



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DESARROLLO DE UN ASISTENTE PARA LOS  
REBALANCEOS DE  
UNA PLANTA DE MONTAJE DE AUTOMÓVILES Y  
PUESTA EN  
PRÁCTICA

Trabajo Fin de Grado

Grado en Ingeniería Mecánica

AUTOR/A: Rodríguez Part, Víctor

Tutor/a: Sanabria Codesal, Esther

CURSO ACADÉMICO: 2022/2023



## Resumen

En las líneas de montaje de las fábricas de automoción, es esencial la optimización de los procesos que no aportan un valor añadido. En este sentido, Ford focaliza su compromiso con la mejora continua a través de los llamados “Key Projects (KP)”.

Este Trabajo de fin de grado se desarrolla dentro del “KP2: Workstation Optimization & Line Rebalancing” y tiene como objetivo mejorar la eficiencia en los desplazamientos de los trabajadores, minimizando el número de pasos recorridos, optimizando la distribución y secuencia de las operaciones en la cadena de montaje.

Para ello, se ha decidido utilizar la metodología Ágil Scrum, definiendo sprints semanales y cuatro “Minimum Viable Product (MVP)” que deben entregarse al final de cada trimestre del año.

Los MVPs son los siguientes:

1. Crear un simulador de pasos que muestre una visualización (para demostrar la fiabilidad del simulador) y que proporcione Indicadores Clave de Rendimiento (KPI) relevantes para ayudar a los ingenieros industriales a decidir cómo rebalancear las líneas de montaje.
2. Crear un optimizador de layout automático funcional para todas las estaciones de uno de los sistemas de la planta de montaje.
3. Crear un optimizador del orden y la simultaneidad de las operaciones.
4. Combinar ambos optimizadores para obtener la secuencia de operaciones y el plano de la distribución (layout) óptimo para cada estación.

Este TFG abarcará únicamente hasta el MVP1. Y adicionalmente se utilizará la aplicación para rebalancear algunas estaciones de trabajo.

## Resum

En les línies de muntatge de les fàbriques d'automoció, és essencial l'optimització dels processos que no aporten un valor afegit. En aquest sentit, Ford focalitza el seu compromís amb la millora contínua a través dels anomenats “Key Projects (KP)”.

Aquest Treball de fi de grau es desenvolupa dins del “KP2: Workstation Optimization & Line Rebalancing” i té com a objectiu millorar l'eficiència en els desplaçaments dels treballadors, minimitzant el nombre de passos recorreguts, optimitzant la distribució i seqüència de les operacions en la cadena de muntatge.

Per a això, s'ha decidit utilitzar la metodologia Ágil Scrum, definint sprints setmanals i quatre “Minimum Viable Product (MVPs)” que s'han d'entregar al final de cada trimestre de l'any.

Els MVP consisteixen en:

1. Crear un simulador de passos que mostre una visualització (per a demostrar la fiabilitat del simulador i que ens indique KPI rellevants per ajudar als enginyers industrials a decidir com rebalancejar les línies de muntatge.
2. Crear un Optimitzador de layout automàtic i funcional per a totes les estacions d'un dels sistemes de la planta de muntatge.
3. Crear un Optimitzador de l'ordre de les operacions i simultaneïtat.
4. Juntar tots dos optimitzadors per obtenir la seqüència d'operacions i el Layout òptim per a cada estació.

Aquest TFG sols abastarà fins l'MVP1. Adicionalment s'utilitzarà l'aplicació per a rebalancejar algunes estacions de treball.

## **Abstract**

In automotive factory assembly lines, optimizing processes that do not add value is essential. In this regard, Ford focuses its commitment to continuous improvement through the so-called Key Projects (KPs).

This Bachelor's thesis is developed within the KP2: Workstation Optimization & Line Rebalancing and aims to improve efficiency in worker movements by minimizing the number of steps taken and optimizing the distribution and sequence of operations in the assembly line.

To achieve this, the Agile Scrum methodology has been chosen, defining weekly sprints and four Minimum Viable Products (MVPs) that must be delivered at the end of each quarter of the year.

The MVPs consist of:

1. Create a step simulator that shows us a visualization that demonstrates the reliability of the simulator and that indicates relevant KPI to help industrial engineers decide how to rebalance assembly lines.
2. Create a functional automatic layout Optimizer for all stations of one of the assembly plant systems.
3. Create an optimizer for the order of operations and simultaneity.
4. Combine both optimizers to obtain the optimal sequence of operations and Layout for each station.

This TGF will cover just the MVP1. And additionally, the application will be used to rebalance some work stations.

Quiero expresar mi sincero agradecimiento a las siguientes personas y entidades:

- A mis tutores, Sergio Casado y Esther Sanabria, por su invaluable ayuda y paciencia durante las prácticas y la realización de mi TFG. Su guía y apoyo han sido fundamentales en mi proceso de aprendizaje y desarrollo.
- A mis compañeros de proyecto y de los equipos de ingeniería y de innovación. En especial a Sergio Mallasén, Pablo Molina, Enrique Monlleo y Cristina Rabadán, por su colaboración y apoyo incondicional a lo largo de este tiempo. Su trabajo en equipo ha sido imprescindible para alcanzar nuestros objetivos.
- A FORD ESPAÑA S.L., por brindarme la oportunidad de realizar mis prácticas y llevar a cabo mi TFG en un entorno tan enriquecedor. Quiero agradecer también a todos mis compañeros con quienes he tenido el placer de interactuar, por su inclusividad, cercanía y hospitalidad.
- A todas las personas que han dedicado su tiempo y esfuerzo en leer mi Trabajo Fin de Grado (TFG) y brindarme su valioso feedback.
- A los amigos que he hecho en la universidad y a mis padres por todas las experiencias y apoyo mostrados, haciendo mi paso por la universidad una experiencia fantástica e inolvidable.

Estoy profundamente agradecido/a a todos ellos por su contribución y por hacer de esta experiencia una etapa de crecimiento y aprendizaje invaluable.

# Índice general

## I Memoria

<b>1. Objeto del proyecto</b>	<b>3</b>
<b>2. Antecedentes</b>	<b>5</b>
2.1. Características de la planta de Montaje de Ford Almussafes. . . . .	5
2.2. Motivación y justificación . . . . .	8
<b>3. Factores a considerar</b>	<b>11</b>
3.1. Características fundamentales . . . . .	11
3.2. Limitaciones y condicionamientos . . . . .	12
<b>4. Descripción de la solución adoptada</b>	<b>15</b>
4.1. Inputs . . . . .	16
4.2. Funcionalidades del simulador . . . . .	18
4.3. Outputs . . . . .	34
<b>5. Validación</b>	<b>37</b>
<b>6. Soluciones alternativas</b>	<b>41</b>

## II Pliego De Condiciones

<b>1. Definición y alcance</b>	<b>45</b>
1.1. Alcance del proyecto . . . . .	45
1.2. Alcance del TFG . . . . .	45
<b>2. Condiciones Técnicas</b>	<b>47</b>
2.1. Problemas a abordar . . . . .	47
2.2. Solución adoptada y condiciones . . . . .	47
2.3. Condiciones exigibles: . . . . .	48
2.4. Software, permisos y licencias requeridas . . . . .	49
<b>3. Condiciones Facultativas</b>	<b>51</b>
3.1. Controles de Calidad y Ensayos . . . . .	51
3.2. Recepción Definitiva . . . . .	51
<b>4. Ejecución</b>	<b>53</b>

4.1. Metodología y organigrama del proyecto . . . . .	53
<b>III Presupuesto</b>	
1. Presupuesto	57
2. Costes y ahorro para la empresa	59
<b>IV Anexos</b>	
1. Puesta en práctica	63

# Índice de figuras

2.1. Fotografía Aerea de la factoría FORD ESPAÑA S.L. . . . . .	5
2.2. Coche desplazandse por la electrovía en la línea de Chasis . . . . .	8
3.1. Esquema de la funcionalidad “Esquivador” . . . . .	11
3.2. Crab Line . . . . .	13
3.3. Happy Seat . . . . .	13
3.4. Planta de Salpicaderos . . . . .	14
4.1. Plantilla hoja Worksteps . . . . .	17
4.2. Plantilla para la anotación de las operaciones Relativo 1 . . . . .	18
4.3. Coordenadas consideradas en la cadena de montaje . . . . .	20
4.4. Desplazamiento con relativo 0 . . . . .	22
4.5. Desplazamientos cuando monta una pieza en el coche . . . . .	23
4.6. Triángulo asociado a la ley de los cosenos . . . . .	24
4.7. Modelización del problema . . . . .	24
4.8. Desplazamiento con relativo = 2 . . . . .	25
4.9. Desplazamiento con relativo = 1 . . . . .	26
4.10. . . . .	27
4.11. . . . .	27
4.12. . . . .	28
4.13. Ejemplo de grafo . . . . .	31
4.14. Ejemplo de starved . . . . .	31
4.15. Ejemplo de blocked . . . . .	32
1.1. Layout de partida de la estación piloto. . . . .	63
1.2. Tabla resumen primer escenario . . . . .	64
1.3. Gráfico de posiciones primer escenario . . . . .	64
1.4. Tabla resumen segundo escenario . . . . .	65
1.5. Gráfico de posiciones segundo escenario . . . . .	65
1.6. Tabla resumen tercer escenario . . . . .	66
1.7. Gráfico de posiciones tercer escenario . . . . .	66
1.8. Tabla resumen cuarto escenario . . . . .	66
1.9. Gráfico de posiciones cuarto escenario . . . . .	67
1.10. Tabla resumen quinto escenario . . . . .	67
1.11. Gráfico de posiciones quinto escenario . . . . .	67
1.12. Tabla resumen sexto escenario . . . . .	68
1.13. Gráfico de posiciones sexto escenario . . . . .	68
1.14. Tabla resumen septimo escenario operación antes del soporte. . . . .	68



1.15. Gráfico de posiciones séptimo escenario operación antes del soporte. . . . .	69
1.16. Tabla resumen séptimo escenario operación después del soporte. . . . .	69
1.17. Gráfico de posiciones séptimo escenario operación después del soporte. . . . .	69
1.18. Tabla resumen octavo escenario. . . . .	70
1.19. Gráfico de posiciones octavo escenario. . . . .	70
1.20. Tabla resumen noveno escenario. . . . .	70
1.21. Gráfico de posiciones noveno escenario. . . . .	71
1.22. Tabla resumen décimo escenario. . . . .	71
1.23. Gráfico de posiciones décimo escenario. . . . .	71
1.24. Layout optimizado de la estación piloto. . . . .	72

# Índice de Tablas

4.1. Tabla ejemplo del conversor . . . . .	33
5.1. Tabla comparativa con tiempos APT . . . . .	37
5.2. Tabla comparativa con tiempos operario. . . . .	38
5.3. Tabla resumen de la extensión de la línea Trim A1. . . . .	38



# Listado de siglas y lenguaje especializado.

**APT** En las factorías Ford, el programa “Allocations Planning Tool (APT)” es donde se guarda la información que define la línea de montaje, se ordenan las operaciones de trabajo y se calcula la carga de trabajo.

**Buffer** Pequeños depósito donde se acumulan coches para estabilizar la cadencia de montaje. Acumulando la sobreproducción para evitar paros por bloqueos o abasteciendo estaciones para evitar paradas por falta de coches. Existen pequeños buffers en los nexos entre sublíneas (Entre A1 y A2, entre A2 y A3...) y uno más grande entre las líneas Chasis y Trim para conseguir independizar estas lo máximo posible.

**CLL** La base de datos “Cycle Line Layout (CLL)” es donde se guarda la información del mapa de la línea de montaje, así como los puntos de uso de cada pieza, las características de la estación, las herramientas, papeleras, etc.

**Data Frame** Estructura de datos bidimensional en la programación y análisis de datos. Es una forma de organizar y manipular datos en forma de tablas compuestas por filas y columnas, similar a una hoja de cálculo o una tabla de base de datos.

**FHEV** “Full Hybrid Electric Vehicle (Vehículo Eléctrico Híbrido Total)”. Vehículo con un motor eléctrico completamente funcional, independiente del motor de combustión y una batería que permite hacer recorridos cortos. No se puede conectar a la red eléctrica y se recarga mediante la energía cinética sobrante del coche, como puede ser la disipada por los frenos del coche. Esta tecnología permite realizar eléctricamente una cuarta parte del kilometraje aproximadamente.

**Happy Seats** Sillas robóticas asistentes que ayudan a los operarios en ciertas estaciones a entrar dentro del coche para no pisar por dentro, y así no ensuciarlo, en operaciones ergonómicas como agacharse repetidas veces o trabajar en puntos bajos del coche cuando no hay fosos.

**KP** “Key Project”. Una serie de proyectos que desarrolla el equipo de innovación para las diferentes plantas de la empresa FORD ESPAÑA S. L. Estos proyectos son parte del proyecto “Digital Twin” que pretende replicar toda la información de la factoría de forma digital para poder tener mayor trazabilidad, uso y accesibilidad de datos, así como mejor gestión y predicción de la planta.

**KPI** “Key Performance Indicator”. Indicadores clave para medir el rendimiento de nuevas funcionalidades.

**Layout** Disposición física o configuración de los elementos dentro de un espacio de trabajo o de una instalación industrial.

**Lean Manufacturing** Enfoque de gestión y organización de procesos industriales que busca maximizar el valor para el cliente eliminando desperdicios y actividades innecesarias en la producción. Se centra en la eficiencia y la mejora continua, con el objetivo de eliminar todo aquello que no agrega valor y optimizar los recursos disponibles.

**Modapts** Sistema de medición estandarizado y de análisis de tiempos utilizado en la industria para evaluar y mejorar la eficiencia de los procesos de trabajo. La sigla “Modapts” significa “Modular Arrangement of Predetermined Time Standards (Arreglo Modular de Normas de Tiempo Predeterminadas)” y se basa en la premisa de que cada movimiento (MODO) y actividad tiene un tiempo predeterminado asociado. Los MODOS son movimientos mínimos divididos principalmente en acciones de: Get (G), Move (M), Turn (giro) y Put (P). Entre otros. Podríamos añadir también como fundamental en Ford el Walk (W). A estos movimientos mínimos se le añade un número multiplicador dependiendo de la dificultad de este movimiento (mover un dedo será M1 mientras que mover todo el brazo desde el hombro será M4). Una vez definido el Modapt de la operación se sumarán (y multiplicará las acciones repetidas) todos los números asociados a los MODS y se multiplicará por aproximadamente 1/8 de segundo (en este caso, exactamente por 0.128s). Con este cálculo obtenemos el tiempo que se tarda en realizar una operación. Debido a motivos de confidencialidad, Ford no permite compartir información sobre los Modapts. Por lo que toda la información referida esta será ficticia.

**MVP** “Minimum Value Product”. Versión incompleta pero funcional de un producto que ya aporta valor.

**PHEV** “Plug-in Hybrid Electric Vehicle (Vehículo Eléctrico Híbrido Enchufable)”. Vehículo con un motor eléctrico completamente funcional e independiente del motor de combustión y una batería que permite hacer recorridos relativamente largos y se puede conectar a la red eléctrica. La capacidad de la batería del Kuga PHEV es de 14.4kWh, lo que permite realizar unos 60 km en modo eléctrico.

**QPS** “Quality Process Sheets”. Se refiere a las hojas de procesos de calidad. Su función es proporcionar a los operarios instrucciones detalladas sobre las operaciones y actividades necesarias para garantizar la calidad de un producto o servicio.

**Rebalanceo** Acciones tomadas por el equipo de Ingeniería Industrial de cualquier línea de montaje para llegar a una solución más óptima, mediante el análisis detallado del proceso de producción y la distribución de material en la línea. Su principal objetivo es ahorrar carga de trabajo y reducir los costes del proceso.

**Tiempo “blocked”** Tiempo que el coche está parado debido a que el operario no ha tenido tiempo de finalizar las tareas asignadas.

**Tiempo “starved”** Tiempo que el operario está esperando a que entre el coche en la estación de trabajo, ya que ha completado el proceso asignado en menos tiempo del esperado.

**Time Study** Base de datos donde se agrupan y ordenan las operaciones elementales que deben realizarse en una misma estación. En el Time Study, se muestran datos como los modaps, las piezas referenciadas a cada operación, las herramientas necesarias, la descripción del proceso, etc.

**Parte I**

**Memoria**





# Capítulo 1

## Objeto del proyecto

El presente proyecto nace de la necesidad de hacer más y mejores rebalanceos con menos recursos en la planta de Montaje que la empresa FORD ESPAÑA, S.L tiene en Almussafes (Valencia). El proyecto “Workstation Optimization and line rebalancing”, tiene como objetivo desarrollar y poner en marcha un asistente para los Rebalanceos que realizan los ingenieros industriales de la planta de Montaje de Ford . Este, se enmarca dentro de los proyectos de innovación en los que la empresa está trabajando, llamados ”key project”(KP), y más específicamente en el KP2: Workstation Optimization & Line Rebalancing. Estos objetivos, son parte del proyecto “Digital Twin”que pretende replicar la información de la factoría de forma digital para poder tener mayor trazabilidad, uso y accesibilidad de datos, gestión y predicción de la planta.

Se ha intentado desarrollar anteriormente pero, por falta de un equipo propio y recursos no se había podido indagar en el asunto. En el momento en el que se crea un equipo de innovación y se inician los proyectos del “Digital Twin”se decide rápidamente abordar de nuevo este problema.

El propósito principal de este proyecto es minimizar el número de pasos que hacen los trabajadores, mediante la optimización de la distribución del mapa de la planta y la secuencia de operaciones. Para llevarlo a cabo, se utilizará la metodología Ágil Scrum, creada para resolver problemas complejos y con requisitos cambiantes y así obtener resultados periódicos en un proyecto. Se decide tener sprints semanales y cuatro “Minimum Viable Product (MVP)” trimestrales.

La creación del asistente para los Rebalanceos deberá contemplar los siguientes MVPs:

0. Abordaje del problema y debatir qué datos se necesitan, de dónde se cogerán y en qué formato se deben tomar.
1. Creación de un simulador de pasos que proporcione una visualización confiable y muestre Indicadores Clave de Rendimiento (KPI) relevantes como los que se muestran a continuación:
  - Tiempo de ciclo de la estación de trabajo (Tiempo que pasa desde que el operario empieza a trabajar en un coche hasta que realiza todas las operaciones).
  - Tiempo de pasos.
  - Tiempo “starved”. (Tiempo que el operario espera hasta que el coche entra en la estación de trabajo).

- Tiempo “blocked”. (Tiempo que el coche está parado debido a que al operario no ha tenido tiempo de acabar de montar el coche).
- Frecuencia con la que pasa cada opción. Siendo las opciones, operaciones que solo se realizan en los coches que se han encargado con ciertas características. Una posible opción sería el coche con el volante a la derecha o alguna de las diferentes opciones híbridas que ofrece la compañía.
- Peso de la opción. Considerando el peso como un número que indica la prioridad de rebalancear en función de la influencia que tenga esa operación en el total de acciones del puesto de montaje.

Estos datos servirán como herramienta para ayudar a los ingenieros industriales en la toma de decisiones sobre cómo redistribuir los materiales en las líneas de montaje.

2. Desarrollo de un optimizador de diseño automático de Layout que sea funcional para todas las estaciones de la planta de Montaje. Este optimizador permitirá mejorar el rendimiento y productividad de las operaciones en cada estación.
3. Implementación de un optimizador del orden y la simultaneidad de las operaciones, con el objetivo de actualizar la secuencia en la que se realizan las tareas en la planta de montaje para afinar la eficiencia.
4. Integración de los optimizadores desarrollados en los puntos anteriores para obtener una secuencia de operaciones y un diseño óptimo para cada estación de trabajo.

Por tanto, a través de este trabajo se busca mejorar la eficiencia de una planta de Montaje de automóviles, así como reducir los tiempos de producción y optimizar los recursos disponibles. Además, de proporcionar herramientas efectivas para la toma de decisiones informadas en cuanto a los Rebalanceos necesarios en las líneas de montaje.

