

# Índice

<b>1. INTRODUCCIÓN</b>	<b>3</b>
1.1. OBJETIVOS . . . . .	5
<b>2. MARCO TEÓRICO</b>	<b>6</b>
2.1. SIMULACIÓN DE EVENTOS DISCRETOS . . . . .	6
2.1.1. Terminología y funcionamiento de una DES . . . . .	7
2.1.2. Exposición de un caso ficticio usando una DES . . . . .	9
2.1.3. Etapas de desarrollo de una DES . . . . .	15
2.2. TEORÍA DE LA PROBABILIDAD Y DES . . . . .	19
2.2.1. Variables aleatorias o random . . . . .	19
2.2.2. Distribuciones de probabilidad . . . . .	20
2.2.3. Tests de hipótesis . . . . .	21
2.3. EL LABORATORIO DE EVALUACIÓN DEL PRODUCTO . . . . .	22
2.3.1. Ensayos Endurance . . . . .	23
2.3.2. Ensayos Rolling Resistance . . . . .	23
2.3.3. Análisis dimensional . . . . .	26
<b>3. DESARROLLO</b>	<b>26</b>
3.1. FORMULACIÓN DEL PROBLEMA, ESTABLECIMIENTO DE OBJETIVOS Y PLAN GENERAL . . . . .	26
3.2. CONCEPTUALIZACIÓN DEL MODELO . . . . .	27

3.3. RECOPIACIÓN DE DATOS . . . . .	29
3.4. TRADUCCIÓN DEL MODELO . . . . .	29
3.5. VERIFICACIÓN Y VALIDACIÓN . . . . .	30
<b>4. RESULTADOS</b>	<b>30</b>
<b>5. CONCLUSIONES</b>	<b>38</b>
<b>A. ANEXO I: CÓDIGO FUENTE</b>	<b>40</b>
A.1. EXPOSICIÓN DE UN CASO FICTICIO DE UNA DES . . . . .	40
A.2. CÓDIGO FUENTE DE LA SIMULACIÓN PRINCIPAL . . . . .	46
<b>B. ANEXO II: DIAGRAMAS DE FLUJO</b>	<b>58</b>

# 1. INTRODUCCIÓN

El sector de la automoción, engloba una gran variedad de industria y servicios dedicados a servirla. Se estima que la aportación económica total de las actividades relacionadas, en este sector, asciende a un 11 % del PIB, lo que la convierte en la industria manufacturera que mas ingresos aporta después del 18,8 % del PIB que posee la industria agroalimentaria española (Díaz y Montorriol Garriga, 2021). Esta cifra puede ser contrastada frente a la carta emitida por el presidente de la ANFAC, adherida al informe anual, donde se aprecian estimaciones similares.

En este sector, el vehículo de propulsión autónoma, es generalmente utilizado tanto por los servicios de transporte a pasajero, como por los servicios logísticos dedicados al transporte de mercancías. Entre los componentes que forman el vehículo autónomo, se encuentran las cubiertas o neumáticos, los cuales se comportan como enlace entre el vehículo y el pavimento. Este nexo permite una transmisión eficiente de la energía producida por el motor de combustión interna, y a su vez, sus propiedades elásticas atenúan las irregularidades de la vía.

La industria manufacturera de cubiertas, ha mantenido un incremento sostenido en su producción en los últimos años, como puede observarse en la Figura 1. El mercado de neumáticos es liderado por dos empresas: Bridgestone y Michelin (Rodgers, 2020). Las demás empresas compiten entre ellas a un magnitud inferior a los lideres del sector, como se aprecia en la Figura 2.

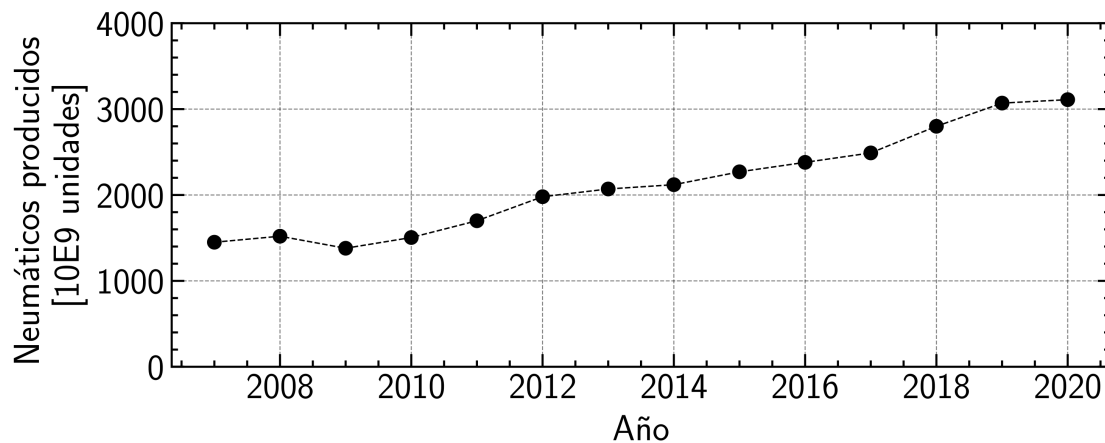


Figura 1: Evolución de la producción mundial de cubiertas.

La elevada competitividad característica del mercado, sumada a la emergente co-

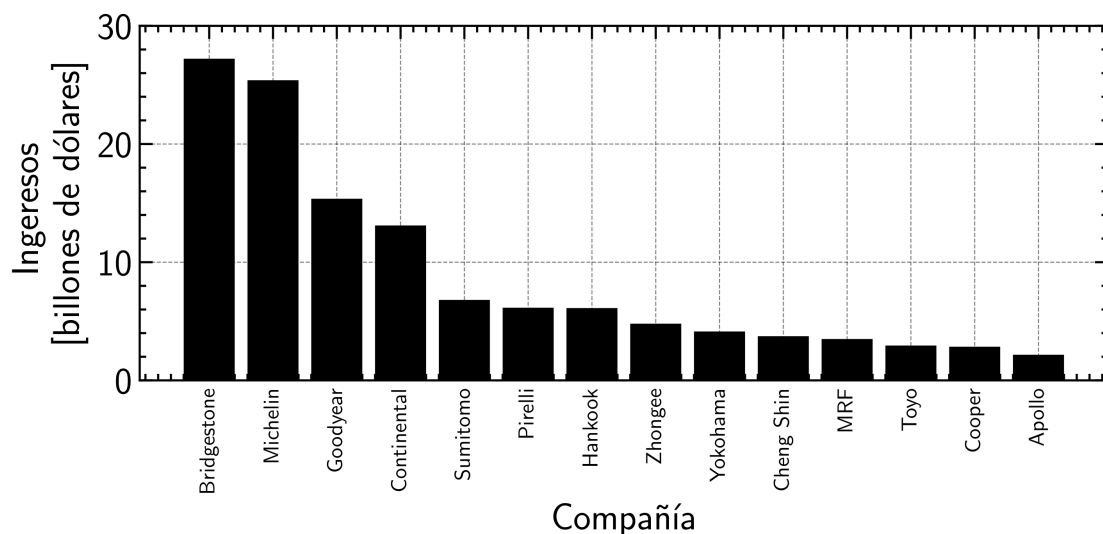


Figura 2: Ingresos de los fabricantes de neumáticos más relevantes en 2019.

mercantilización de producto asiático, incentiva a las establecidas multinacionales, a tomar un enfoque innovativo, con la intención de mantener su liderazgo (Chicu et al., 2020). El esfuerzo invertido en innovación, por una parte, intenta alcanzar sus objetivos mediante el rediseño y la mejora del producto. Mientras que por la otra, Trata de optimizar sus procesos y reducir desperdicios, mediante la automatización y el uso inteligente de los recursos disponibles.

A la hora de implementar mejoras, ya sea a un producto o a un proceso, la monitorización de los resultados y del cambio percibido se vuelve esencial. El producto o proceso experimental debe satisfacer las expectativas del cliente, y a su vez, cumplimentar la legislación vigente, como los estándares de calidad y medio ambiente. La monitorización de estas mejoras implica un coste elevado, a nivel temporal y económico, ya que estas deben ser puestas a prueba. En el caso del producto, su industrialización, supone reservar recursos que podrían ser destinados a la producción regular. Desde la materia prima utilizada en la elaboración de estos productos experimentales, transitando por cada maquina ocupada para transformarlo, hasta llegar a el laboratorio de calidad, donde se realizan ensayos para otorgar feedback a el nuevo proceso.

El laboratorio de Calidad del Producto (LCP), es el encargado de llevar a cabo los ensayos relevantes para asegurar la conformidad de la producción. A diario se cerciora de que los neumáticos manufacturados dentro de la planta cumpla con los estándares de calidad definidos por la compañía. El laboratorio logra asegurar

la calidad del producto, mediante un control estadístico de las muestras elegidas al azar por cada gamma de producto. Paralelamente, una fracción de los recursos disponibles, es demandada por los proyectos de industrialización. Obligando al departamento a equilibrar ambas necesidades.

En un entorno con recursos limitados, donde la demanda no cesa de incrementar, es cuestión de tiempo que los recursos se agoten y la capacidad de el laboratorio se vea paralizada. Para hacer frente a esta situación, y poder mantener el esencial ritmo marcado por los estándares de control de calidad e industrialización, debe realizarse un escalado de los recursos del LCP. A fin de que esta expansión, de este sistema complejo, sea realizada de la manera optima, debe realizarse un estudio de el impacto de las inversiones que podrían realizarse, en la capacidad del LCP. Para realizar esta tarea, se ha optado por diseñar una Simulación de Eventos Discretos (DES) que prevea los posibles futuros escenarios, al cambiar algunas de las variables independientes de entrada.

El autor de este trabajo, se ha decantado por este método, debido a la capacidad que ha demostrado a la hora de resolver problemas de optimización en sistemas estocásticos. Como menciona el autor (Allen, 2011), su versatilidad ha sido demostrada en numerosos ámbitos, como aplicaciones militares, sistemas sanitarios, problemas logísticos y optimización de procesos de manufacturación.

