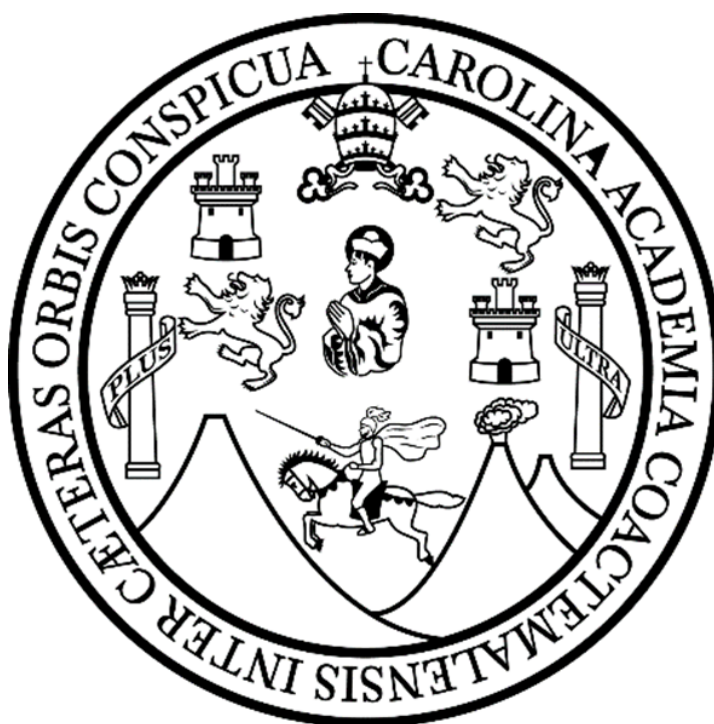


# UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Facultad de Ingeniería  
Escuela de Ciencias y Sistemas



## PRÁCTICA #2

MODELACIÓN Y SIMULACIÓN 1  
GRUPO No. 06

NOMBRE	CARNET
Erick Abdul Chacon Barillas	201807169
Maria Isabel Masaya Córdoba	201800565
Juan Francisco Urbina Silva	201906051
Steven Josue González Monroy	201903974
Douglas Darío Rivera Ojeda	201122881
Luis Fernando Sánchez Santos	3348212820901

# ÍNDICE

1. Carátula .....	1
2. Índice .....	2
3. Introducción.....	3
4. Diseño del Sistema con Justificación.....	4-6
5. Descripción de los Procesos.....	7
6. Descripción de los Estados.....	8
7. Conclusiones.....	9
8. Propuestas de Mejora del Sistema.....	10

