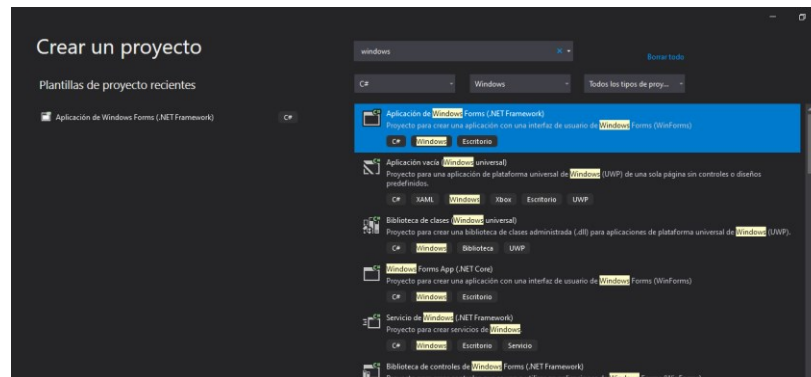
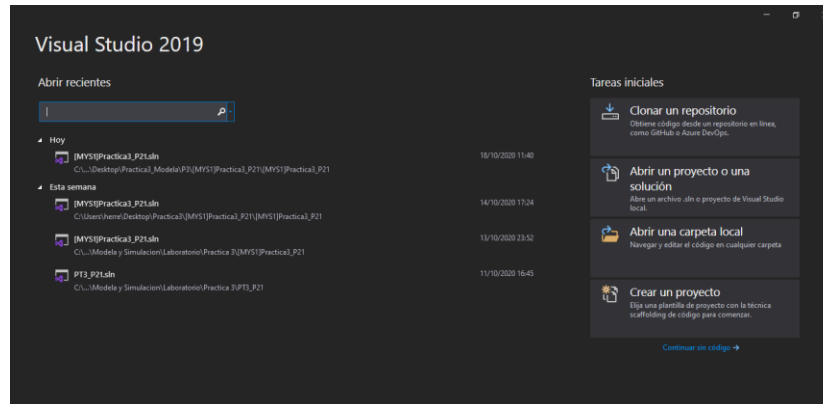
Pasos para crear una Api en visual studio (C#) con simio.

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.

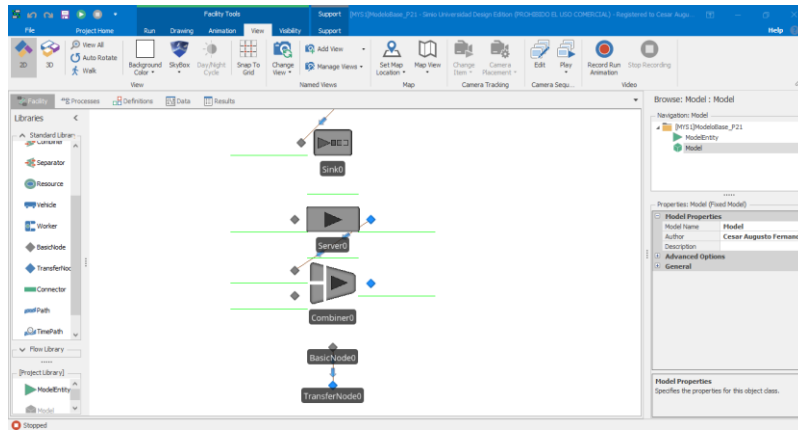
Paso 2.

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

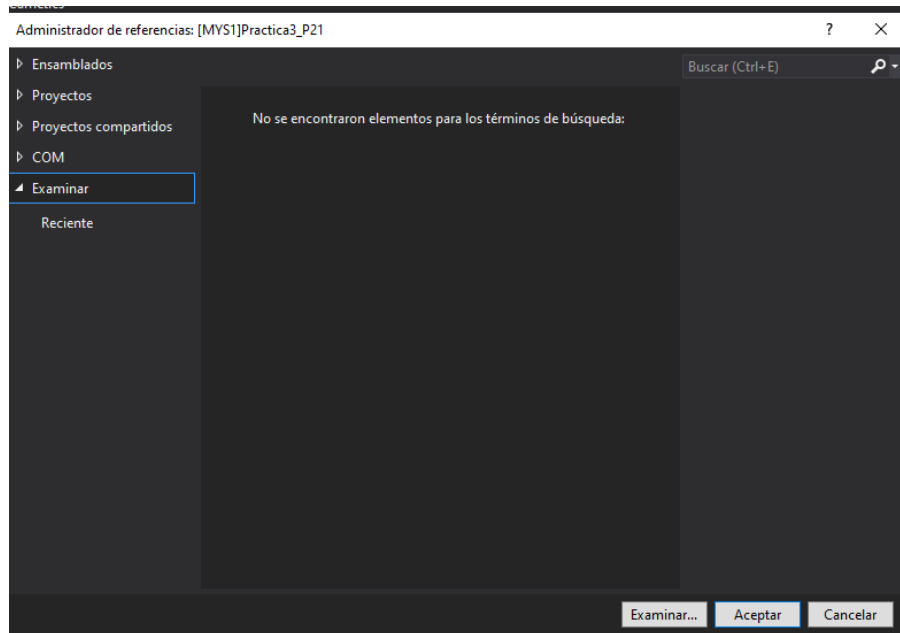
QImLicenseLib.dll	14/10/2020 17:39	Extensión de la ap...	988 KB
Simio.resources.dll	14/10/2020 17:39	Extensión de la ap...	1,139 KB
SimioAPI.dll	14/10/2020 17:39	Extensión de la ap...	64 KB
SimioAPI.Extensions.dll	14/10/2020 17:39	Extensión de la ap...	21 KB
SimioAPI.Graphics.dll	14/10/2020 17:39	Extensión de la ap...	5 KB
SimioDLL.dll	14/10/2020 17:39	Extensión de la ap...	8,582 KB
SimioEnums.dll	14/10/2020 17:39	Extensión de la ap...	26 KB
SimioReplicationRunnerContracts.dll	14/10/2020 17:39	Extensión de la ap...	14 KB
SimioTypes.dll	14/10/2020 17:39	Extensión de la ap...	46 KB