## **1.1. OBJETIVOS**

El objetivo principal de este trabajo es proveer a la planta hipotética de cubiertas con la información necesaria para la optima expansión del LCP. Este trabajo se limitara a analizar los distintos escenarios investigados, ofreciendo métricas e información cualitativa sobre las distintas posibilidades, para finalmente decantarse por la mejor opción.

Para lograr el objetivo final se deberán completar los siguientes subobjetivos:

- Modelar los procesos del LCP de acuerdo a los fundamentos de una DES, obteniendo un modelo ajustado a la realidad.
- Definir las variables independientes del proceso, que posteriormente serán usadas en la simulación.
- Estimar el tiempo de ciclo de cada proceso, a través de la asignación de distribuciones ajustadas a cada subproceso.

- Definir las variables dependientes que otorgara el sistema a la salida.
- Desarrollar un programa de simulación en el entorno de Python mediante el uso de la librería Simpy. Dicha simulación, sera capaz de emular los distintos escenarios propuestos durante el desarrollo.
- Proponer una alternativa de la distribución de los recursos actuales para asegurar la capacidad del LCP cuando en el futuro aumente la demanda de ensayos.

## 2. MARCO TEÓRICO

### 2.1. SIMULACIÓN DE EVENTOS DISCRETOS

Con el avance tecnológico experimentado desde los inicios de la computación, nuevas maneras para afrontar la resolución de problemas han surgido, que un siglo antes, habrían sido descartadas por su falta de viabilidad. La capacidad de calculo y el diseño de inteligencias artificiales, a proporcionado las herramientas necesarias para solucionar aquellos problemas, que con métodos clásicos, definitivamente demasiado complejos para modelar. La simulación, hoy en día, es una de las tres metodologías consolidadas, en el ámbito científico e ingenieril, para la resolución de problemas (Banks, 1998). Siendo esta metodología descrita como “la técnica del ultimo recurso”, por el autor (Garzia et al., 1986), debido a la intensa demanda computacional que requería en el momento de su publicación, el transcurso de los años ha mermado esta desventaja.

Diversos autores describen la simulación de la siguiente manera

- “Una simulación es el establecimiento de un modelo lógico-matemático de un sistema y la manipulación experimental de este en una computadora digital (Pritsker, 1974)”.
- “La simulación es el proceso de diseñar el modelo de un sistema real y realizar experimentos con este modelo con el propósito de, o bien, entender el comportamiento del sistema, o evaluar distintas estrategias (dentro de los limites impuestos por un criterio o conjunto de criterios) para la operación del sistema” (Shannon and Johannes, 1976).

- “Una simulación es la imitación de el modo de operación de un proceso real o sistema durante el transcurso del tiempo” (Banks, 1999).

Como se puede observar, el modelo, es un termino recurrente en el ámbito de la simulación. Banks sostiene que el modelo es la representación del sistema simulado (Banks, 1998). El autor matiza que la virtud de un modelo radica en su complejidad, siendo necesario el balance entre una representación ajustada, sin complicar en exceso el modelo. Por tanto, aquellos factores que no influyan lo suficiente en los resultados de la simulación, deberían ser eliminados, ya que únicamente extienden el proceso de desarrollo.

El modelo, por tanto, constituye la base de la simulación, define las variables, y criterios para las decisiones tomadas en la simulación. Banks señala, la distinción entre un modelo matemático convencional y el modelo de una simulación (Banks, 1998). Los modelos matemáticos y estadísticos suelen representar las variables de manera explícita, relacionando variables independientes y variables dependientes para obtener un resultado. En el caso de una DES, el modelo utilizado se enfoca en la representación de sus componentes internos y sus interacciones..

El funcionamiento de la DES, se basa en registrar los cambios que van ocurriendo en el estado del sistema durante el transcurso del tiempo. En el intervalo entre alternaciones de estado, ocurren el concepto de eventos, propios de esta rama de simulación. Varga (Varga, 2001) describe estos eventos como instantáneos, siendo la duración de los mismos nula. Durante el evento no ocurre nada especial, salvo la alteración del estado del sistema. Una sencilla representación de este concepto puede ser observada en la Figura 3, donde el sistema representado es un coche. El coche posee 3 estados, los cuales se van sucediendo, durante el transcurso del tiempo.

### 2.1.1. Terminología y funcionamiento de una DES

Para describir detalladamente este tipo de simulaciones, es necesario definir previamente algunos conceptos. Los siguientes conceptos son la síntesis de la información redactada por (Banks, 1998) aplicados en el entorno de Python y su librería Simpy.

**Las variables del estado** del sistema proporcionan información acerca de el comportamiento de los procesos que se llevan a cabo durante la simulación. Determina las variables de estado que el programa debe retornar depende de el objetivo

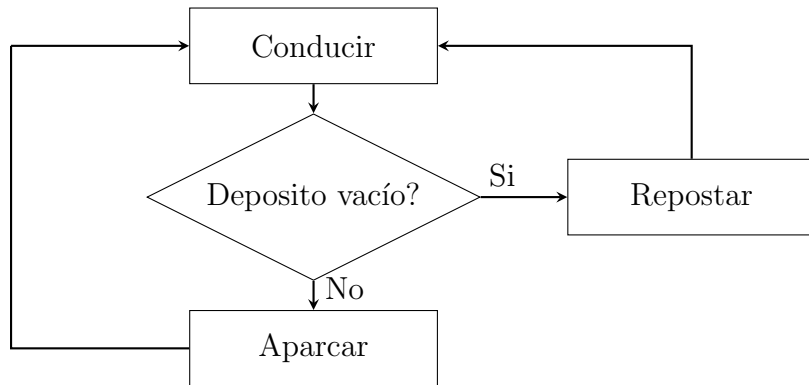


Figura 3: Representación del cambio de estado en una DES, los rectángulos representan el estado del sistema, mientras que las flechas indican el suceso de un evento instantáneo.

del proyecto. El programa deberá ser diseñado entorno a ellas, puesto que, serán usadas para el posterior calculo de indicadores del rendimiento. Un ejemplo, es el momento de inicio y fin de un subproceso.

**Las entidades** son los objetos representados por el modelo. Se distinguen dos tipos de entidades, las dinámicas y las estáticas. Las primeras, avanzan en el sistema a través de los procesos simulados, mientras que las segundas, dan servicio a las primeras permaneciendo ocupadas en el proceso. Considerando una gasolinera como sistema, la entidad dinámica seria un coche repostando combustible, mientras que la estática, seria la estación de repostaje. Estos objetos pueden poseer atributos, como el consumo de combustible o el tamaño de su deposito. Dependiendo de el objetivo de la simulación, algunos atributos serán relevantes en diseño del modelo.

**Un recurso** es una entidad que provee de servicio a una entidad dinámica. Los recursos pueden servir a múltiples entidades simultáneamente. De manera similar, una entidad puede solicitar varios tipos de recursos en distintas cantidades. Si la demanda de una entidad no se satisface, esta entrara en una cola, a la espera de la liberación del recurso. Cuando la entidad tome el recurso, lo mantendrá ocupado durante la duración del proceso. Los recursos pueden tener varios estados, como disponible, ocupado, averiado, en mantenimiento...



**Una actividad** es un periodo de tiempo, el cual es conocido previo a su inicio. Su duración puede ser determinada de varias maneras; mediante una constante, una distribución estadística, el resultado de una ecuación, un archivo o una decisión lógica. El conocimiento de su duración, supone saber cuando un recurso sera liberado, lo que facilita el trabajo de computación.

**Una demora** es un periodo de espera indefinido antes del comienzo de una actividad. Este depende de la ejecución de el resto de actividades prioritarias, por lo que el final de la espera no se puede determinar mientras el recurso siga ocupado.

**Un evento** ocurre al principio y al final de una actividad o demora. Marcan la transición de el sistema de un estado a otro.

Considerando todos estos conceptos, las entidades del sistema compiten entre ellas por el uso de los recursos, si estos no se encuentran disponibles, quedan a la espera de poder solicitarlos. Tanto las actividades como las demoras reclaman estas entidades durante los intervalos entre eventos, siendo liberadas para el siguiente proceso una vez finalizado el periodo. Una DES, puede describirse como el avance de una entidad o conjunto de entidades a través de las actividades que modelan el proceso real. Este avance es accionado mediante el transcurso de tiempo virtual dentro de la simulación.

### **2.1.2. Exposición de un caso ficticio usando una DES**

Con el objetivo de ilustrar los conceptos descritos en la sección 2.1.1 y demostrar la capacidad de resolución de problemas de la DES, se ha procedido a emular un caso práctico.

Supóngase que un ayuntamiento ficticio desea disminuir los problemas causados por el congestionamiento del trafico en la ciudad. El ayuntamiento solicita a una consultoría ingenieril desarrollar un proyecto para observar el desempeño de las distintas soluciones planteadas. La consultoría ve apropiado usar una DES para valorar el impacto de las distintas propuestas y proponer un plan de acción optimo a la entidad contratante.

Entre los objetivos del contratista se encuentra el objetivo principal de minimizar la cantidad de vehículos que circulan por la ciudad. Una de las causas de congestión apunta a una estación de repostaje con unos precios particularmente competitivos

que causa complicaciones a la entrada de la ciudad. Se cree que su capacidad de abastecimiento no es suficiente durante los picos de alta actividad. Por otra parte, la ciudad es un potente centro de actividad comercial. Muchos empleados se desplazan a diario en coche, lo que sumado a los espacios limitados de estacionamiento, y la alta densidad poblacional, hace sospechar que los conductores se demoran en exceso a la hora de buscar parking. Se teme que este tiempo adicional, sea un factor considerable en las emisiones totales de los vehículos.

Con las siguientes dos medidas se espera obtener una reducción del congestionamiento del tráfico en la ciudad.

- Primeramente, negociar un ampliamiento del servicio de distribución de combustibles con la empresa propietaria de la estación de repostaje.
- Paralelamente, ampliar la capacidad de un parking público gratuito ubicado cerca del núcleo de actividad laboral.