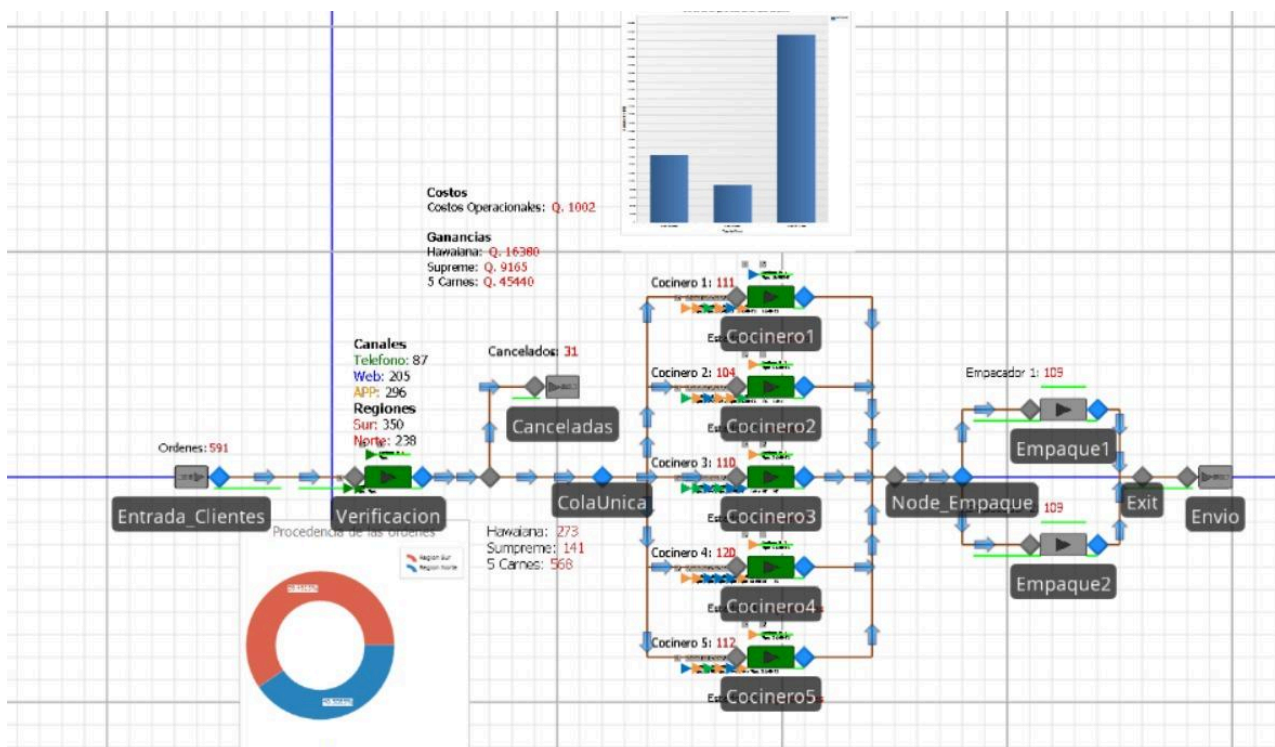
# INTRODUCCIÓN

En esta práctica se modela el funcionamiento automatizado de una pizzería, simulando todas las etapas operativas desde la recepción de pedidos hasta la entrega final al cliente. Para ello, se utiliza el software de simulación Simio, que permite representar de manera visual y precisa los procesos internos del sistema, como la validación de órdenes, la preparación en cocina, el empaque y el control de métricas.

El objetivo es analizar el comportamiento del sistema ante diferentes condiciones, evaluando aspectos como el flujo de trabajo, tiempos de atención, cantidad de pedidos por canal, y ganancias por tipo de pizza. A través del modelo se identifican posibles cuellos de botella y áreas de mejora, permitiendo proponer soluciones basadas en datos. Esta simulación fortalece la comprensión del estudiante sobre el uso práctico de herramientas de simulación discreta en sistemas reales.

# DISEÑO DE LA JUSTIFICACIÓN

El diseño del modelo simulado en Simio representa el flujo completo de atención y producción dentro de la pizzería automatizada *Panucci's*, desde el ingreso del pedido hasta su empaque final y salida del sistema. En la siguiente imagen se observa una vista general del modelo implementado:



## COMPONENTES PRINCIPALES UTILIZADOS

Elemento del sistema	Componente Simio utilizado	Justificación
Entrada de Clientes	Source	Genera las órdenes con distribución exponencial (media de 40 por hora).
Verificación de pago	Server	Simula la inspección de pagos con tiempo Uniform(60, 90). Un 5% es cancelado.
Canales de origen y región	Status Label y lógica interna	Muestra las estadísticas de entrada: Teléfono, Web, App y regiones (Norte/Sur).
Cancelaciones	Decision + Sink	Desvía el 5% de pedidos no pagados al nodo "Canceladas".
Cola única para cocina	Queue (InputBuffer) compartida	Distribuye pedidos a los cocineros según disponibilidad.
Cocineros (1 al 5)	Server (paralelos)	Cada uno sigue el flujo: amasado (Uniform), ingredientes (fijo), horneado (fijo).
Empaque	Server (Empaque1 y Empaque2) + Queue	Dos servidores con cola compartida. Tiempo Triangular(30s, 40s, 50s)
Salida del sistema	Sink (Envío)	Marca que la pizza fue entregada y contabiliza la venta.

## INTERPRETACIÓN DE LAS GRÁFICAS Y RESULTADOS

Según los resultados del modelo, se puede observar que las tres pizzas generan ganancias, siendo la más rentable la pizza de 5 Carnes (Q.45440), seguida de la Supreme (Q.9165) y por último la Hawaiana (Q.16380). Aunque esta última es la que más se vende después de la 5 Carnes, su ganancia es menor. Esto podría deberse a que su precio es más bajo o su costo de producción más alto en proporción.

