

[193046] Diseño de un sistema de control para AGVs a utilizar en el marco de una emulación en un sistema de almacenamiento y recuperación automática (AS/RS).

Andrés Camilo Rincón<sup>c</sup> , Daniel Rolando Rodríguez<sup>a,c</sup> ,  
Yenny Alexandra Paredes<sup>b,c</sup> , Jose Fernando Jiménez<sup>b,c</sup>

<sup>a</sup>Estudiante de Ingeniería Industrial

<sup>b</sup>Profesor, Director del Proyecto de Grado, Departamento de Ingeniería Industrial

<sup>c</sup>Pontificia Universidad Javeriana, Bogotá, Colombia

---

## Resumen

In recent decades, technological systems have played an influential role in companies' competitiveness. These systems have reduced execution time. Automated logistics, specifically in storage and collection systems, has provided customers with prompt responses, while having a better performance within the tasks of a company. In the modern manufacturing industry, the time spent on material transport represents almost 80-90% of the total time of verification, incurring more than 30% of the total cost of the process. For this reason, this paper proposes the design of a control system for AGVs to be used in the framework of an emulation in an automatic storage and retrieval system (AS/RS), which carries out the collection of packages in a way that optimizes their collection, concurring with minimization in the response time in cases of disturbance. For the emulation, a control architecture based on the multi-agent paradigm will be used, and it will be parameterized with a predictive-reactive decision model. Specifically, an optimization tool will be used for collection scheduling, as well as reactive control modeling to respond to possible disturbances. Different scenarios will be considered to validate the response of the model to each situation, emulating the process in a virtual environment.

---

## 1. Justificación y planteamiento del problema

La competitividad de las empresas en las últimas décadas se ha visto complementada por sistemas tecnológicos que les han permitido optimizar sus procesos. Lo anterior se ve reflejado en la reducción de esfuerzos físicos de los trabajadores, y a su vez, en un menor tiempo de operaciones. Actualmente en la industria, la logística automatizada en el almacenamiento y recolección de materiales ha brindado una ventaja en el tiempo de respuesta al cliente; lo cual se debe a que los robots pueden ejecutar tareas de manera precisa sin descanso. Además, incrementa la capacidad de procesamiento, llevando a una mejor sincronización en el desarrollo de tareas de suministro (Meneses, Velasquez, y Grisales, 2014).

La metodología clásica para la manipulación de cargas, y transporte de estas ha empleado sistemas mecánicos fijos y móviles, como son las cintas transportadoras de rodillos, bandas, transpaletas, apiladores, carretillas elevadoras, entre otros; sin embargo, dichos sistemas se encuentran limitados dada su forma, proceso y esfuerzo físico requerido algunas veces. En la industria manufacturera moderna, el tiempo empleado en el transporte de material representa casi el 80-90 % del tiempo total de verificación de rutas, incurriendo en más del 30 % del costo total del proceso; de acuerdo con Escudero (2014), la eficiencia de fabricación y los costos pueden estar directamente influenciados por el nivel de automatización del equipo de transporte de material.

Existen algunos sistemas automatizados de recolección y almacenamiento en bodega que permiten un control adicional en cada artículo transportado y garantizan facilidad en el manejo de dimensiones considerablemente grandes de inventario. Entre ellos está el sistema AS/RS (Sistemas de Almacenamiento y Recuperación Automática por sus siglas en inglés). Este sistema es controlado y manejado desde una computadora, donde el

mecanismo de almacenamiento y recuperación se mueve entre las estanterías, y tiene como objetivo reducir al máximo la intervención humana, minimizando errores y accidentes en el proceso (Rodríguez y Martín, 2010).

Son diversos los tipos de ASRS, entre ellos, los de carga de unidad tienen sistemas grandes para el manejo de cargas sobre contenedores o pallets, y son totalmente automatizados y rápidos adquiriendo la carga. Otros son los ASRS Depp lane, que almacenan en una sola ubicación varias cantidades de materiales. Estos son diseñados especialmente para flowthrough, que es una estantería inclinada levemente para que por gravedad bajen los artículos almacenados, manejando inventario LIFO (último en entrar, primero en salir). También se encuentran los sistemas de recuperación automatizada de artículos en los que se recuperan materiales por unidad, estos materiales se van empujando unos a otros cayendo sobre la banda transportadora la cual entrega los materiales en el punto determinado, los artículos tienen un sistema de inventario FIFO. Además, están los almacenamientos de levantamiento vertical que funciona como un pasillo horizontal para acceder a todos los materiales o artículos. Finalmente, se encuentran sistemas de control de AGVs, donde los robots se comunican mediante sistemas de red, se mueven a lo largo de una cuadrícula encima de pilas de contenedores, cuya función es levantar y transportar cargas ligeras hasta una estación de picking y almacenar nuevos contenedores (Laurence y Tobón, 2013).

El sistema de control de AGVs es una alternativa que permite recuperar y ubicar objetos de inventario utilizando dispositivos mecatrónicos acoplados a su estructura, razón por la cual pueden actuar como robots en el entorno de un ASRS. Los vehículos de guiado automático constituyen un sistema de carros de arrastre centrados en trasladar los materiales desde el punto de fabricación hasta el de almacenaje sin la intervención directa de personas. Como vehículos de transporte, tienen capacidades de programación separada y gestión unificada a través del sistema de control. El AGV puede correr a lo largo de las rutas planeadas, de acuerdo con las instrucciones, llegar al destino y completar una serie de tareas previstas, tal como se muestra en la figura 1. Este sistema, integra tecnologías clave como, por ejemplo, recopilación y procesamiento de información, adquisición de información de guía, localización precisa, comunicación inalámbrica y control de movimiento del vehículo (Kelvin, J y Arcentales, M, 2018).



Figura 1: Sistema de control para AGVs en el marco de un ASRS. Fuente: Robotics & Automation News

Considerando la versatilidad y flexibilidad de este tipo de sistemas, esta investigación propone diseñar un sistema de control para AGVs a utilizar en el marco de un ASRS a través de una emulación. Estos dispositivos logran un debido funcionamiento a través de sensores y diversos medios electrónicos para guiarse por rutas

preestablecidas o programables dependiendo de las tareas que deban realizarse y de las perturbaciones que puedan ocurrir, logrando así mismo un cubrimiento total de recorridos obligatorios y una reacción inmediata.

La emulación del sistema de control de AGVs diferentes variables en el entorno, haciendo de un vehículo, un agente múltiple con automatización. Así mismo, se deben tener en cuenta técnicas para establecer rutas y restricciones que darán vida a un sistema simulado en la recolección y almacenamiento en bodega. El objetivo de esta investigación es proponer el diseño de un sistema de control para AGVs a utilizar en el marco de una emulación de un AS/RS, que considera ciertas rutas predeterminadas, y probarlo con algunas perturbaciones que se podrían presentar en un espacio real determinado como la capacidad de reacción ante el daño de un vehículo. De acuerdo con lo anterior es pertinente proponer un sistema de control para un escenario previamente planeado y de igual manera capaz de responder ante imprevistos minimizando el tiempo de recolección de las órdenes.

## 2. Antecedentes

El entorno mundial de la industria está en continua transformación debido a la utilización de nuevos recursos, el cambio climático y tecnologías e innovaciones dinámicas (Lins, Oliveira, Correia, y Sa Silva, 2018). De esta manera, nuevas necesidades aparecen en los sistemas productivos. Existen conceptos modernos como lo es el de la Industria 4.0, la cual se basa en la utilización de sistemas ciber-físicos, computación en la nube en las fábricas e internet de las cosas, lo cual permite recogida de datos, toma de decisiones en tiempo real, control total y máxima eficiencia en todos los procesos de la cadena de valor (Lins et al., 2018).

Una ventaja es que la migración a la Industria 4.0 puede ser menos costosa y más incisiva con el uso del concepto de adaptación, que es la actualización y adaptación de equipos industriales a las nuevas tecnologías. Puede existir una readaptación, siendo ésta la forma más rápida y rentable de llegar a la Industria 4.0, incrementar la eficiencia del equipo, reducir el costo de producción y aumentar la conectividad de la industria. Los resultados después de una actualización son medibles y también incluyen la vida útil del equipo y su inserción en la primera etapa de la Industria 4.0, caracterizada por la conectividad y la consiguiente utilización de los datos capturados en beneficio de la productividad (Lins et al., 2018).

La aplicación no sólo es evidente en la industria, sino en todo tipo de organizaciones; por lo tanto, se está frente a una revolución digital en donde está presente la hiperconectividad de las redes, los avances científicos en bio, nano o nuevos materiales. Además de la globalización, se estará frente a una transformación de la economía y ante un cambio mucho mayor que el de las revoluciones industriales previas (Carro, Flores, Flores, Hernández, 2019). Es así, que la autora Jacobsen (2011), expone que, en la actualidad, la automatización es más flexible y adaptable a los cambiantes requisitos operacionales y es que los AGVs proporcionan flexibilidad, mejoran los procesos en las compañías, reducen costos y facilitan la utilización de servicios para los clientes de una manera simple y práctica.

Partiendo de lo anterior el concepto de sistemas automatizados de almacenamiento y recuperación (AS/RS), es un factor fundamental y combinado con vehículos de guiado automatizado forman una composición difícil de superar. La relación entre traslado y la correcta ubicación de los productos en los puntos de reorden, muestran en la industria una buena relación entre el ruteo y la zona de picking, dando lugar a una creación de AGV como carretilla con el proceso interno de asignación del AS/RS. Tanto la movilidad como la asignación de las tareas están conjuntas en los AGVs de nueva generación y mantienen de esta manera la eficiencia en la productividad que caracteriza a los AS/RS (Schwind, 1998).

Entre los principales sistemas automatizados de almacenamiento y recuperación se encuentran los AS/RS de carga única, evidenciados generalmente como sistemas de elevación media anclados a los racks, encargados de realizar la extracción del paquete seleccionado. Este sistema tiene ciertas desventajas, ya que prohíbe la

movilidad de las torres debido al anclaje que genera poca flexibilidad de cambios en el sistema. sin embargo, también genera una eficiencia en los tiempos, pues se especializan los pedidos por punto de recolección (Knill, Schwind, & Witt, 1993). A diferencia del AS/RS único, el Palé está enfocado a cargas donde la capacidad media sobrepasa constantemente los requerimientos de pedido. En este sistema la sobrecarga que genera el transportador puede llegar a causar fallas, por lo que en la implementación el sistema de Palé para cargas elevadas o de magnitudes superiores es más eficiente. Por último, el sistema de control por operario es el menos eficiente, pues requiere hablando de costos, costos adicionales y en ocasiones tiempos de retardo en la recolección de paquetes (Kator, 2007).

Jacobsen (2011), muestra una postura donde se resalta una herramienta importante para las compañías porque anteriormente la reducción de costos se centraba en la optimización de procesos de producción, pero ahora con la implementación de AGVs se está logrando obtener menores costos en otras áreas que no están relacionadas directamente con los productos, sino que hacen parte de la cadena de una forma indirecta como son las áreas de instalación, envío, recepción y almacenamiento (Jacobsen, 2011). Cabe recalcar que las participaciones de los vehículos de guiado autónomo con las condiciones ya mencionadas y sus beneficios han sido comprobadas únicamente en empresas que aplicaron el sistema.

Partiendo de lo anterior, al buscar el objetivo de minimizar tiempos ya sea de recorridos totales en un sistema o simplemente de respuesta entre los AGVs, comienzan a implementarse diferentes tipos de técnicas para obtener la ruta más eficiente teniendo en cuenta restricciones que surgen o aquellas que los autores proponen para la elaboración del modelo. Dicho modelo puede ser inicializado ubicando los puntos críticos o puntos de partida de los AGVs para obtener una mejor respuesta de estos según las demandas de pedidos y la ruta que cada uno deba tomar (Gademann y Van de Velde, 2000).