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#

```
using SimioAPI;
using SimioAPI.Extensions;
using SimioAPI.Graphics;
using Simio;
using Simio.SimioEnums;
```

Paso 7

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

```
private ISimioProject proyectoApi;  
private string rutaBase = "[MYS1]ModeloBase_P21.spfx";  
private string rutaFinal = "[MYS1]Tienda2_P21.spfx";  
private string[] warnings;  
private IModel model;  
private IIntelligentObjects intelligentObjects;  
private ILengthUnit lengthUnits;
```

Paso 8

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

```
1 referencia  
public ApiSimio()  
{  
    proyectoApi = SimioProjectFactory.LoadProject(rutaBase, out warnings);  
    model = proyectoApi.Models[1];  
    intelligentObjects = model.Facility.IntelligentObjects;  
}  
1 referencia
```

Descripción de las funciones mas importantes de visual studio

CreateModel

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

```
public void createModel()  
{  
    createMap();  
    SimioProjectFactory.SaveProject(proyectoApi, rutaFinal, out warnings);  
}
```

createMap

Este método nos sirve para mandar a llamar el método de creación de todo el modelo como tal.

```

public void CreateMap()
{
    /*****ENTRADA CLIENTE*****/
    createSource(-9, 0);
    updateName("Source1", "EntradaCliente");
    updateProperty("EntradaCliente", "InterarrivalTime", "Random.Uniform( 1.2 , 1.8 )");

    /*****SALIDA CLIENTE*****/
    createSink(-9, 3);
    updateName
    void Modelo.createSink(int x, int y)

    /*****BARRA LATERAL IZQUIERDA*****/
    createServer(2, -7);
    updateName("Server1", "BarraLateralIzquierda");

    /*****CAJA *****/
    createServer(8, 0);
    updateName("Server1", "Caja");
    updateProperty("Caja", "ProcessingTime", "Random.Uniform( 0.58 , 1.75)");

    /*****BODEGA*****/
    createSource(12, -6);
    updateName("Source1", "Bodega");
    updateProperty("Bodega", "InterarrivalTime", "Random.Uniform( 0.01 , 0.014)");
}

```

[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 posición en Y.

```

12 references
public void createSource(int x, int y)
{
    this.createObject("Source", x, y);
}

8 references
public void createServer(int x, int y)
{
    this.createObject("Server", x, y);
}

11 references
public void createSink(int x, int y)
{
    this.createObject("Sink", x, y);
}

47 references
public void createTransferNode(int x, int y)
{
    this.createObject("TransferNode", x, y);
}

```

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.

```
4 references
public void createObject(String type, int x, int y)
{
    intelligentObjects.CreateObject(type, new FacilityLocation(x, 0, y));
}
```

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.

```
2 references
public void createLink(String type, INodeObject nodo1, INodeObject nodo2)
{
    intelligentObjects.CreateLink(type, nodo1, nodo2, null);
}
```

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.

```
56 references
public void updateProperty(String name, String property, String value)
{
    model.Facility.IntelligentObjects[name].Properties[property].Value = value;
}
```

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.

```
99+ references
public void updateName(String oldName, String newName)
{
    model.Facility.IntelligentObjects[oldName].ObjectName = newName;
}
```


getNode

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

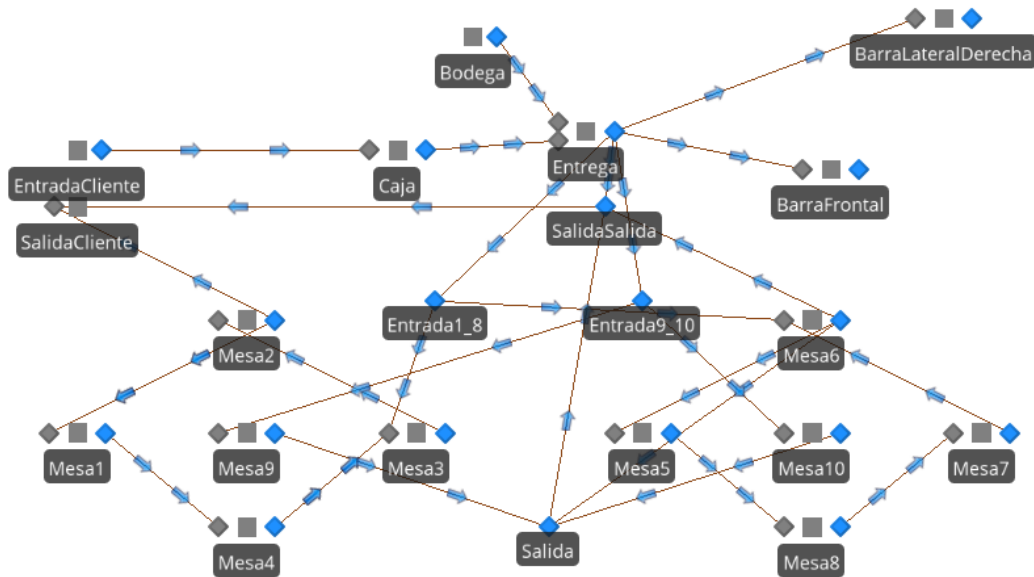
```
86 references
public INodeObject getNode(String name, int nodo)
{
    ...
    return ((IFixedObject)model.Facility.IntelligentObjects[name]).Nodes[nodo];
}
```

getNodebasico

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

```
94 references
public INodeObject getNodebasico (String name)
{
    ...
    return (INodeObject)model.Facility.IntelligentObjects[name];
}
```

Modelo Final



En este modelo final para la Tienda 2 se hizo esta distribución para que tenga una mejor visión de las entidades que van recorriendo dentro del modelo.