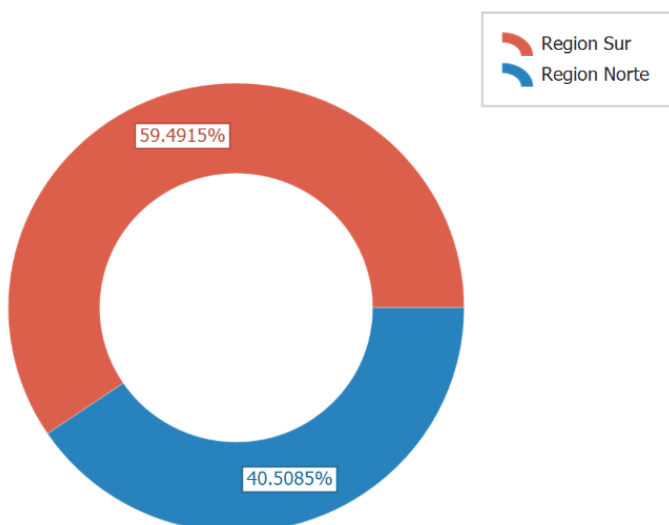
Los costos operacionales totales son de Q.1002, asociados principalmente a los pedidos que ingresan por el sitio web y la app. Al compararlos con las ganancias, se nota que el sistema sigue siendo rentable.

El gráfico circular indica que la mayoría de las órdenes provienen de la región sur, lo cual puede ser útil para tomar decisiones de logística o estrategias de mercadeo.

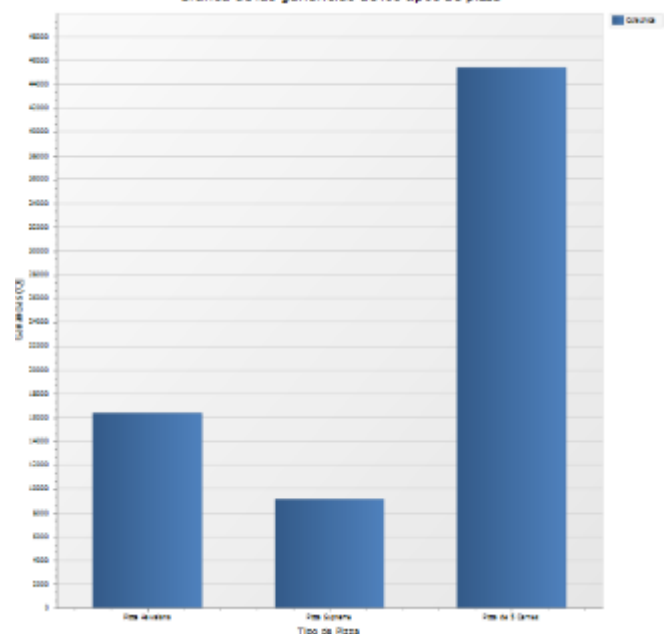
En cuanto a los canales, la mayoría de pedidos ingresaron por la app (296), seguido de la web (205) y por último por teléfono (87). Esto confirma que los usuarios prefieren los canales digitales, especialmente móviles.

Por último, se reportaron 31 órdenes canceladas, lo que corresponde aproximadamente al 5% del total de pedidos (591), lo cual es coherente con el comportamiento esperado en el modelo.

Procedencia de las ordenes



Grafica de las ganancias de los tipos de pizza



# DESCRIPCIÓN DE CADA UNO DE LOS PROCESOS UTILIZADOS

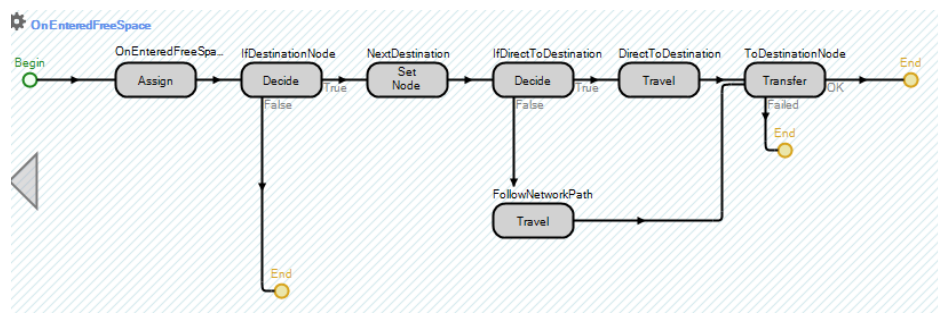
Para la elaboración de la segunda práctica, fueron empleados diferentes procesos, para poder cumplir con cada una de las especificaciones utilizadas.