Al delimitar los puntos o puertos de partida de cada AGV el paso a seguir es definir la cantidad de agentes necesarios para una correcta implementación. Los autores Tu, Qian, y Lou (2017) proponen una red de ruta guiada unidireccional, con el fin de poder controlar la cantidad de agentes que estén en un área determinada para evitar el embotellamiento de los AGVs y de esta forma poder delimitar el correcto número de vehículos, controlando los faltantes o demoras en el recorrido. Los autores plantean un algoritmo genético de nicho mejorado (INGA) que permite encontrar la cantidad correcta de vehículos dependiendo el área y las demandas a despachar.

Seguido del ajuste correcto de AGVs por bucle, los autores Fazlollahtabar, Saidi-Mehrabad, y Masehian (2015), Ventura y Lee (2008), y Xu, Guo, Li, Guo y Peng (2019) buscan minimizar el tiempo de recorrido entre nodos y el tiempo de recorrido total. Para lograr esto, los autores de los dos primeros artículos plantean un algoritmo de polinomios de tiempo para identificar la cantidad de AGVs por bucle y así conseguir la minimización de recorridos, y al igual que Tu et al. (2017) al obtener el número de vehículos correctos para un bucle, logran disminuir el tiempo de recorrido y los tiempos muertos.

Buscando el mismo objetivo de los autores ya mencionados, Liu, Zhong, Peng, Tian y Zou (2018) proponen un modelo de optimización de un algoritmo tabú y MILP, buscando siempre la minimización de recorridos. Se fortalece el modelo relacionándolo con un algoritmo híbrido entre genético y enjambre al igual que los autores Saidi-Mehrabad, Dehnavi-Arani, Evazabadian y Mahmoodian. (2015) que también desarrollan un algoritmo de colonia de hormigas (ACA) de dos etapas, (similar al enjambre) para buscar según la secuencia de soluciones, la mejor solución entre las encontradas.

El procedimiento continúa con el mejoramiento de la función objetivo (minimizar tiempos de recorrido) y es por esto que los autores Fazlollahtabar et al. (2015) realizan la búsqueda de soluciones iniciales factibles., las cuales van a caracterizar un punto de partida en el desarrollo del objetivo y a partir de esta se generan una

serie de soluciones verificando la factibilidad y comparándolas entre ellas mismas para encontrar la mejor solución entre las ya buscadas.

De igual forma, Ventura y Lee (2008) mejoran las soluciones iniciales y basan su teoría en el posicionamiento de los agentes, gracias a esto se enfocan en el concepto de bucles tanto únicos como en tándem. El bucle único genera una solución unidireccional, que para un caso real se considera ineficiente, pues cada vehículo tiene que hacer su ruta y la de los demás para evitar el choque entre ellos; sin embargo, el bucle tándem divide el área en sub áreas identificando los AGVs que tienen que estar en cada una de ellas y ser responsables de los pedidos que se realicen en las mismas teniendo en cuenta que, a diferencia del bucle único, el bucle tándem logra presentar movimientos bidireccionales (Castillo, Reyes y Peters, 2001). De esta forma, Ventura y Lee (2008) generan un algoritmo de polinomios de tiempo con un solo AGV como prueba y van aumentando el número de vehículos generando mayor dificultad en las intersecciones y en la toma de decisiones del modelo.

Partiendo de la creación de una heurística, que minimiza los tiempos de recorrido y los tiempos de respuesta, el paso a seguir para algunos de estos autores es la implementación simulada para recrear el modelo ya formulado y poder verificar los resultados. Entre los autores ya mencionados se encuentran Ventura y Lee (2008) los cuales desarrollan 4 simulaciones con diferentes localizaciones de los bucles, un único pedido por estación y de un solo paquete cada uno con el objetivo de no incrementar la complejidad en el estudio. A diferencia de Ventura y Lee. (2008), Xu et al. (2019) presenta una simulación para el algoritmo con el fin de identificar la cantidad de AGVs pertinentes en la implementación y demuestra las características de movilidad y las dificultades que puede conllevar tener mayor o menor cantidad de agentes de los estipulados en la simulación, ya que, si son menos, la ruta de cada AGV es más larga, y si son más el tiempo ocioso aumenta considerablemente. Los autores Xu et al. (2019) desarrollan cinco experimentos con el fin de identificar las tareas máximas o restricciones que puede tener cada uno de los AGVs para no forzar el modelo y hacerlo eficiente. Se han utilizado múltiples AGVs para transportar tantos paquetes como sea posible en un momento dado; sin embargo, el autor llega a la conclusión partiendo de la simulación de que pueden surgir colisiones y puntos muertos cuando dos o más AGVs intentan pasar por un área compartida al mismo tiempo. Para resolver este inconveniente, los autores Sun, Gu, Wan, Huang, y Jia (2018) proponen un algoritmo de generación de ruta, de modo que se puedan minimizar las colisiones y los puntos muertos, luego, se basan en la construcción de un modelo de red de Petri temporizado orientado a recursos (CROTPN) para cambiar dinámicamente las rutas de AGVs después de dividir el período de programación en intervalos de tiempo (Sun et al., 2018). Los resultados de los experimentos muestran que el método puede lograr un alto rendimiento de clasificación utilizando la menor cantidad de agentes posibles.

Dentro de los autores que han abordado los cuestionamientos relacionados con arquitecturas de control desde la simulación agentes, se encuentran Estrada, Ramírez, y Uribe (2019), quienes proponen el diseño de un sistema automático de recolección y suministro (AS/RS) de producto en bodega, generando un modelo predictivo-reactivo basado en agentes que simulan el control del sistema. Dentro de las fases establecidas, existió la programación de un modelo matemático predictivo para medir el comportamiento de la entidad global con base en instancias más pequeñas. Seguido de esto, se llevó acabado el modelo reactivo, en donde se delegó la responsabilidad de la toma de decisiones a los diferentes componentes de control, resultando así una arquitectura de control híbrida. Lo anterior involucra las operaciones de un agente global, el cual es el encargado de minimizar la tardanza en cada una de las trayectorias establecidas inicialmente. Seguido de esto, cabe mencionar la otra parte del modelo basado en multiagentes, los cuales constituyen un agente local encargado de realizar las tareas proporcionadas por el modelo predictivo inicial.

Un punto importante para resaltar es la pertinencia de dos tipos de modelamiento, uno global y otro local. El primero, hace referencia a las tareas de los AGVs, como la destinación de productos a recoger; sin embargo, la interna o local hace referencia a las decisiones que puede llegar a tomar el agente en el trayecto como una interferencia o una irregularidad según el trazo de rutas predeterminado (Saidi-Mehrabad et al., 2015). De

acuerdo con la metodología empleada, los autores Saidi-Mehrabad et al. (2015) proponen un algoritmo ACA (Algoritmo de colonia de hormigas), en donde se busca la minimización del tiempo de ejecución de los trabajos utilizando AGVs con rutas de procesamiento diferentes. Las rutas y las órdenes son transmitidas a cada vehículo por parte de un agente global, sin embargo, después de asignar un trabajo y una ruta a un AGV, ese AGV no es asignado a otro trabajo hasta que el anterior termine su orden, haciendo que nunca colisionen. En este punto la literatura lo define como un enrutamiento predictivo libre de conflictos.

Ahora bien, dentro del contexto de un modelo reactivo presente en diversos entornos y en el marco de un AS/RS, Muñoz Alcázar (2019) hace referencia a la utilización de un gemelo digital para representar aquellos agentes reactivos que ejecutan el ruteo enviado por el agente global, en donde es necesario tener una representación gráfica de lo que realmente está sucediendo. Teniendo en cuenta lo anterior, un gemelo digital consiste en una representación digital de un sistema físico siguiendo la máxima fidelidad posible. Se puede utilizar con varios propósitos aprovechando la sincronización en tiempo real de los datos recolectados originados en el sistema físico, lo que le da una ventaja de tomar decisiones sobre un conjunto de acciones con el objetivo de estructurar y asegurar el funcionamiento del conjunto del sistema de una forma óptima.

De acuerdo con Muñoz Alcázar, J., (2019), la representación digital debe incluir toda la información posible sobre el sistema físico que sea relativa a su funcionamiento y que influya en este de manera óptima. Además, la relación entre la representación digital y el sistema físico está dada por el flujo de datos en donde si está totalmente automatizado nos encontramos ante un gemelo digital. Con esta combinación, el objeto digital puede actuar también de controlador del sistema físico, sin embargo, puede haber otros objetos, físicos o digitales que actúen sobre el objeto digital.

Finalmente, en los autores consultados, la mayoría de sus aportes están dirigidos al planteamiento y solución del problema de la minimización de tiempos de recorrido o tiempos de respuesta de los AGVs, es por esto que, por su aporte enfocado netamente al direccionamiento y de manera predictiva de las rutas, se tendrán en cuenta los artículos que renombren las heurísticas y metaheurísticas como parte de la solución propuesta por los autores para la respuesta a la minimización de los tiempos. Del mismo modo, para la simulación del sistema, se tomarán como referencia los artículos propuestos por los autores Muñoz Alcázar, J. (2019), y a su vez, a Estrada Bernal et al (2019), ya que plantean conceptos importantes de acuerdo con arquitecturas de control referentes al experimento propuesto, proponiendo soluciones para la emulación de las rutas de recolección. Estrada et al. (2019) en su trabajo, diseñan un modelo basado en multiagentes utilizando como herramienta el software NetLogo, en el cual se trabajará para la simulación del modelo predictivo-reactivo.

### **3. Objetivos**

#### **Objetivo General**

Diseñar un sistema de control para AGVs a utilizar en el marco de una emulación en un sistema de almacenamiento y recuperación automática (AS/RS), que permita programar la recolección de paquetes minimizando el tiempo de reacción considerando perturbaciones.

#### **Objetivos Específicos**

1. Diseñar el modelo de despacho y enrutamiento de AGVs dentro de un AS/RS.
2. Validar un sistema de control del AS/RS con emulaciones basadas en la simulación de agentes.
3. Probar el desempeño del sistema propuesto y el modelo de despacho de enrutamiento de AGVs, considerando posibles perturbaciones.
4. Realizar una comparación de los resultados del sistema de control de AGVs propuesto con un sistema de control basado en reglas de despacho.

## 4. Propuesta de emulación de un AS/RS utilizando AGVs

### 4.1. Declaración de Diseño

El diseño que abarca este trabajo trata de diseñar un sistema de control para AGVs a utilizar en el marco de una emulación en un sistema de almacenamiento y recuperación automática (AS/RS), que permita programar la recolección de paquetes minimizando el tiempo de reacción considerando perturbaciones. Cabe aclarar que la intención es dejar listo el modelo para posteriormente probarlo en una emulación, sin embargo, para el desarrollo se trabaja en una simulación del mundo físico.

El proyecto presenta la integración de dos partes, primeramente, la metaheurística (diseño teórico), y en segunda instancia, la aplicabilidad en condiciones reales del modelamiento de rutas previamente obtenido, a través de una experimentación. El modelo recibirá como parámetros de entrada, la cantidad de AGVs que se pueden encontrar en la planta y su distribución, identificando así las distancias que hay entre puntos de recolección. Del mismo modo, parámetros como tiempos por distancia recorrida y cantidad de AGVs en un bucle. En la aplicabilidad los parámetros de entrada son prácticamente los mismos teniendo en cuenta que la velocidad del vehículo se asumirá constante y por lo mismo pueden llegar a variar un poco los tiempos entre la experimentación y el modelamiento.

El resultado que se espera obtener es la minimización de tiempos de las rutas por AGV y así el tiempo de finalización de recorrido por cantidad de pedidos. También se busca comparar el modelo predictivo con el modelo reactivo y observar sus variaciones.