Debido a que mis prácticas finalizarán antes de la entrega del MVP2, el enfoque principal del Trabajo de Fin de Grado (TFG) se centrará en el desarrollo y la implementación del MVP1. En este sentido, se pondrá especial énfasis en la creación de un simulador de pasos que muestre una visualización confiable y proporcione KPI relevantes para ayudar a los ingenieros industriales en la toma de decisiones respecto a los Rebalanceos de las líneas de montaje.

Si bien el desarrollo completo de los MVPs planteados en el proyecto no estará cubierto en el TFG, se espera que los resultados obtenidos hasta el momento sirvan como una sólida base para futuros trabajos y contribuyan a la mejora de la eficiencia y productividad de la planta de Montaje.

En conclusión, este proyecto representa una oportunidad de aplicar las competencias adquiridas durante mi formación académica, como Graduado en Mecánica, e iniciarme en la práctica profesional, dentro del ámbito de la automoción, a través de la optimización de procesos industriales.

## Capítulo 2

# Antecedentes

### 2.1. Características de la planta de Montaje de Ford Almussafes.

La planta de Ford Almussafes, perteneciente a FORD ESPAÑA S.L., fundada en 1976, destaca como una de las factorías automovilísticas más extensas y productivas de Europa. Con una capacidad de producción de 2100 coches diarios, se han llegado a fabricar en esta planta cinco modelos diferentes: Kuga, Mondeo, Galaxy, S-Max y Transit-Connect, obteniendo el título de manufacturera automoviística con más complejidad de Europa. Esta diversidad de modelos implica un proceso productivo altamente complejo, debido a la gran cantidad de variables y características específicas que cada uno de ellos posee.



**Figura 2.1: Fotografía Aérea de la factoría FORD ESPAÑA S.L.**

Además de su destacada capacidad de ensamblaje de vehículos, FORD ESPAÑA S.L. se posiciona como una empresa manufacturera de motores de renombre. La planta de Almussafes tiene una capacidad máxima de producción de hasta 2250 motores diarios, incluyendo los modelos 2.0 EcoBoost SCTi de 203, 240 y 250cv, el 2.0L GDI de 150cv, el 2.5L HEV de 175 cv y el corazón del emblemático Ford Mustang, el motor 2.3L EcoBoost SCTi de 310 y 350cv.

El proceso de producción en la planta de Almussafes se desarrolla en una factoría dividida en cuatro áreas: Motores, Body and stamping (estampado y soldadura de la chapa de la carrocería), Pinturas y Assembly (Montaje final). Asimismo, la planta cuenta con un parque de proveedores propio y dedicado casi exclusivamente a Ford. Este parque está conformado por más de 120 empresas pro-

veedoras, empleando a más de 8500 trabajadores. Estos proveedores están encargados de realizar el preensamblado, presecuenciado, fabricación y suministro de las piezas que se utilizarán en los vehículos de la marca Ford. Emplean diferentes métodos de suministro, siendo algunos de ellos mediante camiones, mientras que otros suministran piezas estratégicas, como salpicaderos, radiadores y tubos de escape, a través de un sistema que en su momento fue pionero en el mundo que conecta el parque de proveedores con la factoría mediante un túnel y balancinas para transportar las piezas directamente en secuencia desde el parque de proveedores hasta la estación en la que se ensambla en la línea de montaje. Este modelo de suministro, ha permitido optimizar la eficiencia y agilidad del proceso de producción.

En la planta de Almussafes, Ford opera con dos sistemas independientes: el sistema A y el sistema B. Actualmente, el sistema A se dedica exclusivamente a la fabricación del modelo Kuga, aunque hasta hace un año también se producían los modelos Mondeo, S-Max y Galaxy. Por otro lado, el sistema B se encarga de la producción de los modelos Kuga y Transit-Connect. Estos sistemas se distinguen tanto en la planta de Ford como en la logística de los proveedores, quienes deben asegurar que las piezas sean enviadas de manera secuencial y en un orden específico, ya sea mediante camiones o balancinas, con el objetivo de agilizar el proceso de ensamblaje y garantizar la correcta selección de las piezas correspondientes.

Cómo se ha dicho anteriormente, la factoría de Ford se divide en cuatro plantas; Motores, Body, Pinturas y Assembly. Debido a que he realizado las prácticas en Montaje, únicamente explicaré el funcionamiento de esta.

La planta de Montaje, no es que no sea una excepción para el funcionamiento mediante dos sistemas, sino que es fundamental esta diferenciación para la distribución de las operaciones a lo largo de todo el proceso de ensamblaje.

La planta de Montaje de Ford Almussafes se compone de diferentes secciones y áreas que abarcan todas las etapas del proceso de ensamblaje del vehículo. A continuación, se detallan las secciones presentes tanto dentro como fuera de la línea de producción ordenadas siguiendo el flujo desde que las carrocerías entran a la planta de Montaje hasta que sale el coche terminado:

Fuera de la línea de montaje:

- **MLV (Motor Line Vehicle):** Formada por una única línea conjunta para el sistema A y sistema B. En esta sección se monta la parte electrónica del motor y se realiza el ensamblaje del motor con la caja de cambios.
- **Baterías:** Se cuenta con dos líneas independientes para el montaje de las baterías destinadas a alimentar los motores independientes, tanto para vehículos PHEV como FHEV. No se realiza distinción entre los sistemas A y B de la planta. Las baterías se envían posteriormente a la línea de montaje.
- **Salpicaderos:** En esta sección se lleva a cabo el montaje completo de los salpicaderos, que luego son ensamblados directamente en la línea de Trim. Se cuenta con dos líneas independientes diferenciadas por modelos, no por sistemas. Actualmente, debido a la desaparición del S-Max/Galaxy, únicamente está activa la línea del Kuga ya que el salpicadero de la Transit lo monta un proveedor. Esta línea está en el parque de proveedores aunque pertenece a Ford. Se envían los salpicaderos mediante balancinas a través del túnel.
- **Foamizados:** Aquí se producen las espumas de los asientos, que posteriormente son ensam-

bladas por un proveedor externo.

Dentro de la nave de montaje:

- **ASRS (Automated Storage and Retrieval System):** Es un almacén encargado de resecuenciar los vehículos provenientes de la sección de Pinturas. Aquí se asigna el VIN (Número de Identificación del vehículo) y los coches llegan al lugar con el chasis pintado y con los patines ya instalados. Salen de aquí todos los coches en la misma línea y más adelante se separarán.
- **Pretrim:** De la secuencia proveniente de ASRS, se separan los coches que tienen las opciones especiales, de ventanas fijas en Transit y techo solar en Kuga, para realizarles ciertas operaciones. Luego se devuelven a su sitio en la secuencia
- **Mezzanine:** Además de actuar como un buffer a la entrada de Montaje, aquí se realizan operaciones para quitar las puertas y se separan los coches que van al sistema A y al sistema B.
- **Línea de puertas:** Se dispone de dos líneas independientes para el montaje de las puertas, una para Trim A y otra para Trim B, donde se realizan los montajes necesarios para cada tipo de puerta. Una operación característica podría ser poner los cristales, cerraduras, altavoces...
- **Trim A y Trim B:** Junto a Chasis, las líneas más importantes de la planta de Montaje. Estas son dos líneas de ensamblaje independientes divididas por los Sistemas A (Trim A) y B (Trim B). El coche se desplaza sobre patines por la línea a través de una cadena, pasando por las estaciones de trabajo designadas a lo largo de las sublíneas ExtA1, A1, A2, A3, B1, B2, B3. Aquí se realizan los primeros montajes, ruteados y atornillados de cables y recambios, especialmente los del habitáculo y la parte superior del coche. Estas líneas están altamente automatizadas, siendo completamente montadas por robots piezas clave tan pesadas y delicadas como el ensamblaje del salpicadero, el parabrisas y los asientos, entre otros.
- **Buyoff:** Es un Buffer (almacén) separado, automatizado e independiente para desvincular Trim y chasis. En esta sección se retiran los patines y el coche es tomado por un pulpo (gancho que coge el coche desde arriba para permitir el acceso al coche por la parte inferior) mediante varios robots.
- **Chasis A y Chasis B:** Aquí se continúa con el montaje del coche. El coche se desplaza suspendido por una electrovía y cogido por los carriers o “pulpos”, los cuales se regulan la altura pudiéndola personalizar cada operario según sus necesidades mejorando la ergonomía. El coche pasa por las estaciones de trabajo designadas a lo largo de las sublíneas Ch01A, Ch01B, Ch04A, Ch04B, Ch05A, Ch05B, Ch06A y Ch06B. Aquí se realizan los montajes de la parte mecánica del coche, especialmente las que están ubicadas en la parte inferior y los laterales. Algunas de las operaciones más características de la línea de Chasis son el montaje de las ruedas, ensamblaje de faros, el montaje del tubo de escape o el ensamblaje de las baterías en los Kuga PHEV.
- **Pallet-conveyor:** Es una única línea que combina los dos sistemas donde se acoplan el motor, el radiador y el eje delantero y se rutean cables necesarios para el control del coche.



**Figura 2.2: Coche desplazandse por la electrovía en la línea de Chasis**

- **Decking:** En esta sección se unen los sistemas, combinando la carrocería con la transmisión y el eje trasero. Se ensambla con el conjunto proveniente del pallet. Este punto de la planta es crucial, ya que el tiempo de ciclo de la línea es de 30 segundos en lugar de 60 segundos, y si se detiene, se paraliza toda la producción.
- **Wheel alignment:** Aquí se lleva a cabo la alineación correcta de las ruedas para evitar desviaciones. Se utilizan rodillos y controles, y cinco equipos mezclan las dos líneas, sin separación entre Sistema A y Sistema B. Ya no se vuelven a separar los coches entre el Sistema A y el Sistema B.
- **OK line:** Se realiza la comprobación de calidad en esta sección. Existen dos líneas que permiten realizar más chequeos de calidad y aumentar el tiempo de ciclo.
- **Test de agua:** En una línea dedicada, se verifica que el coche sea completamente hermético y no permita el ingreso de agua.
- **FAL:** Esta sección está destinada a realizar más chequeos de calidad, con una única línea y un tiempo de ciclo de 30 segundos. Además, se descarga el software del coche.
- **Pista de pruebas:** Se trata de una prueba en condiciones reales de conducción para verificar la ausencia de vibraciones extrañas y asegurarse de que el coche cumple con los estándares de conducción.

## 2.2. Motivación y justificación

El proyecto se fundamenta en la necesidad de abordar los desafíos asociados al proceso de Rebalanceo en una planta de producción de la empresa FORD ESPAÑA, S.L. Tradicionalmente, los Rebalanceos en esta planta, se han llevado a cabo con un enfoque intuitivo basado en la experiencia de los ingenieros industriales y los equipos de producción y corroborado mediante tablas, lo que ha generado una falta de datos cuantitativos precisos. Esta falta de información confiable ha originado

ineficiencias considerables y ha requerido una inversión significativa de tiempo y trabajo manual de los operarios que trabajan en ella.

Para poder realizar esta tarea con éxito es necesario que el ingeniero industrial tenga perfecto conocimiento del proceso del montaje de los diferentes vehículos, de ergonomía y seguridad, del método Lean Manufacturing, de implicaciones en coste de los cambios de layout.

Dentro del contexto industrial, es bien conocida la dificultad y complejidad inherente que supone la tarea de rebalancear las operaciones de una planta de producción. Esta complejidad se deriva de múltiples factores, como la variabilidad en la demanda, la asignación inadecuada de recursos, la enorme variabilidad entre dos productos del mismo tipo y/o la falta de un marco sistemático para la toma adecuada de decisiones, entre otros.

En este sentido, el objetivo principal del proyecto es automatizar el proceso de Rebalanceo, eliminando en el mayor grado posible la dependencia del factor humano y buscando una mayor precisión y eficiencia en la asignación de recursos. Dejando el factor humano únicamente para validar las soluciones obtenidas, así como su viabilidad económica y ergonómica. Mediante la aplicación de técnicas y algoritmos avanzados, se desarrollará un sistema inteligente capaz de analizar los datos de producción en tiempo real, así como de evaluar la carga de trabajo y proponer ajustes óptimos para lograr una distribución equilibrada de las tareas.

Durante los últimos años, la empresa ha realizado varios intentos para abordar este desafío, incluyendo un proyecto colaborativo con la Universitat de València hace 4 o 5 años. Sin embargo, debido a la complejidad inherente al problema, ninguno de estos intentos anteriores ha logrado una solución satisfactoria y aplicable a gran escala.

En el pasado, los Rebalanceos se realizaban de forma manual, lo que implicaba introducir los cambios en el APT, el software de programación de la producción, calcular las cargas de trabajo correspondientes mediante el sistema Modapts, realizar modificaciones en el diseño del Layout según la metodología Lean Manufacturing y definir la secuencia de operaciones basándose en la experiencia del ingeniero industrial y teniendo en cuenta factores como la ergonomía o que, en muchos casos, para poder montar algunas piezas, se tiene que haber montado otras previamente. La validación de los cambios se realizaba posteriormente, corrigiendo errores en caso de que se produjeran. Esto podía causar problemas en la línea de producción ya que, un error en la carga de trabajo de una estación ralentizaría la línea, creando un efecto en cadena que podría paralizar completamente al resto de estaciones. Situación que afectaría muy negativamente al rendimiento, producción, productividad, KPI y rentabilidad de la planta.

Con la automatización planteada, se busca también superar las limitaciones de las soluciones previamente implementadas, por ejemplo, el uso de un panel de control con gráficos que proporcionaba información basada en datos históricos. Este enfoque pasivo solo permitía una visión retrospectiva y no facilitaba la planificación anticipada de los Rebalanceos. En contraste, el proyecto actual proporcionará una herramienta predictiva que permita tomar decisiones informadas, ágiles, automatizadas y basadas en datos, así como anticiparse a posibles desequilibrios en la carga de trabajo de la planta de producción.

En resumen, el principal motor de impulso en este proyecto de ingeniería ha sido la necesidad de optimizar el proceso de Rebalanceo en una planta de producción, eliminar la subjetividad en la toma de decisiones, mejorar la eficiencia en la asignación de recursos y obtener resultados confiables y precisos. Todo ello, a través de la aplicación de metodologías y técnicas avanzadas, con el objetivo de obtener soluciones innovadoras que contribuyan a la mejora continua de los procesos de

producción en la industria automovilística de la Comunidad Valenciana.



## Capítulo 3

# Factores a considerar

Puesto que nuestro principal objetivo es la creación de un simulador de pasos con que proporcione una visualización confiable y muestre los KPI adecuados, en este apartado vamos a analizar las características que debemos considerar para realizar los cálculos de este simulador, así como los condicionamientos asociados.

### 3.1. Características fundamentales

A continuación, damos el listado de las características necesarias para realizar los cálculos del simulador. El análisis de todas ellas nos facilitará encontrar la solución más conveniente.

El simulador debe imitar de forma fiable las tareas realizadas por los operarios, por lo que debe tener la funcionalidad de ser:

- **Esquivador:** El operario virtual no debe atravesar el coche durante las operaciones.

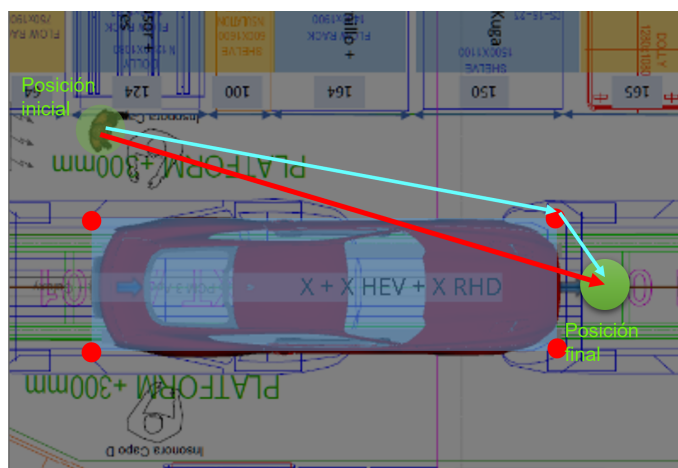


Figura 3.1: Esquema de la funcionalidad “Esquivador”

- **Realista:** El operario virtual debe realizar las operaciones indicadas en las QPS (“Quality Process Sheets”) y en el APT (“Allocations Planning Tool”).

El simulador debe considerar las características propias de los operarios, como:

- **Capacidad en las manos:** El operario virtual debe considerar la habilidad de los operarios para manipular diferentes componentes y herramientas.
- **Inputs Necesarios en cada operación:**
  - Tiempo de operación.
  - Posición de la operación (x e y).
  - Opciones en las que se realiza la operación.
  - Lugar donde se realiza la operación: en las estanterías o en el coche (relativo).
  - Descripción de la operación.
- **Parámetros fijos:**
  - Dimensiones y límites de la estación.
  - Velocidad del operario y de la línea de producción.
  - Dimensiones del coche.
  - Distancia entre coches.
- **Tiempo “starved” y Tiempo “blocked”** (definidos el glosario)

### 3.2. Limitaciones y condicionamientos

A la hora de abordar este proyecto, es importante considerar que en una factoría tan grande hay estaciones poco estandarizables, que no se han considerado, documentación errónea o desordenada en algunos ámbitos o procesos, entre otras cuestiones. Esto impiden que la información de la que disponemos sea totalmente completa o fidedigna, por lo que existen ciertas condiciones que debemos tener en cuenta. Destacamos a continuación las que afectan más directamente al desempeño de nuestra propuesta.

- **Operaciones negras:** Existen operaciones que no están documentadas en la hoja de procesos (QPS) y referencias de piezas que no están registradas en el APT.
- **Procesos poco estandarizados:** Aunque el orden de las tareas está definido en la QPS, en las operaciones con bajo riesgo de producir defectos de calidad, los operarios tienen libertad de cambiar el orden al que consideren conveniente y más eficaz. Los ingenieros industriales redactan el APT manualmente, de acuerdo a sus preferencias.
- **Desorden:** Las acciones en el APT están ordenadas por operaciones completas de trabajo (Vete y coge el material / Vete al coche y monta el material / Vuelve al stock) y el simulador necesita el orden en el que se realizan las operaciones. Debido a esto, el Time Study se debe reescribir ordenadamente para que su uso sea ágil.
- **Modaptss no fiables al 100 % :** Los Modaptss, utilizados para medir el tiempo de ejecución de las operaciones, pueden presentar cierta variabilidad y no proporcionar una precisión absoluta.

- **Estaciones atípicas:** Son estaciones de trabajo con características inusuales o especiales que deben ser consideradas en el diseño del sistema.
  - **Crab Line:** Líneas dónde el coche se desplaza lateralmente. Hay que poner la condición de que el coche pueda girar cada  $90^\circ$  grados y así poder desplazarse hacia adelante, hacia atrás y lateralmente. En esta línea hay tres operarios y hace cada uno 3 estaciones seguidas. → El tiempo de ciclo será de aproximadamente 180s.



**Figura 3.2: Crab Line**

- **Carros HOFU:** Estanterías que se mueven con el coche. En el simulador indicaremos relativo = 1 cuando vaya a coger la pieza.
- **Cajas Kitting:** Cajas que se rellenan previamente con el material que hace falta para varias operaciones en cada coche, por lo que no se tiene que ir a las estanterías a recoger el stock. Indicaremos relativo = 1 cuando vaya a coger la pieza.



**Figura 3.3: Happy Seat**

- **Happy Seats:** Sillas robóticas que ayudan al operario a entrar dentro del coche sin dañarlo. Puesto que el operario tiene el material en la silla, no irá a las estanterías y por tanto no hay desplazamientos que optimizar.
- **Manriders:** Cintas que transportan al usuario a la velocidad del coche para evitar que este tenga que andar mientras avanza el coche. No influye en nuestro diseño, puesto que despreciaremos estos desplazamientos ya que no son optimizables.

- **Pallet, Salpicaderos, MLV...:** No se montan coches, solo componentes de estos. Hay que definir los ejes de coordenadas.