La simulación planteada, tratara de emular el comportamiento de los vehículos que circulan por la ciudad durante el transcurso de una jornada laboral normal. Es por ello que el comienzo de la simulación se estima entorno a las 7:00 a.m. y se prologa hasta 9:00 p.m. El modelo producirá 2 tipos de entidades, el vehículo de tipo 'A', que sera el medio de transporte para los trabajadores de la zona, y el de tipo 'B' que englobará todo tipo de usuarios ajenos a el ámbito laboral. Esta distinción se ha considerado necesaria, debido a la uniformidad que representa el primer grupo, respecto al segundo, algo más caótico. Los vehículos generados seguirán la rutina descrita en la Figura 4.

La consultoría, por una parte, establece las propiedades generales de su modelo en la Tabla 1. Mientras que, por otra parte, decide recopilar información clave para el desarrollo del modelo de simulación. A raíz de un análisis obtiene los datos mostrados en la Tabla 2.

Una vez finalizado el proceso de diseño de la simulación, la consultoría posee el código fuente expuesto en la Sección A.1 del Anexo I. Este programa, devuelve la Tabla, la cual contiene datos acerca del inicio y final de cada proceso. Con el correcto tratamiento de datos recibidos, se obtienen los siguientes resultados.

Como puede observarse en la Figura 5, la adición de 2 estaciones de combustible evita el colapso de la gasolinera en intervalos críticos. En el gráfico situado a la izquierda, el pico de actividad azul entorno al minuto 150 de la ejecución de la simulación, se ve drásticamente reducido, mientras que el repunte secundario es

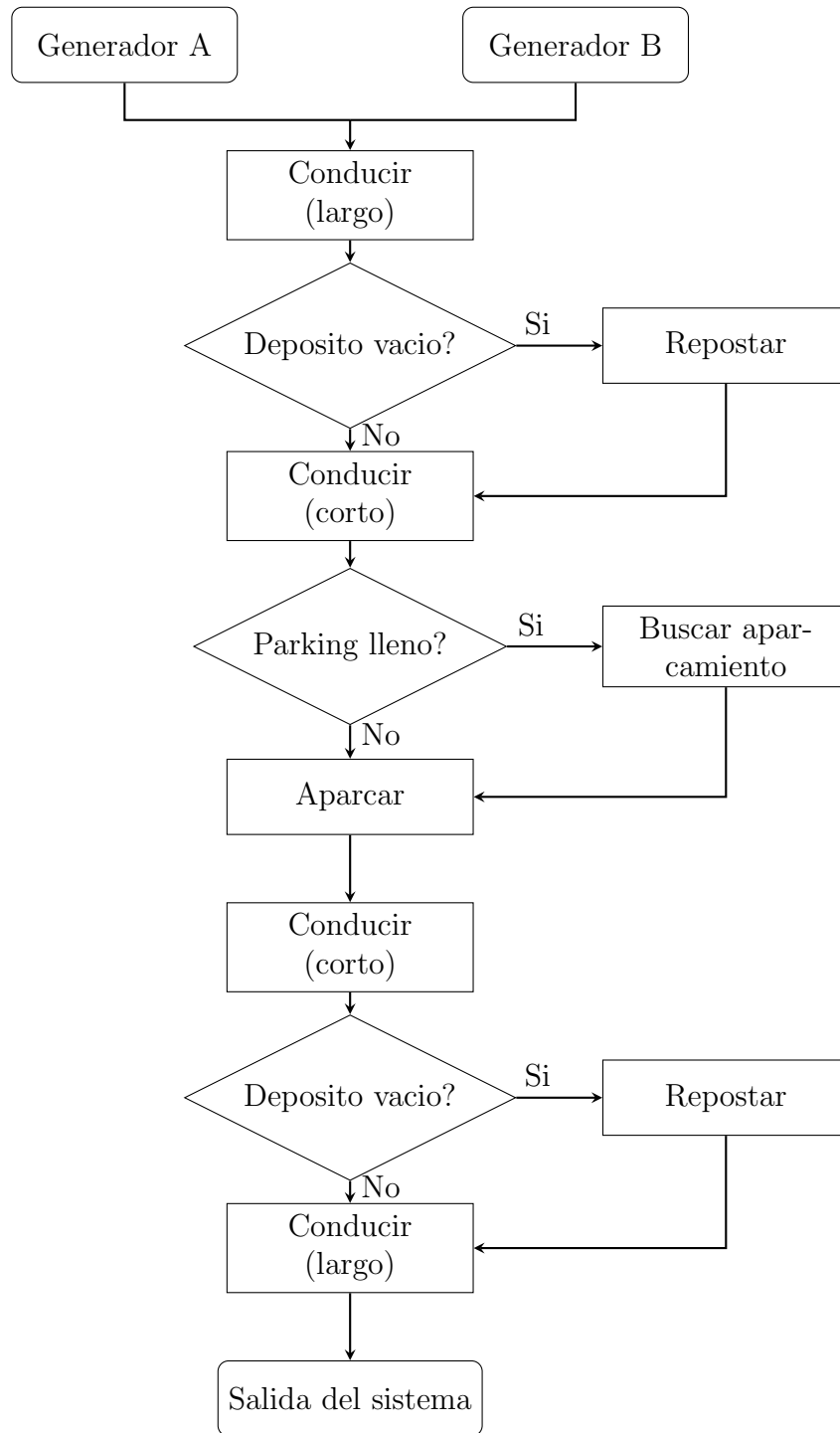


Figura 4: Modelo del caso practico descrito en el apartado 2.1.2

Cuadro 1: Especificaciones generales.

Nombre	Valor
Duración	960 min
Ejecuciones	20 uds.
Capacidad del parking	100 uds.
Estaciones de repostaje	4 uds.
Probabilidad de repostaje	50 %

Cuadro 2: Propiedades de las distribuciones observadas en los procesos reales.

Variable	Tipo de vehiculo	Distribución	Constantes (min)
Conducir largo	A y B	Gamma	$k = 8, \theta = 3$
Conducir corto	A y B	Triangular	$min = 2, moda = 3, max = 4$
Buscar parking	A y B	Gamma	$k = 10, \theta = 1$
Repostar	A y B	Triangular	$min = 4, moda = 5, max = 8$
Entre llegadas	A*	Exponencial	$\lambda = 1$
	B	Exponencial	$\lambda = 15$
Estacionamiento	A	Normal	$\mu = 560, \sigma = 60$
	B	Lognormal	$\mu = 5, \sigma = 0,7$

\*La generación de entidades tipo 'A' únicamente se prolonga desde el comienzo de la simulación hasta el minuto X. Así, se obtiene una aproximación, a el horario típico de un día laboral.

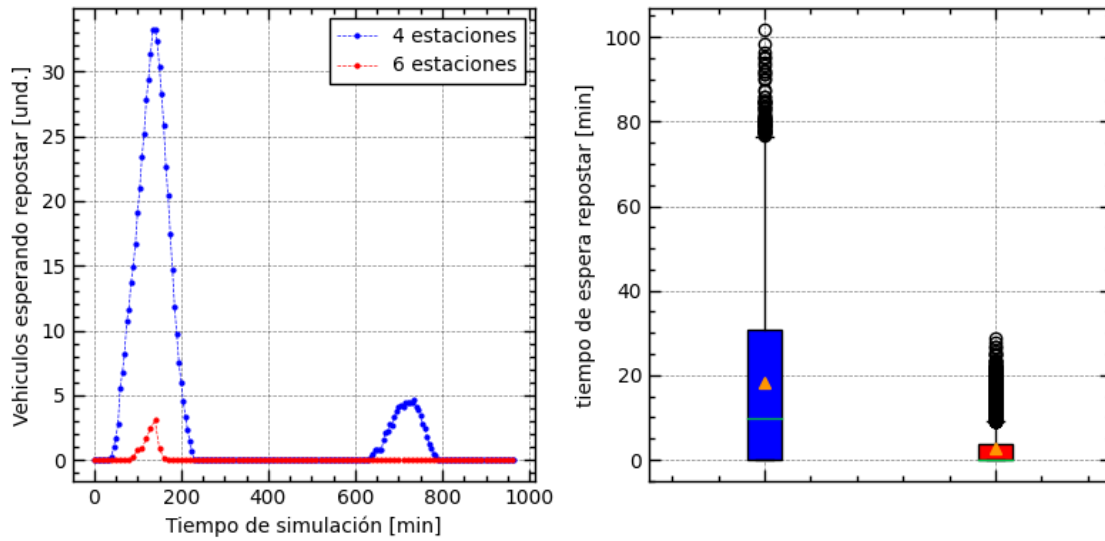


Figura 5: Izquierda: Evolución de los vehículos a la espera de repostar para distintos escenarios. Derecha: Distribución de los tiempos de espera de repostaje para los distintos escenarios

completamente eliminado. A su vez, en el gráfico adyacente, La desviación estándar de tiempos de espera queda reducida y su media queda desplazada hacia el eje.

Continuando con el análisis, en la Figura 6, El efecto de incrementar en un 50 % las plazas de parking disponible, no es apreciable en el gráfico expuesto arriba. La cantidad de vehículos en carretera permanece similar, mientras que en el gráfico inferior los vehículos saturan el parking hasta su nuevo limite. Al no haber una clara correlación entre ambos indicadores, se ilustra la representación de la distribución de tiempos de conducción en la Figura 7. En esta exposición, puede llegar a apreciarse una ligera diferencia en los tiempos de conducción, aunque un test de hipótesis sería necesario para determinar si estas diferencias son estadísticamente significativas.

En conclusión, los resultados muestran una clara ruta de inversión a la hora de ampliar la estación de combustibles, mientras que la expansión de el parking parece irrelevante. Esto demuestra la capacidad de recopilación de datos que ofrece una DES. Simultáneamente subraya la importancia del tratamiento de los datos obtenidos, y el uso de pruebas estadísticas, para la obtención de conclusiones certeras.