## OnEnteredFreeSpace:

Al iniciar el proceso, se le realiza una asignación a la variable *FreeSpaceSteeringToken.EnteredFreeSpaceFromNode*, en la cual primero se realiza una verificación si existe un nodo previo, esta es una comprobación de seguridad para evitar errores si se intenta usar un objeto que no está definido. En caso de existir, se le calcula una distancia directa desde el objeto actual hasta el nodo previo, luego comprueba si esa distancia es exactamente 0.0, lo que generalmente significa que ambos están en la misma ubicación o en una posición equivalente. En caso de existir se regresa automáticamente el valor de cero.

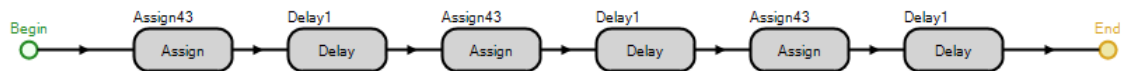
Una vez realizada la asignación, se procede a verificar si el nodo de destino es divergente de *nothing*, en caso de serlo, se cambia el nodo de destino. Luego se decide si no se puede o no se necesita verificar una ruta de navegación. Si no hay necesidad (por comportamiento, falta de red, o entrada no válida), devuelve True. Si sí hay red y entrada válida, devuelve True solo si no hay un camino entre el nodo anterior y el de destino.

En caso que la decisión sea verdadera, se realiza un viaje hacia el nodo de destino. Y por último se realiza una transferencia hacia el nodo de destino. En caso que la decisión sea falsa, se hace un seguimiento a la red, de la red actual y por último se realiza una transferencia hacia el nodo de destino.



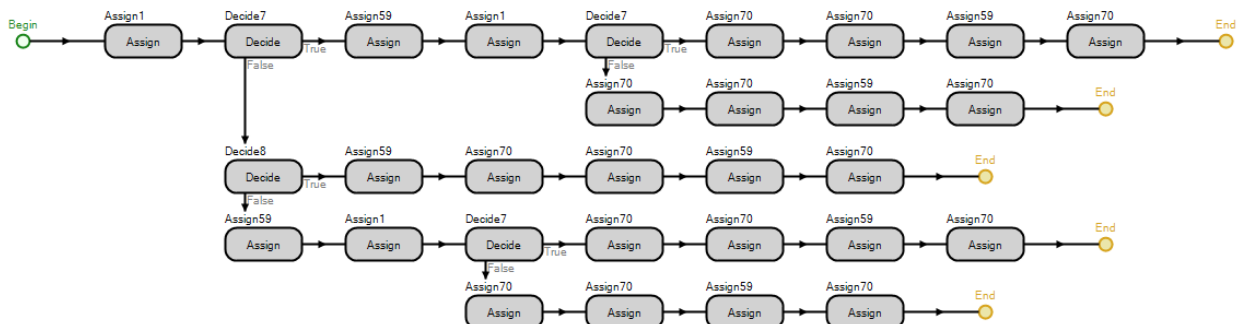
**Cocinero\_Processing:** Al iniciar el proceso, se le asigna un valor a la variable *EC#*, con el estado de “Amasando”, luego se le aplica un delay entre 4 y 6 minutos, luego se le cambia el estado a “Ingredientes” y nuevamente se le aplica un delay de 5 minutos y por último se le aplica de nuevo un cambio de estado a “Horneado” y se le aplica un último delay de 10 minutos. Este proceso es aplicado a todos los cocineros.

Cocinero1\_Processing



**ColaUnica\_Entered:** Al momento de iniciar la cola, se le asigna al valor de la variable de *TipoEspecialidad* un valor random entre 0 y 1. Luego se verifica si el tipo de especialidad tiene un valor menor a 0.5; en caso de ser cierto se le asigna un valor de 1 y al TipoCantidad se le asigna un valor aleatorio entre 0 y 1 y se verifica si el TipoCantidad tiene un valor menor a 0.5, se le asigna una cantidad al *Contador\_PH* y en caso contrario se le asignan dos valores al Contador\_PH. En dado caso que ninguno de estos se llegase a cumplir, se aumenta el *Contador\_P5C* en dos unidades.