Distribución de Pedidos

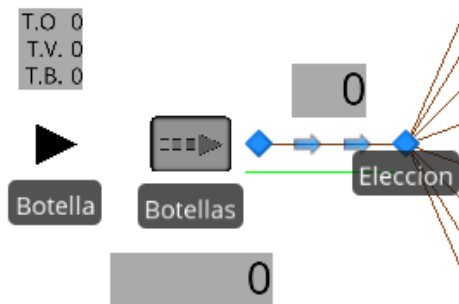
Justificación de Modelos

Se realizó el diseño del modelo del cual quedo de la siguiente manera.

- Esta parte del modelo es donde se leerá el archivo de Excel, se tomarán todas las órdenes para después ejecutar todo el proceso de distribución de vino.



- De la recepción de orden se realiza el despacho de botellas de vino, las cuales son sacadas por medio de eventos y se le asignan sus respectivos estados.



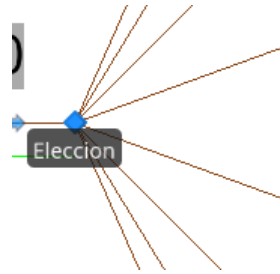
Properties: Botellas (Source)

Entity Arrival Logic	
Entity Type	Botella
Arrival Mode	On Event
Initial Number E...	0
Triggering Even...	SacarBotellas
Triggering Even...	1
Entities Per Arrival	1
+ Stopping Conditions	
+ Buffer Logic	
+ Table Row Referencing	
- State Assignments	
Before Exiting	4 Rows
On Balking	0 Rows
On Reneging	0 Rows

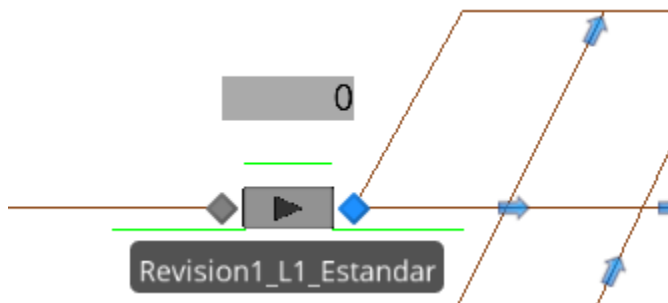
Before Exiting - Repeating Property Editor

Items:	Properties:
BotellaEntity.Picture, TipoVino	Basic Logic
BotellaEntity.TipoO, TipoOrder	
BotellaEntity.TipoV, TipoVinoA	State Variable ... BotellaEntity.Picture
BotellaEntity.TipoB, TipoBotella	New Value TipoVinoActual - 1
Basic Logic	
State Assignments (Basic Logic)	
Close	

- Seguido se realiza un transferrnode para realizar la separación por tipo de botella y enviarla a su respectivo canal de revisión, se cuenta con dos líneas de producción.



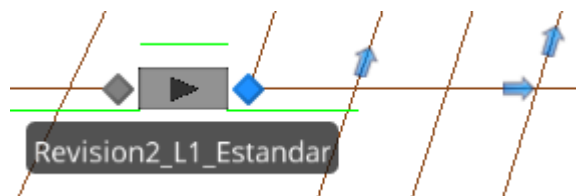
- Luego se hace la revisión de botella la cual tiene una probabilidad de 88% de pasar la prueba y 12% de ser rechazada. Las que son rechazadas son enviadas a reciclaje y se tienen que reponer. Esto se hace para los 4 tipos de botella y en las dos líneas de producción.



Properties: Revision1_L1_Estandar (Server)

Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
Dynamic Selection Rule	None
Transfer-In Time	0.0
Process Type	Specific Time
Processing Time	Random.Uniform(0.5 , 1)
Units	Seconds
Off Shift Rule	Suspend Processing
Other Processing Options	
Buffer Logic	
Reliability Logic	
Table Row Referencing	