### 4.2. Metodología

Inicialmente es importante definir algunos conceptos que se tuvieron en cuenta para el diseño del sistema de control empleado. Además, se describirá la arquitectura de control propuesta, mencionando cada uno de sus componentes y la comunicación que tienen entre ellos.

En primera instancia, se aborda el concepto de agente, el cual se define como una entidad física o virtual que puede percibir su entorno a través de sensores, evaluando dichas percepciones y tomando decisiones (Morales, 2008). Dentro de los agentes que componen esta propuesta, inicialmente se encuentra el agente global decisional, el cual proporciona el modelo predictivo y es el encargado de transmitir instrucciones, órdenes y rutas establecidas para cada agente local, minimizando el tiempo de recolección, llevando así la información a los agentes producto y AGV virtuales. Seguido de esto, participan los agentes locales virtuales quienes almacenan la información definida por el agente global y cumplen la asignación de instrucciones y órdenes. Finalmente se encuentran los agentes producto y AGV físicos, quienes estarán conectados a los virtuales y reciben la información que estos envíen, permitiendo la visualización de los movimientos enviados.

Existen características propias de los agentes locales inteligentes, como lo mencionan los autores Ángel, Moreno, Pérez y Vargas (2019). Dentro de estas, se encuentra la proactividad, en donde los agentes presentan un comportamiento dirigido a objetivos, tomando la iniciativa de satisfacerlos. También tienen habilidad social, interactuando con otros agentes incluyendo la percepción de su entorno de manera reactiva. En la siguiente figura podemos observar las características de los agentes utilizados en esta investigación.

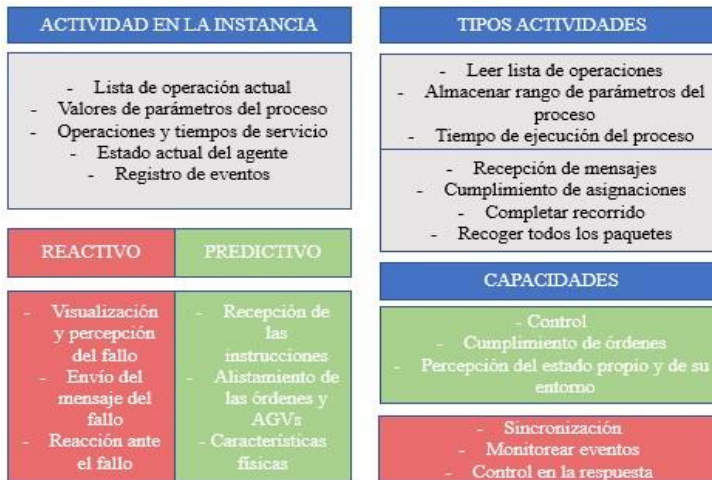


Figura 2: Características de un agente. Fuente: Los Autores

En segunda instancia, de acuerdo con la relación entre los agentes físicos y virtuales, en este diseño se propone una comunicación basada en el concepto de gemelo digital. Esta estructura consiste en la representación digital de un sistema físico que sigue fielmente el funcionamiento de un sistema virtual, sincronizando en tiempo real los datos recolectados de acuerdo con Muñoz Alcázar, J., (2019). Para esto, se construyó un mundo virtual y un mundo físico, en donde este último permite visualizar el entorno del sistema, los productos físicos y los AGVs con sus respectivos movimientos. El modelo virtual será aquel que ejecute cada instancia que se quiera ver reflejada de manera física.

En la figura 3, se puede observar el agente AGV local virtual y su respectivo agente local físico. Como el flujo de datos entre el sistema físico y su reflejo digital está totalmente automatizado, encontramos que este conjunto es el gemelo digital donde el agente físico replica los movimientos del agente virtual en tiempo real. El agente físico proporcionará datos frente a algún incidente que pueda ocurrir, y reaccionará gracias a las decisiones que el virtual le envía para que replique su comportamiento. Esta es la manera en que el gemelo digital puede cumplir objetivos a nivel local como la recolección total de las órdenes y completar un recorrido. De igual manera, podrá cumplir objetivos que contribuyen a nivel global con indicadores como el porcentaje de utilización, tiempo total de recolección y demanda satisfecha.

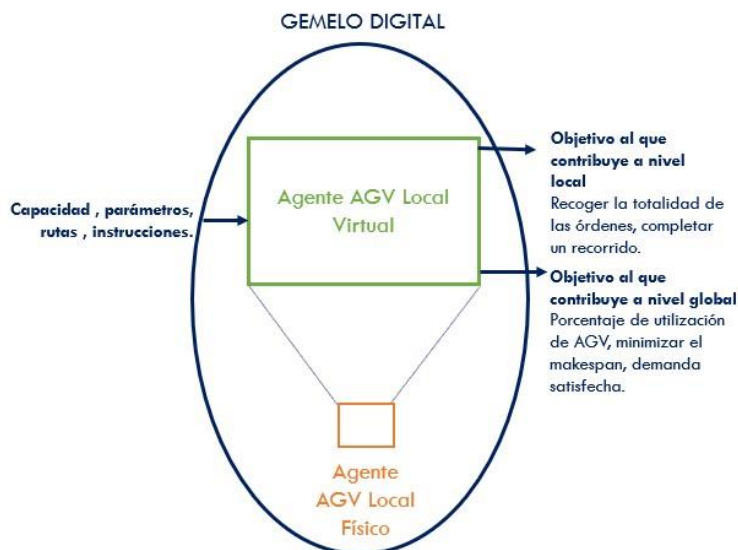


Figura 2: Comunicación gemelo digital de un AGV. Fuente: Los Autores



El diseño del sistema de control definido en este trabajo se basa en la programación de un modelo predictivo-reactivo, en donde la primera etapa no se enfoca en prever ningún tipo de imprevisto, difiriendo de la segunda en donde el entorno es cambiante y los agentes deben ser capaces de reaccionar para cumplir los objetivos. Cabe resaltar, que la arquitectura de un sistema de control puede considerarse como el estándar más importante que debe establecerse ya que coordina y dirige el manejo de piezas y actividades de procesamiento en cualquier tipo de modelo. El sistema de control, por lo tanto, incorpora las responsabilidades de toma de decisiones, incluida la programación parcial, el enrutamiento parcial, y asignaciones de recursos dentro de cada etapa del proceso (Dilts, D. M., Boyd, N. P., & Whorms, H. H, 1991).

De acuerdo con el concepto definido por Dilts & Whorms (1991), y considerando la simulación de agentes, se propone desarrollar inicialmente una arquitectura de control jerárquica como se ve en la figura, donde participa agente global decisional, el cual envía la información de manera directa a los agentes producto y AGV virtuales como lo representan las líneas azules (Figura 4) y estos de igual forma envían las instrucciones a los agentes físicos que replican estos movimientos. Lo anterior, compone el modelo predictivo sin perturbaciones.

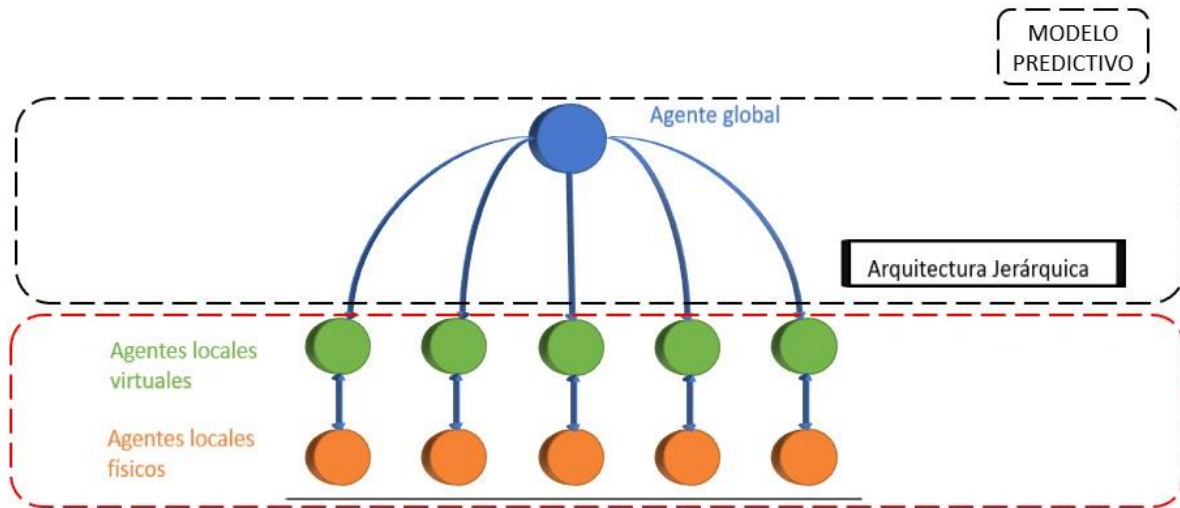


Figura 4: Modelo Predictivo. Fuente: Los Autores

Ahora bien, para la parte reactiva, la estructura cambia y pasa a ser heterárquica entre los agentes locales, en donde se da una negociación colaborativa representada por las líneas violeta, para responder ante las posibles perturbaciones que se puedan presentar en la ejecución de sus tareas y que no se tienen en cuenta en un entorno virtual (figura 5). En dicha hipótesis como lo proponen Gasquet & Romano (2008), se hace referencia a las condiciones y restricciones necesarias para elaborar un modelo reactivo, en donde al existir un programa en curso se decide ejecutar un nuevo programa que lo sustituya cuando se presente un imprevisto.

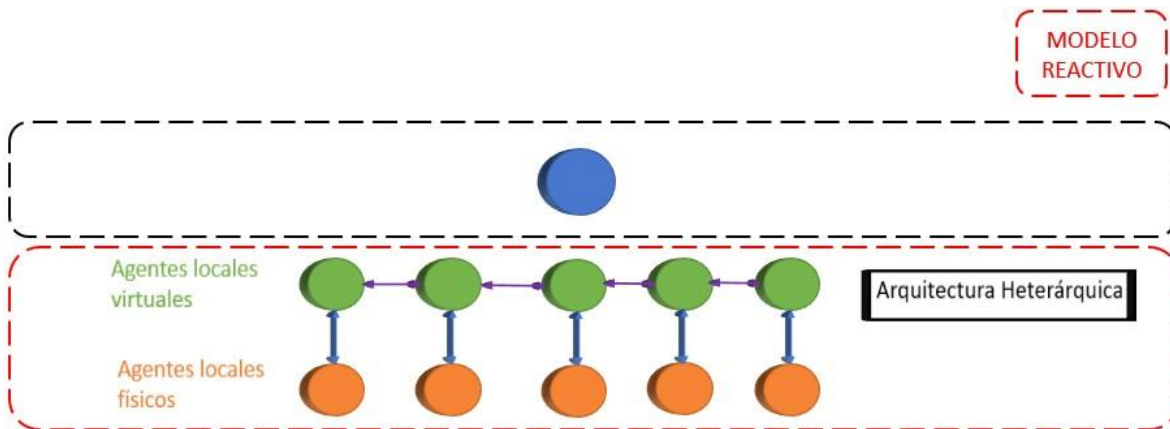


Figura 5: Modelo Reactivo. Fuente: Los Autores

En otra instancia, este diseño abordó la estructura observada en la figura 6, teniendo un orden lógico que permitió la ejecución y visualización eficiente del experimento. En esta figura se pueden evidenciar los componentes de la estructura empleada, así como los eventos en los que se ven involucrados con las respectivas entradas y salidas. Inicialmente existe un operario que debe accionar el botón para dar inicio al modelo, luego de esto encontramos el modelo predictivo definido por un agente global como se mencionó anteriormente, en donde están presentes parámetros fijos como la capacidad, los pedidos y la cantidad de AGVs. Estos, representan las entradas correspondientes a la creación de un mundo virtual en donde se almacena toda la información necesaria y las decisiones a ejecutar. En esta parte, hay una visualización de las rutas definidas por el agente global en el mundo virtual que gracias a una representación gráfica podrán ser observadas, y es ahí donde se da paso a la creación del mundo físico.