**Figura 3.4: Planta de Salpicaderos**

- **Estaciones estáticas:** El coche llega a la estación, se para y los operarios empiezan a montar. Aquí la problemática es complicada por la naturaleza del funcionamiento de los tiempos "starved" y "blocked". Una opción para forzar al programa a parar el coche en el punto que nos interesa, es poner el límite de la estación donde tiene que pararse el coche. De la misma forma, para que empiecen a montar el coche, se tendría que añadir el inicio de la estación en el punto donde interesa comenzar a montar la primera pieza. Esto podría crear un bucle infinito, si hay alguna pieza que se monte antes, ya que el operario esperará que el coche entre en la estación para hacerlo, pero no entrará nunca debido a que el coche esperará a que el operario termine de montarlo para poder seguir avanzando. Además, estas soluciones modificarían los datos de output. La solución más sencilla es añadir una operación al principio con:  $t = (x_{coche} + d_{separacion})/v_{coche}$
- **Ergonomía:** Se debe tener en cuenta las condiciones de las tareas y garantizar la correcta adaptación del puesto, para garantizar que las operaciones realizadas en él sean seguras y cómodas para los operarios.
- **Discriminación de procesos:** Distinguir cuando el operario realiza desplazamientos optimizables (generalmente referidos al stock) y cuando anda mientras hace una operación junto al coche.

## Capítulo 4

# Descripción de la solución adoptada

La solución adoptada para abordar los desafíos identificados en el proceso de rebalanceo de la planta de Montaje de FORD ESPAÑA S.L., consiste en el desarrollo de un optimizador de layout automático que organice estos procesos de la mejor forma posible basándose en datos objetivos. Esta solución se divide en dos subaplicaciones interconectadas: una que se encarga de la optimización del layout y otra que organiza la secuenciación de operaciones.

Para lograr este objetivo, se ha propuesto el proyecto descrito y desarrollado en este TFG: la creación de un software de simulación que proporcione datos relevantes para comparar distintas casuísticas.

Previamente al desarrollo de este simulador virtual se abordará el problema y se verá los datos y cauísticas necesarias para el correcto funcionamiento del simulador. Este análisis previo permitirá crear el simulador de manera mucho más robusta y con las diferentes funcionalidades más integradas entre sí. El simulador debe ser utilizable desde etapas tempranas del proyecto y proporcionar información valiosa que permita adaptar las herramientas existentes en la empresa Ford, como el APT (“Allocation Process Tools”) y el CLL (“Cycle Line Layout”), a los requisitos del simulador. Además, su uso anticipado permitirá identificar problemáticas no consideradas durante su elaboración.

La elección del lenguaje de programación R se basa en su simplicidad y capacidad para el análisis de datos, el análisis estadístico y la visualización de gráficos. Esto es porque R está creado pensando en este problema y tiene estas funcionalidades altamente optimizadas e integradas sin necesidad de instalar librerías no integradas como serían “pandas” en Python. Alternativamente en R hay una gran comunidad y muy activa, lo cual nos permitirá comunicarnos con gente que ha tenido problemas similares a los que tendremos nosotros así como muchas librerías gratuitas con funciones que resuelven problemas. Como la e1070 que permite resolver problemas de grados y utilizaremos en el desarrollo del simulador. Sin embargo, una vez construido valoramos traducirlo a Python. Esta transcripción asegurará la integración de todas las soluciones y mejoras detectadas durante el proceso de desarrollo, consiguiendo de esta forma un simulador más robusto y eficiente.

La decisión de migrar el código a Python también se justifica por el hecho de que el optimizador estará escrito en este lenguaje ya que Python está mucho más extendido que R y se integra fácilmente con otras tecnologías y herramientas populares. Python también nos permite programar por objetos, este tipo de programación es muy útil para la lógica que se pretende utilizar en el optimizador, lo que lo convierte en una excelente opción para proyectos que requieren interacción con

bases de datos, servicios web, API y otras tecnologías. Aunque haya que utilizar librerías externas como puede ser “pandas”, al tener el simulador y el optimizador en el mismo lenguaje de programación, se evitarán llamadas repetidas a archivos externos, lo que mejorará significativamente la eficiencia del sistema. El optimizador creará y evaluará soluciones aleatorias que se iterarán con un algoritmo genético que generará 40 descendientes en cada una de las 30 iteraciones previstas, con un resultado total de 1200 iteraciones. Estos descendientes se obtendrán en función de los mejores “padres” aunque, con poca probabilidad, se seguirán creando escenarios aleatorios que permitan valorar opciones que se puedan haber desconsiderado en las primeras iteraciones. Este algoritmo genético nos permite analizar un amplio espectro de soluciones sin requerir excesiva potencia computacional y obtener una solución suficientemente optimizada.

A pesar de que el desarrollo del optimizador no es objeto de este TFG, es interesante conocer a grandes rasgos su previsión de funcionamiento para enfocar el simulador a los requerimientos del futuro optimizador.

## 4.1. Inputs

Dentro del marco del proyecto de ingeniería, se requieren varios inputs para llevar a cabo los cálculos necesarios en el simulador de procesos. Estos inputs se almacenarán en un Libro de Excel con varias hojas. A continuación, se describen las diferentes hojas que componen este libro y su función dentro del simulador:

1. **Hoja “Worksteps”**: Contiene la información principal de cada estación de trabajo, incluyendo los siguientes campos:
  - Step: Número de secuencia de la operación.
  - Option: Opción en la que se realiza dicha operación. Entendemos como diferentes opciones las operaciones, relacionadas con las especificaciones pedidas por el cliente, que no se montan en todos los coches del mismo modelo.
  - Description: Descripción de la operación.
  - Relativo: Parámetro que indica si la operación se realiza en la línea (0). Siendo esta una operación que se realiza en el mismo punto independientemente de la posición del coche. O dependiendo de la ubicación del coche (1). Por lo que siempre se irá a puntos diferentes de la línea.
  - Part number: Referencia de la pieza asociada cuando el valor de “Relativo” es 0.
  - x: Posición en el eje x, en formato relativo o absoluto.
  - y: Posición en el eje y, en formato relativo o absoluto.
  - Tiempo: Tiempo requerido para realizar la operación.
2. **Hoja “Mix”**: Se utiliza para definir la secuencia de coches con las diferentes opciones que se simularán.

Permite analizar un solo coche, la secuencia deseada por el ingeniero o la secuencia real del día elegido tanto pasado como futuro. Las opciones especificadas en esta hoja solamente se tendrán en cuenta si coinciden exactamente con las opciones establecidas en el “Time Study”.

A	B	C	D	E	F	G	H
Step	Option	Description	relativo	x	y	Tempo	
2	20 X	#FTB00100 (CTD) Va + Coge bracket pilar A (#Part LV4B-S026A52-A*) + Va a vehiculo I (#WES A1_1485) (#F)	0	LV4B-S026A52-A*	1,416	-1,918	0,54
3	30 X	#FIL03021 (CTD) Coge 2 clips cable tirador capo (#Part W719610-S300) (#F)	0	-W719610-S300	1,416	-1,918	1,02
4	40 X	#FIL03021 (CTD) Posiciona + Asienta 2 clips a dieta izquierda	1		0,75	-1	3,12
5	50 X	#FTB00100 (CTD) Alinea y asienta bracket pilar A a 2 esparragos carroceria I	1		1,25	-1	1,44
6	60 X	#FTB00100 (CTD) Va a stock + Coge Pistola + Va a vehiculo + Deja pistola I (#F)	0	Pistola	2,282	-1,918	1,02
7	70 X	#FTB00100 (CTD) Coge 2 buercas (#Part W707137-S437) (#F)	0	-W707137-S437	1,416	-1,918	1,02
8	80 X	#FTB00100 (CTD) Asienta 2 buercas a esparragos soporte pilar A I	1		1,25	-1	5,94
9	90 X	#FTB00100 (CTD) Asienta (2) buercas soporte pilar A (#DC 4468 Boca 10mm) I (#WES A1_511)	1		1,25	-1	6,06
10	230 X	#FTN00300 (CTD) Coge 1 "Mastick Pad" foam veniente (#Part LV4B-S278F00-A*) I (#WES A1_1906) (#F)	0	LV4B-S278F00-A*	1,416	-1,918	0,39
11	160 X	#FEF00000 (CTD) Va a stock + Coge base antena (#Part HS7T-19K351-B*) + Va a vehiculo I (#WES A1_1731) (#F)	0	HS7T-19K351-B*	1,416	-1,918	1,14
12	240 X	#FTN00300 (CTD) Quita papel protector I (#F)	0	-Papel	1,416	-1,918	1,44
13	250 X	#FTN00300 (CTD) Tra papel protector I (#F)	0	-Papel	1,416	-1,918	0,54
14	260 X	#FTN00300 (CTD) Posiciona + Asienta parche en zona superior pasa de rueda trasero I	1		-0,75		1,32
15	170 X	#FNEGRA (CTD) Coge protector para asentar base antena (evitar daños) (#F)	0	-Protector	3,409	-1,5	0,66
16	180 X	#FNEGRA (CTD) Monta protector para asentar base antena (evitar daños)	1		-0,75		1,02

Figura 4.1: Plantilla hoja Worksteps

3. Hoja “Parameters”: Proporciona información sobre diferentes parámetros relacionados con el coche, el operario y la estación de trabajo. Algunos de los parámetros incluidos son:

- Posición inicial del coche y del operario.
- Dimensiones del coche.
- Dimensiones de la estación de trabajo.
- Velocidad del coche y del operario.
- Separación entre coches.

Una vez validado el simulador, se han añadido ciertas funcionalidades adicionales al libro de Excel:

#### ■ Hoja “Copia Time Study”:

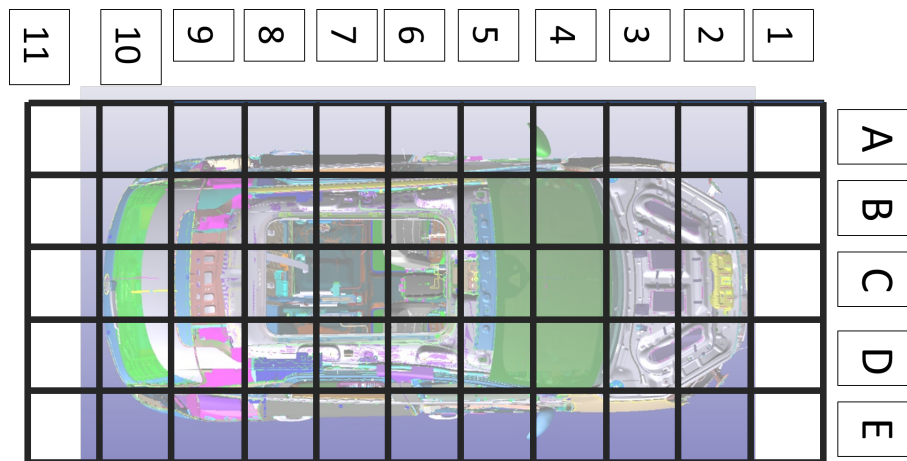
Facilita al ingeniero la introducción de de los datos necesarios y que automáticamente se transformen al formato de la hoja “Worksteps”. Para evitar modificaciones que puedan afectar al simulador, se copian y pegan los datos del “Time Study” a partir de la celda B2. En la columna A, se introduce manualmente el número de pasos considerados en el “Time Study”. Estos pasos los cogemos de la columna “Modapts” (N) y teniendo en cuenta que se debe considerar el número anterior a cualquier “W5”(número que indica el número de pasos). Por ejemplo, si para una operación se tiene “4W5; M5G4M5P0; 4W5”, se considerarán 4W5 + 4W5, lo que resulta en 8 pasos. Esta hoja proporciona directamente los datos de Step, Option, Description, Part Number y relativo. Además, se obtiene el tiempo en una columna titulada “tiempo”, donde se convierte de minutos a segundos y se resta el número de pasos según la fórmula:  $Tiempo_{total} = (Time * 60) - (n_{pasos} * 5 * 0,128)$ .

#### ■ Hoja “Ubicación Relativo 1”:

Se almacena la posición de las coordenadas x e y. El proceso actual implica ir a la línea, analizar la posición y anotar, mediante “coordenadas alfanuméricas de filas y columnas”, en qué casilla se realiza cada operación.

Sin embargo, se pretende guardar las coordenadas en una base de datos para no tener que hacer este proceso en futuras consultas. A continuación, se leen estas coordenadas en la hoja “Worksteps” y se registran las letras correspondientes en el eje OY y los números en el eje OX. Finalmente, se convierten a coordenadas cartesianas utilizando las siguientes fórmulas:

- **Coordenadas x:** se multiplica el número de la derecha por 0.5 (tamaño de cada casilla) y se resta 0.75 ya que el 0 se ubica en el centro de la casilla C2.



**Figura 4.2: Plantilla para la anotación de las operaciones Relativo 1**

- **Coordenadas y:** se utiliza una concatenación de las funciones COLUMN e INDIRECT para convertir la letra en un número de columna. Luego, se multiplica por 0.5 (dimensión de las celdas) y se resta 1.5 (el eje de coordenadas se encuentra en el centro de la casilla C2).

- **Hoja “Puntos Uso Estación”:**

El ingeniero debe escribir el nombre del punto de uso (por ejemplo, estantería, bebedero, flow rack) y, mediante la fórmula “Buscar X”, se obtiene desde el CLL la posición absoluta del punto de uso respecto al layout completo de toda la planta, así como las piezas que contiene.

- **Hoja “Ubicación Relativo 0”:**

Filtra las piezas, piezas negras (piezas que no están registradas por no ser definitivas. Algunas podrían ser los protectores o los útiles de pinturas), herramientas y papeleras. Busca coincidencias en el campo “Piezas” de la hoja “Puntos Uso Estación”. Una vez encontradas, se obtiene el punto de uso en el que se encuentran y su posición (absoluta respecto al 0 del layout completo) en milímetros con respecto al punto de inicio de la estación. Esta posición se convierte de milímetros a metros y se resta a la posición inicial de la estación para obtener la posición relativa al punto de inicio.

Utilizando estas hojas, simplemente copiando el “Time Study”, ingresando el número de pasos y anotando los puntos de uso de la estación, además de comprobar los parámetros en la hoja “Parameters”, se completará automáticamente la plantilla del simulador.

## 4.2. Funcionalidades del simulador

A continuación, se presenta paso a paso la descripción del funcionamiento del programa que realiza la simulación, desarrollado para resolver el problema planteado.

### 1. Parámetros iniciales del simulador:



Con el objetivo de abordar el problema propuesto, se ha desarrollado una aplicación web que permite a los trabajadores del departamento de Ingeniería Industrial insertar el archivo descrito a lo largo de la sección de “Inputs”. La tabla debe contener los parámetros necesarios completados para llevar a cabo con éxito la simulación:

- Posición relativa o absoluta respecto al coche.
- Secuencia de coches.
- Posición de la operación.
- Tiempo requerido para realizar la operación.
- Tamaño del coche.
- Velocidades del coche y del operario.
- Descripción y orden de las operaciones.

## 2. Cálculos iniciales:

Una vez se insertan los datos en la aplicación web, el programa genera los datos necesarios para realizar los cálculos y los guarda en un Data Frame. Este archivo incluye información como el tiempo de cada paso, la posición actual del operario y del coche, la velocidad del operario, el tiempo total de los pasos y el tiempo acumulado.

Mediante un bucle que tiene en cuenta diversas condiciones, se procede a rellenar automáticamente este Data Frame para calcular todos los pasos de los coches deseados. Los parámetros de la fila elegida se toman como parámetros finales, mientras que los parámetros iniciales corresponden a los de la fila anterior.

Este apartado se desarrollará con detalle en el punto 4.

## 3. Ejes de coordenadas:

Indicamos a continuación los ejes de coordenadas en las estaciones de la cadena de montaje y en el coche.

- **Estación:** En la línea, hay estaciones en las que el coche entra por la derecha y estaciones en las que entra por la izquierda. Esto sucede porque la línea está configurada de forma que, cuando llega al extremo final de la nave, se traslada verticalmente el coche hasta la siguiente línea. En esta línea, se arrastra el coche hasta el otro extremo de la nave, mientras se realizan las operaciones pertinentes, en dirección contraria a la anterior.

Se considerará para la simulación que los coches que pasan por las estaciones siempre se desplazan de izquierda a derecha, paralelamente al eje de las x. Por lo que la posición en x será un número positivo independientemente de la posición en el layout de la estación y en la dirección en la que se mueva el coche.

Se define el origen de coordenadas de la estación en el punto situado donde entra el coche, y en el centro de las vías por las que se desplaza el coche.

El eje negativo de las ordenadas (eje OY) se extiende desde el origen hacia arriba, y el eje positivo desde el origen hacia abajo (es decir, con signos opuestos a los habituales).

Esto es debido a que en nuestra estación piloto el coche avanzaba de derecha a izquierda y decidimos rotar la estación 180° respecto al eje x para visualizarla mejor. Además,

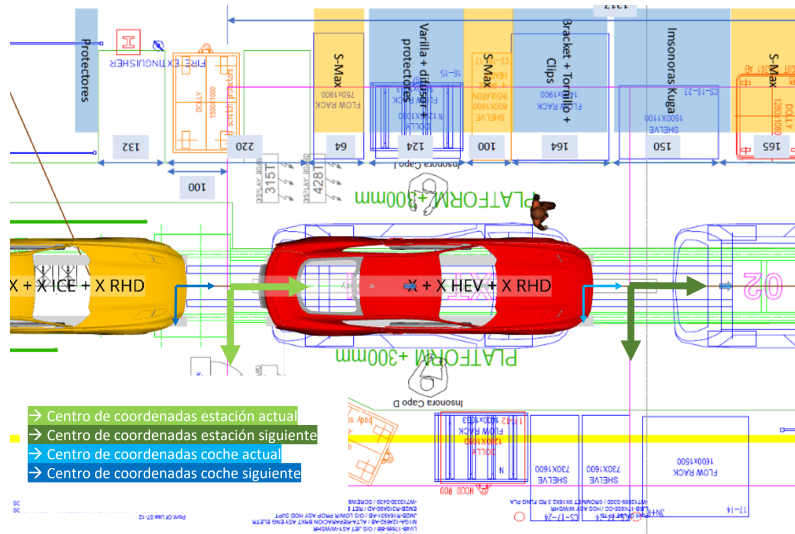


Figura 4.3: Coordenadas consideradas en la cadena de montaje

en el layout de toda la planta, nuestra estación se encuentra en la parte de abajo de la vía, por lo que sería el eje negativo de esta, con la orientación habitual.

Por esta razón, el puesto de trabajo negativo de la estación será el que monte el lado izquierdo del coche, mientras que el positivo será el que monte el lado derecho.

Generalmente, cada estación de trabajo tiene dos puestos de trabajo asignado a un único operario. Uno a la derecha de la línea y otro a la izquierda y separados por la vía por la que se desplaza el coche. Los operarios van rotando por los diferentes puestos para evitar que su trabajo sea excesivamente repetitivo. Estos cambios se realizan cada hora y media aprovechando las pausas de 15 minutos.

- **Coche:** El origen de coordenadas del coche se ubica en la posición del logotipo en el capó del vehículo y, por tanto, la dirección positiva en el eje OX se extiende desde el capó hacia el maletero. Debido a que la dirección positiva del eje OX es opuesta en el coche y en la estación, es importante tener en cuenta esta discrepancia en el código. El eje OY del coche es paralelo al eje OY de la estación y mantiene los mismos signos.
- **Ajuste del coche:** Con el objetivo de evitar posibles errores y complicaciones futuras, y mantener los ejes de coordenadas alienados, el centro de coordenadas se traslada al logotipo ubicado en el maletero, que es la parte del coche que ingresa primero a la línea. Para que el operario sepa a dónde dirigirse, se realiza una conversión de las ubicaciones relativas al coche considerando:

$$x = \text{largo}_{\text{car}} - x$$

$$y = -y$$

#### 4. Tabla “Calculos Todos”:

Se crea un Data Frame que contiene las columnas del input original:

- *step*: Número de operación.
- *tiempo*: Tiempo que se tarda en realizar la operación
- *x* e *y*: Posición (absoluta o relativa) dónde se realiza la operación.