ColaUnica\_Entered



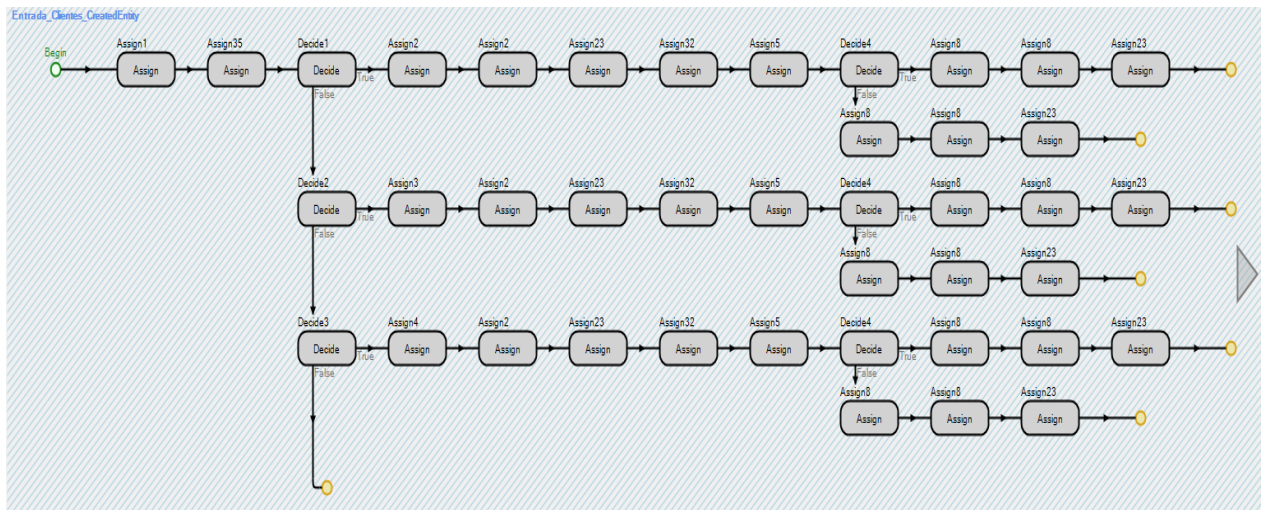
**Entrada\_Clientes Add\_On Process:** Este método se encarga de agregarle un valor aleatorio al tipo de canal que van a llegar a emplear los clientes y se aumenta la cantidad de clientes que acaba de ingresar al local. Se verifica si el tipo de canal tiene un valor menor a 0.15. En caso de ser así se incrementa el valor del *ContadorTelefono* y se verifica el valor que tiene el tipo de región para incrementar en uno la región norte o la región sur.

En caso de no cumplirse se verifica si el canal tiene un valor entre 0.15 y 0.55, lo cual significa que los clientes están cursando por el canal 2. Se

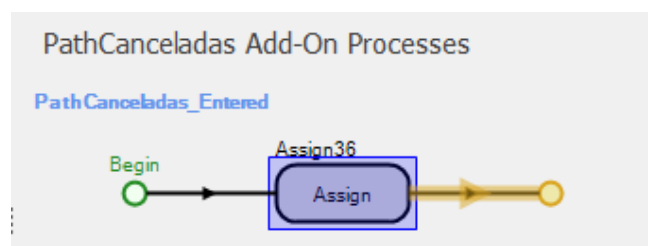


aumenta el contador web y en base al tipo de región, se decide a incrementar la región norte o la región sur.

En caso de que este no cumpla con ninguno, se opta por asignarle el canal 3 a los clientes para que estos puedan transitar por ahí, se incrementa el valor de la variable del *ContadorApp* y se verifica el tipo de región para ver si se llega a incrementar la región norte o la región sur.

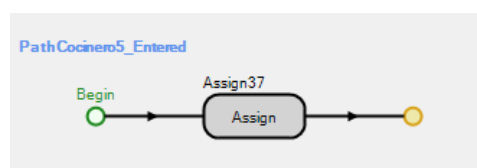


**Path\_Canceladas Add\_On Process:** Este proceso, se encarga de incrementar en una unidad, la variable de contador de órdenes canceladas.