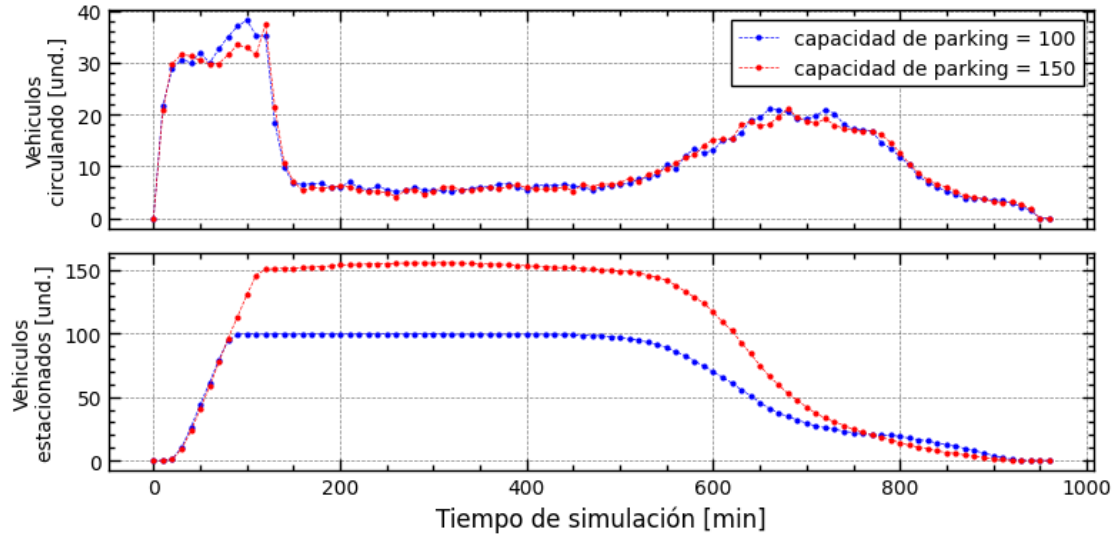


Figura 6: Evolución de la cantidad de vehículos a lo largo de la simulación, en distintos escenarios. Arriba: Vehículos en circulación. Abajo: Vehículos estacionados.

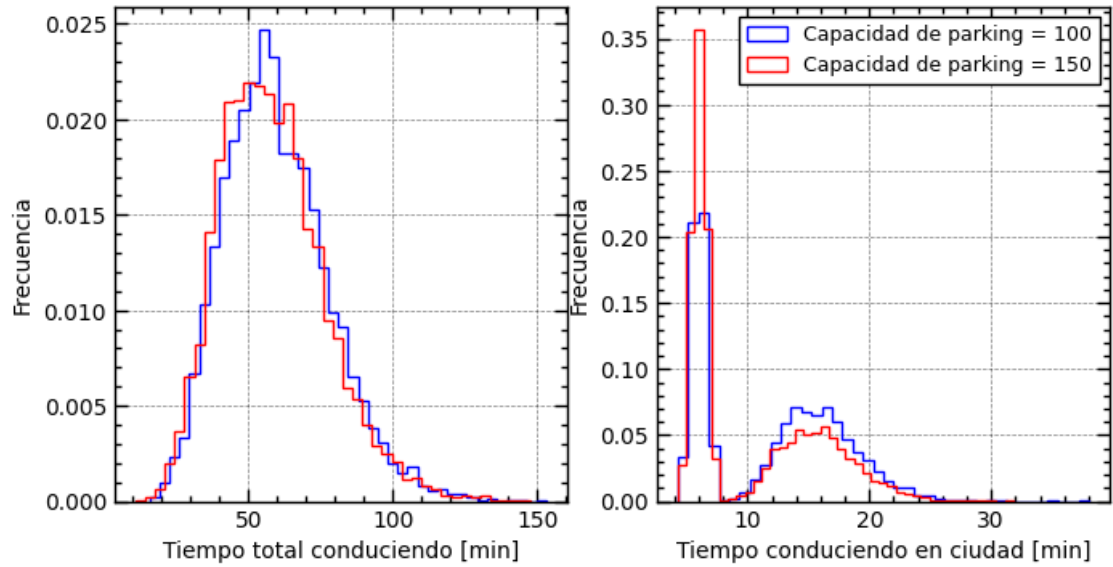


Figura 7: Distribución de el tiempo de conducción para distintos escenarios. Izquierda: Tiempo total en circulación. Derecha: Tiempo circulando dentro de la ciudad.

### 2.1.3. Etapas de desarrollo de una DES

Para que un proyecto tan complejo como una DES prospere, es fundamental estructurarlo. El diagrama expuesto en la Figura 8 publicado por (Banks, 1998) define las etapas que deberían seguir este tipo de proyectos. Al ser este proyecto un simple Trabajo de Fin de Grado, este esquema servirá como orientación, más que como plano para la construcción de la simulación. Por ejemplo, la recopilación de datos ha resultado imposible en el momento de redacción del trabajo, pero se espera desarrollar el proyecto a partir de las más ajustadas distribuciones de probabilidad supuestas para el desarrollo del modelo. Se espera que este trabajo asiente una base, y que las imperfecciones actuales puedan ser corregidas en el futuro.

#### **Formulación del problema, establecimiento de objetivos, y plan general.**

Cada DES comienza con la declaración del problema que se tratará de solucionar. Esta etapa es crucial debido a que el resto de el trabajo se asentara sobre esta base. Se debe asegurar que el problema que se esta formulando sea el correcto, la comunicación entre el cliente y el equipo desarrollador debe ser optima. Una descripción precisa y un conocimiento extenso del sistema son necesarios para evitar trabajar en solucionar el problema equivocado. Es por ello que debe haber comunicación y feedback constante entre cliente y equipo.

En el caso de este trabajo, algo similar sucede. Al ser un único autor, la comunicación no sera un factor importante, pero esto conllevara otras complicaciones. El conocimiento de el autor debe ser extenso en cada ámbito, debe comprender profundamente el sistema a modelar, y a su vez, dominar el uso de las herramientas de simulación. Manejar ambas, requerirá recordar la formulación del problema inicial más frecuentemente para no desviarse de la dirección.

Respecto a los objetivos, estos deben ayudar a asentar la dirección de trabajo actuando como plan general. El establecimiento de varios objetivos alcanzables, cohesivos y correctamente secuenciados, guiará el proceso de desarrollo. Esto objetivos quedan listados en la sección 1.1.

**Conceptualización del modelo.** El sistema propuesto sugiere conceptualizar el modelo en orden inverso, es decir, comenzar identificando los objetivos del proyecto, tal y como se ve en la Figura 9.

El siguiente paso es hallar las preguntas clave a las que se les quiere dar respuesta.

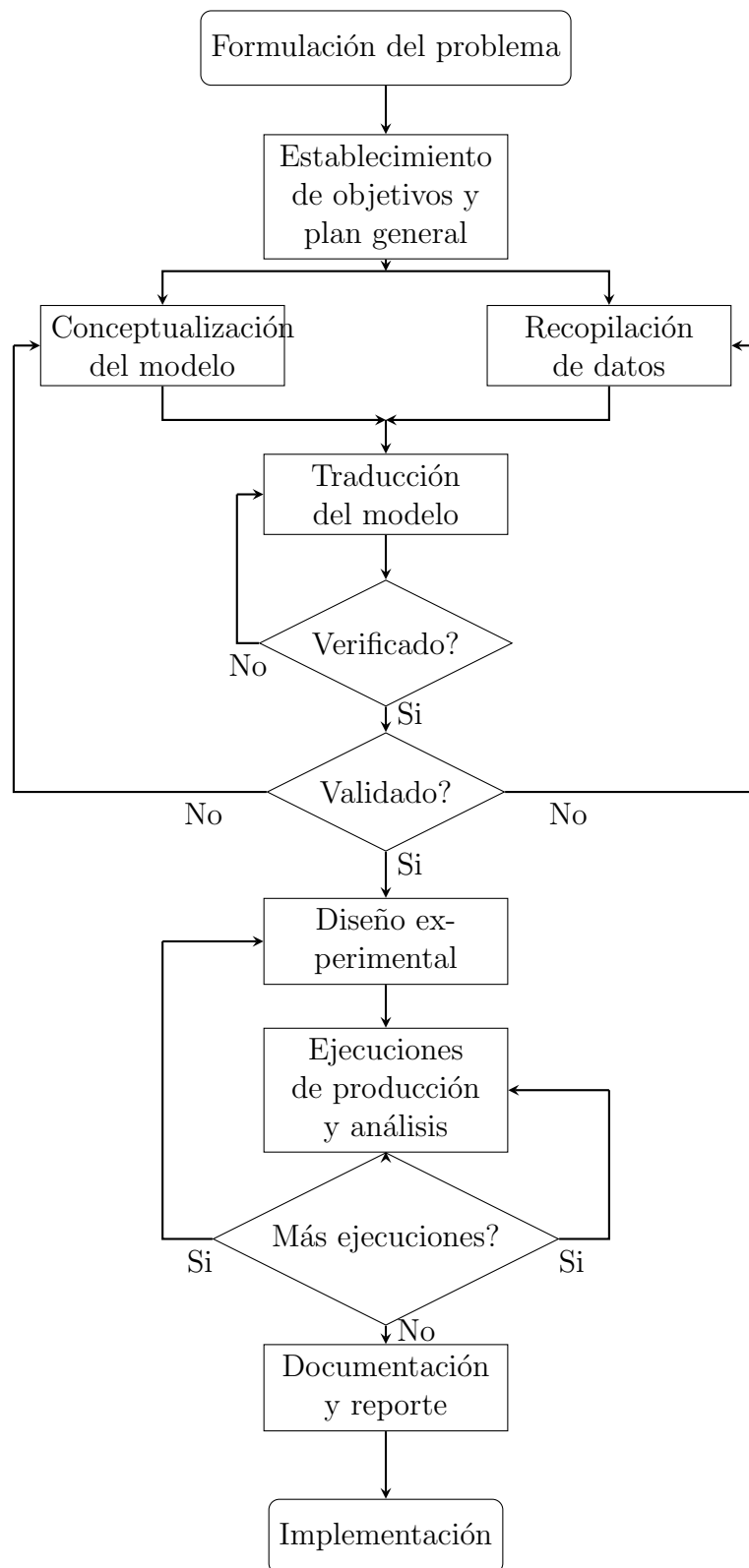


Figura 8: Esquema del proceso de desarrollo de una DES.