- *relativo*: Si se realiza la operación en la estación o en el coche. Si relativo=0 esto indica que el operario va a las estanterías y se espera un tiempo mientras coge las piezas necesarias. Si relativo=1, entonces el operario se está desplazando junto al coche mientras monta una pieza y si relativo=2, el operario está yendo hacia el coche.
- *description*: Descripción de la operación.

Además, se añaden las siguientes columnas:

- $x_{op}$  e  $y_{op}$ : Representan la posición absoluta a la cual debe dirigirse el operario y en la cual se encontrará al finalizar la operación.
- $x_{car}$  e  $y_{car}$ : Indican la posición absoluta a la que debe dirigirse el coche y en la cual se encontrará al finalizar la operación.
- $dist_{desplaz}$ : Se refiere a la distancia recorrida por el operario que no aporta valor añadido, es decir, los pasos que anda mientras no realiza operaciones en el coche. Estos pasos no son optimizables y no se consideran en el cálculo.
- $t_{desplaz}$ : Corresponde al tiempo necesario para realizar el desplazamiento mencionado anteriormente.
- $t_{step}$ : Es el tiempo de proceso más el tiempo de desplazamiento.
- $t_{acum}$ : Representa el tiempo acumulado a lo largo de la columna  $t_{step}$  hasta el momento actual. Siempre se calculará como  $t_{acum} = t_{acum}[i - 1] + t_{step}$
- *vel*: Indica la velocidad a la cual debe desplazarse el operario. Esta velocidad varía según las condiciones, siendo 0 si está detenido (blocked), la velocidad del coche ( $v_c = 0,1$  m/s) cuando el valor relativo es 1 y la velocidad del operario ( $v_{op} = 1,4$  m/s) cuando se desplaza hacia el stock o hacia el layout (relativo 0 o 2).
- *VIN*: Se refiere al número de VIN del coche, que corresponde al recuento de los coches montados en la secuencia proporcionada.

Una vez definida la estructura de trabajo y sabiendo qué datos queremos obtener, se crea un bucle que se repetirá tantas veces como filas tenga el Data Frame y se empieza a resolver los cálculos con las siguientes funcionalidades.

1. La primera funcionalidad consiste en calcular los desplazamientos del operario en función de la posición a la que se debe dirigir. A continuación, se detallan las tres posibles condiciones que pueden darse durante la intercepción del coche:

- El operario se dirige a las estanterías para obtener el material necesario (relativo = 0):

En esta situación, el operario debe desplazarse hacia las estanterías para recoger el material requerido. Las acciones realizadas por el programa son las siguientes y en el siguiente orden:

$$x_{op} = x$$

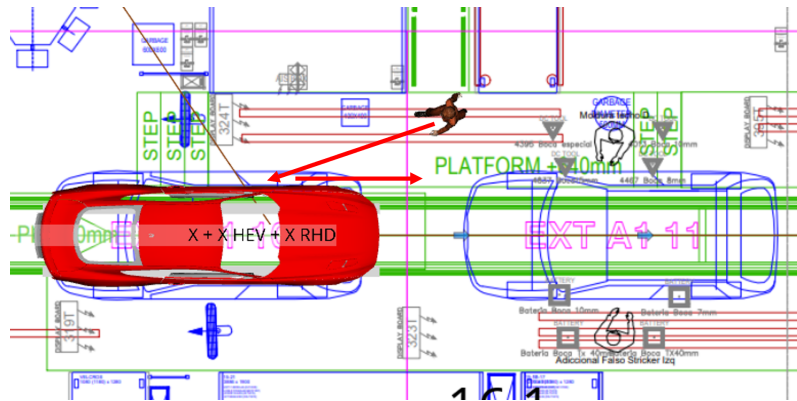
$$y_{op} = y$$

$$vel = v_{op}$$



- El operario va a montar una pieza (relativo = 1, donde  $pos_{ini} \neq pos_{fin}$ ):

Cuando el operario debe ir al coche a montar una pieza, ya sea desde una posición inicial relativo = 0 o relativo = 1, donde  $pos_{ini} \neq pos_{fin}$ , se requiere que el operario se desplace con el coche, por lo que este paso combinará dos velocidades (la de andar para alcanzar el coche y la del coche para realizar la operación).



**Figura 4.5: Desplazamientos cuando monta una pieza en el coche**

Para lograr diferenciar ambas velocidades, se crea una fila intermedia en el Data Frame con relativo = 2 que describe el desplazamiento desde la posición inicial hasta la posición donde debe comenzar a montar la pieza. Por otro lado, cuando se realiza un desplazamiento con relativo=0, una sola línea es suficiente, ya que se puede indicar que el operario “va al stock” y “espera x segundos”(esta última acción se indicará en la columna *tiempo*).

Para conseguir este objetivo, se modeliza matemáticamente el proceso de manera que es equivalente a un problema de geometría analítica que puede ser resuelto utilizando el teorema de los cosenos, mediante las siguientes acciones:

- Creación y cálculo de la fila de relativo=2: Esta fila adicional permite al programa calcular la fila del relativo=1 en la siguiente pasada del bucle, ingresando en una condición específica que se explicará más adelante.

Para calcular esta fila, se completan los siguientes parámetros:

$$relativo = 2$$

$$x[i] = x[i + 1]$$

$$y[i] = y[i + 1]$$

$$tiempo = 0 \quad (\text{No se realizan operaciones})$$

$i_{max} = i_{max} + 1$  Esto se hace para que el bucle se repita una vez más ya que, si no se lo indicamos, el programa no calculará todas las filas.

- Seguidamente, calculamos el resto de parámetros creando la función “interceptar”. Esta función calcula el tiempo de desplazamiento  $t_{desplaz}$  y la creamos siguiendo los pasos del problema físico-matemático de alcance entre dos objetos en movi-

miento (“chaser” y “runner”) que como veremos se puede modelizar utilizando geometría analítica y el teorema de los cosenos o formula de d’Al-Kashi.

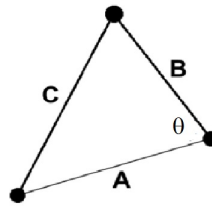
Para ello modelizamos vectorialmente los puntos considerando como “chaser” al operario (que intenta alcanzar al coche) y como “runner” al coche (que se desliza huyendo del operario), como sigue:

- Definiendo las posiciones del operario ( $P_{op}$ ) y del coche ( $P_{car}$ ).
- Definiendo las velocidades escalares del operario y el coche ( $v_{op}$  y  $v_{car}$ ).
- Definiendo el vector de velocidad del coche ( $V_{car}$ ).

$$\begin{aligned} V_{car_x} &= v_{car} \\ V_{car_y} &= 0 \end{aligned} \tag{4.1}$$

- Calculando de la distancia inicial ( $d$ ) y el vector de distancia ( $D$ ) entre el operario y el coche:

$$\begin{aligned} d &= \sqrt{D_x^2 + D_y^2} \\ D_x &= x_{operario.ini} - (x_{coche.ini} - x) \\ D_y &= y_{operario.ini} - (0 - y) \end{aligned} \tag{4.2}$$

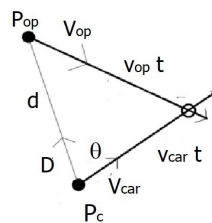


**Figura 4.6: Triángulo asociado a la ley de los cosenos**

- Aplicando la ley de los cosenos al triángulo de la figura 4.6 sabemos que:

$$C^2 = A^2 + B^2 - 2AB \cos(\theta) \tag{4.3}$$

En la figura 4.7 vemos que en este caso podemos considerar  $C = v_{op} t$ ,  $B = v_{car} t$  y  $A = d$ .



**Figura 4.7: Modelización del problema**

Utilizando la fórmula obtenemos:

$$(v_{op}^2 - v_{car}^2) t^2 + 2d v_{car} \cos(\theta) t - d^2 = 0 \tag{4.4}$$

Por la interpretación geométrica del producto escalar de vectores:

$$\cos(\theta) = \frac{D \cdot V_{car}}{d v_{car}} \quad (4.5)$$

luego

$$2 d v_{car} \cos(\theta) t = 2 d v_{car} \frac{D \cdot V_{car}}{d v_{car}} \quad (4.6)$$

y así obtenemos una ecuación cuadrática con parámetros

$$\begin{aligned} 0 &= a t^2 + b t + c \\ a &= v_{op}^2 - v_{car}^2 \\ b &= 2 D \cdot V_{car} \\ c &= -d^2 = -(D_x^2 + D_y^2) \end{aligned} \quad (4.7)$$

y por tanto programamos la siguiente ecuación:

$$t = \frac{-2(D_x V_{car_x} + D_y V_{car_y}) \pm \sqrt{4(D_x V_{car_x} + D_y V_{car_y})^2 + 4(v_{op}^2 - v_{car}^2)(D_x^2 + D_y^2)}}{2(v_{op}^2 - v_{car}^2)} \quad (4.8)$$

- o Una vez calculado el tiempo de desplazamiento ( $t_{desplaz}$ ), es fácil obtener el resto de variables en el siguiente orden:

$$\begin{aligned} x_{coche} &= x_{coche}[i - 1] + v_{car} * t_{desplaz} \\ vel &= v_{op} \\ x_{op} &= x_{car} - x \\ y_{op} &= y \\ dist_{desplaz} &= t_{desplaz} * vel \\ t_{step} &= t_{desplaz} \end{aligned}$$

- Cerramos la función dejando que pase a la fila de relativo = 1 que hemos dejado sin rellenar. Observamos que ahora el operario se encuentra en el punto del coche en el cual quiere montar la pieza, por lo que el programa entrará en la tercera condición del bucle.

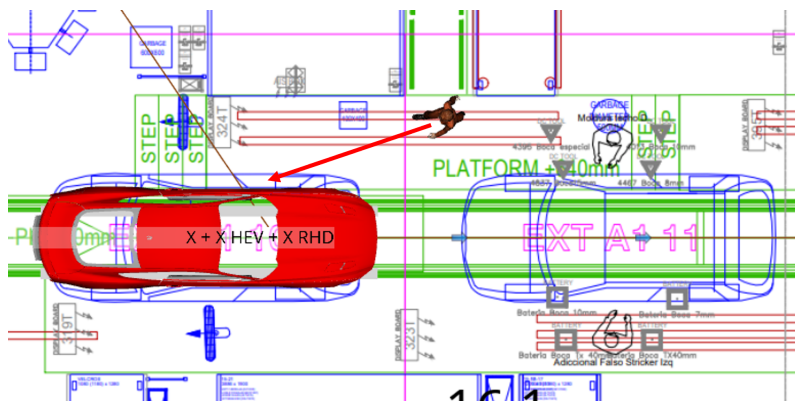


Figura 4.8: Desplazamiento con relativo = 2

- El operario va a montar una pieza ubicada en la misma posición en la que se encuentra actualmente, es decir,  $pos[i] = pos[i-1]$  (este caso solo se considerará para relativos  $\neq 0$  puesto que, para relativo=0 entrará en la primera condición):

En esta condición, el operario debe realizar operaciones mientras se desplaza. Por lo tanto, los campos se completan de la siguiente manera:

$$x_{car} = x_{car}[i - 1] + v_{car} * tiempo$$

$$y_{op} = y$$

$x_{op} = -x + x_{car}$  (Se utiliza un valor negativo para la coordenada x debido a la inversión de los ejes de coordenadas)

$t_{desplaz} = dist_{desplaz} * tiempo = 0$  (este desplazamiento no se contabiliza, ya que no se puede optimizar)

$$t_{step} = tiempo$$

$tiempo = 0$  Cuando exportemos la tabla indicaremos al simulador que lea los segundos que el operario debe estar parado de esta fila. Por lo que solo habrá tiempo cuando la velocidad del operario sea 0 (relativo=0 después del desplazamiento y Tiempo “blocked”

$$vel = v_{car}$$

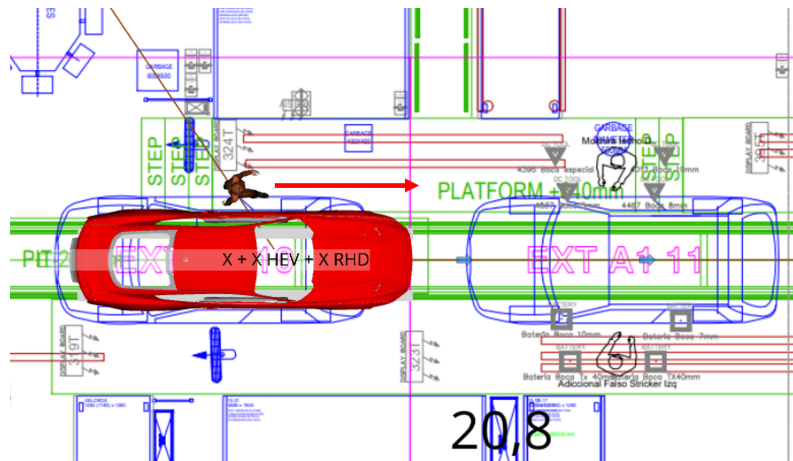


Figura 4.9: Desplazamiento con relativo = 1

Mediante estos cálculos, el programa garantiza que el operario se dirija al punto deseado de manera eficiente.

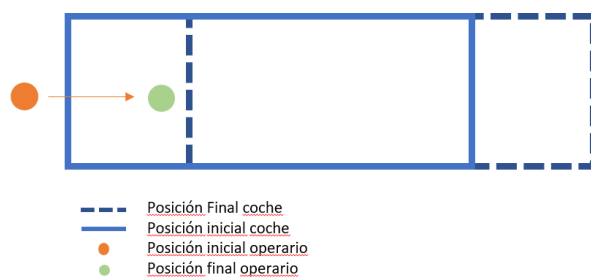
- Una vez calculada la posición objetivo del vehículo, es necesario verificar si existen posibles colisiones con las paredes del mismo. Esta verificación se realiza debido a que, en la mayoría de las estaciones de trabajo, no se permite ingresar dentro de los límites del vehículo, salvo en casos específicos, como estaciones con “Happy Seats” o en estaciones de la línea de Chasi, donde se trabaja desde debajo del vehículo.

El proceso para detectar colisiones se lleva a cabo mediante el cálculo de intersecciones entre segmentos. Este cálculo se repite seis veces, correspondiendo a cada una de las paredes del vehículo, además de dos intersecciones adicionales para evitar el maletero del vehículo anterior y el capó del vehículo siguiente.



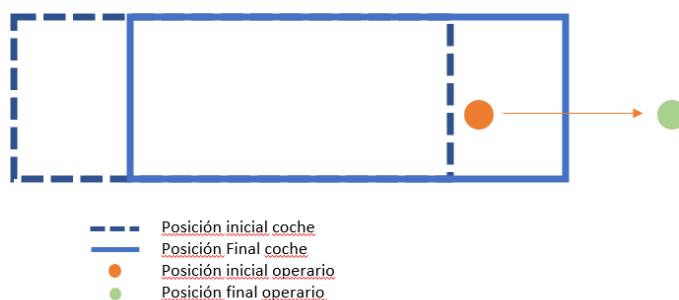
Para llevar a cabo estas comprobaciones de intersección, se crea una matriz de puntos con dos columnas (la primera para las coordenadas x y la segunda para las coordenadas y) y diez filas. La primera fila representa la posición inicial del operario, la segunda fila representa la posición final del operario, las filas del 3 al 6 corresponden a las cuatro esquinas del vehículo, y las filas del 7 al 10 representan los límites del capó y el maletero del vehículo anterior y siguiente, respectivamente.

Para evitar bucles infinitos en los cuales el operario entra dentro del vehículo y no puede salir debido a una intersección, se establecen límites reducidos para el vehículo. Decidimos esto ya que, utilizando la posición inicial del coche, cuando el operario va a montar algo a la parte trasera, entra dentro de los límites del coche y en el siguiente paso dará error por no poder salir.



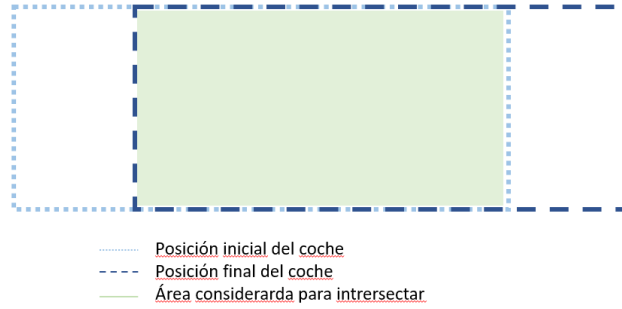
**Figura 4.10**

Y utilizando la posición final cuando está montando algo en el capó, entra dentro de los límites del coche y en el mismo paso no puede salir o incluso no podrá ir al punto objetivo.



**Figura 4.11**

Por tanto, la mitad delantera del vehículo, incluyendo el maletero del vehículo delantero, se representa mediante las posiciones iniciales del vehículo, mientras que la mitad trasera, incluyendo el capó del vehículo siguiente, se representa mediante las posiciones finales del vehículo. Aunque esto puede ocasionar que el operario cruce el vehículo cerca de las esquinas, el error introducido es mínimo y se considera insignificante en comparación con los coeficientes de seguridad aplicados en otros puntos del programa.



**Figura 4.12**

Adicionalmente, creamos una función llamada “check intersections” para comprobar si se produce alguna intersección con alguna de las esquinas. Para la resolución de este problema [calculamos si existe una intersección](#) entre dos segmentos construidos a partir de sus extremos. Esta función coge los diferentes puntos iniciales y finales de los segmentos creados por las paredes del coche para comprobar las seis posibles intersecciones. La función opera tal que así:

- Se calcula la diferencia entre la posición inicial y final del coche (del segmento elegido) y del operario.

$$\begin{aligned}
 s_{x_{op}} &= x_{op_{ini}} - x_{op_{fin}} \\
 s_{y_{op}} &= y_{op_{ini}} - y_{op_{fin}} \\
 s_{x_{car}} &= x_{car_{ini}} - x_{car_{fin}} \\
 s_{y_{car}} &= y_{car_{ini}} - y_{car_{fin}}
 \end{aligned} \tag{4.9}$$

- Las variables  $s_x$  darán como resultado 0 en caso de un movimiento vertical y las  $s_y$  serán 0 en caso de un desplazamiento horizontal.
- Utilizando las diferencias construimos los segmentos de forma paramétrica, considerando el parámetro  $\lambda_1$  para el segmento del operario:

$$(x, y) = (x_{op_{ini}} - \lambda_1 \cdot s_{x_{op}}, y_{op_{ini}} - \lambda_1 \cdot s_{y_{op}})$$

observamos que si  $\lambda_1 = 0$ , estamos en el extremo inicial del segmento  $(x_{op_{ini}}, y_{op_{ini}})$  y si  $\lambda_1 = 1$  en el extremo final  $(x_{op_{fin}}, y_{op_{fin}})$ . Y el parámetro  $\lambda_2$ , para el segmento del coche:

$$(x, y) = (x_{car_{ini}} - \lambda_2 \cdot s_{x_{car}}, y_{car_{ini}} - \lambda_2 \cdot s_{y_{car}})$$

si  $\lambda_2 = 0$ , estamos en el extremo inicial del segmento  $(x_{car_{ini}}, y_{car_{ini}})$  y si  $\lambda_2 = 1$  en el extremo final  $(x_{car_{fin}}, y_{car_{fin}})$ . De esta forma, igualando las variables x e y de

ambos segmentos, obtenemos los parámetros  $\lambda_1$  y  $\lambda_2$  que nos indicarán si existe o no intersección.

$$\lambda_1 = -\frac{-s_{y_{op}} * (x_{op_{ini}} - x_{car_{ini}}) + s_{x_{op}} * (y_{op_{ini}} - y_{car_{ini}})}{-s_{x_{car}} * s_{y_{op}} + s_{x_{op}} * s_{y_{car}}}$$

$$\lambda_2 = -\frac{s_{x_{car}} * (y_{op_{ini}} - y_{car_{ini}}) - s_{y_{car}} * (x_{op_{ini}} - x_{car_{ini}})}{-s_{x_{car}} * s_{y_{op}} + s_{x_{op}} * s_{y_{car}}}$$

Como el coche siempre se mueve horizontalmente  $s_{y_{car}}$  siempre será 0, los valores de estos parámetros se simplifican como sigue:

$$\lambda_1 = -\frac{-s_{y_{op}} * (x_{op_{ini}} - x_{car_{ini}}) + s_{x_{op}} * (y_{op_{ini}} - y_{car_{ini}})}{-s_{x_{car}} * s_{y_{op}}} \quad (4.10)$$

$$\lambda_2 = -\frac{s_{x_{car}} * (y_{op_{ini}} - y_{car_{ini}})}{-s_{x_{car}} * s_{y_{op}}} \quad (4.11)$$

Analizamos a continuación los casos en los cuales el resultado del cálculo es 0 o indefinido.