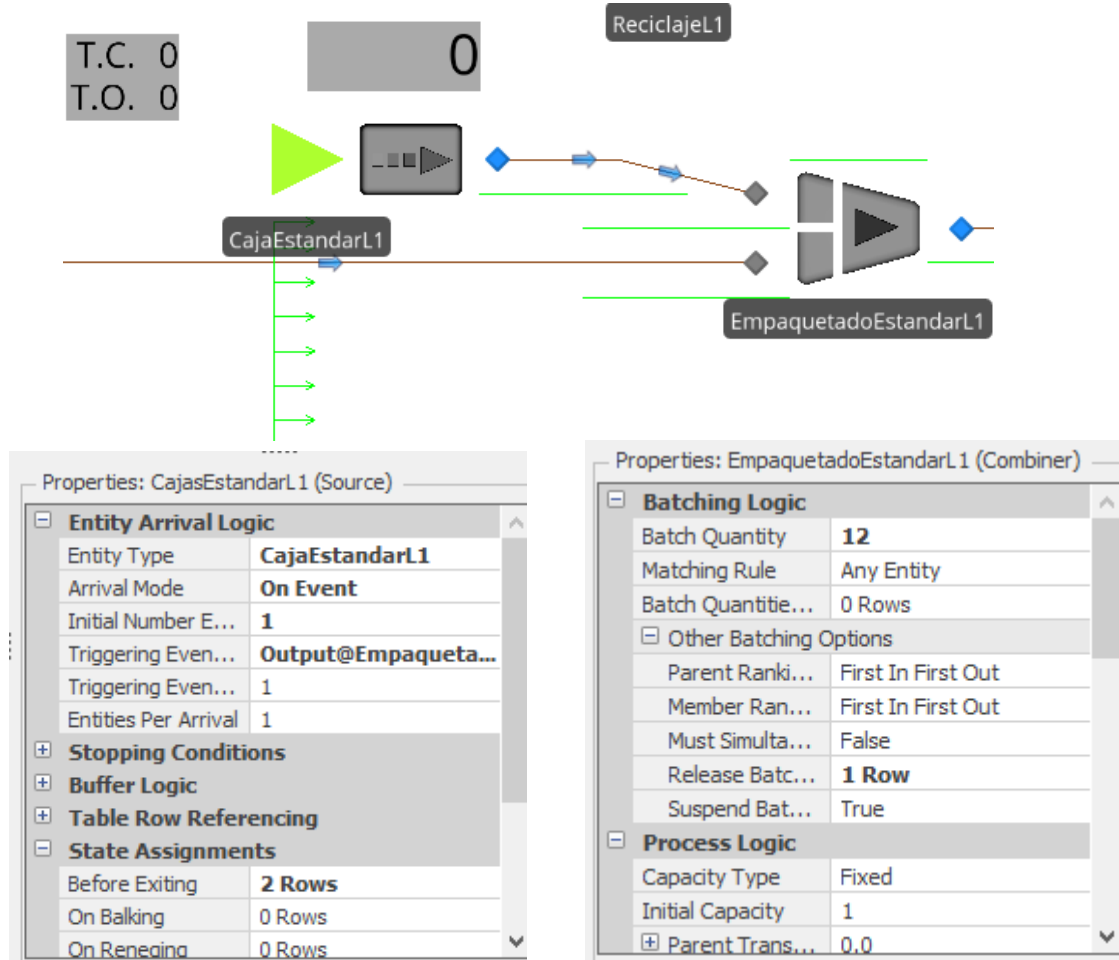
- Se realiza una segunda revisión esta vez es para verificar el estado del oxígeno en la botella, tiene una probabilidad de pasar de 92% y de ser rechazada del 8%. Igualmente las rechazadas son enviadas a reciclaje y las aceptadas son enviadas a empaquetar. (Este proceso es para las 4 botellas también)



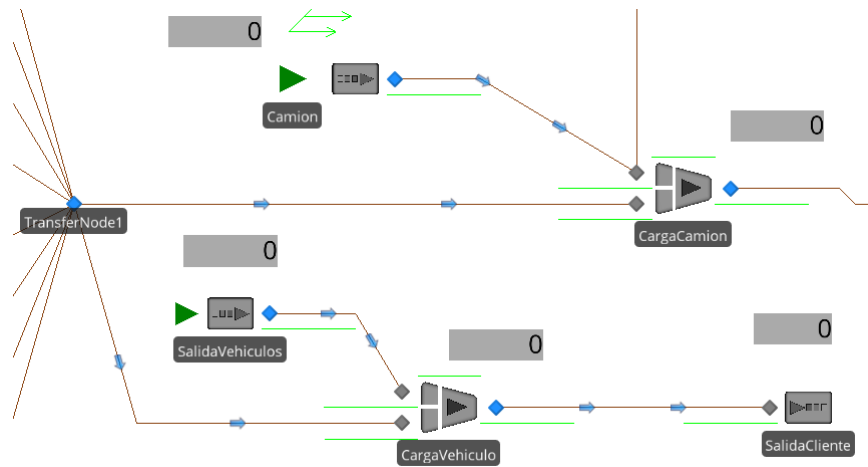
Properties: Revision2_L1_Estandar (Server)

Process Logic	
Capacity Type	Fixed
Initial Capacity	1
Ranking Rule	First In First Out
Dynamic Selection Rule	None
Transfer-In Time	0.0
Process Type	Specific Time
Processing Time	Random.Uniform(2 , 3)
Units	Seconds
Off Shift Rule	Suspend Processing
Other Processing Options	
Buffer Logic	
Reliability Logic	
Table Row Referencing	

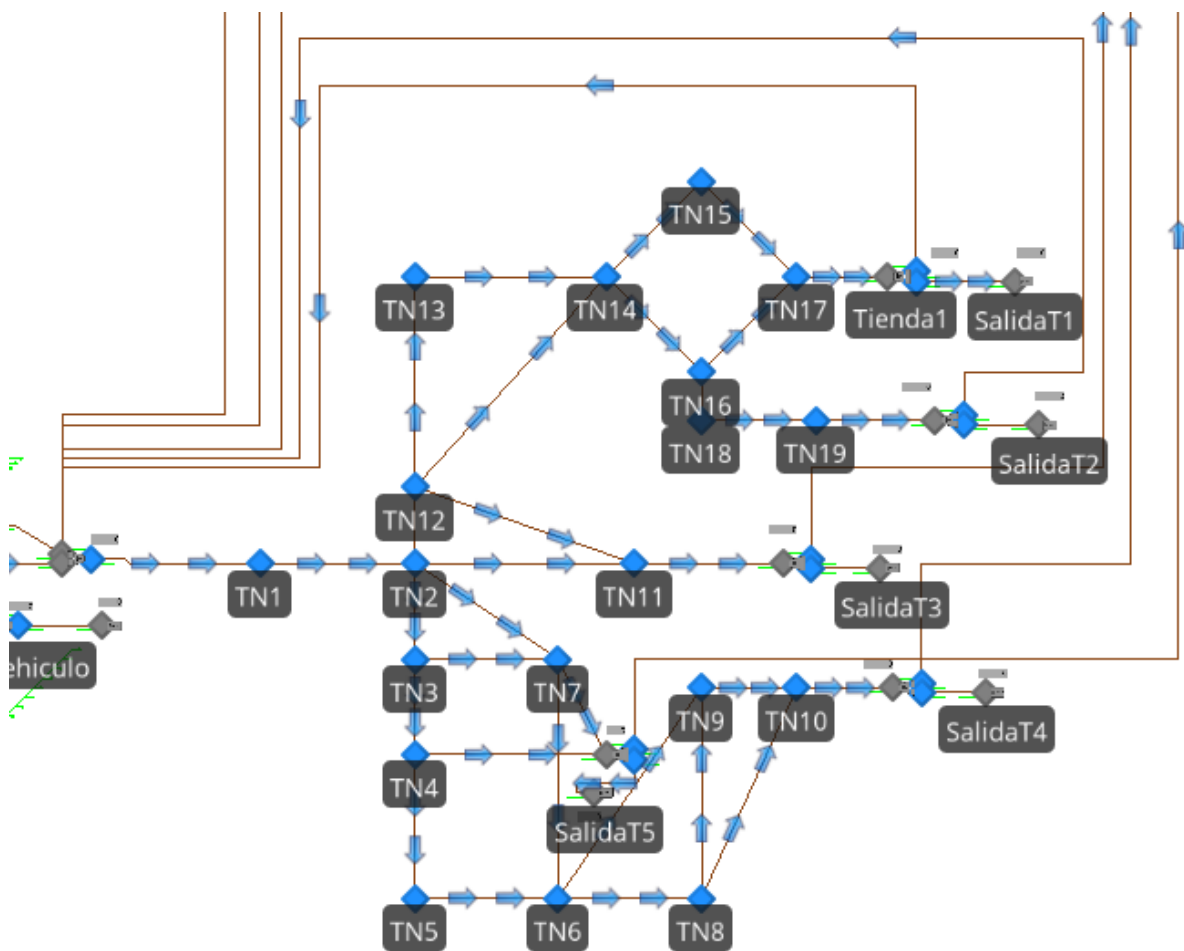
- Todas las botellas que pasaron las dos revisiones se dirigen al área de empaquetado, en el cual se van colocando en cajas dependiendo su tipo de vino y tamaño de botella. (Este proceso también es realizado en los 4 tipos y en las dos líneas).