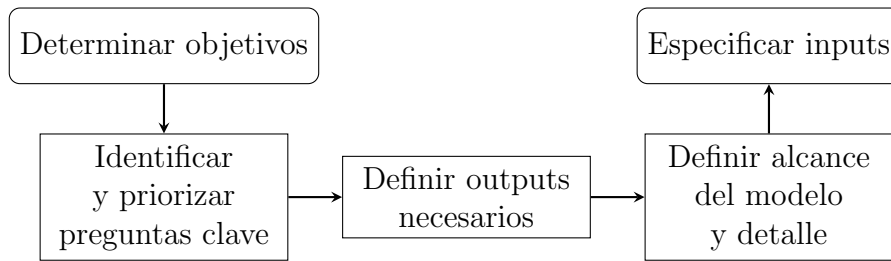


Figura 9: Diagrama de pasos a seguir en la conceptualización de un modelo.

Generar una lista, y ordenar las preguntas respecto a su relevancia es lo ideal. Además, cuantificar los beneficios asociados a estas, aclara su jerarquía.

Después, se debe determinar que outputs son necesarios para contestar a las preguntas previamente formuladas. Introduciendo en la simulación únicamente la complejidad necesaria, agilizará el proceso de desarrollo y mantendrá el enfoque original.

Continuando con el alcance de la simulación, se deben detallar los procesos críticos en el cálculo del output, descartando o simplificando aquellos que no tengan gran impacto en la salida. Por ejemplo, modelar recursos abundantes en la simulación, no tendrá impacto en el output, mientras que conocer los tiempos de espera de un proceso sí lo hará.

Se Finaliza, especificando las variables independientes que se introducirán al sistema. Estas variables tienen que ser de alta calidad, ya que, de lo contrario su introducción puede perjudicar al sistema al completo. Un ejemplo sería el de distinguir entre distintas distribuciones para entidades con atributos pobremente caracterizados.

**Recopilación de datos.** Cualquier DES requiere datos. La recopilación de la información del sistema, o la estimación en caso de ser un sistema ficticio, es una necesidad.

En este trabajo, al no disponer de datos reales, se obviara el proceso de recopilación. El enfoque se situara en la correcta elección de las distribuciones, recopiladas en la sección 2.2.2, para representar el sistema ficticio a modelar.

**Traducción del modelo.** El modelo de simulación se construye a partir de la información desarrollada en el anterior punto. Este proceso consiste en traducir de manera eficaz el esqueleto previamente desarrollado, a un lenguaje que el ordenador pueda procesar. Para ello el libro sugiere lo siguiente:

- Enfoque en el problema. El proyecto no debería limitarse a desarrollar el modelo, si no que debería tratar de invertir aproximadamente el mismo tiempo en la exploración del sistema y la solución de problemas.
- Comienzo simple. El comienzo del proyecto debe ser simple, adquiriendo complejidad hasta alcanzar el detalle necesario.
- Manejo de la inteligencia del modelo. El modelo desarrollado puede desarrollar ventajas respecto a su contrapartida real, como conocer la ubicación exacta de una pieza instantáneamente. Los humanos que gestionan el proceso difícilmente van a poder lograr este control, y supondrá una característica inalcanzable en el sistema real.
- Revisión. Revisar el funcionamiento del proyecto frecuentemente. Esto evitará desvíos respecto al objetivo principal y facilitará el proceso de verificación y validación.

**Verificación y validación.** Comprobar el correcto funcionamiento de el proyecto desarrollado es fundamental. Esta es la labor de la verificación y validación (VV). El autor sugiere que este paso se realice frecuentemente a lo largo de el proceso de desarrollo. Esto evita complicaciones una vez el proyecto se vuelve demasiado extenso y demasiado complejo. Realizar la VV en cada pieza funcional de el modelo cada vez que suceda algún cambio es lo recomendable.

Por una parte, la verificación es el proceso encargado de comprobar que la simulación se ejecute de la manera esperada. En este proceso se intentará hallar bugs en el programa. Esto se logra comprobando que la ejecución de cada paso en la simulación sea la correcta y dé los resultados esperados.

Por otra parte, la validación consiste en comparar, el comportamiento del modelo respecto a el comportamiento real del sistema. En el caso de poseer datos sobre el sistema simulado, el libro sugiere utilizar herramientas estadísticas como, el test Student t o análisis de regresión donde el ajuste a la realidad sera comprobado. En el caso de no poseer los datos del sistema, como es este, se optara por un enfoque cualitativo. Aunque no haya datos, el comportamiento de un sistema puede predecirse cualitativamente, al cambiar ciertas variables. Por ejemplo en un tanque

de agua cilíndrico, aunque el volumen del tanque y el caudal de entrada del agua sea desconocido, el nivel del agua disminuirá si el flujo neto es negativo. Si el programa, ejecutado obtiene el resultado contrario, este será invalidado comenzando la revisión del mismo.

**Análisis.** Mediante el análisis del output obtenido de la simulación, se obtienen las conclusiones a las preguntas planteadas al inicio del proyecto. Primeramente, se deberán plantear los distintos escenarios que serán simulados. Estos escenarios serán ejecutados en la simulación obteniendo su output. El correcto tratamiento de los datos, junto con una exposición significativa de ellos dará claridad a los resultados. No obstante, para obtener las conclusiones deseadas, deberán realizarse testes estadísticos a los resultados para certificarlos. En este proyecto, se aplicarán los test descritos en la sección 2.2.3.

**Documentación y reporte.** Con la intención de que este proyecto pueda ser continuado en el futuro por otra persona, la documentación se vuelve necesaria. El proyecto se considera documentado, por una parte, mediante este trabajo, y por la otra mediante un código fuente auto-explicativo, con abundantes comentarios, que se haya en el Anexo A.2.

**Implementación.** La implementación de este proyecto queda fuera de su alcance, por lo que no se profundizará en este tema.

## 2.2. TEORÍA DE LA PROBABILIDAD Y DES

La probabilidad está estrechamente relacionada con la DES. Todo el cálculo realizado en una simulación tiene el objetivo de calcular el valor esperado o la media de las variables dependientes de un proceso que se está intentando medir. Definir ciertos conceptos estadísticos se vuelve imprescindible para el correcto desarrollo de este tipo de proyecto.

### 2.2.1. Variables aleatorias o random

Según el autor (Meester, 2008), una variable random ( $X$ ) se puede explicar como un número cuyo valor no es conocido en el momento de su planificación. Aunque

el valor ultimo de esta variable sea desconocido, el rango de valores que pueden tomar estas variables viene determinado por una función.

### 2.2.2. Distribuciones de probabilidad

Una distribución de probabilidad es una función estadística que determina la probabilidad de que una variable random (Simon, 2002), obtenga todos los valores posibles que contiene la propia distribución. Las distribuciones comúnmente usadas en los modelos de una DES son las siguientes:

**Distribución uniforme.**

$$P(x) = \frac{1}{b - a} \quad (1)$$

**Distribución normal.**

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

**Distribución lognormal.**

$$P(x) = \frac{1}{\sigma x \sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}} \quad (3)$$

**Distribución triangular.**

$$P(x; a, m, b) = \begin{cases} \frac{2(x-a)}{(b-a)(m-a)} & \text{para } a \leq x \leq m, \\ \frac{2(b-x)}{(b-a)(b-m)} & \text{para } a \leq x \leq m, \\ 0 & \text{para el resto} \end{cases} \quad (4)$$

**Distribución exponencial.**

$$P(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0 \end{cases} \quad (5)$$

### Distribución gamma.

$$P(x) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)}, \text{ siendo: } \Gamma(k) = (k-1)! \quad (6)$$

#### 2.2.3. Tests de hipótesis

La comprobación de los resultados obtenidos en las DES, es una parte fundamental de la simulación. Para ello se suelen emplear los test de hipótesis como muestra de la validez de las conclusiones obtenidas.

El test de hipótesis involucra formular, 2 o más hipótesis, que serán puestas a prueba mediante el empleo de tests estadísticos, a las muestras poblacionales obtenidas (Martin, 2022).

Un formato común en el ámbito de test de hipótesis es el siguiente:

- Hipótesis nula,  $H_0$ , trata de demostrar que el fenómeno observado es únicamente resultado del azar.
- Hipótesis alternativa,  $H_1$ , trata de demostrar la hipótesis que se desea probar. Existen 3 tipos:
  - parámetro de población  $>$  valor hipotetizado.
  - parámetro de población  $<$  valor hipotetizado.
  - parámetro de población  $\neq$  valor hipotetizado.

Para realizar un test de hipótesis, primero debe escogerse un test, y aplicarlo a las muestras de la población a analizar. Después, la hipótesis nula es rechazada o no dependiendo de el valor-p obtenido.

Un test de hipótesis muy usado es el test t-student 7.

$$t = \frac{\tilde{x} - \mu_0}{s/\sqrt{n}} \quad (7)$$

El valor-p, es la probabilidad condicional de los valores a los extremos de un test estadístico, asumiendo que la hipótesis nula es cierta. Por convención, si la

probabilidad de la hipótesis nula en  $< 5 \%$ , o  $< 0,05$ , la hipótesis nula es rechazada. De lo contrario, no es posible rechazar la hipótesis nula, lo que no supone que esta sea cierta.

### **2.3. EL LABORATORIO DE EVALUACIÓN DEL PRODUCTO**

A lo largo del proceso de fabricación de una cubierta, son varios los estándares de calidad que debe cumplir el producto para que pueda ser vendido en los distintos mercados a nivel global. Desde la composición química de las materias primas usadas, hasta las propiedades físicas del producto finalizado, la certificación de que el producto se halle dentro de los límites especificados, internamente como externamente, asegura un producto de calidad.

En el proceso final de control de calidad que se lleva a cabo en un laboratorio de evaluación del producto, estos son los ensayos principales.