- Numerador = 0:
  - Cuando el segmento del coche y el operario empiezan en el mismo punto. → En este caso no nos importa que no entre en el condicional ya que, en caso de intersectar, devolverá TRUE con cualquiera de las otras barreras.
  - Cuando el operario se queda en el mismo sitio → no aplica para el problema
- Denominador = 0
  - El operario se queda en el mismo sitio → realmente no intersecta
  - El coche se queda en el mismo sitio (blocked) → Puede intersectar pero si lo tenemos en cuenta colapsaríamos el simulador. Puesto que es un target a reducir, es más problemático tener un blocked que el hecho de que el operario atraviese el coche. Por lo que, en caso de blocked la estación se tendrá que optimizar o reducir carga para evitarlo y entonces ya no se tendrá este problema.
  - Cuando el coche y el operario se desplazan ambos horizontalmente (el coche siempre) → realmente no intersectan.

Observamos que cuando el denominador de la fórmula de intersección es igual a 0, el programa devuelve NaN (not a number) y esto da problemas al simulador.

Como los casos analizados no representan situaciones problemáticas, excepto en el caso de un bloqueo del vehículo, situación que se considera despreciable. Se incluirá una condición que devolverá FALSE en el caso de que el resultado de  $\lambda_1$  o  $\lambda_2$  sea NaN.

Para evitar que el programa considere intersecciones cuando el operario está en una esquina del coche, es decir, cuando los **extremos** de los segmentos coinciden, se establece una condición de que  $0 < \lambda_1 < 1$  y  $0 < \lambda_2 < 1$ . Si queremos incluir también la coincidencia de los extremos como una intersección, la condición sería  $0 \leq \lambda_1 \leq 1$  y  $0 \leq \lambda_2 \leq 1$ .

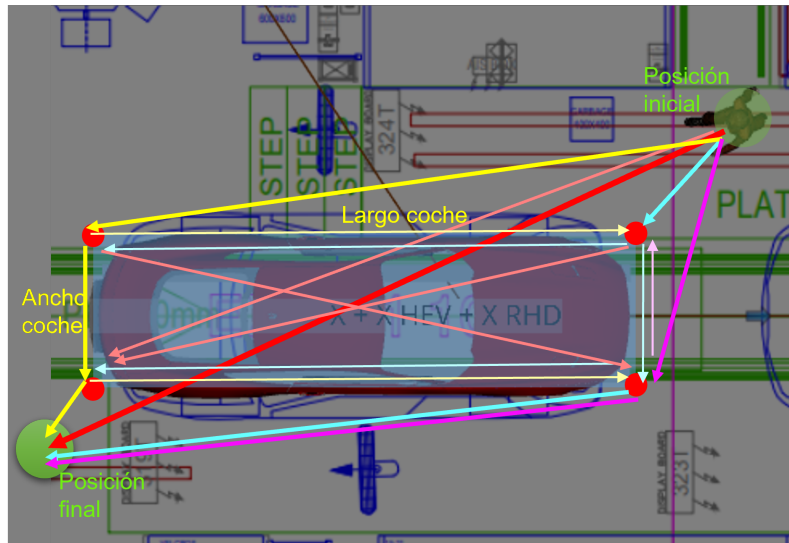
Esta función se repetirá 6 veces con los diferentes puntos considerados en el coche y los mismo puntos del operario; una por cada uno de los seis segmentos que contamos del coche.

- Una vez determinada la presencia de intersecciones, se construye un grafo cuyos vértices son las esquinas del vehículo y los límites de las paredes de los vehículos anterior y posterior. Las aristas del grafo representan los trayectos entre estos puntos, y las distancias entre ellos definen los pesos del grafo y, por ende, la matriz de adyacencia del mismo. Se utilizan funciones para calcular los caminos más cortos en este grafo, como “allShortestPaths” y “extractPath” de la librería de R “e1071”. Préviamente a estas dos funciones, se deben hacer los siguientes cálculos.

  - Antes de comenzar el bucle principal, se llenará la matriz con las distancias constantes (las concebidas entre los puntos del coche). Se dejarán vacías las distancias que no es posible resolver (NaN) y la diagonal principal.
  - Puesto que el grafo es no dirigido, su matriz de pesos será simétrica, luego solo calcularemos el triángulo superior y el inferior lo rellenaremos con los resultados del superior.
  - Modificaremos las filas 1 (distancia entre el punto inicial del operario y los puntos del coche) y 2 (distancia entre el punto final del operario y los puntos del coche).
  - Como ya hemos calculado que de 1 a 2 intersecta, solo rellenaremos desde la columna 3 a la 10. Por lo que solo tenemos que hacer un doble bucle (bucle anidado) dónde rellenemos 8 puntos en cada pasada.
  - Para rellenar estas 8 distancias, modificaremos la matriz de puntos (cambiaremos el objetivo del coche a la segunda fila) y le pasamos la matriz de puntos a la función “check intersections” para ver si puede ir a ese punto o no. En caso de que pueda ir, calcularemos la distancia entre los dos puntos y rellenamos el hueco de la matriz y su simétrico.
  - Se repite este paso con las 8 distancias al punto inicial y se repite el bucle poniendo en la primera fila de la matriz de puntos el objetivo que se quiere alcanzar.
- A continuación, se llama a la función “allShortestPaths”, la cual, toma como entrada la matriz de distancias y calcula todos los caminos más cortos entre todos los pares de puntos en la matriz, devolviendo un objeto de tipo grafo que contiene esta información. Por otro lado, la función “extractPath” se utiliza para extraer el camino más corto de un objeto de tipo grafo generado por “allShortestPaths”. Se proporcionan los índices de los puntos de origen y destino, y la función devuelve un vector que representa el camino más corto.
- Cuando ya tenemos el camino creado, hacemos otro bucle para que cree una fila por cada punto encontrado con relativo=2 y tiempo 0 en la que x e y son los diferentes puntos encontrados para el camino más corto (en relativo). Haciendo esto, le indicamos el camino con esquinas que no intersectan y evitamos tener que volver a rellenar la matriz de distancias. Sumamos al número de veces que se tiene que repetir el bucle, el número de filas que se hayan creado y se vuelve al inicio del bucle para rellenar los parámetros restantes.

Con los cálculos realizados, se logra que el operario se dirija al punto deseado sin atravesar el vehículo. Esto se consigue creando filas adicionales en las que se indica el camino a seguir con esquinas que no intersectan. Estas filas contienen la posición de los puntos encontrados en el camino más corto, con un relativo= 2 y tiempo 0. Al realizar este proceso, se evita tener que rellenar nuevamente la matriz de distancias.

En resumen, el programa desarrollado en este proyecto de ingeniería mecánica permite cal-



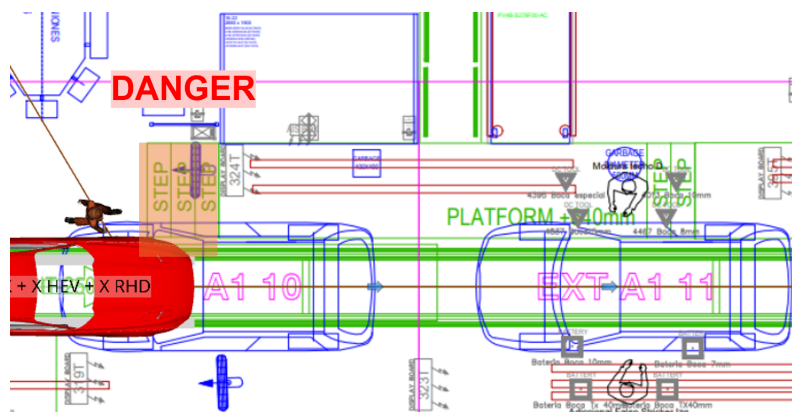
**Figura 4.13: Ejemplo de grafo**

cular y controlar el recorrido del operario en la manipulación de vehículos, evitando colisiones con las paredes del mismo. Mediante el cálculo de intersecciones, la construcción de un grafo y la determinación de los caminos más cortos, se logra una navegación eficiente y segura del operario en el entorno de trabajo.

### 3. Tiempos “Blocked” y “Starved”:

- **Tiempo? “Starved”:**

El tiempo “starved” se produce cuando el operario está esperando en el sitio a que el punto del coche donde se monta la siguiente pieza entre dentro de la estación. Hay estaciones, como el la mostrada en la figura 4.14 en la que, por seguridad o por imposibilidad física (paredes, limitaciones por el cable de la herramienta, etc.), el operario no se puede adelantar del límite de la estación.



**Figura 4.14: Ejemplo de starved**

Para simular esto, se implementa un condicional al inicio del bucle. Cuando los relativos son iguales a 1, se verifica si la posición absoluta a la que el operario debe dirigirse

es menor que un límite específico. Este límite puede ser establecido para evitar que el operario avance más allá del límite de la estación o para esperar a que el coche alcance una posición determinada. Es importante que sea únicamente para los relativos 1 ya que, aunque no esté el coche en la estación, el operario puede ir adelantando operaciones con relativo=0.

Cuando se cumple la condición, se crea una nueva fila en la que se asigna un valor relativo de 3 (para poder sumar fácilmente el tiempo total de “starved”). Si el relativo [i-1] es igual a 0, se establece que  $y_{op}[i]$  sea igual a  $y_{op}[i - 1]$  para que el operario no se mueva. Además, se indica al operario que se mueva verticalmente hasta alcanzar la posición  $y = ancho_{car}/2$  si el relativo anterior es igual a 1. Esto evita que el operario permanezca quieto y entre dentro del coche y luego no pueda salir debido a que intersectaría con las paredes.

Además, se calcula la nueva posición x del coche utilizando la fórmula:  $x_{car} = lmite_{estacin} + x[i + 1] - t_{desplaz}$  (para apartarse del camino del coche). El tiempo de desplazamiento se resta para indicar el tiempo que se necesita para apartarse del camino del coche.

El resto de parámetros se calculan así:

$$vel = v_{op}$$

$$dist_{desplaz} = y_{op}[i] - y_{op}[i - 1] \text{ En el caso del relativo 0 será 0}$$

$$t_{desplaz} = \frac{dist_{desplaz}}{vel}$$

$$tiempo = \frac{(x_{car}[i] - x_{car}[i-1])}{v_{car}} - t_{desplaz}$$

$$t_{step} = \frac{(x_{car}[i] - x_{car}[i-1])}{v_{car}}$$

■ **Tiempo? “Blocked”:**

Hay estaciones, como el la mostrada en la figura 4.15 en la que, por seguridad o por imposibilidad física (paredes, limitaciones por el cable de la herramienta,etc.), el operario no puede superar el límite de la estación.

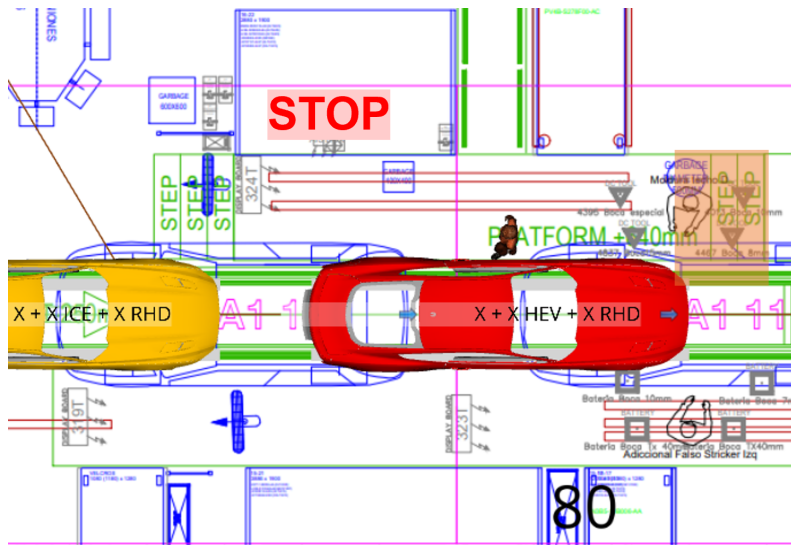


Figura 4.15: Ejemplo de blocked

Esta función es un poco más compleja. Se utiliza para evitar que una posición del

operario se salga de los límites de la estación. La condición para activar la función “blocked” es que la coordenada  $x$  del origen de coordenadas del coche sea menor que el límite de la estación ( $x_{car} < limite_{estacion}$ ). Esta función se debe incluir dentro de cada uno de los condicionales mencionados anteriormente, como la intercepción del coche. Se incluirá esta revisión después de calcular la  $x_{car}$ . Para evitar hacer cálculos y modificaciones innecesarias se calculará lo más pronto posible la  $x_{car}$  y luego el resto de parámetros en función a esta. Alternativamente, se guarda el tiempo que el coche está bloqueado en una variable para más adelante poder mostrarlo en los output. Este tiempo también servirá para hacer una tabla de 2 filas y 2 columnas con el tiempo y la velocidad a la que se mueve el coche y imputarlo en el visualizador.

	Tiempo	Velocidad
En movimiento	65s	$v_{car}$ m/s
Blocked	12s	0 m/s

**Tabla 4.1: Tabla ejemplo del conversor**

Después de calcular la posición  $x$  del coche, se verifica si supera el límite de la estación y, en caso afirmativo, se establece que  $x_{car} = limite_{estacion}$ . De esta manera, se evita que el coche se salga de los límites permitidos.

Los tiempos de “blocked” y “starved” se suman y se guardan en variables separadas, ya que son parámetros que se desean optimizar en el programa.

#### 4. Crear secuenciación de coches.

Para esta función, tenemos que hacer uso de la hoja “MIX”. Primero, crearemos un bucle que cuente cuantos coches llevamos. Este número (en la tabla será la columna VIN) servirá para conocer las opciones que lleva cada coche. Adicionalmente, filtramos estas opciones en el Data Frame “Cálculo” y se lo inputamos a la función Workstation. Esta función agrupa todo el bucle explicado anteriormente y encargado de realizar los cálculos fila por fila.

Al finalizar los cálculos, se filtrará la primera operación del siguiente coche para ver a qué punto de la estación tiene que ir. Se reinicia el bucle calculando las posiciones iniciales del coche y del operario que serán:

$$\begin{aligned} x_{op_{ini}} &= x_{op_{fin-VIN[i-1]}} \\ x_{car_{ini}} &= x_{car_{fin-VIN[i-1]}} - largo_{car} - espaciamiento \end{aligned} \quad (4.12)$$

Este bucle se repite tantas veces como el ingeniero quiera o hasta que no encuentra más coches en el el Mix.

#### 5. Girar el coche:

En algunas estaciones, el maletero del coche entra primero, lo que requiere girar el coche para ajustar las posiciones y mantener el orden adecuado. Para implementar esta funcionalidad, se utiliza una función simple que se activa cuando se indica:

Si se desea girar el coche 180 grados, se aplican las siguientes transformaciones:

$$x = largo_{car} - x$$

$$y = -y$$

Si se desea girar el coche 90 grados en sentido horario, se aplican las siguientes transformaciones:

$$largo_{car} = ancho_{car}$$

$$ancho_{car} = largo_{car}$$

$$x = y$$

$$y = ancho_{car} - x$$

Estas transformaciones permiten ajustar las posiciones y mantener la coherencia en el programa sin necesidad de realizar cambios adicionales en el código existente.

### 4.3. Outputs

En el proceso de simulación definido, se generan diversos tipos de output que proporcionan información relevante sobre el desempeño y los resultados obtenidos. A continuación, se describen los principales outputs generados:

Primero, se exportan varias tablas en formato CSV que representan los datos para cada VIN de la secuencia:

1. Tabla “Calculos Todos”: Contiene los siguientes parámetros:
  - Tiempo: Representa el tiempo que debe estar parado el operario realizando una acción. En el caso de las operaciones con relativo 1, se modifica este valor y se iguala a 0. En su lugar, se guarda el tiempo en la columna “ $t_{step}$ ”.
  - $x$  e  $y$ : Indican las posiciones del operario y del coche, que se obtienen de la tabla de inputs. Dependiendo de la dirección en la que se encuentre el coche, las posiciones con relativo 1 se transforman para que coincidan con la posición adecuada.
  - $x_{op}$  e  $y_{op}$ : Representan la posición final (absoluta) en la que se ubicará el operario al finalizar la operación actual.
  - $x_{car}$  e  $y_{car}$ : Indican la posición final en la que se encontrará el coche al finalizar la operación actual.
  - $dist_{desplaz}$ : Corresponde a las distancias de desplazamiento entre dos puntos. No se tienen en cuenta los desplazamientos realizados mientras se monta una pieza en el coche.
  - $t_{desplaz}$ : Es el tiempo dedicado a los desplazamientos mencionados anteriormente.
  - $t_{step}$ : Representa el tiempo total de desplazamiento más el tiempo de operación.
  - $t_{acum}$ : Indica el tiempo total que ha transcurrido para el montaje del coche.
  - $vel$ : Es la velocidad a la que se desplaza el operario en cada momento.
  - VIN: Número de coche de la secuencia inputada que se está montando.
2. Tabla “ $Calculos_{lista}$ ”: Se genera a partir de la tabla anterior y se utiliza en el software de visualización Simio. Contiene los siguientes parámetros:



- $x$  e  $y$ : Son las posiciones a las que se desplazará el operario en términos absolutos.
  - $vel$ : Indica la velocidad a la que se desplaza el operario.
  - $Tiempo_{Espera}$ : Representa el tiempo que el operario espera al llegar al objetivo antes de iniciar el siguiente paso.
  - Descripción: Proporciona una descripción de la operación.
3. Tabla “ $Calculos_{Conveyor}$ ”: Esta tabla también se crea a partir de la tabla inicial y se utiliza en el software Simio. Contiene los parámetros Tiempo, velocidad y VIN. Para cada VIN, se generan dos filas: una con la velocidad de la línea y el tiempo necesario para llegar al final de la línea, y otra en caso de que el coche esté bloqueado, indicando el tiempo en que el coche está parado con velocidad 0.

Estas tablas adicionales se emplean para realizar cálculos y visualizaciones más avanzadas, así como para alimentar la simulación en el software Simio. Proporcionan información detallada sobre posiciones, velocidades, tiempos y descripciones de operaciones, lo cual permite realizar análisis exhaustivos y evaluaciones del rendimiento del sistema.

A partir de estos outputs, se obtiene información valiosa que contribuye a mejorar la eficiencia y optimización de la estación de trabajo y las operaciones realizadas en ella, en línea con los objetivos del proyecto de ingeniería.