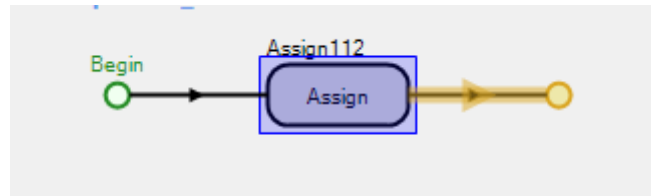
**Path\_Cocinero Add\_On Process:**

Este proceso, se encarga de incrementar en una unidad, la variable de contador de cocinero, con la finalidad de poder analizar cuantas órdenes fueron atendidas por un cocinero.



## Path\_Empacador Add\_On Process

Este proceso, se encarga de incrementar en una unidad, la variable de contador de empaque, con la finalidad de poder analizar cuantos elementos fueron empacados mediante esas rutas.

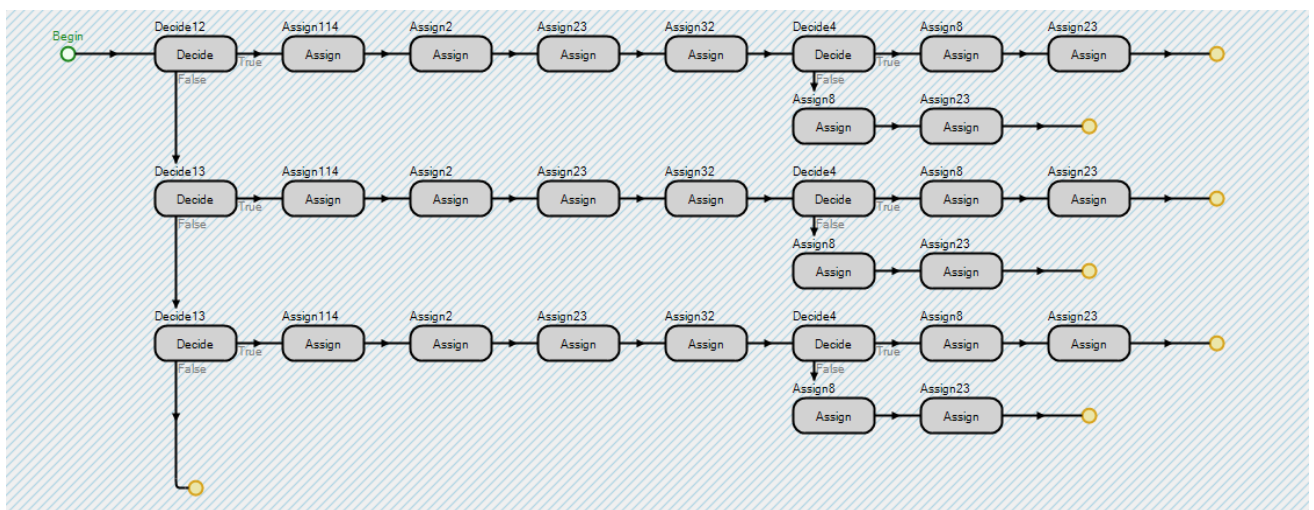


## Verificación Add\_On Process

Este proceso se encarga de verificar el canal en el cual van a transitar las entidades. En caso de estar transitando por el canal 1. Se le asigna el valor a pizza a realizar de Hawaiana y se procede a verificar si posterior a este se movió de canal la entidad y se procede a asignarle el contador norte o sur acorde a la verificación.

En caso de estar transitando por el canal 2., se le asigna el valor a pizza a realizar de Supreme y se procede a verificar si posterior a este se movió de canal la entidad y se procede a asignarle el contador norte o sur acorde a la verificación.

En caso de estar transitando por el canal 3, se le asigna el valor a pizza a realizar de 5 carnes y se procede a verificar si posterior a este se movió de canal la entidad y se procede a asignarle el contador norte o sur acorde a la verificación.