- Ensayos Endurance
- Ensayos Rolling Resistance
- Ensayos dimensionales

En el laboratorio que ha sido el sujeto analizado en este trabajo, son dos clientes internos los que comparten los recursos disponibles. Por una parte, se encuentran los procesos de conformidad de la producción (CP), encargados de aseguramiento de la calidad de el producto destinado a la venta a clientes. Por otra parte, se encuentran los procesos de desarrollo de nuevos productos, o el departamento de industrialización (IND), encargados de desarrollar nuevas líneas del producto, y optimizar los procesos de producción mediante cambios controlados que suceden a bachas. Finalmente, se reciben solicitudes de ensayo de otras plantas (EXT) para dar soporte y abastecer sus necesidades de aseguramiento de la calidad. Estas 3 fuentes de muestras de ensayo suman la cantidad descrita en la tabla 3. Debido al incremento en la actividad en los años recientes, parece que se ha llegado a saturar la capacidad actual del laboratorio para realizar ensayos. Las perspectivas de futuro, prevén que el incremento de la actividad continúe, por lo que la fábrica experimentará una evolución al alza tanto de la producción regular como de los procesos de industrialización.

Cuadro 3: Suma de la demanda actual de ensayos por tipo de ensayo.

Ensayo	Demanda (unds.)
Endurance	280
Rolling	760
Dimensional	140

El flujo de los procesos llevados a cabo en el laboratorio, es representado en la Figura 10.

### 2.3.1. Ensayos Endurance

Estos ensayos someten a las cubiertas a unas condiciones particularmente extremas que no se llegan a alcanzar dentro del margen de uso habitual para el que están diseñadas. El ensayo consiste en hacer rodar el objeto de ensayo durante un periodo aproximado de 72 horas a una temperatura constante de 40 °C, aumentando la carga en pasos discretos a lo largo del test.

El laboratorio dispone de 4 maquinas similares a la que aparece en la Figura 11, cada una de ellas teniendo la capacidad de montar hasta 2 cubiertas.

El ensayo se reporta como exitoso si el objeto de ensayo no ha sufrido deformaciones antes de el tiempo especificado internamente para cada producto. Una vez finalizado el test, 2 secciones son cortadas y pulidas por un operario y son almacenadas para la elaboración del reporte.

### 2.3.2. Ensayos Rolling Resistance

El objetivo de este ensayo es determinar el consumo energético de la cubierta debido a la fricción causada mientras esta rueda. La maquina que aparece en la Figura 12 mide la resistencia que ofrece el objeto de ensayo a 25 °C durante alrededor de 3 horas. La particularidad de este ensayo consiste en que el resultado es sensible a la presión de inflado y temperatura de la sala.

El laboratorio dispone de una maquina capaz de realizar estos ensayos. Dependiendo del cliente interno, las muestras de ensayo serán almacenadas después del test durante 2 semanas para después ser inutilizadas por el operario mediante un corte.

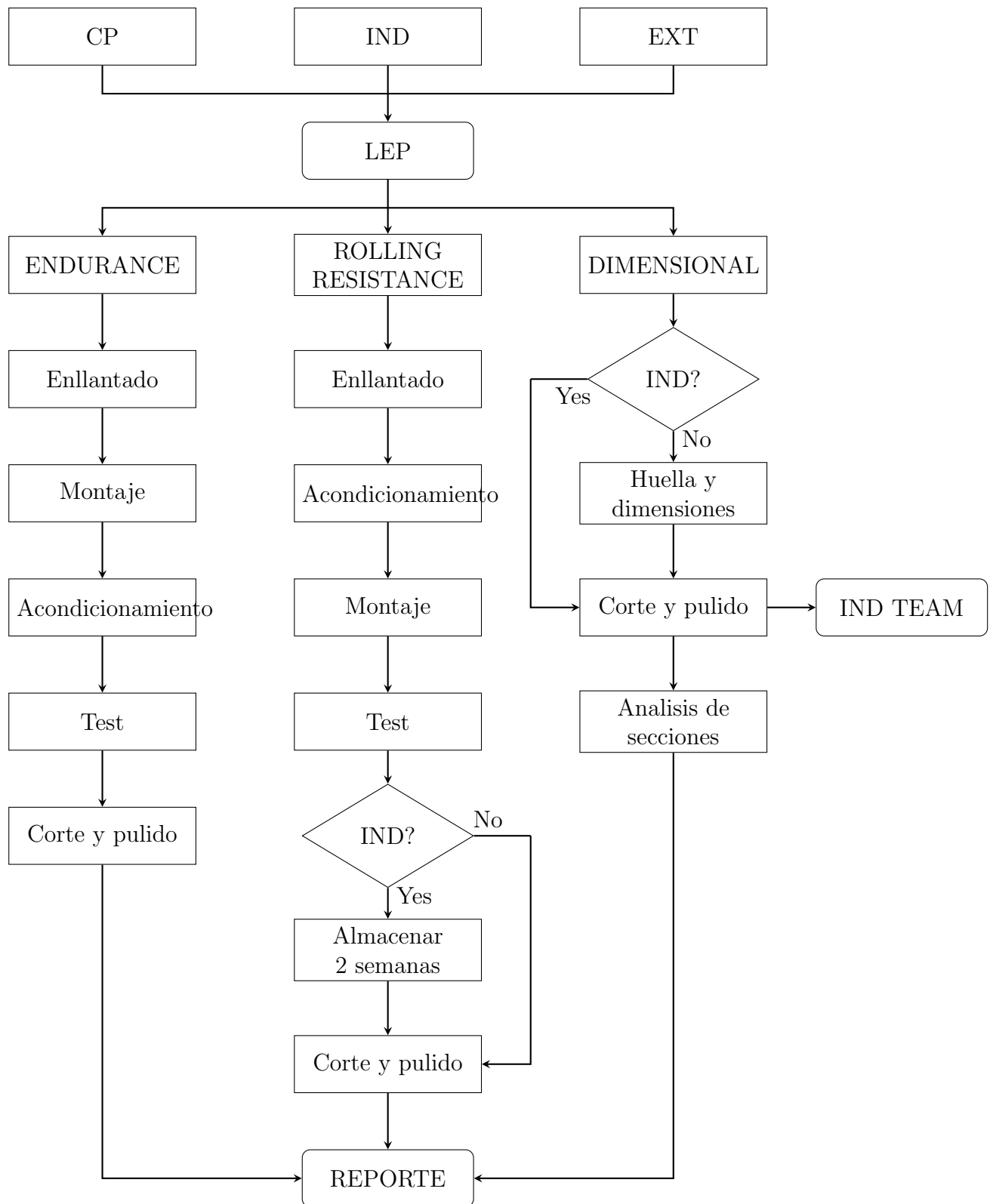


Figura 10: Diagrama de flujo de los procesos llevados a cabo en el Laboratorio de Evaluación del Producto.





Figura 11: Maquina utilizada en los ensayos Endurance

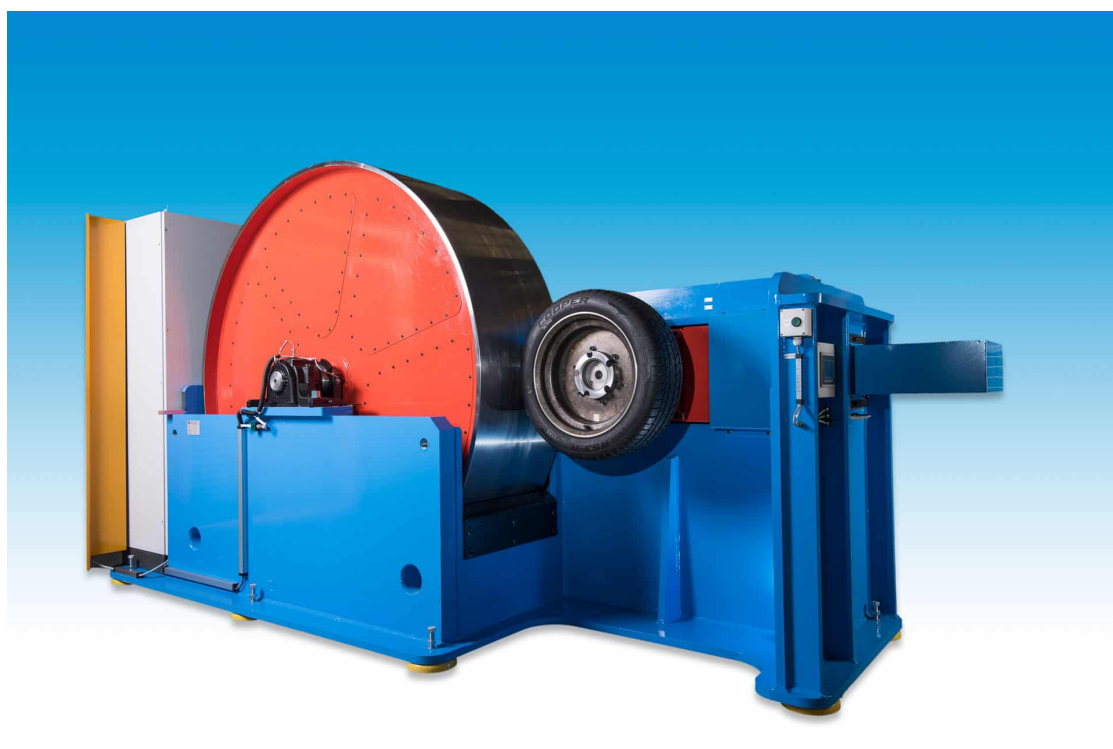


Figura 12: Maquina utilizada en los ensayos Rolling Resistance

### **2.3.3. Análisis dimensional**

El objetivo de este ensayo es verificar que tanto las dimensiones generales de la cubierta, como sus componentes internos quedan dentro de la tolerancia de las especificaciones.

Para ello, el laboratorio dispone de 2 técnicos adicionales que se dedican únicamente a llevar a cabo estos análisis. Cada análisis implica una alta carga de trabajo para el operario dedicado a preparar las secciones analizadas.