El mundo físico tiene como entradas la creación de aquellos componentes del mundo virtual, quienes conforman un grupo de agentes que serán los protagonistas del experimento. De este modo, aparece primeramente un agente AGV virtual, que almacenará las rutas definidas por el agente global, y después, aparece un agente de pedido virtual que representa las órdenes que serán recogidas. Estos dos tipos de agente tienen gemelos digitales físicos, que replican sus movimientos y hacen parte del mundo físico.

Referente a la segunda parte de la estructura, se busca dar reactividad al modelo, pero para esto es necesario añadir nuevas restricciones al mundo virtual y apoyarse de los beneficios que ofrece un gemelo digital. De acuerdo con el comportamiento del mundo virtual, cuando existe un fallo del AGV, se hace un reconocimiento de esta perturbación por parte del agente virtual y es visualizada gracias al agente físico. Con esta información el AGV virtual puede generar una reacción para dar respuesta oportuna, y proporcionar una visualización del fallo por parte del agente AGV físico. Esta respuesta se evidencia en la generación de un algoritmo de ruteo Dijkstra que permite a los AGVs físicos cumplir el objetivo de recoger todos los pedidos en un menor tiempo. En esta parte, existe una señal enviada desde el mundo virtual al mundo físico, donde se puede observar gráficamente como es la recogida de cada pedido.

Finalmente, en el mundo físico se observan también los nuevos movimientos de cada AGV, siguiendo el nuevo modelo de ruteo donde se hace la recolección de pedidos faltantes por el fallo. Dentro del mundo físico también existen ciertos indicadores que permiten observar en tiempo real si dicha recolección se está ejecutando.

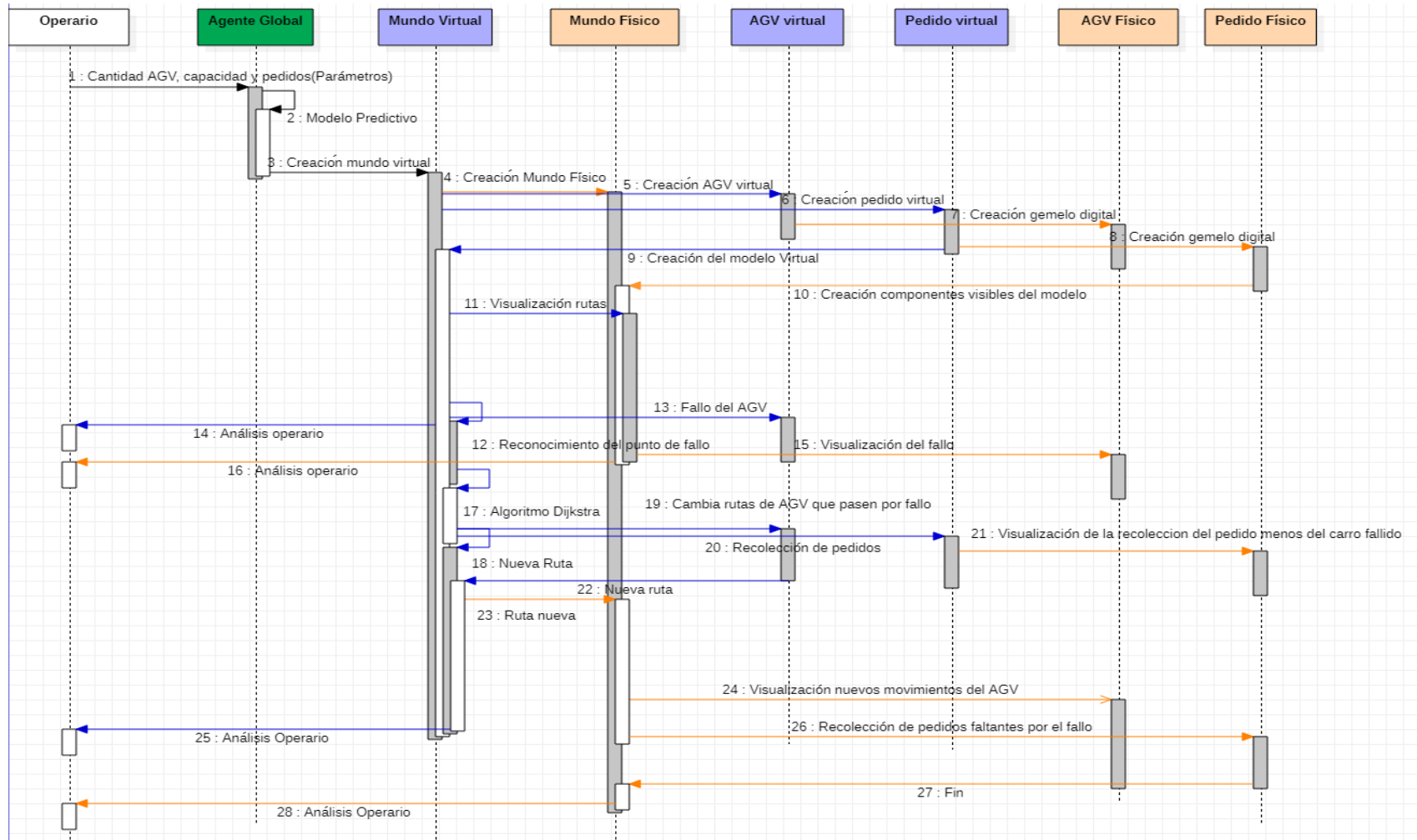


Figura 6: Diagrama UML del sistema de control para AGVs en un AS/RS (Ver anexo 1. Diagrama UML). Fuente: Los Autores

## AGENTE GLOBAL: MODELO PREDICTIVO

Con la intervención de los autores Estrada Bernal, Ramírez Loza, y Uribe Espinoza (2019) partimos hacia el concepto de la predictividad o solución predictiva, la cual para este caso logra encontrar un ruteo basado en agentes y el cual será la base de construcción del modelo reactivo. En la predictividad es necesario encontrar una buena solución al problema de enrutamiento de los AGVs, pues lo que se busca es minimizar los tiempos totales de recolección y entrega de productos (tal como pasa en la vida real), es así como se llega a un complemento entre la metaheurística de búsqueda la cual permite encontrar la mejor solución entre las exploradas y las Heurísticas de ruteo. El proceso de la predictividad en VBA se encuentra en la figura 7.

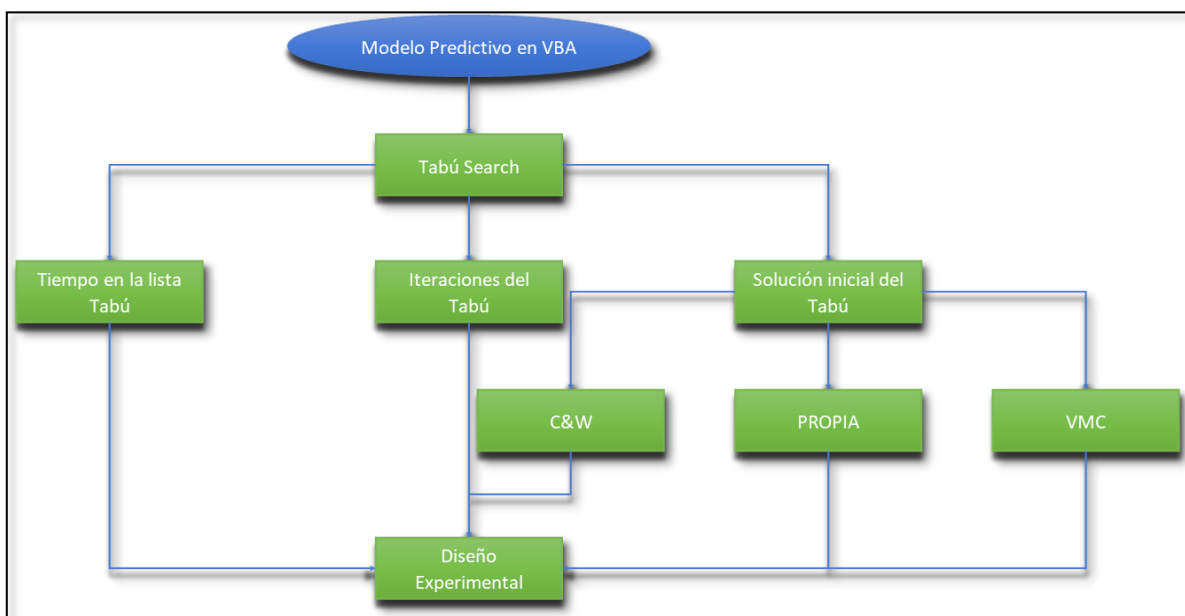


Figura 7: Flujograma Modelo Predictivo. Fuente: Los Autores

En la selección de la metaheurística más acertada para el modelo propuesto, se escogió la búsqueda Tabú ya que como se verá a continuación se parte de una solución inicial y lo que buscamos es mejorarla a diferencia de un algoritmo poblacional el cual busca un espectro global de soluciones. Partiendo de los autores Liu, Zhong, Peng, Tian y Zou (2018) los cuales desarrollaron un algoritmo Tabú para la solución del problema de la minimización de tiempo con rutas basadas en el movimiento de recolección y entrega de productos, se describe el mismo planteamiento en donde la globalidad del problema predictivo se resuelve en una Búsqueda Tabú.

Como punto de partida para la generación del Tabú es necesario encontrar una buena solución en el problema de enrutamiento ya que será la base para posteriormente encontrar una mejora. Asemejándolo con un problema logístico logramos encontrar tres soluciones para las trayectorias de los AGVs, entre estas tenemos la solución de la heurística Vecino más cercano, la heurística Clarke & Wright y una solución propia siendo estas definidas por parámetros escalables los cuales nos permiten recrear soluciones dependiendo de instancias generadas por el usuario, es así como la cantidad de AGVs de recolección es cambiante partiendo de la decisión del usuario, al igual que la capacidad de los agentes y la cantidad de nodos de demanda. Un punto para resaltar es la

generación de demandas, pues con el fin de asemejarlo a la realidad se presentan para cada nodo como un número aleatorio entre uno y la capacidad de los AGVs.

Al igual que los parámetros escalables se procede a la creación de parámetros fijos y supuestos del problema, entre estos se encuentran, la distribución de la planta o en otras palabras las longitudes matriciales siendo para este caso 30 posiciones en las coordenadas horizontales y 35 en las verticales. El nodo de partida de los carros estará situado en la posición (0,0) y del mismo modo la llegada al punto de embarque. Además, la capacidad de los carros sobrepasa la demanda individual de todos los nodos, es decir, el nodo con la demanda máxima o el pedido más grande podrá ser recogido por un AGV. La teoría nos indica que heurísticas como el C&W son soluciones que permiten la recolección de pedidos si el pedido es menor que la capacidad de los carros, gracias a esto se fundamenta el supuesto de capacidad.

La cantidad de AGVs esta defina como un parámetro escalable, aunque para las pruebas se representa como un parámetro fijo, pues las pruebas son realizadas siempre con cinco carros, sin embargo, se realiza un diseño experimental con el objetivo de definir un modelo de regresión que permita estimar el tiempo de ejecución de recorridos totales dependiendo la iteración y en esta la cantidad de vehículos. El diseño que se realizó es un diseño de bloques en donde el Factor es la cantidad de agentes y el bloqueo son las iteraciones. La ANOVA es concluyente al determinar que la cantidad de vehículos, las iteraciones y los supuestos son comprobables. (Ver Anexo 2. Diseño de experimentos cantidad AGV).