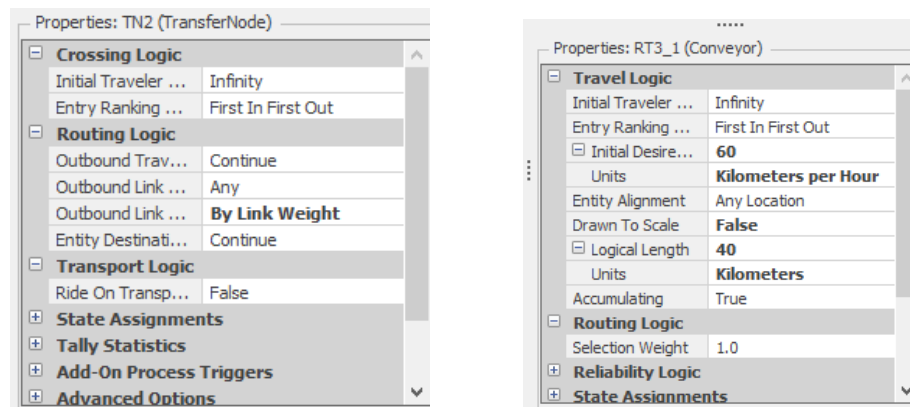
- Luego de empaquetar todas las botellas se dirige al área de carga, aquí hay dos opciones, cargar al vehículo del cliente mayorista o cargar a un camión de la empresa. Si se carga al vehículo del cliente parte de la empresa con rumbo desconocido, si se carga al camión tiene 5 opciones hacia dónde dirigirse las cuales son las 5 tiendas donde se vende el vino. Esta se realiza por medio de un transfunde y selección por peso dependiendo el tipo de orden.



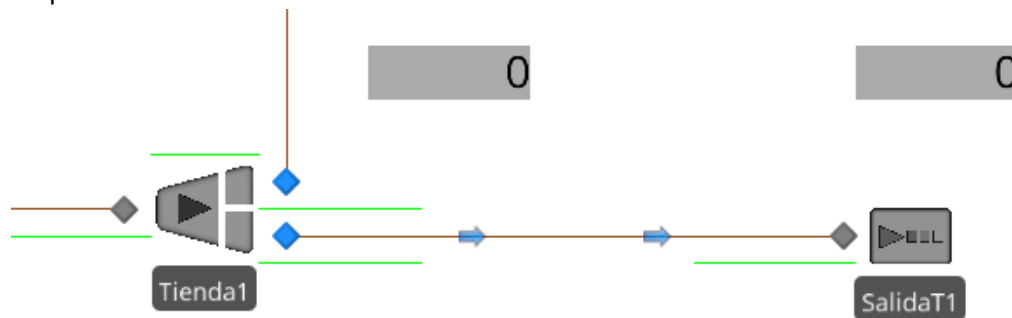
- Si el pedido hacia las tiendas este es el mapa de los posibles recorridos del camión, la empresa únicamente cuenta con dos camiones para esta distribución.



- Para realizar este recorrido en cada transnodo tiene la configuración de selección por peso y dependiendo el numero de destino el camión tomara la ruta mas optima. Para simular las calles se usaron conveyor y cuentan con determinada velocidad y distancia.



- Al llegar a su destino el camión deja el producto en la tienda y se regresa nuevamente a la empresa.