## **3. DESARROLLO**

### **3.1. FORMULACIÓN DEL PROBLEMA, ESTABLECIMIENTO DE OBJETIVOS Y PLAN GENERAL**

El problema que ha llevado a la necesidad de redactar este trabajo es la aparente escasez de recursos para acoger el futuro incremento de la demanda de ensayos en el LEP. Desde la gerencia, existen dudas sobre cual puede ser el factor limitante en los procesos actuales. Se observan varios cuellos de botella, entre los cuales están; el escaso tiempo de utilización de las maquinas endurance y rolling, el operario de corte y pulido saturado de trabajo y la falta de flexibilidad que industrialización reclama al laboratorio.

Con el fin de dar respuesta al que parece ser el factor mas relevante, el tiempo de utilización de maquina, se ha propuesto la idea de desarrollar una DES. El propósito sera determinar la causa de el bajo rendimiento de ensayos y sugerir alternativas al flujo de trabajo actual para satisfacer la futura demanda de ensayos.

Para ello primeramente, se determinara el alcance de la simulación que se realizará. Se desarrollará un modelo en base a el alcance determinado, y se tratara de sacar conclusiones en base a los resultados obtenidos.

Los objetivos de este proyecto quedan mas detallados en la sección ??.

Cuadro 4: Características generales de la simulación.

Propiedad	Valor
Duración	365 días
Jornada laboral	8 h
Turno(s)	1
Ejecuciones	100 unds.

### 3.2. CONCEPTUALIZACIÓN DEL MODELO

Empezando desde el final, una vez han quedado los objetivos descritos, se ha procedido a determinar las cuestiones clave para dar con una solución. La siguiente lista de preguntas ha sido la que al autor le ha parecido de mayor relevancia:

- Que sucedería si el tiempo disponible para el montaje de las máquinas fuese mayor? Aumentar el número de turnos cuantos ensayos más acabaría proporcionando?
- Cual sería el efecto de que dos técnicos trabajasen simultáneamente en los ensayos?
- Como afectaría la instalación de máquinas Endurance adicionales a la cantidad de ensayos realizados?

Para responder a las preguntas planteadas, se ha programado la simulación de tal manera que devuelva una tabla conteniendo los tiempos de inicio y fin de cada proceso. A través de análisis de los datos indexados, se han podido obtener estadísticas como; número de ensayos totales realizados por tipo de test, o el tiempo total de máquina en funcionamiento por cada ejecución de la simulación.

El modelo a reproducir para obtener los datos descritos en el anterior párrafo, ha sido deliberadamente diseñado de manera simple. Esto ha facilitado la obtención de datos representativos y su posterior análisis. Se ha tomado parte del diagrama de la Figura ??, Excluyendo ciertos procesos irrelevantes para las preguntas previamente formuladas. Estos diagramas se encuentran en el Anexo ??.

Los inputs necesarios para el funcionamiento del modelo se han recopilado en las Tablas 4, 5, 6 y 7.

Cuadro 5: Recursos representados en el modelo.

Recurso	Cantidad (unds.)
Máquina endurance	4
Máquina rolling	1
Máquina butler	1
Personal técnico	1
Llantas endurance	14
Llantas rolling	20

Cuadro 6: Características del proceso de ensayos endurance.

Proceso	Prioridad	Distribución (min)
Tiempo entre llegadas	-	Exponencial $\lambda = 120$
Enllantado	0	Triangular $a = 12, m = 15, b = 20$
Montaje	-1	Normal $\mu = 45, \sigma = 10$
Acondicionamiento	-3	Constante $c = 180$
Ajuste de presión	-4	Constante $c = 5$
Ensayo	-	Normal $\mu = 4320, \sigma = 360$

Cuadro 7: Características del proceso de ensayos rolling.

Proceso	Prioridad	Distribución (min)
Tiempo entre llegadas	-	Exponencial (min) $\lambda = 48$
Enllantado	0	Triangular $a = 12, m = 15, b = 20$
Acondicionamiento	-2	Constante $c = 360$
Montaje	-2	Normal $\mu = 45, \sigma = 10$
Ensayo	-	Normal $\mu = 180, \sigma = 20$

### 3.3. RECOPIACIÓN DE DATOS

Al carecer de la disponibilidad de datos, se ha tratado de representar cada proceso mediante distribuciones acordes a sus propiedades. Para los tiempos entre llegadas de objetos en la simulación, se ha elegido la distribución exponencial debido a su asociación a tiempos de espera entre eventos.

Para los procesos de los que menos información se poseía se ha optado por representarlos mediante una distribución triangular. Ya que aunque no hubiera información precisa, se ha podido estimar correctamente sus duraciones típicas, y sus límites superiores e inferiores.

Finalmente para los procesos cuya información era algo mas extensa, se han tratado de representar mediante distribuciones normales, debido a la sencillez y versatilidad de las mismas.

### 3.4. TRADUCCIÓN DEL MODELO

El modelo traducido, que se encuentra en el Anexo B, puede dividirse en 3 procesos generales:

**La generación de entidades** se encarga de crear instancias de las cubiertas que serán alimentadas a los procesos maestros de cada ensayo. Son 2 los generadores que posee la simulación, uno para los ensayos Endurance, y otro para los ensayos Rolling.

**Los procesos maestros** son los responsables de ejecutar los subprocesos en el orden correspondiente mientras demandan los recursos necesarios durante su transcurso. Al igual que los generadores, hay 2 procesos maestros, uno para cada tipo de ensayo.

**Los subprocesos** son las subrutinas llamadas por los procesos maestros. Crear módulos para cada tarea mejora la legibilidad del código.

**La obstrucción del técnico** es el proceso que simula el inicio y el fin de una jornada laboral, y hace transcurrir el tiempo, incluyendo los fines de semana.

### 3.5. VERIFICACIÓN Y VALIDACIÓN

El proceso de verificación se ha ido realizando a lo largo de la simulación. Se ha comprobado que los pasos se ejecutan en el orden esperado, y se ha revisado la lógica de los algoritmos propuestos.

Respecto al proceso de validación, se ha tomado como referencia el recuento ensayos realizados durante un año, y el tiempo de funcionamiento de maquina, y se ha comprobado que fueran similares.

## 4. RESULTADOS

Una vez simulados los distintos escenarios, se ha procedido a analizar las tablas de datos obtenidos por la simulación.

En la Figura 13 se han graficado las distribuciones de el numero de ensayos realizados en un año simulado para cada escenario. Cada escenario ha sido ejecutado 100 veces, y se ha trazado la frecuencia con la que ocurría un determinado numero de

Cuadro 8: Capacidad máxima de ensayos por maquina.

Ensayo	Capacidad maxima (unds.)
Endurance	934
Rolling	2502

ensayos en el histograma. La figura consta de 2 gráficas, correspondiendo el histograma (A) a los ensayos Endurance y el histograma (B) a los ensayos Rolling. De izquierda a derecha se pueden observar 4 distribuciones de distintos colores. Cada uno de ellos corresponde a una configuración de turnos del técnico encargado de estas maquinas. Los turnos en los cuales el operario esta disponible como recurso van desde 1 a tiempo completo, incluidos los fines de semana.

En el histograma superior puede observarse la distinción que ocurre entre las distintas configuraciones. Cada configuración tiene un amplio rango de ensayos, mostrando el comportamiento estocástico de este tipo de ensayos. Exceptuando la configuración a tiempo completo, la cantidad de ensayos, parece variar dentro de un rango común de 100 ensayos. Esta tendencia cambia al disponer a tiempo completo de un operario que monte la maquina. Al eliminarse la posibilidad de que el ensayo finalice fuera del horario laboral, el margen de ensayos realizados se estrecha al mismo tiempo que la cantidad de ensayos se aproxima a la capacidad ideal.

En el histograma inferior, las 3 primeras configuraciones expuestas, muestran un comportamiento similar observado en la configuración a tiempo completo observada en el anterior gráfico. Esto se debe, a que el técnico solamente puede llegar a lanzar un determinado numero de ensayos por turno trabajado. Esta cantidad se mantiene uniforme a lo largo de las ejecuciones debido a que la duración de este ensayo es inferior a la duración del turno del técnico. Mientras el horario de trabajo sea limitado la tendencia de ensayos corresponderá a la ecuación 8. En el escenario en el que el técnico esta disponible durante 3 turnos al día, la acumulación de eventos aleatorios se vuelve notable en la desviación de ensayos observada respecto a los anteriores 2 escenarios. Finalmente, a tiempo completo, la varianza aumenta considerablemente, ya que no hay un descanso que regule la aleatoriedad de los ensayos.

$$TestRealizados = 365 \cdot 8 \cdot \underbrace{\left( \frac{N_{turnos}}{\mu_{test}} + 1 \right)}_{\text{Entero truncado}} \quad (8)$$

Cuadro 9: Media y desviación estándar de ensayos realizados por cada escenario y tipo de ensayo.

Escenario	Ensayos endurance (unds.)		Ensayos rolling (unds.)	
	$\mu$	$\sigma$	$\mu$	$\sigma$
1 Turno	293.35	17.27	739.35	5.51
2 Turnos	516.83	26.38	1293.40	5.00
3 Turnos	701.07	27.25	1782.26	11.15
Tiempo completo	855.06	5.70	2202.28	73.18

Una representación mas significativa ocurre cuando se normaliza la cantidad de ensayos respecto a la capacidad máxima de trabajo. De esta manera, se obtiene la fracción de tiempo de utilización de maquina representado en la Figura 14. Esta conversión logra asentar la base necesaria para comparar el grado de utilización de maquinas de manera directa.

Si se grafica la fracción de tiempo de utilización de maquina, respecto a la fracción del tiempo trabajado, con los escenarios expuestos hasta ahora, se obtiene la Figura 15. La dispersión de los datos se ajusta perfectamente a una regresión lineal a partir de la fracción de tiempo trabajado de 0.25 en adelante. Para ambas maquinas, la relación entre ensayos realizados y tiempo trabajado es directamente proporcional. La obtención de esta relación, junto con las capacidades máximas de las maquinas expuestas en la tabla 8 permite a el gerente anticipar si se llegará a cumplir con la demanda de ensayos.