Partiendo de lo anterior se tratan la generación de los datos, siendo esta un modelo variable donde la demanda de la cantidad de nodos suministrados puede cambiar y del mismo modo la posición en donde está ubicado cada uno. Seguido de la generación de puntos se procede al cálculo de distancias para determinar en las heurísticas que tan alejados se encuentran un punto del otro, (El cálculo de distancias se desarrolla mediante el procedimiento de distancias rectilíneas o distancias de Manhattan). Para todas las heurísticas el primer punto que será recogido por cada AGV después de salir de la zona de descargue será un punto aleatorio dentro de la matriz solución, aclarando que este punto no fue visitado y recolectado en su totalidad en recorridos anteriores.

Ya conocidos los parámetros y supuestos, el paso a seguir es la ejecución de las heurísticas. Inicialmente se desarrolla una heurística C&W la cual es conocida por generar una matriz de ahorros que determina una buena solución, esta heurística se encuentra en la figura 8. Como segunda solución se desarrolla una heurística Propia pues combina conceptos de un VMC con modelos propios dando así una solución de segmentación, en la cual la recogida de un pedido puede ser dividida o recogida en varios viajes con el fin de llenar siempre la capacidad del carro, esta heurística se encuentra en la figura 9. Para terminar, se tiene en cuenta el desarrollo de un VMC el cual simplemente recoge el pedido que este más cercano si la capacidad del carro alcanza para recoger la totalidad del pedido, esta heurística se encuentra en la figura 10.

**C&W**

**Inicio**

**Si (existe demanda en el nodo) Entonces**  
 Crea la ruta con la que se inicializan todos los vehículos  
 Cuenta que capacidad a en el instante

**SiNo**  
 se bloquea la fila y la columna

**Fin si**  
 Inicializar matriz de ahorros

**Hacer**

**Si (no se ha bloqueado la fila) Entonces**  
**Si (no se ha bloqueado la fila ni la columna) Entonces**  
 busca el mayor ahorro

**Fin si**

**Fin si**

**Si (se/cumplen las restricciones) Entonces**  
 Se agrega a la ruta  
 Se bloquea la fila y la columna

**Fin si**

**Mientras que por lo menos una fila y columna estén desbloqueadas**

**Fin**

**Vecino Más Cercano**

**Inicio**

**Hacer**  
 Mínimo es igual a un número muy grande

**Si (mínimo es mayor que la distancia actual && no se ha visitado el nodo donde estamos ubicados) entonces**  
**Si (la Demanda puede ser satisfecha) Entonces**  
 Mínimo es igual a la nueva distancia  
 Cercano es igual a la posición en donde estamos ahora

**Fin si**

**Fin si**

**Si (cercano es diferente al nodo donde estamos ubicados) Entonces**  
 Donde estamos ubicados es igual al más cercano  
 Se visitó el nodo en donde estamos  
 La nueva posición en la ruta es la que se acaba de visitar

**SiNo si (todos los nodos no han sido visitados) Entonces**  
 Vaya a la siguiente posición

**SiNo**  
 Se termina

**Fin si**

**Mientras que no se haya terminado**

**fin**

Figura 8: Pseudo código C&W. Fuente: Los Autores

Figura 9: Pseudo código Propia. Fuente: Los Autores

**Heurística propia**

**Hacer mientras no se haya cumplido todo**

**Si (demanda es mayo que 0) Entonces**  
 no se ha cumplido todo

**Fin si**  
 Ruta = ruta + 1

**Si (cantidad de nodos ahora >= cantidad de AGVs) Entonces**  
 Se retorna de manera aleatoria cada AGV a un nodo diferente

**Si (demanda de un nodo es menor o igual a cero) Entonces**  
 No se asigna el nodo

**Fin si**  
 Actualizar la ruta

**Si (No se ha visitado el nodo de la ruta) Entonces**  
**Si (demanda del nodo es menor que lo que puede llevar el AGV en ese momento) Entonces**  
 El AGV se puede llevar toda la demanda

**SiNo**  
 Solo se puede llevar lo que alcance

**Fin si**  
 Lo que tiene el carro es igual a lo que tenía + la demanda que llevo  
 El nodo tiene una demanda de lo que tenía menos lo que el AGV se llevo

**Fin si**

**Mientras que no se haya cumplido todo Hacer**  
 Buscar el Vecino Más Cercano

**SiNo**  
 Sumar la demanda de los nodos  
 Dividir las demandas en la capacidad del carro  
 Enviar esa cantidad de carros a los nodos.

**Fin si**

**Fin**

Figura 10: Pseudo código VMC. Fuente: Los Autores

Teniendo soluciones iniciales para el problema a partir de las tres heurísticas, el paso a seguir es la construcción de la búsqueda tabú como estructura enfocada a mejorar la solución ya planteada. La búsqueda Tabú se basa en

cambios pequeños que determinen si la solución total mejora o no, es importante aclarar que la solución puede mejorar por diversos factores, entre estos el cambio de la primera posición a visitar. Ya sabiendo que la primera posición es aleatoria se puede producir un cambio en la misma y de esta forma mejorar la función objetivo. El Tabú recibe como parámetros por un lado la cantidad de iteraciones o cambios que se pueden hacer en un ciclo sin que esta mejore siendo prácticamente una restricción buscando no llegar a ciclos infinitos, y por otro lado el tiempo en la lista Tabú en donde se determina después de encontrar una mejor solución que tanto tiempo deberá permanecer este en la lista Tabú. A continuación, se encuentra el Pseudo Código del Tabú. (Ver Anexo 4. Pseudo códigos VBA).

```

Inicio
Inicializar Tabu
El mejor vector = vector de prueba
fobest = la nueva solución
Hacer
  Mejor cambio tabú
  Si (vectorprueba(i) <> vectorprueba(j)) Entonces
    Generar cambio
    Si (FOprueba < FOfinal) Entonces
      FOfinal = FOprueba
    Fin si
  Generar cambio
  Fin si
  Nuevafo = la nueva solución
  Si (nuevafo < fobest) Entonce
    Vectorbest = vector de prueba
    Fobest = nuevafo
  SiNo
    Iteraciones sin mejora = iteraciones sin mejora +1
  Fin si
actualizar tabu
Mientras que iteraciones sin mejora < iteración máxima
Fin /

```

Figura 11: Pseudo código Tabú. Fuente: Los Autores

Finalmente encontrada para un total de 5 vehículos, con capacidad 10 y cantidad de nodos de demanda 100 se procede a analizar cuál es la mejor heurística (factor C), la mejor cantidad de iteraciones (factor B) y el tiempo en la lista Tabú (factor A) con el fin de identificar si existe efecto de los factores sobre la variable respuesta, dando lugar así a un diseño experimental factorial  $3^3$  (Ver Anexo 5. Diseño de experimentos  $3^3$ ).

## SIMULACIÓN DEL MODELO PREDICTIVO

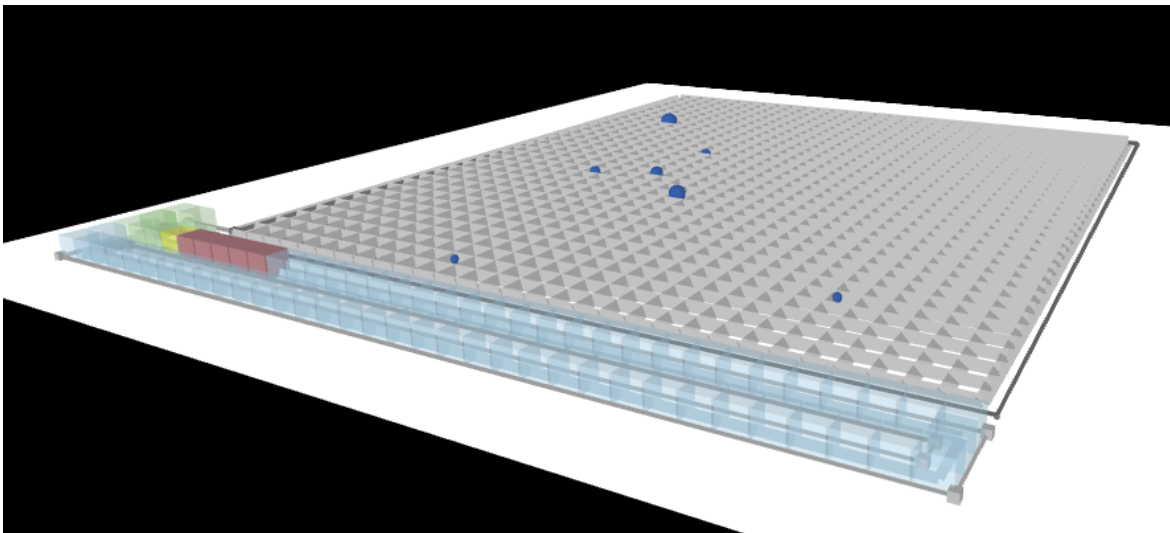
Para poder observar el comportamiento del modelo predictivo de manera gráfica se diseñó un simulador en NetLogo, en donde se probaron cada uno de los escenarios desarrollados en la predictividad, logrando evidenciar un modelo sin perturbaciones.

El simulador se compone de dos archivos diferentes que son un mundo virtual y un mundo físico, en donde el primero es aquel que tiene todas las instrucciones que se deben ver reflejadas en la parte física. La comunicación entre los dos mundos se hace mediante el concepto de gemelo digital, en donde el mundo virtual tendrá los agentes producto y AGVs virtuales y su representación visual serán estos mismos agentes de manera física. La

herramienta permite cargar el número de instancias que se desean observar, en donde se creó el campo para probar diez situaciones diferentes.

En la construcción de los agentes del modelo físico, se generaron parámetros no escalables como dimensiones de la matriz, de los vehículos y productos. Cada agente tiene como parámetros su forma, color, tamaño y posición, facilitando la observación de todo el sistema. Cabe aclarar que todo esto se envía desde el mundo virtual y el mundo físico permite dicha visualización.

De acuerdo con la manipulación del simulador (Anexo 6. Modelo Predictivo NetLogo), existen cinco botones que comandan la puesta en marcha y la ejecución de cada instancia. El orden que se debe tener en cuenta para utilizar de manera eficiente la herramienta es el siguiente: Inicia eligiendo cualquiera de los escenarios los cuales tienen su respectivo TXT leído por el mundo virtual, continuando con el cargue de dicha instancia, dando paso al mundo físico que es el responsable de mostrar gráficamente toda la información del mundo virtual. Finalmente, al iniciar la simulación, el programa comienza a correr con la velocidad preestablecida a un paso, sin embargo, se tiene la opción de correr el modelo cada 30 pasos y aumentar la velocidad de cada vehículo. Se puede observar en la imagen, la representación de como cargan los componentes del mundo físico, llamados por el sistema virtual. Estos componentes corresponden a los agentes físicos de producto y AGV.



*Figura 12: Mundo físico sin órdenes recogidas. Fuente: Los Autores*

Esta simulación, se hizo con el fin de poder observar un modelo sin perturbaciones, pero aproximado a un entorno real y así comparar la función objetivo frente al modelo en VBA y próximamente frente a posibles imprevistos en los mismos escenarios. En el mundo físico, se diseñaron indicadores que evidencian el tiempo total de recolección con una velocidad constante para cada carro y así mismo la cantidad de pedidos que se van dejando en el punto de desembarque.

## **MODELO REACTIVO**

Teniendo en cuenta las características de agentes, en este caso se tendrán como agentes reactivos a los AGV. Los AGVs físicos-local reciben información del gemelo digital mediante sensores y, una vez que analiza la información recibida, actúan para garantizar la consecución de objetivos. La manera en que se transmite dicha información se basa en que, durante la ejecución de las actividades, el agente hace repetidas tomas del sensor y luego compara cada una de ellas con las condiciones en ese instante para ejecutar las acciones pertinentes. Al recibir la información, cada agente local debe llevar a cabo las órdenes planteadas desde el agente global y así



mismo ser capaz de reaccionar ante algún tipo de perturbación. El resultado exitoso de lo mencionado anteriormente se debe a que el agente local presente en el mundo virtual tiene un gemelo físico al que a través de sensores se le envían las instrucciones para que se refleje en un mundo real.