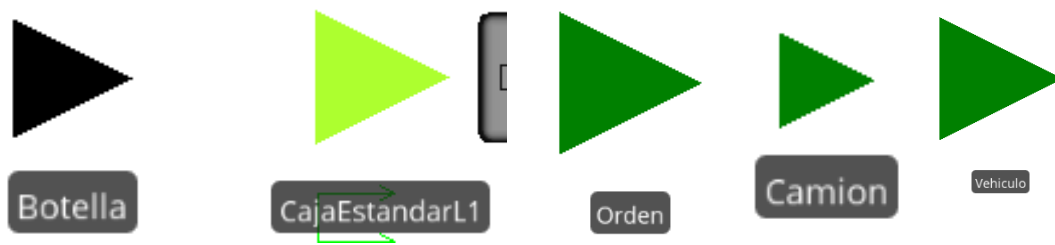


Objetos Utilizados

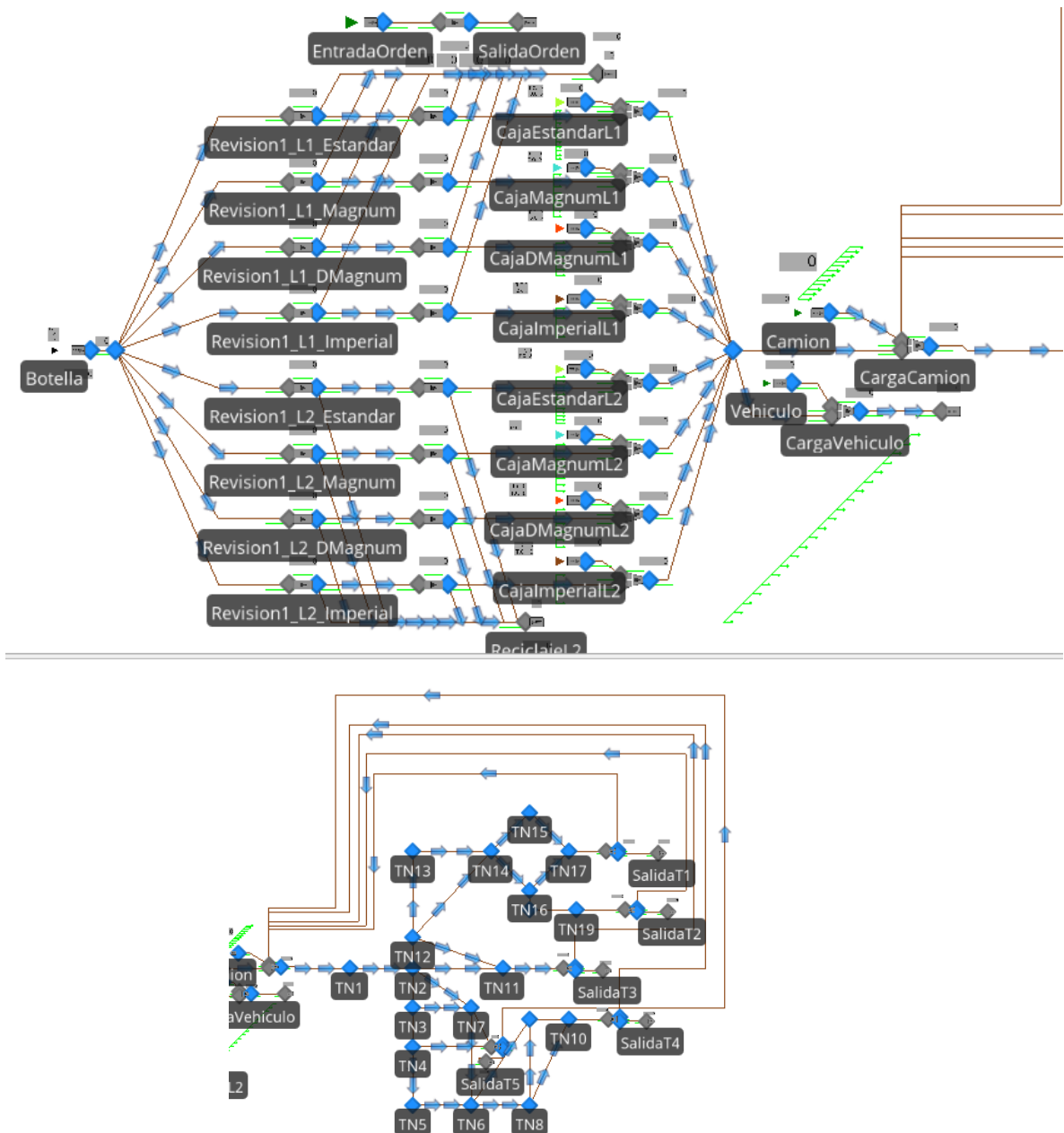
- Source: Se utilizo para modelar la creación de botellas, cajas, camiones, vehículos y creación de órdenes.
- Server: Se utilizo para la recepción de ordenes y revisiones de las botellas.
- Combiner: Se realizo el empaquetado por medio de combiners. También fueron usados en la simulación de carga de cajas a los vehículos.
- Separator: Se utilizo para realizar la separación del camión repartidor con el pedido.
- Sink: Se utilizo para la salida de cada orden, para destruir las botellas que eran enviadas a reciclaje y para simular que el vino había llegado a su destino.
- Conveyor: Fueron utilizados para realizar los distintos enlaces entre los objetos anteriormente mencionados.

Entidades Usadas

Se utilizaron diferentes entidades durante el proceso las cuales son:



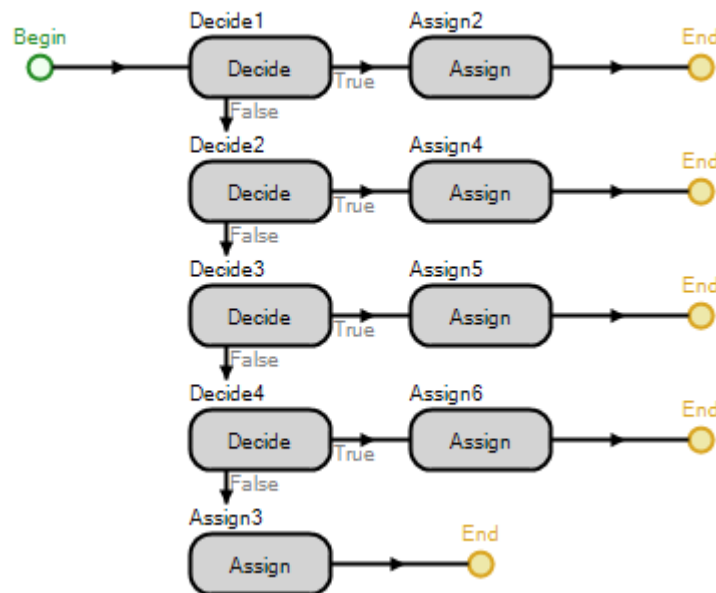
Diseño General



Descripción de Procesos

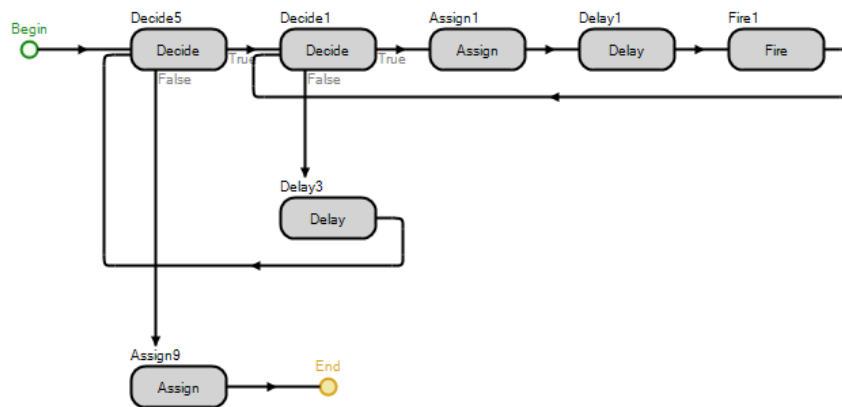
- AsignarNombre: Se le asigna el nombre del tipo de vino que son.

AsignarNombre



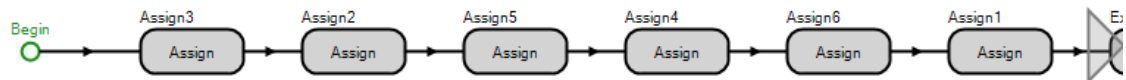
- Despacho de Botellas: Este proceso sirve para sacar la cantidad de botellas solicitadas.

DespachoBotellas



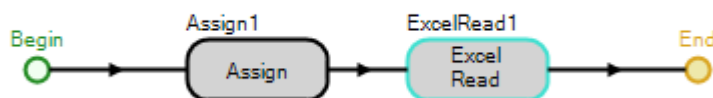
- DeterminarFila: Sirve para guardar en estados la lectura hecha desde el archivo de Excel.

DeterminarFila



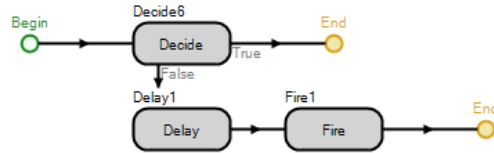
- LecturaExcel: Este proceso sirve para realizar la lectura desde el Excel.

LecturaExcel

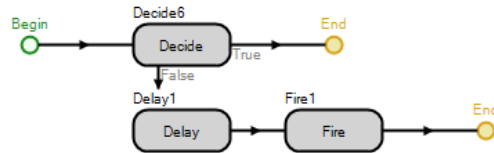


- Prueba2 y Prueba2_1: Sirven para determinar cuando una orden esta completa en camión y vehículo respectivamente.

Prueba2

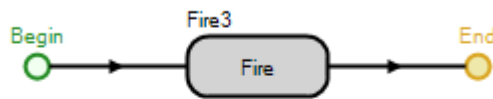


Prueba2_1

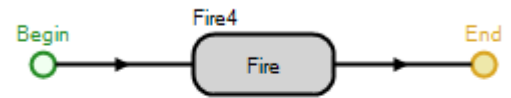


- Reponer (TipodeBotella): Sirve para reponer las botellas que se van a reciclaje.

ReponerDMagnum



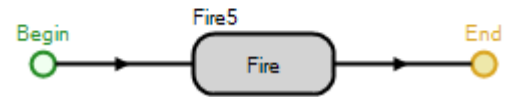
ReponerImperial



ReponerEstandar

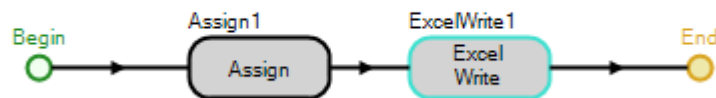


ReponerMagnum



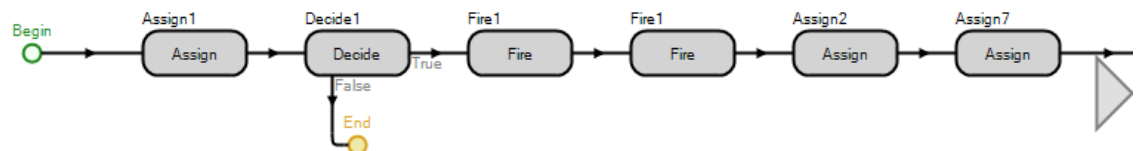
- Reporte: Este proceso sirve para generar el reporte de las botellas recicladas.

Reporte










- SacarPaquete(TipodeCaja): Este proceso sirve para sacar las botellas empaquetadas, este es aplicado a los cuatro tipos.

SacarPaqueteEstandar











Descripción de Estados:

Estados entidad Orden

State Variables		
 Picture	Real State Variable	Picture
 Animation	String State Variable	Animation
 NoOrden	Integer State Variable	NoOrden
 TipoOrden	Integer State Variable	TipoOrden
 TipoVino	Integer State Variable	TipoVino
 TipoBotella	Integer State Variable	TipoBotella
 CantidadBotellas	Integer State Variable	CantidadBotellas

- Picture: Sirve para cambiar de color a la entidad.
- NoOrden: Sirve para almacenar el número de orden leído del Excel.
- TipoOrden: Sirve para almacenar el tipo de orden.
- TipoVino: Sirve para almacenar el tipo de vino.
- TipoBotella: Sirve para almacenar el tipo de botella.
- CantidadBotellas: Sirve para almacenar la cantidad de botellas solicitadas.