# DESCRIPCIÓN DE CADA UNO DE LOS ESTADOS

## Procesos generales

- **ContadorClientes:** Variable entera que lleva el conteo total de clientes generados en el sistema.
- **TipoCanal:** Variable real que almacena un número aleatorio utilizado para determinar el canal de entrada del cliente.
- **Canal:** Variable entera que indica el canal por el que se realizó la orden: Teléfono (1), Web (2), App (3).
- **TipoRegion:** Variable real que guarda un valor aleatorio para asignar la región del cliente.
- **Region:** Variable entera que representa la región de origen del pedido: Sur (1) o Norte (2).
- **ContadorSur:** Variable entera que acumula el número de órdenes provenientes de la región sur.
- **ContadorNorte:** Variable entera que acumula el número de órdenes provenientes de la región norte.
- **ContadorTelefono:** Variable entera que cuenta los pedidos realizados por teléfono.
- **ContadorWeb:** Variable entera que cuenta los pedidos realizados vía sitio web.
- **ContadorApp:** Variable entera que cuenta los pedidos realizados desde la aplicación móvil.
- **ContadorCanceladas:** Variable entera que registra la cantidad de órdenes canceladas tras no pasar la verificación.
- **ContadorCocinero1:** Variable entera que cuenta las pizzas procesadas por el Cocinero 1.











- ContadorCocinero2: Variable entera que cuenta las pizzas procesadas por el Cocinero 2.
- ContadorCocinero3: Variable entera que cuenta las pizzas procesadas por el Cocinero 3.
- ContadorCocinero4: Variable entera que cuenta las pizzas procesadas por el Cocinero 4.
- ContadorCocinero5: Variable entera que cuenta las pizzas procesadas por el Cocinero 5.
- EC1, EC2, EC3, EC4, EC5: Variables de tipo cadena (string) que indican el estado actual del proceso de cada cocinero, como por ejemplo "Amasando", "Horneando" o "Colocando ingredientes".
- TipoEspecialidad: Variable real que guarda un valor aleatorio utilizado para determinar la especialidad de la pizza.
- Especialidad: Variable entera que identifica el tipo de pizza solicitada: Hawaiana (1), Supreme (2), 5 Carnes (3).
- TipoCantidad: Variable real que determina aleatoriamente cuántas pizzas se solicitan en la orden.
- Cantidad: Variable entera que guarda la cantidad de pizzas en una orden (1, 2 o 3), determinada por la especialidad.
- Contador\_PH: Variable entera que acumula la cantidad de pizzas hawaianas vendidas.
- Contador\_PS: Variable entera que acumula la cantidad de pizzas supreme vendidas.
- Contador\_PSC: Variable entera que acumula la cantidad de pizzas de 5 carnes vendidas.
- ContadorEmpaque1: Variable entera que lleva el conteo de pizzas empacadas por el Empacador 1.
- ContadorEmpaque2: Variable entera que lleva el conteo de pizzas empacadas por el Empacador 2.

State Variables	
ContadorClientes	Integer State Variable
TipoCanal	Real State Variable
Canal	Integer State Variable
TipoRegion	Real State Variable
Region	Integer State Variable
ContadorSur	Integer State Variable
ContadorNorte	Integer State Variable
ContadorTelefono	Integer State Variable
ContadorWeb	Integer State Variable
ContadorApp	Integer State Variable
ContadorCanceladas	Integer State Variable
ContadorCocinero1	Integer State Variable
ContadorCocinero2	Integer State Variable
ContadorCocinero3	Integer State Variable
ContadorCocinero4	Integer State Variable
ContadorCocinero5	Integer State Variable
AaEC1	String State Variable
AaEC2	String State Variable
AaEC3	String State Variable
AaEC4	String State Variable
AaEC5	String State Variable
TipoEspecialidad	Real State Variable
Especialidad	Integer State Variable
TipoCantidad	Real State Variable
Cantidad	Integer State Variable
Contador_PH	Integer State Variable
Contador_PS	Integer State Variable
Contador_P5C	Integer State Variable
ContadorEmpaque1	Integer State Variable
ContadorEmpaque2	Integer State Variable

## Procesos a las entidades

- **Picture:** Variable numérica real, posiblemente utilizada para animación o visualización.
- **Animation:** Cadena de texto que podría indicar el estado animado o visual de una entidad.
- **ME\_Canal:** Número entero que identifica el canal por el que fue realizada la orden (ej. 1 = Teléfono, 2 = Web, 3 = App).