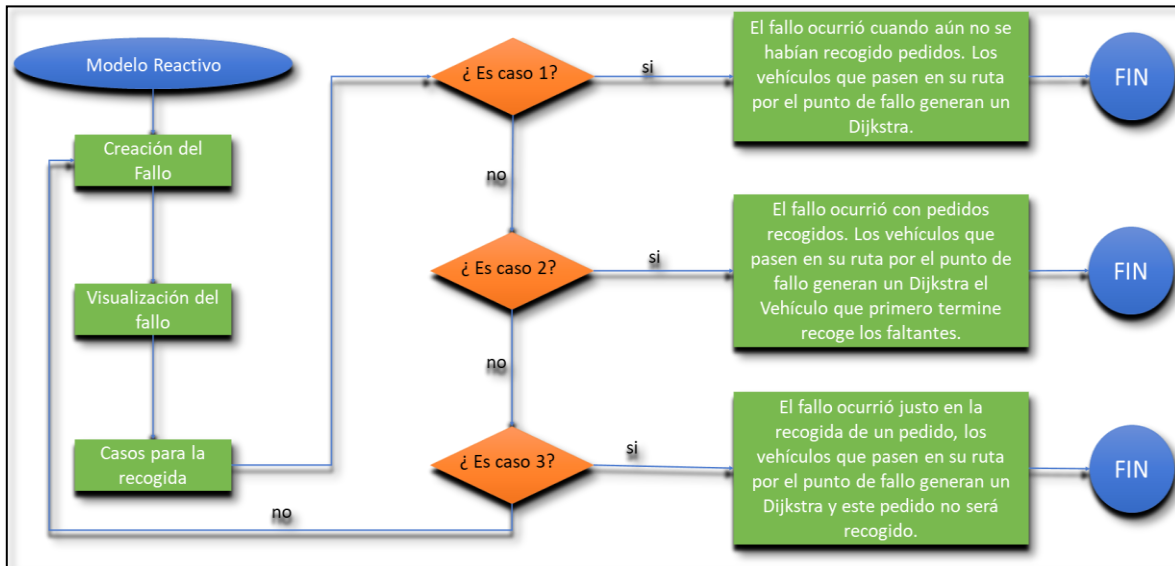


Figura 13: Flujograma Modelo Reactivo. Fuente: Los Autores

Partiendo de lo anterior se construye un modelo de reactividad que consiste en la ejecución con interacciones de la simulación, es decir, el fallo de algún vehículo originando así un error en el sistema y la posible desconexión del movimiento de los AGVs con el modelo predictivo. Como se observa en la Figura 13, la reactividad será probada para el caso de un agente fallido dando lugar así a la solución lineal consistente de tres etapas: fallo del AGV, la visualización del vehículo que genera el bloqueo en la matriz del mundo físico y por último el nuevo enrutamiento de los vehículos.

Para el primer caso en la generación del fallo se sabe que el enrutamiento de los AGV estará designado por el modelo predictivo o agente global en donde se digita para el mundo virtual la cantidad de Ticks o movimientos que el sistema hará antes del fallo y así forzar un error en la ejecución. El carro bloqueará la posición en su totalidad y visualmente será un muro para los demás agentes ya que si su recorrido tiene este punto tendrán que cambiar la ruta. El siguiente paso es la conexión del carro fallido con los demás vehículos generando una visualización global del sistema con el fin de conectarlos entre sí y lograr determinar la posición que estará bloqueada. Los AGVs que se encuentren en la ruta con la posición bloqueada pasaran al tercer pilar el cual consiste no en cambiar un punto de llegada B sino la ruta de llegada al punto B, este recorrido se realiza con un algoritmo Dijkstra.

Basados en el concepto de Ramé y Ahmet (2017), donde se determina el algoritmo Dijkstra como una solución ideal para la formulación del problema de ruteos entre un punto A y un punto B se toma como referencia del mismo modo la generación de rutas para la futura aplicación del proyecto. Este algoritmo también contempla la minimización de costos de transporte y cargas en ruta al igual que el ruteo con la base única de distancias y capacidades, de este modo nos permite encontrar una buena solución al ruteo de un punto a otro con la condición de no pasar por el bloqueo, es decir creando rutas alternas para la creación de la reactividad. Sin embargo, los vehículos que no son afectados por el bloqueo tienen la potestad de seguir por las rutas generadas en el modelo predictivo. Cabe aclarar que el algoritmo Dijkstra se ejecutara tanto en los recorridos de ida como de regreso si el vehículo en su trayectoria tiene el punto de bloqueo.

Partiendo de lo anterior, el ruteo de los vehículos y el fallo del AGV se dividen en casos que se contemplan para el desarrollo del modelamiento. El primer caso, (Figura 14), se visualiza en el bloqueo de un agente que falla al momento que se encuentre en ruta y además no lleve carga. El segundo caso, se evidencia cuando un vehículo falla en trayectoria de un punto a otro, pero a diferencia de la primera instancia este si ha recogido pedidos. El tercer caso, (Figura 15), se identifica cuando un carro falla justo en la recogida de un pedido, el cual no podrá ser recogido y se tomará como incumplimiento del pedido ya que al estar el vehículo ubicado justo en el punto de recolección impide el ingreso de otro AGV. Finalmente, cada AGV continua con las rutas establecidas desde el inicio y al finalizar todos sus recorridos el AGV que culmine primero será el encargado de recoger el pedido del agente fallido. (Ver Anexo 7. Modelo Reactivo NetLogo).

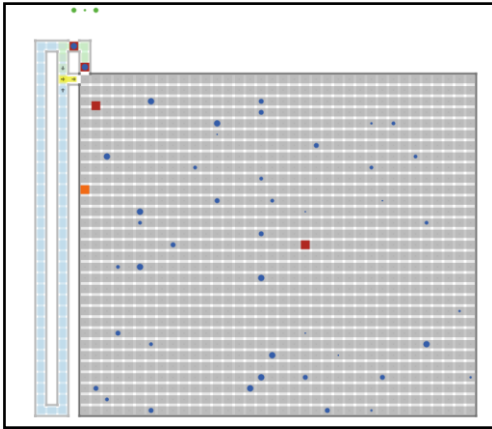


Figura 14: Caso 1 Reactividad. Fuente: Los Autores

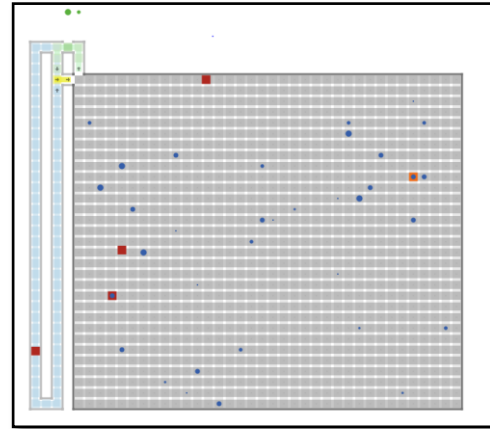


Figura 15: Caso 3 Reactividad. Fuente: Los Autores

Referente a la comunicación que tienen los agentes AGV en el momento del fallo, se basa en el envío de un mensaje de manera automática a los demás agentes, avisando que existió una perturbación y reflejando así la arquitectura heterárquica existente. Sabiendo que los dos archivos están conectados, se creó un atributo binario en el mundo físico que cuando toma el valor de 1 refiere a la existencia de un fallo y es leído por el mundo virtual que inicia la reacción. Lo anterior, muestra la importancia de la utilización del gemelo digital, en donde el flujo de datos entre los dos archivos es automático en tiempo real.

## 5. Resultados

Inicialmente se realiza el estudio para determinar los parámetros a utilizar tanto en la estructura del proyecto como en la solución de la metaheurística, para esto se desarrollan dos diseños de experimentos que determinan si hay diferencia significativa a la hora de saber la cantidad de agentes que trabajaran en el mundo virtual dependiendo la instancia trabajada y la mejor combinación de parámetros en el Tabú. Partiendo de los resultados del diseño experimental se desarrolla una regresión la cual permite estimar el tiempo que puede tardar cierta cantidad de vehículos en culminar el ruteo.

Para el caso propuesto anteriormente se realiza un diseño de experimentos el cual es concluyente al determinar que si hay diferencia significativa en el tiempo de ejecución si se cambian la cantidad de carros, arrojando en la ANOVA un P valor para este factor de  $2.58e^{-06}$  y para el factor de la Iteración un P valor de  $4.48e^{-13}$ . Partiendo del rechazo de la hipótesis nula se determina una regresión pertinente para cada una de las instancias, (las instancias determinan la cantidad de pedidos que hay al inicio del problema de enrutamiento), trabajadas dando como ecuaciones finales las siguientes.

<i>Instancias</i>	<i>Ecuación de la regresión</i>
<i>Instancia 30</i>	$Y = 6679.6 - 386.8 * x$
<i>Instancia 40</i>	$Y = 7320.5 - 386.8 * x$
<i>Instancia 50</i>	$Y = 8345.9 - 386.8 * x$
<i>Instancia 60</i>	$Y = 9153.1 - 386.8 * x$
<i>Instancia 70</i>	$Y = 1004.9 - 386.8 * x$
<i>Instancia 80</i>	$Y = 9707.7 - 386.8 * x$
<i>Instancia 90</i>	$Y = 11725.9 - 386.8 * x$
<i>Instancia 100</i>	$Y = 13067.7 - 386.8 * x$
<i>Instancia 110</i>	$Y = 12927.5 - 386.8 * x$
<i>Instancia 120</i>	$Y = 13579.6 - 386.8 * x$

Regresión cantidad AGV. Tabla 1. Fuente: Los Autores

La cantidad de AGVs que se colocan como parámetros para la ejecución de los modelos alternos (modelo reactivo), se trabajan con un escenario de 5 vehículos y de esta forma se corren las instancias en el modelo predictivo de VBA. Ya que este modelo predictivo se realiza con una meta-heurística es necesario saber qué factores son significativos a la hora de su ejecución, es así que se realiza un segundo diseño experimental para la solución descrita donde los resultados de la ANOVA son concluyentes al determinar que la mejor solución está dada por los siguientes parámetros: La Heurística C&W con un número de iteraciones de 1000 y un tiempo en la lista Tabú de 50, puesto que para la prueba paramétrica los factores A, C y las interacciones entre todos son medias significativamente diferentes, es decir, que se rechaza la hipótesis nula. Partiendo de esto se logra observar que el tiempo más reducido es la combinación previamente señalada. Al validar el supuesto de normalidad no es posible concluir con los resultados de la ANOVA dando lugar a una prueba no paramétrica con el fin de identificar los factores significativos.

La prueba de Kruskal-Wallis nos da como resultados P valores por debajo de 0,05 para el factor A y el factor C, al igual que la interacción entre los mismos, contemplando también en las interacciones al factor B, sin embargo, para el factor B como único factor el P valor es superior a 0,05. Teniendo en cuenta los resultados se desarrolla un gráfico de interacciones (Figura 16) el cual nos permite visualizar resultados al finalizar la prueba no paramétrica.