1. Tabla resumen con información relevante sobre los tiempos de cada VIN:

Esta tabla contiene información detallada sobre los tiempos relacionados con cada VIN (número de coche en la secuencia inputada). Los parámetros incluidos son los siguientes:

- Tiempo ciclo: Este valor representa el tiempo total dedicado al coche sin tener en cuenta los tiempos de espera. Es decir, es el tiempo que el operario está realizando operaciones o desplazándose mientras monta el coche.
- Número de pasos: Para calcular este valor, se suman todas las distancias de desplazamiento entre dos puntos en el layout y se dividen por la longitud de un paso. Según la metodología Modapts, se asume que un paso de W5 equivale a 0.65 metros.
- Tiempo operación: Es la suma de los tiempos de cada paso, el tiempo de desplazamiento y el tiempo de espera. Representa el tiempo total dedicado a realizar las operaciones en el coche.
- Tiempo pasos: Se calcula sumando los tiempos de desplazamiento entre dos puntos en el layout. Este tiempo no incluye los tiempos de espera ni el tiempo de operación en sí mismo.
- Tiempo espera: Se obtiene sumando los  $t_{step}$  correspondientes a los relativos 3 (starved). Estos tiempos indican el tiempo de espera del operario cuando no tiene una operación que realizar.
- Tiempo bloqueo: Cuando un coche está bloqueado, se guarda un registro del tiempo que el coche permanece detenido. Esta información se exporta y se muestra en la tabla. Esta tabla proporciona una visión detallada de los tiempos de cada VIN y permite analizar el rendimiento de la estación de trabajo en términos de tiempos de operación, desplazamiento y espera.

2. Tabla resumen con datos numéricos:

Además de la tabla detallada anterior, se genera una tabla resumen que muestra los valores medios de los parámetros mencionados anteriormente. Esta tabla tiene como objetivo proporcionar una visión general y comparativa del rendimiento de los diferentes VIN y opciones.

En la tabla resumen se incluyen las medias totales y las medias para cada opción. Aquí es importante tener en cuenta que el objetivo principal es reducir el tiempo de pasos. Sin embargo, dado que el tiempo de operación es constante, una reducción en el tiempo de pasos puede aumentar el tiempo de espera. Esto se debe a que el tiempo de ciclo también es constante, lo que significa que entra un coche a la estación cada 65 segundos. Por lo tanto, se puede asumir que la intención es maximizar el tiempo de espera, ya que esto permitiría agregar más operaciones a la estación en el futuro.

La tabla resumen proporciona una vista más compacta de los tiempos promedio y facilita la comparación y evaluación del rendimiento entre diferentes opciones y VIN.

3. Gráfico de ubicación óptima del stock:

Este gráfico tiene como objetivo ayudar a determinar la mejor ubicación para colocar el stock asociado a cada operación en el layout. El proceso para generar el gráfico es el siguiente:

- Se traza un segmento entre el punto donde termina una operación y el punto donde comienza la siguiente operación en el layout.
- Este segmento se divide en intervalos de 0.25 metros.
- Se cuenta el número de veces que cada posición se repite a lo largo del segmento.
- Utilizando estos recuentos, se crea un gráfico, posiblemente un histograma, donde la altura de las barras representa el número de repeticiones de cada posición.
- El punto más alto en el gráfico indica la posición óptima para ubicar el stock asociado a esa operación, ya que es el punto que más veces se repite a mitad de camino entre el inicio y el final del desplazamiento.

Este gráfico proporciona información visual sobre las posiciones óptimas para ubicar el stock, lo que facilita la planificación y organización eficiente de los materiales en el layout.

4. Tabla con el peso de cada operación:

Para ayudar en la toma de decisiones sobre qué material es más conveniente mover, se utiliza una variable llamada “peso”. El peso se calcula multiplicando el tiempo dedicado a una operación por la frecuencia con la que se realiza esa operación. Cuanto mayor sea el peso, mayor será la preferencia a la hora de optimizar la posición en el layout.

La tabla muestra los valores numéricos del peso para cada operación. Esta información permite identificar las operaciones más relevantes y prioritarias en términos de tiempo dedicado y frecuencia.

En resumen, las tablas y gráficos mencionados proporcionan información detallada y resumida sobre los tiempos de cada VIN, promedios de rendimiento, ubicaciones óptimas de stock y pesos de operaciones. Estas representaciones visuales y numéricas ayudan a analizar, evaluar y optimizar el rendimiento de la estación de trabajo y las operaciones realizadas en ella.

## Capítulo 5

# Validación

Para la validación del simulador, se optó por compararlo con un video en tiempo real de un operario, con el objetivo de verificar la similitud en los tiempos entre el simulador y la realidad. Sin embargo, debido a la falta de consentimiento y derechos de imagen de los trabajadores grabados, no es posible mostrar los videos. No obstante, se procederá a describir los resultados y conclusiones obtenidos.

Inicialmente, se realizó un video comparativo detallado, operación por operación, y se observó que existía una notable discrepancia en los tiempos. Para abordar esta disparidad en la primera iteración, se llevó a cabo un reordenamiento del Time Study y se logró reducir el desfase de 12 segundos a 4 segundos.

Una vez mitigado el desfase, se procedió a analizar en detalle los segmentos del video donde se generaban estas diferencias en los tiempos. Los resultados de dicho análisis se presentan en la tabla 5.1. En esta tabla resumen solo se indican las operaciones en las que existe desfase. El desfase positivo es que la simulación tarda más y el negativo que el operario tarda más.

	Desfase de tiempo	Desfase Acumulado	Tipo de operación
Montar varilla	-2s	-2s	Montar
Montar difusor	-1s	-3s	Montar
Coger protector	2s	-1s	Coger
Coger clips	2s	1s	Coger
Montar insonora	-4s	-3s	Montar
Total	t video: 47.11s	t simulación: 44.1s	Desfase total: -3s

**Tabla 5.1: Tabla comparativa con tiempos APT**

Como se puede observar, las diferencias entre el video y las operaciones vienen del tiempo que se ha impuesto a cada operación. Esto es comprensible ya que los modaps son genéricos y valoran la habilidad del operario. Pudiendo este tardar más por tener poca experiencia en el puesto o pudiendo tardar menos por ser una persona muy habilidosa. Para eliminar este error, se cronometrarán los tiempos de cada operación y se inputarán estos tiempos. Los resultados mostrados en la tabla 5.2:

	Desfase de tiempo	Desfase Acumulado	Tipo de operación
Ir a montar protector	-1s	-1s	Andar
Total	t video: 47.11s	t simulación: 45.69s	Desfase total: -1.42s

**Tabla 5.2: Tabla comparativa con tiempos operario.**

Con esta comparación obtenemos aproximadamente 1.5s de diferencia, lo que valida que el simulador es suficientemente preciso a la hora de calcular los desplazamientos.

El siguiente paso a validar será calcular las discrepancias en una estación de la línea de Chasis en la cual andan debajo del coche y por tanto se tiene que desactivar el esquivador. En esta comprobación, imputamos directamente los tiempos de operación obtenidos en el video. Obtenemos unos resultados de 72s que tarda el operario contra 71.5s que tarda el simulador. Un desfase de 0.5s que sigue validando que el simulador funciona bien.

Una vez validado que el simulador contabiliza bien los pasos, se procederá a simular toda la extensión de la línea Trim A1. Para esta comparación, se imputarán los datos tal cual como están en las bases de datos; el orden, tiempo, relativo... de las operaciones se cogerán del Time Study y las ubicaciones de las piezas en el Layout se cogerán del CLL. Con esta comprobación se pretende conocer cuanto error tenemos en los documentos actuales y si podemos evitar cambiar lo que existe. Se harán videos comparativos con la diferencia total y se analizarán únicamente las estaciones con más desfase. Los resultados se muestran en la tabla 5.3.

	Tiempo Operario	Tiempo simul.	Desfase	Desfase corregido	Correcciones realizadas
Insonora Capo D	66s	72s	6s	—	
Insonora Capo I	47s	44s	-3s	—	
Moldura Techo I	68s	82s	14s	1s	Cambiar orden + agrupar operaciones
Burletes tras D	66s	82s	13s	4s	Modificar punto inicial coche
Boca llenado I	54s	53s	-1s	—	
Ruteado Habit D	65s	71s	6s	—	
Ruteado Habit I	61s	49s	-12s	-6s	Operario espera a que entre el coche (6s)
Ins Dash Panel D	55s	54s	-1s	—	
Ins Dash Panel I	69s	70s	1s	—	
Moldura Techo D	70s	106s	36s	6s	Cambiar orden + agrupar operaciones
Falso Stricker I	66s	96s	30s	3s	Cambiar orden + agrupar operaciones
Burletes Tras I	66s	68s	2s	—	

**Tabla 5.3: Tabla resumen de la extensión de la línea Trim A1.**

---

Como podemos observar, en 7/12 estaciones el error es menor del 10 %. Un error asumible para el equipo de ingeniería ya que, se ha demostrado que, aunque no sea exacta la coincidencia en tiempo, si se mejora la distribución del layout y, por tanto, se mejora el tiempo, esta mejora en el tiempo se trasladará a la línea.

Por otra parte, en dos de las estaciones con grandes diferencias, para corregir este error no se ha tenido que cambiar nada más que la posición inicial del coche, este error es notable en el primer coche ya que depende mucho de las posiciones iniciales. Pero aumentando el número de coches que se realizan, este error irá disminuyendo.

Por último, vemos que únicamente cambiando el orden de operaciones al que hace realmente el operario en el video y por tanto agrupando algunos desplazamientos, se corrigen con una precisión suficiente las otras tres estaciones con un desfase demasiado grande.

Haciendo todas estas comparaciones podemos deducir que el simulador es suficientemente fiable siempre y cuando la secuencia de operaciones sea la que realmente está en la línea y que no será necesario hacer nuevos estudios de tiempo ni cambiar los Modapts.



## Capítulo 6

# Soluciones alternativas

A pesar de que revisar la organización y rebalanceo de planta de Almussafes de Ford supondrá una mejora significativa en el desempeño de esta planta de montaje, en el histórico de archivos de la empresa, no aparece contemplado ningún precedente al KP2.

Aunque la solución elegida para este problema ya se probó hace varios años, cuando se estableció contacto con la Universidad de Valencia (UV) para realizar el proyecto en la línea B1, donde se montaban los modelos Kuga y Transit-Connect (el funcionamiento y distribución de la planta de Ford Almussafes se explica en el primer anejo del TFG). En aquel momento, no se logró determinar un abordaje efectivo del problema. La evolución tecnológica y el hecho de trabajar únicamente con un modelo ha simplificado notablemente la complejidad del proyecto. Este hecho, junto con la creación de un equipo interno que conoce perfectamente el problema, así como los datos y especificaciones relevantes, ha permitido que finalmente se haya afrontado la solución a este problema con éxito.

Además de la alternativa presentada en el presente proyecto, existen diversas soluciones que también podrían valorarse como alternativas al enfoque actual.

A continuación, presentamos dos posibilidades interesantes, así como sus dificultades en su posible implementación:

- **Incorporación de cámaras de visión y tecnología de inteligencia artificial**

Alternativa que consistiría en la incorporación de cámaras de visión en cada estación de trabajo, que estén combinadas con sistemas de inteligencia artificial (IA). Estas cámaras analizarían en tiempo real las acciones realizadas por los operarios en cada estación, generando diferentes escenarios para mejorar la distribución del material en el layout. La IA sería capaz de identificar oportunidades de mejora que superen un umbral predefinido (por ejemplo, una mejora superior al x % en la eficiencia). Sin embargo, esta solución plantea desafíos, ya que requeriría obtener el consentimiento de los operarios para grabar sus actividades. Además, su implementación sería costosa, ya que se necesitaría hardware y software potentes, así como una gran capacidad de computación y servidores para almacenar las diversas casuísticas generadas.

- **Desarrollo de un semiasistente basado en una página web**

Otra posible solución consistiría en crear un semiasistente basado en una página web, que

permita visualizar y analizar los datos de las casuísticas en estudio. Esta opción se centraría en el uso del simulador existente, sin requerir un desarrollo adicional. Si bien este enfoque proporcionaría una visión preventiva de posibles problemas en las estaciones de trabajo, no permitiría realizar ajustes ágiles y automáticos en el layout. Además, no ofrecería la capacidad de analizar algunas de las casuísticas más contraintuitivas pero eficaces que podrían resultar en mejoras significativas.



**Parte II**

**Pliego De Condiciones**



# Capítulo 1

## Definición y alcance

El proyecto tiene como objetivo principal desarrollar un sistema automatizado para optimizar el proceso de rebalanceo en una planta de producción del sector del automóvil. El sistema constará de dos subaplicaciones interconectadas: una para el ordenamiento del layout y otra para la secuenciación de las operaciones. Además, se implementará un simulador que proporcionará datos relevantes para la toma de decisiones y permitirá validar y mejorar las soluciones propuestas.

### 1.1. Alcance del proyecto

El objetivo del proyecto es abordar el problema planteado y avanzar en la resolución del mismo, al mismo tiempo que se busca la colaboración con proveedores que puedan ofrecer soluciones adicionales. Se espera que el TFG contribuya al desarrollo de una solución eficiente y efectiva.

### 1.2. Alcance del TFG

El alcance del TFG se limita a la creación del simulador, en línea con los objetivos establecidos para el proyecto. También se analizarán algunas estaciones para poner en práctica el simulador y comprobar su funcionamiento.



## Capítulo 2

# Condiciones Técnicas

En este apartado, presentamos el análisis previo realizado, donde analizamos los puntos críticos del problema, así como las condiciones y las herramientas más adecuadas para abordarlos.

### 2.1. Problemas a abordar

A continuación damos un listado de las principales dificultades a las que nos enfrentamos:

1. **Rebalanceos basados en la experiencia sin datos:** Actualmente, los rebalanceos se realizan de forma intuitiva sin contar con pocos datos que no están relacionados entre ellos y no son precisos ni predictivos, sino correctivos, lo que genera ineficiencias en el proceso.
2. **Rebalanceos que requieren mucho tiempo y trabajo manual:** El proceso manual de rebalanceo implica una inversión considerable de tiempo y esfuerzo, lo que dificulta la eficiencia y agilidad en la asignación de recursos.
3. **Minimizar el factor humano:** Se busca eliminar la dependencia del factor humano en el proceso de rebalanceo para obtener resultados más precisos y consistentes, dejando al ingeniero la tarea de valorar si la solución obtenida es viable y/o amortizable.
4. **Automatización del proceso:** Se pretende desarrollar un sistema automatizado que realice el rebalanceo de manera eficiente, ágil, rápida y precisa.
5. **Validación previa de los cambios:** Es necesario contar con una herramienta que permita validar los cambios propuestos antes de su implementación en la línea de producción.

### 2.2. Solución adoptada y condiciones

A la vista del análisis realizado, considerando todos los factores, así como las soluciones alternativas (descritos en el capítulo 3 y 4 de la memoria) consideramos que la solución más acertada en este caso es desarrollar de un **optimizador de layout automático**.

- **Creación del simulador:** Se desarrollará un simulador que proporcionará datos relevantes para comparar diferentes escenarios y validar las soluciones propuestas. Este simulador será utilizable desde etapas tempranas del proyecto y permitirá adaptar las herramientas existentes en la planta de producción al mismo. También permitirá encontrar casuísticas no obvias y plantear soluciones a los problemas que vayan saliendo.

Se utilizará inicialmente el lenguaje de programación R para la implementación del simulador debido a su facilidad de aprendizaje y capacidad para el análisis de datos.

Posteriormente se transcribirá a Python el simulador. Una vez esté desarrollado y validado se realizará una transcripción del código a Python con la misma lógica fin de ensamblar las diferentes funcionalidades mediante un desarrollo integral de estas. Esto nos permite obtener un simulador más robusto y eficaz, alineado con el lenguaje de programación utilizado para el optimizador automático.

- **Desarrollo de un optimizador automático:** Se implementará un sistema compuesto por dos subaplicaciones, una para el ordenamiento del layout y otra para la secuenciación de las operaciones, que interactuarán entre sí para optimizar el proceso de rebalanceo.

Hasta donde he podido participar por la duración de las prácticas, para el optimizador de layout funcionará utilizando un algoritmo genético que generará 40 descendientes en cada iteración y realizará 30 iteraciones, para un total de 1200 iteraciones.

### 2.3. Condiciones exigibles:

Esta propuesta garantizará que:

- El sistema automatizado proporcione resultados precisos y consistentes en el proceso de rebalanceo.
- La maqueta del simulador sea utilizable y brinde datos relevantes para la toma de decisiones.
- El optimizador automático sea eficiente y rápido en la asignación de recursos.
- El sistema imite la realidad y considere las operaciones indicadas en las QPS (“Quality Process Sheets”) y el APT (“Allocation Process Tools”).
- Sean tenidas en cuenta la existencia de operaciones no documentadas en la hoja de procesos y las referencias no incluidas en el APT.
- Se estandaricen las operaciones y así evitar la subjetividad en el orden de las mismas.
- Se consideren las peculiaridades de las estaciones de trabajo, así como la presencia de “happy seats”.
- La ergonomía y capacidad en las manos de los operarios sean consideradas en el diseño de las tareas.

## 2.4. Software, permisos y licencias requeridas

La implementación de la propuesta será viable, puesto que la empresa proporcionará acceso a los recursos informáticos necesarios para el desarrollo del proyecto, incluyendo:

- Acceso al repositorio GitHub y a la plataforma Azure de FORD ESPAÑA S.L.
- Ordenadores para uso de los empleados asignados al proyecto.
- Licencia o prueba gratuita del software de simulación "SIMIO".
- Acceso a R y RStudio para el análisis de datos y creación de la aplicación.
- Licencia de AutoCAD para el diseño y representación del layout.
- Licencias de Microsoft Office para la elaboración de la tabla de inputs, documentación y presentaciones.
- Acceso a los servidores Anubis de FORD ESPAÑA S.L., que ofrecen una alta capacidad de procesamiento y servirán como repositorio para los servidores de la web.





## Capítulo 3

# Condiciones Facultativas

En este apartado, indicamos los controles de calidad utilizados para la correcta implementación del proyecto, así como el cronograma de su puesta en marcha.

### 3.1. Controles de Calidad y Ensayos

El criterio de evaluación principal para el simulador es su grado de realismo. Para verificarlo se compara la visualización del simulador con videos reales de los operarios trabajando en la línea de producción. Esto garantiza un alto nivel de fidelidad en la representación de las operaciones y tareas.

Otros de los criterios de evaluación que se utiliza para medir el rendimiento y la eficacia del sistema son: la precisión en el rebalanceo, la reducción del tiempo y el trabajo manual, y la optimización del uso de recursos.

### 3.2. Recepción Definitiva

El proyecto se desarrollará a lo largo del año 2023, dividido en trimestres. Cada trimestre se planteará un objetivo específico (MVP) para ser alcanzado. Se espera que el desarrollo de los optimizadores automáticos, la creación del simulador y la migración a Python se completen dentro de este marco temporal.

- MVP 1: Marzo 2023 → Entrega del simulador
- MVP 2: Junio 2023 → Entrega del optimizador de layout
- MVP 3: Octubre 2023 → Entrega del optimizador de la secuencia de operaciones
- MVP 4: Enero 2024 → Entrega del programa completo cohesionado.



## Capítulo 4

# Ejecución

A continuación, indicaremos la metodología y orden previsto de la propuesta.

### 4.1. Metodología y organigrama del proyecto

Para llevar a cabo esta propuesta se seguirá la metodología Agile Scrum. Esta metodología fue creada para resolver problemas complejos y con requisitos cambiantes y así obtener resultados periódicos en un proyecto. Los “sprints”, que es como se llama al periodo transcurrido entre dos reuniones, permiten una comunicación constante entre los miembros del equipo y entre los clientes, consiguiendo proyectos más sólidos, concretos y adaptados a las expectativas de los clientes. También permite dejar constancia del ritmo de avance del proyecto así como de los problemas que han ido surgiendo y de cómo se han resuelto para, en caso de ser de utilidad para futuros proyectos, poder tener esta información disponible. Se decide tener “sprints” semanales y reuniones con el equipo responsable, para revisar el progreso realizado, tomar decisiones sobre las siguientes etapas en la dirección del proyecto y asignar tareas recurrentemente.

En cada “sprint” se realizará una revisión del trabajo realizado y se tomarán decisiones sobre cómo avanzar en la resolución del problema, teniendo en cuenta la búsqueda de proveedores y la definición precisa de los objetivos.