- ME\_Region: Número entero que representa la región de origen del pedido (ej. 1 = Sur, 2 = Norte).
- ME\_Cantidad: Número de pizzas solicitadas en una orden.
- ME\_TCanal: Tiempo total que tomó el canal en procesar la orden (probablemente en minutos o segundos).
- ME\_SCanal: Texto que describe el canal en forma legible (ej. "Teléfono", "Web", "App").
- ME\_CH: Contador de pizzas de tipo Hawaiana vendidas.
- ME\_CS: Contador de pizzas de tipo Supreme vendidas.
- ME\_CSC: Contador de pizzas de tipo 5 Carnes vendidas.

State Variables	
 Picture	Real State Variable
 Animation	String State Variable
 ME_Canal	Integer State Variable
 ME_Region	Integer State Variable
 ME_Cantidad	Integer State Variable
 ME_TCanal	Real State Variable
 ME_SCanal	String State Variable
 ME_CH	Integer State Variable
 ME_CS	Integer State Variable
 ME_C5C	Integer State Variable

# CONCLUSIONES

- Durante la simulación se evidenció una diferencia considerable entre el total de órdenes generadas y las que finalmente fueron entregadas, lo que indica que existen pérdidas o interrupciones en el flujo del sistema que afectan la eficiencia general del proceso.
- Los recursos destinados a la preparación de pizzas mostraron una alta utilización, mientras que los encargados del empaque presentaron una ocupación significativamente baja, lo que sugiere un desbalance en la distribución de la carga de trabajo entre las distintas áreas del sistema.
- Se observaron niveles elevados de acumulación en los buffers previos a la cocina, lo cual refleja que esta etapa enfrenta una alta demanda que no siempre puede ser procesada con fluidez, generando tiempos de espera importantes para los pedidos.
- El área de verificación funcionó con una carga de trabajo considerable, lo que podría estar generando demoras en la validación de órdenes y afectando el flujo hacia los procesos posteriores, convirtiéndose así en un posible cuello de botella.
- El tiempo total que una orden pasa dentro del sistema sugiere que, aunque las etapas individuales son eficientes, existe espacio para optimizar la coordinación general de las actividades con el fin de reducir la permanencia y mejorar el nivel de servicio.

# MEJORAS DEL SISTEMA

- El análisis del recurso de verificación muestra una utilización del 81.3% y un tiempo de espera promedio en el *input buffer* de 0.04669 minutos, con un máximo de hasta 15 órdenes en cola, lo que indica que esta estación opera cercana a su límite de capacidad. Dado que este proceso es obligatorio para todas las órdenes antes de pasar a cocina, se convierte en un cuello de botella crítico. Para aliviar esta carga, se propone duplicar el recurso de verificación o dividirlo en múltiples agentes, lo cual permitiría atender más órdenes en paralelo y reducir los tiempos de espera en esta etapa, disminuyendo así la acumulación general del sistema.
- Los dos recursos de empaque presentan una utilización promedio del 8%, y el contenido en sus input buffers es prácticamente nulo (promedio de 0.00044 y 0.00009 entidades), lo cual indica una infrautilización significativa de esta área. Esta baja carga se explica en parte por la discrepancia entre la cantidad de pizzas producidas (con alto tráfico en cocina) y las que finalmente son empacadas y enviadas. Una mejora viable consiste en reducir a un solo recurso el proceso de empaque, y reasignar el recurso liberado a otra etapa más demandante del sistema, como verificación. Esta estrategia optimiza la utilización global de los recursos sin afectar la eficiencia operativa del empaque.
- Los cinco cocineros tienen una utilización superior al 96%, lo que muestra alta demanda en esta área. Sin embargo, los buffers de entrada de cada cocinero presentan acumulaciones significativas, con promedios de más de 33 órdenes y máximos de hasta 76. Actualmente, las órdenes parecen distribuirse de forma uniforme sin considerar la especialidad o cantidad de pizzas. Se propone implementar una lógica de priorización basada en tiempo de llegada o tipo de orden, que permita procesar primero las órdenes más sencillas o urgentes, reduciendo así el tiempo promedio de permanencia en el sistema (actualmente de 4.67 minutos para pedidos entregados). Esto mejoraría el flujo de trabajo y la percepción de rapidez del servicio.