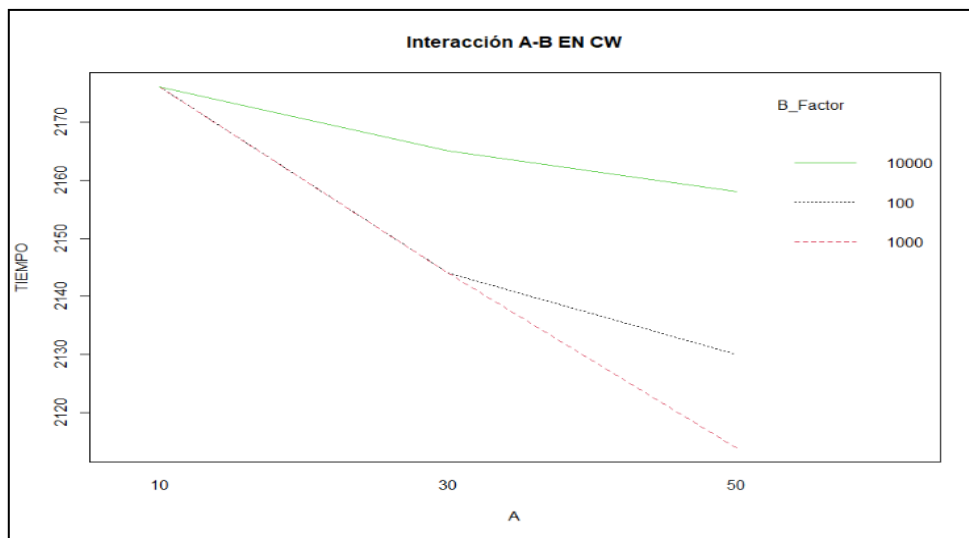


Figura 16: Comparación de interacciones. Fuente: Los Autores

El paso a seguir es el cálculo de los tiempos en cada instancia y su respectiva diferencia en los modelos predictivos y reactivos. Es determinante el incremento de tiempo al ejecutar las corridas incluso comparando el modelo predictivo en VBA y en NetLogo ya que entre los elementos más influyentes los cuales aumentan el tiempo de corrida dependiendo las instancias se encuentran los siguientes.

- Salida de los Agentes: La salida de los agentes en el modelo desarrollado en VBA asume que salen del mismo punto al mismo tiempo, es decir, no se ejecuta un control de salida entre un carro y otro mientras que en el modelo de NetLogo este factor si se tiene en cuenta.
- Cargue y Descargue: El tiempo de cargue y descargue de los pedidos es fundamental pues en VBA solo se ejecuta una ruta que pasa por diversos puntos, sin embargo, en NetLogo se genera un tiempo de espera en la recolección y entrega de pedidos dependiendo también la cantidad de unidades en cargue y descargue.
- Alistamiento de Vehículos: En NetLogo se desarrolla una restricción la cual nos dice que, si en el punto de salida hay por lo menos un vehículo, el AGV que entre al punto de embarque tendrá que hacer todo el recorrido de alistamiento, mientras que si no hay ningún vehículo el AGV estará en potestad de decidir si hace el alistamiento o se dirige directamente a la salida para una nueva ruta.
- Colisión entre AGVs: Se debe tener presente que en la simulación en NetLogo, cuando se encuentran dos AGV, uno se detiene y deja que pase el otro, lo que hace que el tiempo aumente a diferencia del modelo en VBA que no considera esta reacción.

Las diferencias entre los modelos hacen que el tiempo promedio por instancia cambie, los resultados se reflejan en la Figura 17.

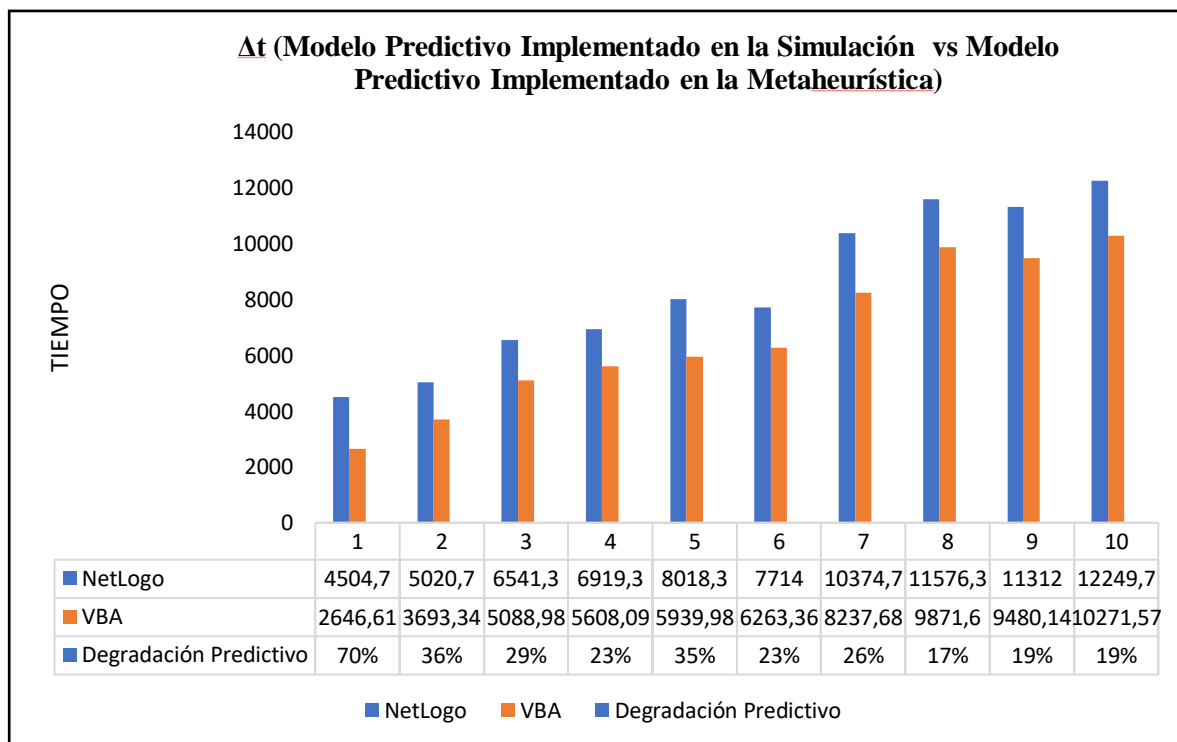


Figura 17: Comportamiento de Instancias Predictividad. Fuente: Los Autores

El gráfico ilustra el crecimiento del tiempo basado en las instancias y así mismo la diferencia entre el tiempo del modelo en VBA y NetLogo. Las diferencias principales son por los elementos mencionados con anterioridad al igual que por la cantidad de nodos respectivos de cada instancia los cuales son determinantes pues el espectro de posiciones es mayor con más nodos, es decir la recogida de los puntos va en el recorrido total de la matriz

solución mientras que con menos nodos se limita a pocos puntos lejanos del punto de embarque. Esta es una razón por la que existen diferencias entre instancias. (Ver anexo 8. Resultados).

Al igual que la comparación entre los modelos predictivos se desarrolla una comparación entre la predictividad y la reactividad, es decir la misma instancia con un fallo en los vehículos (Anexo 8. Resultados). Se realiza un fallo en el primer tercio del recorrido total al igual que en el segundo tercio y se evidencia la diferencia de tiempos aumentando para este caso cuando el error se produce en la primera parte del recorrido en promedio casi un 10% del del tiempo total. Información en el anexo (Reactivo). Finalizando la comparación en la reactividad se desarrolla una tabla donde se demuestra el aumento del tiempo entre el modelo predictivo de NetLogo y el modelo reactivo de NetLogo Figura 18.

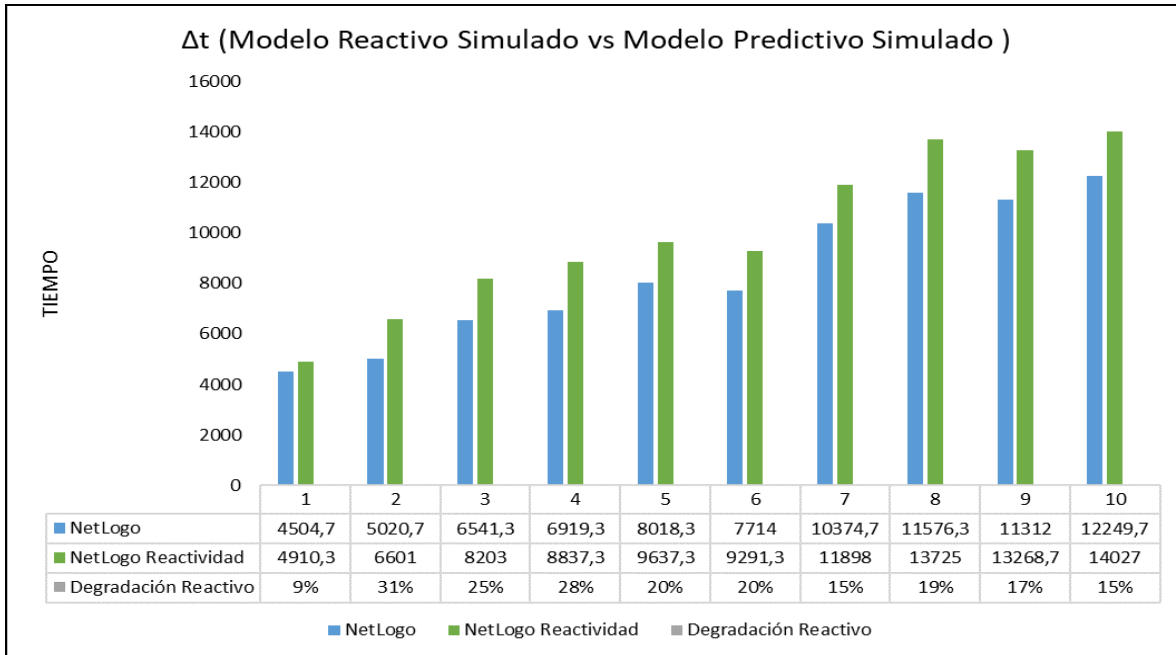


Figura 18: Comportamiento de Instancias Reactividad. Fuente: Los Autores

El aumento del tiempo es en promedio del 20% comparando la reactividad con fallo en el primer tercio del tiempo total de cada instancia y con la predictividad en NetLogo, esto se debe a que el fallo se encuentra al inicio del programa y la cantidad de rutas cambiantes es superior que presentándolo en el segundo tercio de la corrida. Aunque el algoritmo Dijkstra da una solución acertada para el nuevo enrutamiento su respuesta no es comparable con la solución del C&W pues al cambiar la ruta desmejora la solución encontrada anteriormente. El aumento de tiempo es significativo pues al fallar un AGV este dejara de cumplir con su orden de pedido y es allí donde se presenta un aumento considerable de tiempo con respecto al calculado previamente, pues estos productos serán recogidos al finalizar las rutas de los vehículos funcionales por el vehículo que primero finaliza la totalidad de sus pedidos. En el modelo reactivo realizado en NetLogo, se puede observar que el gemelo físico tiene un comportamiento eficiente en el tiempo de reacción según la información enviada por el gemelo digital. Al presentarse el fallo de un AGV, el agente virtual recibe este dato y en tiempo real envía la información de las nuevas rutas a los agentes físicos, quienes replican el movimiento en un tiempo de respuesta aparentemente automático.

Tanto en el modelo predictivo como en el reactivo simulados en NetLogo, se evidenció que entre más número de órdenes estuvieran involucradas en el escenario, se tardaba más tiempo en responder la visualización de los movimientos de cada vehículo. El modelo reactivo al tener mayor número de restricciones y capacidad para responder a posibles perturbaciones hace que el mundo físico tarde más en los movimientos de cada vehículo.

Esto se debe a que además de tener un código con más información, es necesario correr cada escenario en un equipo con amplia capacidad de memoria y procesador.

Finalmente, en la parte reactiva del modelo, al existir un fallo de AGV, se definió que aquellos pedidos que ya habían sido recogidos por ese vehículo, pero aún no estaban en la zona de embarque, automáticamente volvieron a su posición inicial para que otro AGV pasara por ellos. Esto se evidenció de una manera automática, permitiendo concluir una buena capacidad de reacción del agente producto virtual.