Veamos a continuación, el orden en el cual se añadirán las diferentes características al simulador:

- Entender desde qué condiciones partimos y definir los datos de entrada que necesitaremos y entender de dónde tenemos que obtener dichos datos.
- Definir los resultados medibles que queremos obtener de la simulación.
- Generar las especificaciones del evaluador.
- Comenzar la simulación básica en la cual, se calcule dónde tiene que ir el operario y que el operario se mueva entre dos puntos.
- Trabajar con el sistema GIT para un mejor control de versiones.
- Conseguir que el operario esquivе el coche.

- Implementar márgenes al coche y a los soportes (“racks”) así cómo acercar los puntos al chasis del coche.
- Filtrar las diferentes opciones que incluye cada coche.
- Conseguir hacer la secuenciación de varios coches (siempre iguales).
- Conseguir que la secuencia de coches fabrique coches con opciones diferentes.
- Tener en cuenta que hay que evitar el coche anterior.
- Añadir Tiempo “starved” y Tiempo “blocked” al simulador.
- Crear gráficos que muestren la posición ideal del rack y tablas con los resultados que hemos obtenido. Calcular los datos que queremos pero no tenemos directamente.
- Validar el simulador comparándolo con la realidad.
- Crear página web con el simulador y los resultados de la simulación con una interfaz sencilla que permita trabajar a los ingenieros industriales.
- Crear una plantilla de “inputs” sencilla y lo más automatizada posible para que los ingenieros industriales puedan trabajar con el simulador.
- Generar diferentes escenarios con casuísticas nuevas.
- Crear una plantilla para anotar dónde se realizan las operaciones con relativo = 1. Para esto se usarán “coordenadas alfanuméricas de filas y columnas”.
- Crear una funcionalidad que active/desactive el esquivador.
- Crear una funcionalidad que permita cambiar la orientación del coche (girarlo cada 90°)

## **Parte III**

# **Presupuesto**



# Capítulo 1

## Presupuesto

En este apartado, se presenta el presupuesto para la realización del simulador del proyecto de ingeniería KP2: “Workstation Optimization & Line Rebalancing”. Esta previsto llevarlo a cabo desde el 8 de enero hasta el 12 de abril, y a partir de esta fecha, se negociará la participación en el resto del proyecto.

El equipo de trabajo dedicará 30 horas semanales por integrante al desarrollo del proyecto, utilizando sus conocimientos en ingeniería para llevar a cabo las tareas requeridas. La tarifa por hora establecida para el proyecto es de 30€, reflejando el valor y la experiencia aportados por el equipo.

Es importante destacar que durante el desarrollo del proyecto no se requirió la adquisición de licencias adicionales, ya que se utilizó el software de código abierto R y RStudio. Además, se empleó la licencia de prueba gratuita Simio Trial Edition para fines de simulación y para comprobar visualmente el correcto funcionamiento del simulador.

El presupuesto presentado contempla los costos asociados a los servicios de ingeniería, tanto en términos de análisis y diseño como de desarrollo e implementación. Asimismo, se destaca que el presupuesto no incluye costos relacionados con licencias o software, debido a la utilización de herramientas gratuitas y de prueba.

A continuación, se detallarán los costos estimados para cada una de las etapas del proyecto, brindando una visión clara de la inversión requerida. Es importante tener en cuenta que este presupuesto es una estimación y está sujeto a cambios en función de la duración y los requisitos reales del proyecto.

- **Duración del Proyecto:** 9 de enero - 12 de abril
- **Tarifa por hora:** 30 €
- **Número de integrantes:** 2. Un integrante se centrará en el desarrollo de la web y el otro en el desarrollo del simulador.
- **Horas de trabajo semanales por integrante:** 30 horas semanales
- **Horas totales de trabajo por integrante:** 30 horas/semana \* (13 semanas - 1 semana de descanso) = 360 horas por integrante

- **Costos asociados:** Los costos adicionales como licencias, ordenadores, servidores... serán ofrecidos por Ford al equipo.
- **Tarifa por hora por integrante:**  $360 \text{ horas} \cdot 30 \text{ €/hora} \cdot 2 \text{ integrantes} = 10,800 \text{ €}$
- **Licencias y accesos:**
  - R y RStudio (gratuitas y de código abierto): 0 €
  - Simio Trial Edition (versión de prueba gratuita): 0 €
  - Acceso al GitHub de ford \$21 por usuario por mes → \$42 al mes → \$126 en total. Al cambio son 117.48 € al cambio actual.
  - Licencias de microsoft office con acceso al azure information protection 20,60 €/mes · usuario → 41,20 € \* 3 meses → 123,6 €

Total presupuestado: 21,841.08 €

Es importante tener en cuenta que este presupuesto es una estimación basada en las horas de trabajo previstas y la tarifa por hora indicada. Los costos reales pueden variar según la duración real del proyecto, las horas de trabajo efectivas y cualquier cambio o requisito adicional que surja durante el desarrollo del proyecto.



## Capítulo 2

# Costes y ahorro para la empresa

### ■ Costes:

Cabe destacar que los salarios se han obtenido de las tablas salariales públicas, citadas en el convenio laboral que está ubicado en la página web de recursos humanos de FORD ESPAÑA. Se valorarán los salarios de los ingenieros como “Personal técnico de grado cuatro” (14,83 €/h) y el de los operarios como “Personal Obrero de grado uno” (12.96 €/h).

Se realizará una estimación de cuánto costaría rebalancear toda la planta de Montaje.

- Modificar Apt: 40 minutos por estación realizado por ingeniería industrial.  
 $14,83 \text{ €/h} \cdot 0,67 \text{ h/est} \cdot 305 \text{ est} = 3,030,51 \text{ €}$
- Recolectar ubicaciones relativo 1: 16 minutos por estación realizado por ingeniería industrial.  
 $14,83 \text{ €/h} \cdot 0,27 \text{ h/est} \cdot 305 \text{ est} = 1,221,25 \text{ €}$
- Modificar CLL: Despreciable  
0 €
- Coste de servidor Anubis. Por motivos de confidencialidad no se permite ofrecer información del coste y capacidad de los servidores, por lo que se obviará este dato.
- Coste de las reuniones:  
 $14,83 \text{ €/h} \cdot 1,5 \text{ h/reunion} \cdot 12 \text{ reuniones} \cdot 18 \text{ ingenieros} = 4,804,92 \text{ €}$
- Entre 5,000 € y 10,000 € de presupuesto. Cogemos 7,500 € de media por estación.  
 $7,500 \text{ €/est} \cdot 305 \text{ est} = 2,287,500 \text{ €}$
- Coste rebalanceo. 3.5h por estación realizado por ingeniería industrial.  
 $14,83 \text{ €/h} \cdot 3,5 \text{ h/est} \cdot 305 \text{ est} = 15,831,03 \text{ €}$
- Ordenador del almacén de Ford; amortizado → 0€
- Proyecto: 21,841.08 € presupuestados

### ■ Ahorro

- Se utilizará como referencia el ahorro obtenido en la estación piloto utilizada para obtener la validación del simulador; 14.67 s/coche  
 $12,96 \text{ €/h} \cdot \frac{1 \text{ h}}{3600 \text{ s}} \cdot 14,67 \text{ s/(coche} \cdot \text{est)} \cdot 740 \text{ coches/dia} \cdot 305 \text{ est} \cdot 216 \text{ dias/año} = 2,574,648,37 \text{ €/año}$

Por tanto el balance sería de:

$$2,574,648,37 - 3,030,51 - 1,221,25 - 4,804,92 - 2,287,500 - 15,831,03 - 21,841,08 = 240,419,58 \text{ €} \\ (2.1)$$

Por lo que la aplicación sería amortizable en aproximadamente un año.

Esto sin tener en cuenta en el presupuesto que actualmente, el proceso de rebalancear una estación implica el tiempo de un ingeniero industrial de aproximadamente una semana. Este tipo de rebalanceos manuales, actualmente consisten en ver estaciones que son suprimibles porque se piensa en una alternativa para automatizar ciertas operaciones y, cuando la carga de trabajo se reduce por debajo de cierto objetivo, se distribuyen por el resto de estaciones las otras operaciones. No se ha tenido en cuenta en el balance ya que es un cálculo muy difícil de realizar ya que cada estación es muy diferente a las otras y tiene demasiadas variables. Además ya se asume que el proyecto es amortizable en menos de un año.

Es importante destacar que el ratio Costes/Beneficio calculado no será el final y se intenta calcular de forma aproximada y con un único ejemplo el valor económico que aporta únicamente el simulador. Además del citado ahorro monetario, cabe destacar que la herramienta permitirá a los ingenieros trabajar con datos predictivos que no disponían. El proyecto seguirá aportando valor a medida que se avance con los optimizadores automáticos de layout y el de la secuencia de operaciones.

**Parte IV**

**Anexos**



# Capítulo 1

## Puesta en práctica

Para demostrar que la aplicación ya aporta valor, se prueba a rebalancear la estación piloto para ver dónde estamos.

La estación piloto es la llamada Insonora capó Izquierda. Y es la primera estación por la que pasan los Kuga del sistema A después de quitarle las puertas. En esta estación se monta:

- La varilla que sujeta y mantiene abierto el capó.
- El difusor de agua del limpiaparabrisas.
- Una insonora que interrumpe el paso del ruido y las vibraciones del motor al habitáculo.
- Un soporte dedicado a sujetar la batería de 12 voltios en los coches de combustión.

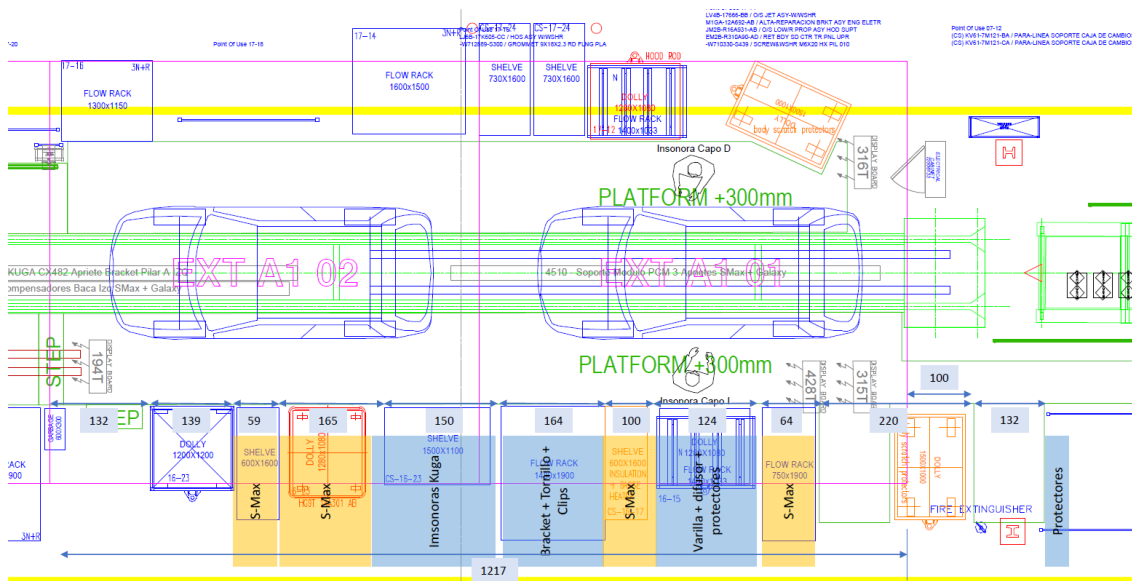


Figura 1.1: Layout de partida de la estación piloto.

1. Partimos de un escenario original previo al runout del S-Max y Galaxy y sin hacer modificaciones en el Time Study. Creamos un mix de 50 coches realista con las opciones que pueden tener influencia en el coche. Las opciones son:ç

- **X:** La operación se realiza en todos los coches Kuga.
- **X PHEV LHD:** La operación se realiza únicamente en los coches Kuga PHEV que tienen el volante en el lado izquierdo.
- **X NO HEV:** La operación se realiza en los coches Kuga de combustión interna puros (sin electrificación de ningún tipo)

La siguiente tabla es la tabla resumen de valores medios ya explicada en el apartado 4.3 de la Memoria; “Outputs”. Respecto al gráfico, también está explicado en dicho apartado. Cabe destacar que, la línea vertical roja indica la posición en la que se encuentra actualmente la pieza. El eje de las x indica el punto del layout y el de las y lo conveniente que es poner la estantería en ese punto.

Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	55.58	30.83	32.64	22.94	10.44	0
X + X PHEV	46.32	25.28	27.51	18.81	9.30	0
X + X NO HEV	74.18	43.44	41.85	32.33	13.88	0
X + X PHEV	46.72	25.81	27.51	19.21	9.86	0
X + X PHEV LHD	49.13	25.10	30.45	18.68	7.89	0

Figura 1.2: Tabla resumen primer escenario

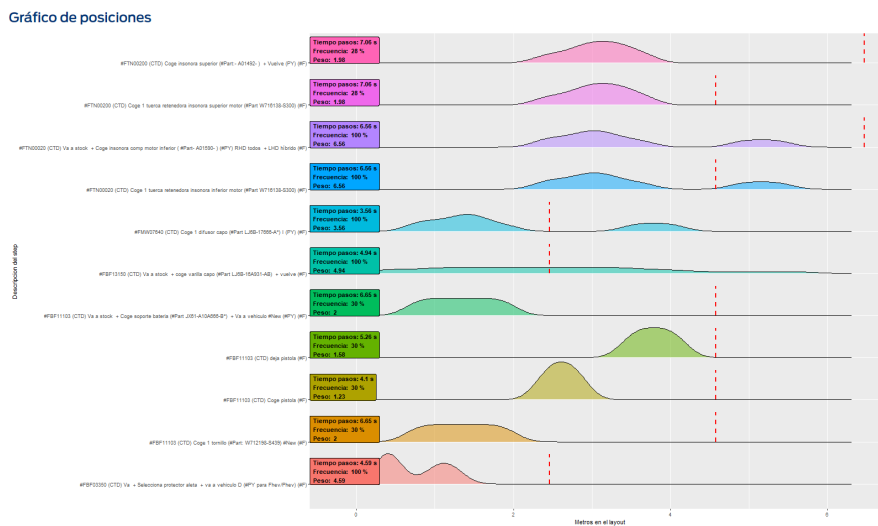


Figura 1.3: Gráfico de posiciones primer escenario

Observamos que el tiempo de pasos es muy alto. Esto no pasa en la línea y, al analizar la tabla de inputs vemos que tenemos que agrupar las operaciones que se pueden hacer juntas. Como coger la varilla capó y el difusor a la vez.

## 2. Agrupamos y ordenamos las operaciones.

### Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	46.11	22.77	29.17	16.94	19.51	0
X + X FHEV	41.98	21.18	26.22	15.76	18.85	0
X + X NO HEV	52.84	26.25	33.30	19.54	22.33	0
X + X PHEV	41.98	21.18	26.22	15.76	19.52	0
X + X PHEV LHD	45.10	21.41	29.16	15.94	16.80	0

Figura 1.4: Tabla resumen segundo escenario

### Gráfico de posiciones

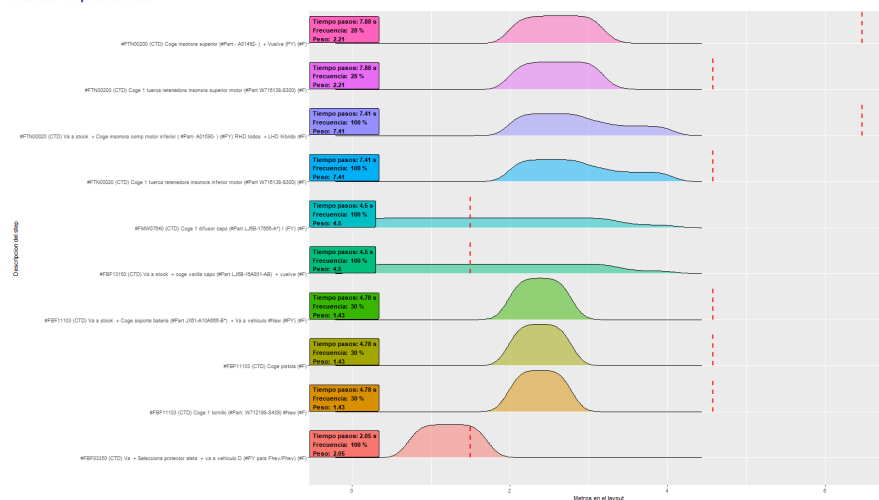


Figura 1.5: Gráfico de posiciones segundo escenario

Únicamente agrupando los desplazamientos, reducimos enormemente el tiempo de pasos. Esto es debido a que reducimos el número de viajes que hace a las estanterías.

También vemos que el tiempo de ciclo es menor a la frecuencia con la que pasan los coches. Esto es debido a que el S-Max y la Galaxy tienen mucha más carga de trabajo y esta se compensa con la menor carga que tienen los Kuga. Generalmente el S-Max y Galaxy tienen un tiempo de ciclo mayor a 65s y el Kuga tiene un tiempo de ciclo menor a 65s. La media de los tiempos de ciclo debe ser cercana a 65s para que no se produzcan paros en la línea y para aumentar al máximo las operaciones realizadas. También se juega con el tamaño de la estación, cuanto mayor sea este, más amplia puede ser la diferencia entre tiempos. Por ejemplo, en esta estación, que el límite está en el metro 9, el tiempo máximo será de aproximadamente 90s, teniendo este que ser compensado en los siguientes coches.

- Como podemos ver en la gráfica 1.7, los stock no están en la posición más adecuada. Por lo que los moveremos a la cima de las gráficas, que es dónde a priori deberían estar para optimizar los desplazamientos.

Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	40.50	15.22	29.17	11.33	24.99	0
X + X FHEV	36.83	14.25	26.22	10.61	24.67	0
X + X NO HEV	46.12	17.23	33.30	12.82	27.46	0
X + X PHEV	36.83	14.25	26.22	10.61	25.11	0
X + X PHEV LHD	39.97	14.52	29.16	10.81	22.37	0

Figura 1.6: Tabla resumen tercer escenario

Gráfico de posiciones

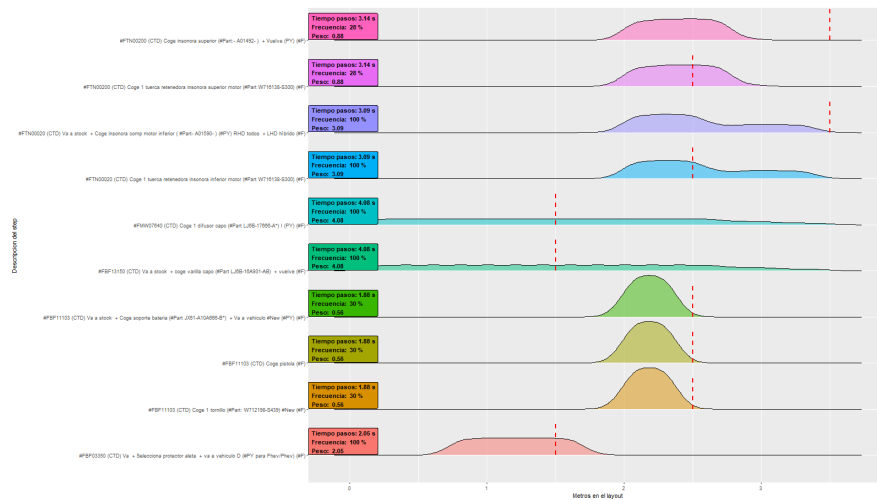


Figura 1.7: Gráfico de posiciones tercer escenario

Hemos reducido 6s del tiempo de ciclo. Podemos mejorar aún más.

- Acercamos más los Racks. Tenemos que tener en cuenta el ancho de los rack para que no se solapen. Debido a las características de la estantería, el clip de la insonora no puede estar junto a las insonoras. Posicionamos primero las insonoras, al tener el mayor peso, luego el protector y después los brackets.

Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	41.72	16.86	29.17	12.55	23.79	0
X + X FHEV	37.99	15.81	26.22	11.77	23.46	0
X + X NO HEV	47.39	18.93	33.30	14.09	26.22	0
X + X PHEV	37.99	15.81	26.22	11.77	23.91	0
X + X PHEV LHD	41.24	16.23	29.16	12.08	21.24	0

Figura 1.8: Tabla resumen cuarto escenario



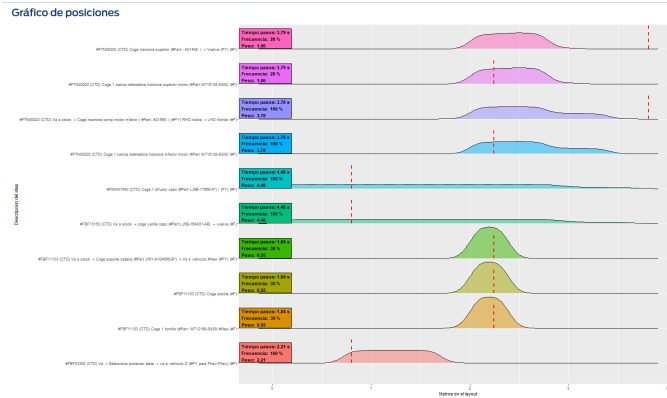


Figura 1.9: Gráfico de posiciones cuarto escenario

5. Probamos a cambiar el orden de las operaciones para ver si es una opción mejor. Ponemos los rack pegados entre sí cambiando el orden y poniendo primero el rack de insonoras.

Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	44.20	18.60	30.36	13.85	21.44	0
X + X FHEV	38.70	16.77	26.22	12.48	20.20	0
X + X NO HEV	54.03	22.54	37.26	16.77	25.56	0
X + X PHEV	38.70	16.77	26.22	12.48	21.49	0
X + X PHEV LHD	41.92	17.14	29.16	12.76	17.61	0

Figura 1.10: Tabla resumen quinto escenario

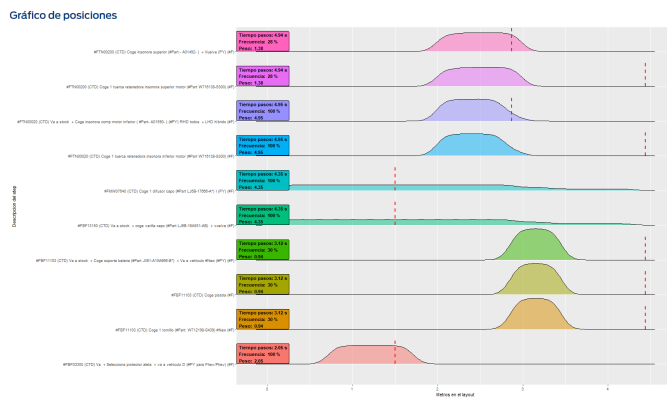


Figura 1.11: Gráfico de posiciones quinto escenario

6. Se puede mover todo medio metro hacia la izquierda.

Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	43.83	18.11	30.36	13.48	21.80	0
X + X FHEV	38.50	16.50	26.22	12.28	20.68	0
X + X NO HEV	53.14	21.34	37.26	15.88	25.67	0
X + X PHEV	38.50	16.50	26.22	12.28	21.85	0
X + X PHEV LHD	41.86	17.06	29.16	12.70	18.14	0

Figura 1.12: Tabla resumen sexto escenario

Gráfico de posiciones

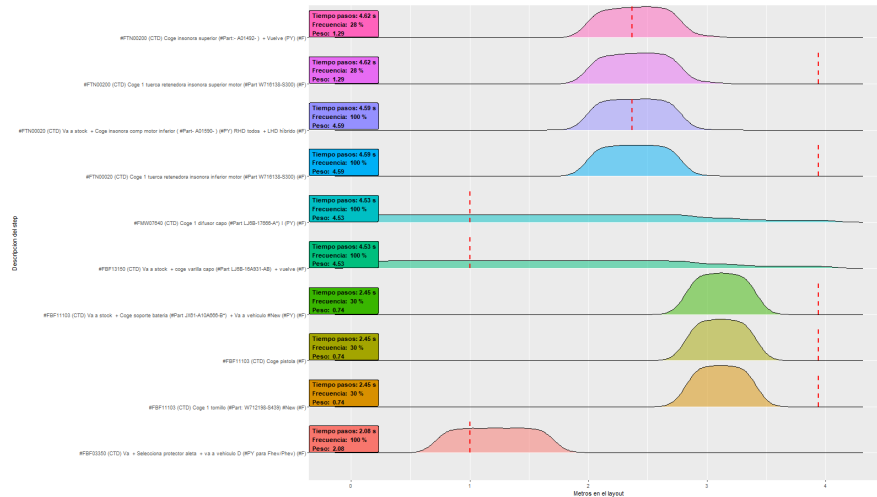


Figura 1.13: Gráfico de posiciones sexto escenario

Obtenemos un resultado mucho peor que el del escenario 3, por lo que la mejor distribución para la estación Insonora Capó Izquierda será la vista en el escenario 3.

- Hemos conseguido reducir el tiempo de ciclo en 15s. Con lo que podemos añadir más operaciones. Se ha valorado que la mejor operación es poner el bracket del pilar A. Se pone después del soporte de batería aprovechando que tiene la pistola en la mano. Se prueba a poner el Bracket del pilar A antes y después del soporte batería.

■ Antes

Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	60.24	22.46	43.53	16.72	5.74	0
X + X FHEV	54.44	19.87	39.66	14.78	5.48	0
X + X NO HEV	70.67	28.04	49.80	20.87	7.14	0
X + X PHEV	54.36	19.76	39.66	14.70	5.08	0
X + X PHEV LHD	57.85	20.49	42.60	15.25	5.02	0

Figura 1.14: Tabla resumen septimo escenario operación antes del soporte.

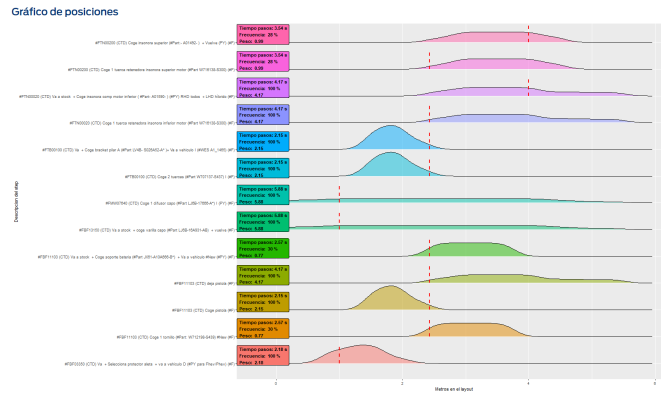


Figura 1.15: Gráfico de posiciones septimo escenario operación antes del soporte.

- Después.

Key Numbers

Variante	⌄ Tiempo ciclo	⌄ Nº pasos	⌄ Tiempo operación	⌄ Tiempo pasos	⌄ Tiempo espera	⌄ Tiempo bloqueo
TOTAL	59.47	21.43	43.53	15.95	6.46	0
X + X FHEV	54.25	19.61	39.66	14.59	5.85	0
X + X NO HEV	68.53	25.17	49.80	18.73	8.52	0
X + X PHEV	54.23	19.59	39.66	14.57	6.01	0
X + X PHEV LHD	57.61	20.17	42.60	15.01	5.02	0

Figura 1.16: Tabla resumen septimo escenario operación después del soporte.

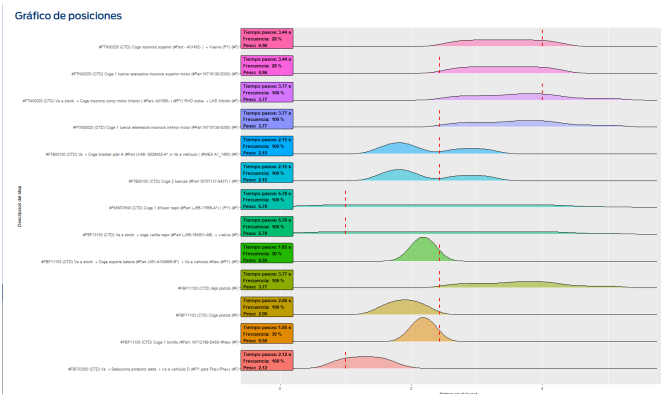


Figura 1.17: Gráfico de posiciones septimo escenario operación después del soporte.

Es ligeramente más conveniente poner las operaciones después.

- Podemos añadir otra operación. Por lo que añadimos montar dos tapas. La tapa del embrague en los coches automáticos LHD y un tapón cerca de dónde irá el panel de control.

Bloqueamos demasiado. Por lo que, en caso de no poder mover el layout sin bloquear, quitaremos la tapa del embrague.

Key Numbers

Variante	⌄	Tiempo ciclo	⌄	Nº pasos	⌄	Tiempo operación	⌄	Tiempo pasos	⌄	Tiempo espera	⌄	Tiempo bloqueo
TOTAL		84.01		49.32		47.31		36.70		0		17.45
X + X FHEV		69.86		37.44		42.00		27.86		0		7.17
X + X FHEV + X LHD AUTO		82.60		51.41		44.34		38.26		0		16.68
X + X LHD AUTO + X NO HEV		95.79		55.50		54.48		41.31		0		26.32
X + X LHD AUTO + X PHEV LHD		80.59		44.76		47.28		33.31		0		14.71
X + X NO HEV		95.89		58.79		52.14		43.75		0		27.37
X + X PHEV		76.61		46.50		42.00		34.61		0		11.27

Variante  Tiempo ciclo  Nº pasos  Tiempo operación  Tiempo pasos  Tiempo espera  Tiempo bloqueo

Figura 1.18: Tabla resumen octavo escenario.

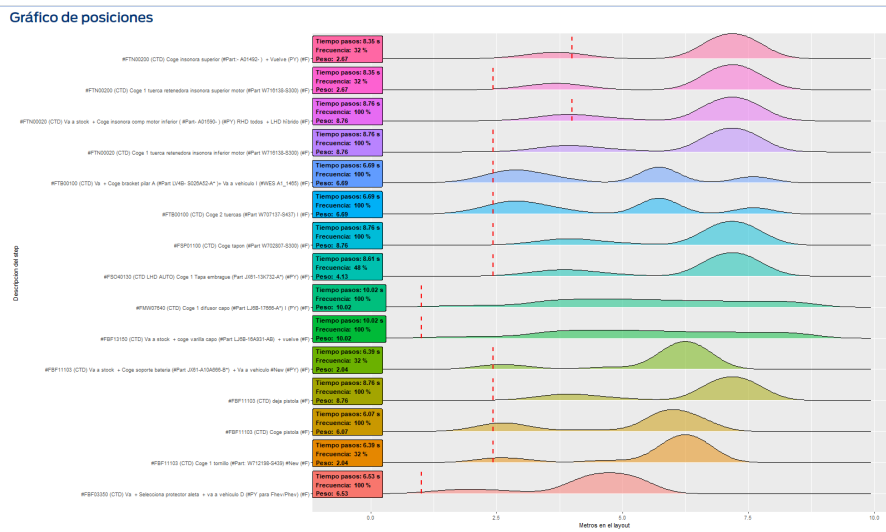


Figura 1.19: Gráfico de posiciones octavo escenario.

9. No es ideal pero no bloqueamos, por lo que haremos una última iteración.

Key Numbers

Variante	⌄	Tiempo ciclo	⌄	Nº pasos	⌄	Tiempo operación	⌄	Tiempo pasos	⌄	Tiempo espera	⌄	Tiempo bloqueo
TOTAL		65.84		24.90		47.31		18.53		0.24		0
X + X FHEV		60.39		24.72		42.00		18.40		0.00		0
X + X FHEV + X LHD AUTO		61.95		23.66		44.34		17.61		0.00		0
X + X LHD AUTO + X NO HEV		74.22		26.52		54.48		19.74		0.00		0
X + X LHD AUTO + X PHEV LHD		65.65		24.68		47.28		18.37		0.58		0
X + X NO HEV		71.90		26.56		52.14		19.76		0.28		0
X + X PHEV		59.45		23.45		42.00		17.45		0.00		0

Variante  Tiempo ciclo  Nº pasos  Tiempo operación  Tiempo pasos  Tiempo espera  Tiempo bloqueo

Figura 1.20: Tabla resumen noveno escenario.

Gráfico de posiciones

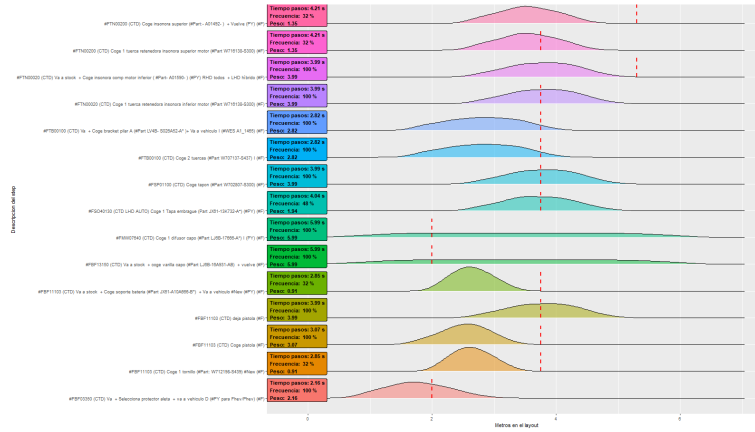


Figura 1.21: Gráfico de posiciones noveno escenario.

10. Mejoramos ligeramente.

Key Numbers

Variante	Tiempo ciclo	Nº pasos	Tiempo operación	Tiempo pasos	Tiempo espera	Tiempo bloqueo
TOTAL	64.88	23.60	47.31	17.57	1.16	0
X + X FHEV	58.83	22.62	42.00	16.83	0.59	0
X + X FHEV + X LHD AUTO	60.75	22.05	44.34	16.41	0.00	0
X + X LHD AUTO + X NO HEV	73.66	25.77	54.48	19.18	0.39	0
X + X LHD AUTO + X PHEV LHD	64.30	22.87	47.28	17.02	1.89	0
X + X NO HEV	71.17	25.58	52.14	19.04	1.31	0
X + X PHEV	58.72	22.46	42.00	16.72	0.79	0

Figura 1.22: Tabla resumen décimo escenario.

Gráfico de posiciones

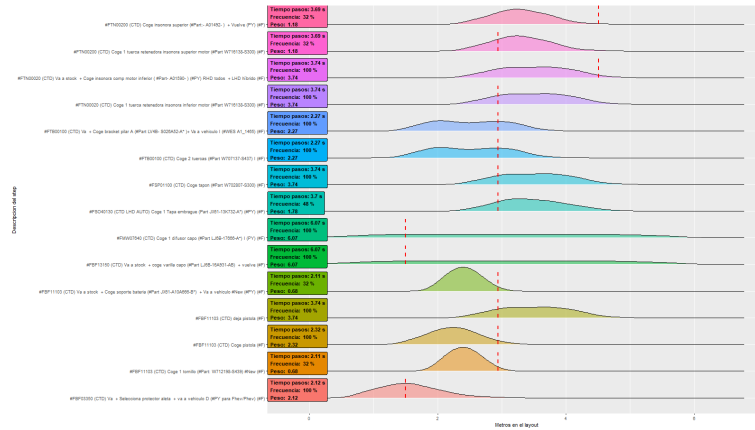


Figura 1.23: Gráfico de posiciones décimo escenario.

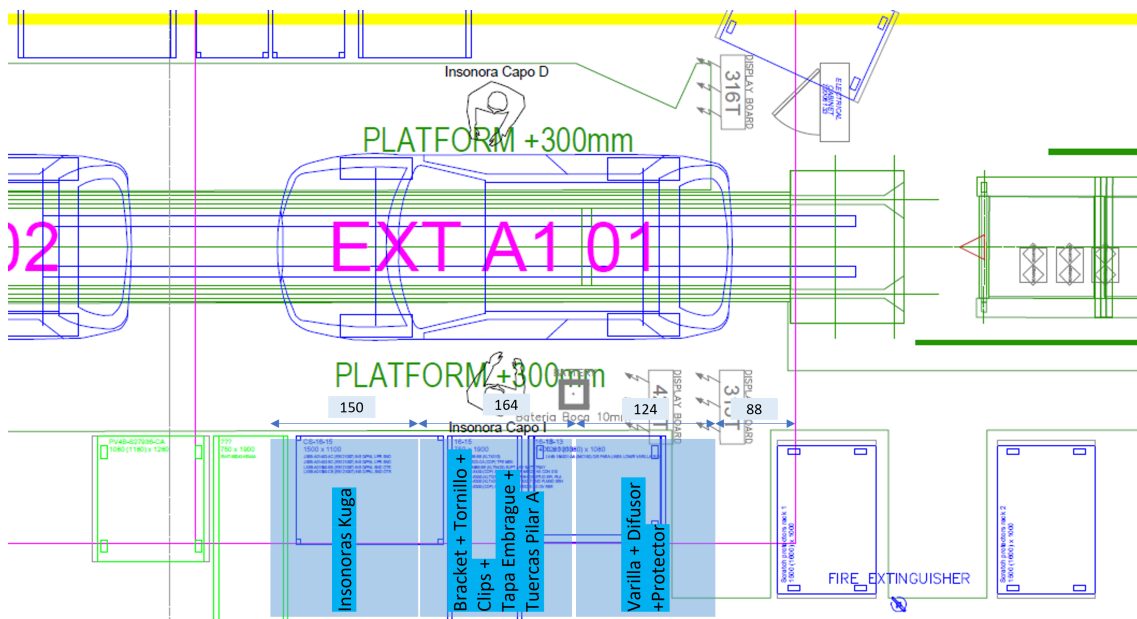
Como observación, vemos que hay coches (los de combustión) que tardan un poco más de 65s. Podemos permitirnos este sobretrabajo por dos motivos.

Primero, las intenciones de Ford, en línea con el ODS de sostenibilidad, pretende reducir al máximo la venta de estos vehículos y aumentar la venta de vehículos electrificados.

Por otra parte, este sobretrabajo se compensa con una carga de trabajo menor de 65s en los coches electrificados, cuya previsión es aumentar su producción. Especialmente en los PHEV de Reino Unido (LHD)

En conclusión, aunque en algunos modelos sobrepasamos los 65s, como la estación es un poco más larga de lo normal, podemos permitirnos poner más trabajo en algunas opciones que en otras.

Finalmente la estación debería quedar como en la figura 1.24:



**Figura 1.24: Layout optimizado de la estación piloto.**

Conseguimos aumentar el tiempo de valor añadido al producto desde 32.64s hasta 47.31s, obteniendo una mejora de 14.67s

# Bibliografía

- [1] Womack, J.P., Jones, D.T., & Roos, D. (2003). The Machine That Changed the World: The Story of Lean Production. Free Press.
- [2] Liker, J.K. (2004). The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer. McGraw-Hill Education.
- [3] Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., ... & Highsmith, J. (2001). Manifiesto for Agile Software Development. Agile Alliance.
- [4] Cohn, M. (2005). Agile Estimating and Planning. Pearson Education.
- [5] Hollmann, J., & Hollmann, K. (2008). MODAPTS: Motion-Time Analysis. Publicaciones APMT.
- [6] Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer.
- [7] Müller, K., & Wickham, H. (2019). tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1.
- [8] Van Rossum, G., & Drake, F.L. (2009). Python 3 Reference Manual. CreateSpace.
- [9] McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.
- [10] Bazaraa, M.S., Jarvis, J.J., & Sherali, H.D. (2013). Linear Programming and Network Flows. Wiley.
- [11] Chong, E.K.P., & Zak, S.H. (2013). An Introduction to Optimization. Wiley.
- [12] Zill, D.G., & Wright, W.S. (2009). Bloques 1: Cálculo. McGraw-Hill.
- [13] Lopez Orti, J.A., & Barreda Rochera, M. (2011). Fundamentos matemáticos de la ingeniería parte 1: Álgebra lineal. Universitat Politècnica de València.

Espero que esta lista sea de ayuda para tu proyecto.