Estados entidad Botella

State Variables		
 TipoB	Integer State Variable	TipoB
 TipoV	Integer State Variable	TipoV
 TipoO	Integer State Variable	TipoO
 Picture	Real State Variable	Picture
 NombreVino	String State Variable	NombreVino
 NombreBotella	String State Variable	NombreBotella
 NombreRevision	String State Variable	NombreRevision
 TipoRevision	Integer State Variable	TipoRevision

- TipoB: Guarda el tipo de botella.
- TipoV: Guarda el tipo de vino.
- TipoO: Guarda el tipo de orden.
- Picture: Sirve para cambiar el color dependiendo el tipo de vino.
- NombreVino: Guarda el nombre del vino.
- NombreBotella: Guarda el nombre de la botella.
- NombreRevision: Guarda el nombre de la revisión aplicada.
- TipoRevision: Guarda el tipo de revisión aplicada.

Estados entidad Caja

State Variables (Inherited)		
State Variables		
TipoCaja	Integer State Variable	TipoCaja
TipoOC	Integer State Variable	TipoOC
Picture	Real State Variable	Picture
NoOrdenCaja	Integer State Variable	NoOrdenCaja

- TipoCaja: Guarda el tipo de caja.
- TipoOC: Guarda el tipo de orden de la caja.
- Picture: Sirve para cambiar color a la entidad
- NoOrdenCaja: Sirve para guardar el numero de orden de la caja.

Estados entidad Vehículo

State Variables		
VehiculoTipoOrden	Integer State Variable	VehiculoTipoOrden

- VehiculoTipoOrden: Sirve para guardar el tipo de orden y saber a que tienda dirigirse.

Globales

State Variables (Inherited)		
State Variables		
Total	Integer State Variable	Total
TipoBotellaActual	Integer State Variable	TipoBotellaActual
CantidadActual	Integer State Variable	CantidadActual
Fila	Integer State Variable	Fila
OrdenActual	Integer State Variable	OrdenActual
TipoVinoActual	Integer State Variable	TipoVinoActual
TipoOrdenActual	Integer State Variable	TipoOrdenActual
ConteoBotellaEstandar	Integer State Variable	ConteoBotellaEstandar
AuxCantidadActual	Integer State Variable	AuxCantidadActual
ConteoReciclajeEstandar	Integer State Variable	ConteoReciclajeEstandar
ConteoBotellaMagnum	Integer State Variable	ConteoBotellaMagnum
ConteoReciclajeMagnum	Integer State Variable	ConteoReciclajeMagnum
ConteoBotellaDMagnum	Integer State Variable	ConteoBotellaDMagnum
ConteoReciclajeDMagnum	Integer State Variable	ConteoReciclajeDMagnum
ConteoBotellaImperial	Integer State Variable	ConteoBotellaImperial
ConteoReciclajeImperial	Integer State Variable	ConteoReciclajeImperial
SepararOrden	Integer State Variable	SepararOrden
FilaEscritura	Integer State Variable	FilaEscritura
Prueba	Integer State Variable	Prueba
AuxTipoOrdenActual	Integer State Variable	AuxTipoOrdenActual

- Total: Guarda el total de las botellas.
- TipoBotellaActual: Guarda el valor de la botella actual.
- CantidadActual: Guarda el valor de la cantidad actual.
- Fila: Sirve para guardar el valor de la fila leída en Excel.
- OrdenActual: Guarda el valor de la orden actual

- TipoVinoActual: Guarda el valor del vino actual.
- TipoOrdenActual: Guarda el tipo de orden actual.
- ConteoBotella(Tipo): Guarda el valor de las botellas que van saliendo del source.
- ConteoReciclaje(Tipo): Cuenta cuantas botellas van a reciclaje de cada tipo.
- AuxCantidadActual: Guarda el valor de la cantidad actual de manera estática.
- SepararOrden: Separa las órdenes.
- FilaEscritura: Sirve para guardar el valor de la fila de escritura de Excel.
- Prueba: Variable utilizada como booleana.
- AuxTipoOrdenActual: Almacena el tipo de orden actual de manera estática.

Mejoras al sistema

- Elaboración de Vinos
 - Tener un poco mas de tanques de vinos con menos capacidad y así poder sacar un tanque que ya haya cumplido con el tiempo de fragmentación y no parte por parte
 -
- Tienda 5
 - Colocar mas sillas en la barra y no tanta mesa
 - Tener un mas trabajadores en el área de bodega para que se reduzca el tiempo de sacar las botellas.
- API (Tienda 2)
 - Reubicar las mesas para que se pueda aprovechar todo el espacio posible
 - Tener mesas más grandes y que tengan mayor capacidad.
- Distribución de Vinos
 - Tener mas camiones para realizar las distintas entregas solicitadas y así generar más ganancias.
 - Eliminar por completo una línea de producción ya que se pudo detectar que su porcentaje de utilización es bastante bajo.

Conclusiones

- Para la elaboración de vino es un proceso muy tardo en el cual si se necesita tener un plan para dicha preparación
- Tienda 5 Y API (Tienda 2): Reubicar las áreas de espera ya que se pierde tiempo al momento de movilizarse dentro de la tienda.
- Para la distribución de vino se cuenta con una línea de producción innecesaria ya que no son muchos los pedidos de vino.
- Se tiene una gran cantidad de botellas que si pasan ambas revisiones.
- Las distancias para repartir los pedidos son bastante lejanas por lo tanto los camiones se tardan bastante en regresar a la empresa.