## **6. Limitaciones, conclusiones y recomendaciones.**

Esta investigación permitió verificar la importancia de un gemelo digital en la emulación de un sistema AS/RS, donde se diseñó el sistema con una arquitectura que inicialmente es jerárquica en la predictividad y heterárquica en la reactividad utilizando el gemelo digital, considerando posibles perturbaciones que probaron el desempeño del sistema propuesto. También, se llevó a cabo la creación de dos archivos independientes, pero del mismo programa donde se encuentra tanto el nivel de coordinación (mundo virtual) como el de ejecución (mundo físico), con el fin de crear una conexión automática y lograr la comunicación entre los agentes virtuales y físicos.

Partiendo de la combinación de parámetros en el Tabú para lograr una minimización del Makespan se encontró que la heurística Clarke and Wright con 1000 iteraciones y 50 en el tiempo de la lista Tabú, dio una mejor solución entre los algoritmos de vecino más cercano y un modelo desarrollado por los autores. Dicha afirmación se soportó con el desarrollo de un diseño experimental, que permitió observar el comportamiento del tiempo de recolección de paquetes con cierta cantidad de nodos y cinco vehículos.

Considerando posibles perturbaciones, se probó el desempeño del sistema propuesto y el modelo de enrutamiento de AGVs, diseñando un modelo reactivo. Este modelo, consiste en ejecutar los escenarios emulados anteriormente, generando el fallo de un vehículo y originando una posible desconexión del movimiento de los AGVs con el modelo predictivo. Se logró evidenciar que el tiempo al generar un modelo reactivo con fallo provocado en el primer tercio del total del tiempo de recorrido aumenta en promedio 21%, ya que el algoritmo controla el enrutamiento de un punto a otro con la restricción de no pasar por el fallo, de esta forma se generan rutas escalonadas que aumentan el tiempo al igual que la recolección de faltantes al finalizar el recorrido de cada AGV.

Dentro de las perspectivas de investigación se propone probar el gemelo digital creado para otro tipo de sistemas como manufactura, logística, salud, educación, entre otros, teniendo en cuenta las variaciones que puede tener cada uno de estos escenarios.

En la parte del modelo predictivo, se propone probar el sistema construido cambiando el modelo predictivo de optimización (Tabú Search) por un algoritmo poblacional como el Algoritmo Genético u otras metaheurísticas. Ahora, en la parte reactiva se plantea probar el sistema con diferentes perturbaciones como el fallo de una línea, fallo de una celda, falta de existencia de un producto, entre otros.

También se propone implementar el sistema en un entorno real, teniendo en cuenta las posibles perturbaciones y nuevos desafíos en la ejecución, como la mecánica y electrónica de los vehículos, la construcción de la matriz y los rieles por donde se desplazarán los carros, etc.

Finalmente, el simulador tiene algunas limitaciones en su funcionalidad en algunos escenarios, como es el caso en el que se supera el número máximo de vehículos y nodos utilizados en este experimento. Bien es cierto que se pueden probar nuevas instancias, su capacidad se limita a la cantidad mencionada anteriormente. Esto se debe a que el espacio se reduce para una cantidad mayor de agentes y exista un aumento en las perturbaciones que no permita que los AGVs prevengan la totalidad de colisiones.

Una segunda limitación que puede existir para esta simulación es el computador donde se corra, ya que al tener agentes físicos respondiendo a diferentes movimientos replicados, es necesario que el equipo cuente con amplia

capacidad en la memoria y en el procesador. Al no contar con dichas condiciones, el modelo puede tardar bastante, especialmente en aquellos escenarios en donde existen varias órdenes, o incluso una cantidad mayor de AGVs.

## Referencias

- Carro, J., Flores, F., Flores, I., y Hernández, R. (2019). Industria 4.0 y manufactura digital: un método de diseño aplicando ingeniería inversa. *Revista Ingeniería*, 24(1), 6-28. doi: zzzz
- Castillo, I., Reyes, S., & Peters, B. (2001). Modeling and analysis of tandem AGV systems using generalized stochastic Petri nets. *Journal of Manufacturing Systems*, 20(4), 236-249. Retrieved from <https://search-proquest-com.ezproxy.javeriana.edu.co/abicomplete/docview/197660819/fulltextPDF/59BDDBC8BB4C4CB4PQ/1?accountid=13250>
- Escudero, J. (2014). *Logística de almacenamiento, España: Paraninfo*, 47-67 recuperado de [https://books.google.com.co/books?id=AnC6AwAAQBAJ&printsec=frontcover&hl=es&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](https://books.google.com.co/books?id=AnC6AwAAQBAJ&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)
- Fazlollahab, H., Saidi-Mehrabad, M., & Masehian, E. (2015). Mathematical Model for Deadlock Resolution in Multiple AGV Scheduling and Routing Network: A Case Study. *Industrial Robot*, 42(3), 252-263. doi: 10.1108/IR-12-2014-0437
- Fedorko, G., Molnar, V., Vasil, M., & Hanzl, J. (2018). Application of the Tecnomatix Plant Simulation Program to Modelling the Handling of Ocean Containers Using the AGV System. *Naše More*, 65(4), 230-236. doi: 10.17818/NM/2018/4SI.12
- Gademann, A., & Van de Velde, S. (2000). Positioning Automated Guided Vehicles in a Loop Layout. *European Journal of Operational Research*, 127(3), 565-573. doi: doi.org/10.1016/S0377-2217(99)00341-0
- Jacobsen, J. (2011). Warehouse Automate with AGVs. *Beverage Industry*, 102(8), 64-66. Retrieved from <https://search-proquest-com.ezproxy.javeriana.edu.co/abicomplete/docview/884326085/fulltextPDF/22348006B96E4B17PQ/2?accountid=13250>
- Kator, C. (2007). AS/RS Basics. *Warehousing Management Edition, Framingham Tomo 62* (8), 50-53. Retrieved from <https://search-proquest-com.ezproxy.javeriana.edu.co/abicomplete/docview/201530298/FFFDF2E2488C428DPQ/6?accountid=13250>
- Kelvin, J y Arcentales, M. (2018). *Diseño e implementacion de un prototipo de robot movil autoguiado controlado por un autómatas programable*. (tesis de pregrado). Universidad politécnica salesiana, Guayaquil
- Knill, B., Schwind, G., & Witt, C. (1993). AS/RS: Full Partner in Manufacturing Material. *Handling Engineering, May 1993*, 48 (5), 43. Retrieved from <https://search-proquest-com.ezproxy.javeriana.edu.co/abicomplete/docview/213288342/fulltextPDF/C101477566AC4736PQ/5?accountid=13250>
- Lins, T., Oliveira, R., Correia, L., & Sa Silva, J. (November of 2018). Industry 4.0 Retrofitting. In Lima, G (President), *Brazilian Symposium on Computing System Engineering (SBESC)*. Symposium carried in Brazil. Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8692029>

- Liu, J., Zhong, X., Peng, X., Tian, J., & Zou, C. (July 25-27, 2018). Design and Implementation of New AGV System Based on Two-layer Controller Structure. In Xie, S. (President), *Chinese Control Conference (CCC)*. Conference carried out in China. Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8483515>
- Laurence, M., y Tobon, J. (2013). Desarrollo de una plataforma para la actualización automática de inventarios del sistema de almacenamiento automático AS/RS del centro tecnológico de automatización industrial CTAI. Retrieved from <https://repository.javeriana.edu.co/bitstream/handle/10554/6328/LaurenceMelissa2013.pdf?sequence=1>
- Meneses, G., Velasquez, G., y Grisales, R. (2014). Prototipo de vehículo de guiado autónomo para sistema de almacenamiento y recuperación automático de libros. *Revista de investigacion universidad de Quindío*, 25(1), 1-73. doi: <https://doi.org/10.33975/riunet.vol25n1.156>
- Miyamoto, T., & Inoue, K. (2013). Random Search for Dispatch and Conflict-free Routing problem of capacitated AGV systems. *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, , 1611-1615.
- Morales, M. (2008). Agentes y sistemas multi-agentes. *Revista de información, tecnología y sociedad, versión impresa ISSN 1997-4044*, 110.
- Muñoz Alcázar, J. (2019). *Aplicación del concepto de Gemelo Digital a un SCADA Industrial* (Doctoral dissertation). Retrived from <https://riunet.upv.es/handle/10251/126015>
- Oboth, C. C. (1997). Dispatching and Conflict-free Routing of Automated Guided Vehicles: An Object-oriented Simulation Study. *International Journal of Flexible Manufacturing Systems (2003) 15(4)*, 223. doi: 10.1023/B:FLEX.0000036033.12568.5e
- Pedroza, A., Sanchez, S., y Rodriguez, H. (2007). Control de un vehículo guiado automáticamente (AGV). *Conciencia Tecnológica*, (34), 10-15. Retrieved from <https://www.mendeley.com/catalogue/control-un-vehiculo-guiado-automaticamente-agv/>
- Ramë , L., y Ahmet, S. (2017). Application of Graph Search Algorithm Dijkstra to Find Optimal Solution for the Problem of Transport. *Faculty of Engineering Hunedoara, Nov 2017, 15 (4)* 69-72, Retrieved from <https://search-proquest-com.ezproxy.javeriana.edu.co/abicomplete/docview/1967314134/abstract/538B5540F7FA47A6PQ/1?accountid=13250>
- Rodríguez, R., y Martín, J. (2010). Simulación de un sistema AS / RS. *Treballs Docents Curs*,(1), 55-58. doi: 10/ISSN:1889-4771
- Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F., & Mahmoodian, V. (2015). An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers and Industrial Engineering*, 86, 2-13. doi: 10.1016/j.cie.2015.01.003
- Schwind, g., (1998). AGVS & AS/RS: A Winning Manufacturing Combination. *Material Handling Engineering 53 (5)* 49-54. Retrieved from <https://search-proquest-com.ezproxy.javeriana.edu.co/abicomplete/docview/213340504/fulltextPDF/12CDBA918BFD464APQ/1?accountid=13250>
- Shao, S., Xia, Z., Chen, G., Zhang, J., Hu, Y., & Zhang, J. (2014). A New Scheme of Multiple Automated Guided Vehicle System for Collision and Deadlock Free. *ICIST 2014 - Proceedings of 2014 4th IEEE International Conference on InShao, S., Xia, Z., Chen, G., Zhang, J., Hu, Y., & Zhang, J. (2014). A New Scheme of Multiple Automated Guided Vehicle System for Collision and Deadlock Free. ICIST 2014 - Proceedings*, , 606-610. doi: 10.1109/ICIST.2014.6920551



- Sun, S., Gu, C., Wan, Q., Huang, H., & Jia, X. (2018). CROTPN Based Collision-free and Deadlock-free Path Planning of AGVs in Logistic Center. *2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018*, , 1685-1691. 10.1109/ICARCV.2018.8581060
- Tu, J., Qian, X., & Lou, P. (2017). Application Research on AGV Case: Automated Electricity Meter Verification Shop Floor. *Industrial Robot*, *44*(4), 491-500. doi: 10.1108/IR-11-2016-0285
- Ventura, J., & Lee, C. (2008). A Study of the Tandem Loop with Multiple Vehicles Configuration for Automated Guided Vehicle Systems. *Journal of Manufacturing Systems*, *20*(3), 153-165. Retrieved from <https://search-proquest-com.ezproxy.javeriana.edu.co/docview/197649911/fulltextPDF/AC27295B077843F4PQ/1?accountid=13250>
- Xu, W., Guo, S., Li, X., Guo, C., Wu, R., & Peng, Z. (2019). A Dynamic Scheduling Method for Logistics Tasks Oriented to Intelligent Manufacturing Workshop. *Mathematical Problems in Engineering*, *2019*, 1-18. doi: 10.1155/2019/7237459