Teniendo en cuenta los datos obtenidos a través de la regresión lineal, la ecuación 9 modela la cantidad de ensayos endurance realizados en función a la fracción de horas trabajadas en un año. Siendo la fracción de horas trabajadas descrita como la ecuación 11. A su vez, la ecuación 10 aplica la misma función pero para los ensayos rolling. Para un determinado objetivo de test realizados, las ecuaciones mencionadas pueden resolverse para obtener la cantidad de horas necesarias para llegar al objetivo. En el caso de requerir más horas de las que una jornada laboral normal dispone, se propone completar dichas horas sobrantes reposicionando uno de los técnicos en un turno adicional durante un periodo. Dicho periodo se puede calcular mediante la ecuación 13, de esta manera se aprovechan al máximo los recursos humanos disponibles. Para ello primeramente se calcula el máximo de turnos anuales mediante la ecuación 12.

$$TestRealizados = 934 \cdot 0,79 \cdot f_{ttrabajo} + 0,15 \quad (9)$$



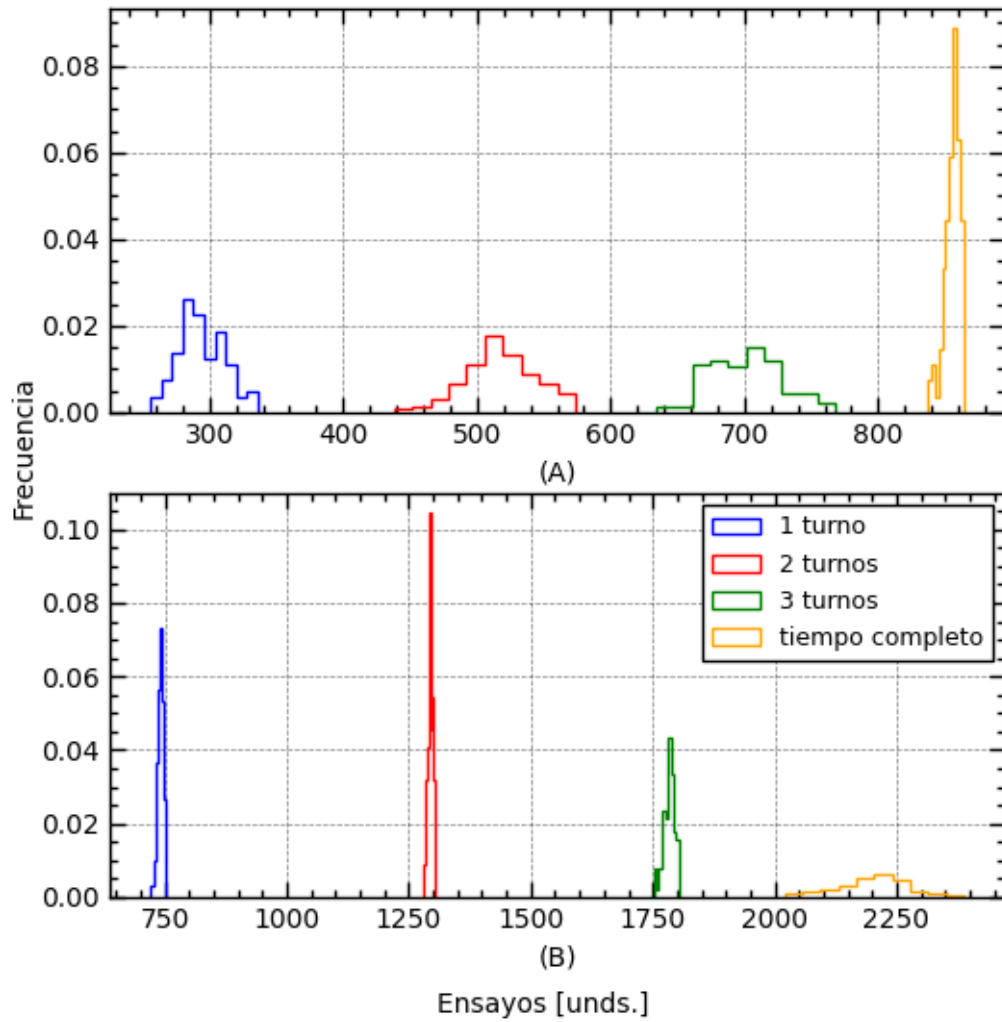


Figura 13: Distribución de ensayos realizados para los escenarios descritos en la leyenda. (A) Ensayos endurance. (B) Ensayos rolling.

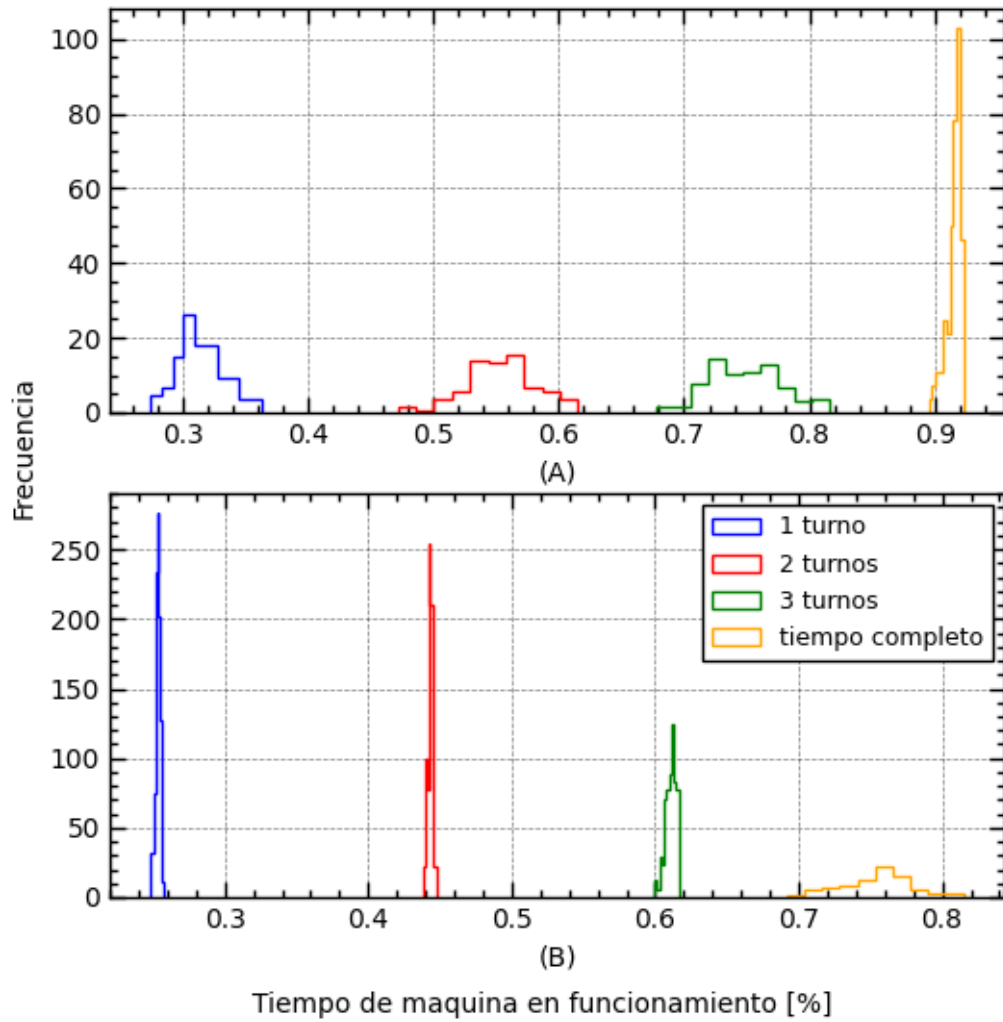


Figura 14: Distribución de el nivel de saturación en tiempo de lo escenarios descritos en la leyenda. (A) Ensayos endurance. (B) Ensayos rolling.

$$TestRealizados = 2502 \cdot 0,66 \cdot f_{ttrabajo} + 0,12 \quad (10)$$

$$f_{ttrabajo} = \frac{t_{trabajo}}{365 \frac{\text{días}}{\text{año}} \cdot 24 \frac{\text{horas}}{\text{días}}} \quad (11)$$

$$G(f_{ttrabajo}) = f_{ttrabajo} \frac{365 \text{ días} \cdot 24 \frac{\text{horas}}{\text{días}}}{8 \frac{\text{horas}}{\text{días}} \cdot 5 \frac{\text{días}}{\text{semanas}} \cdot 52 \frac{\text{semanas}}{\text{año}}} \quad (12)$$

$$Turnos = \begin{cases} 1 & G(f_{ttrabajo}) \leq 1 \\ 2 & 1 < G(f_{ttrabajo}) \leq 2 \\ 3 & 2 < G(f_{ttrabajo}) \leq 3 \\ 4 & G(f_{ttrabajo}) > 3 \end{cases}$$

$$Periodo = 12 \text{ meses} \cdot (G(f_{ttrabajo}) - Turnos - 1) \quad (13)$$

Con el fin de descartar otro tipo de configuraciones, como múltiples técnicos trabajando simultáneamente, se ha graficado la Figura 17. La diferencia en ambos ensayos parece mínima, pero con la intención de descartar la idea de que 2 técnicos trabajando simultáneamente incrementa el numero de ensayos realizados, se desarrolla el siguiente test de hipótesis.

- $H_1: \mu_1 \neq \mu_2$
- $H_0: \mu_1 = \mu_2$

Se ha tratado de descartar la hipótesis nula mediante la prueba de valor-p, y test t-student. Los resultados de la prueba, han sido recogidos en la tabla 10. El bajo valor-p obtenido en ambos tipos de ensayo descarta la hipótesis nula formulada anteriormente. Esto significa que las distribuciones tienen una diferencia estadísticamente significativa. A un siendo este el caso, el escenario en el que se trabaja a 2 turnos, es definitivamente superior, como se puede observar en la Figura 17. Al consumir ambos la misma cantidad de recursos, esto descarta la configuración en la que 2 técnicos trabajan simultáneamente.

Por ultimo, se ha tratado de observar la en el rendimiento de ensayos que aportaría la instalación de nuevas maquinas para el ensayo endurance. En la Figura 18, se

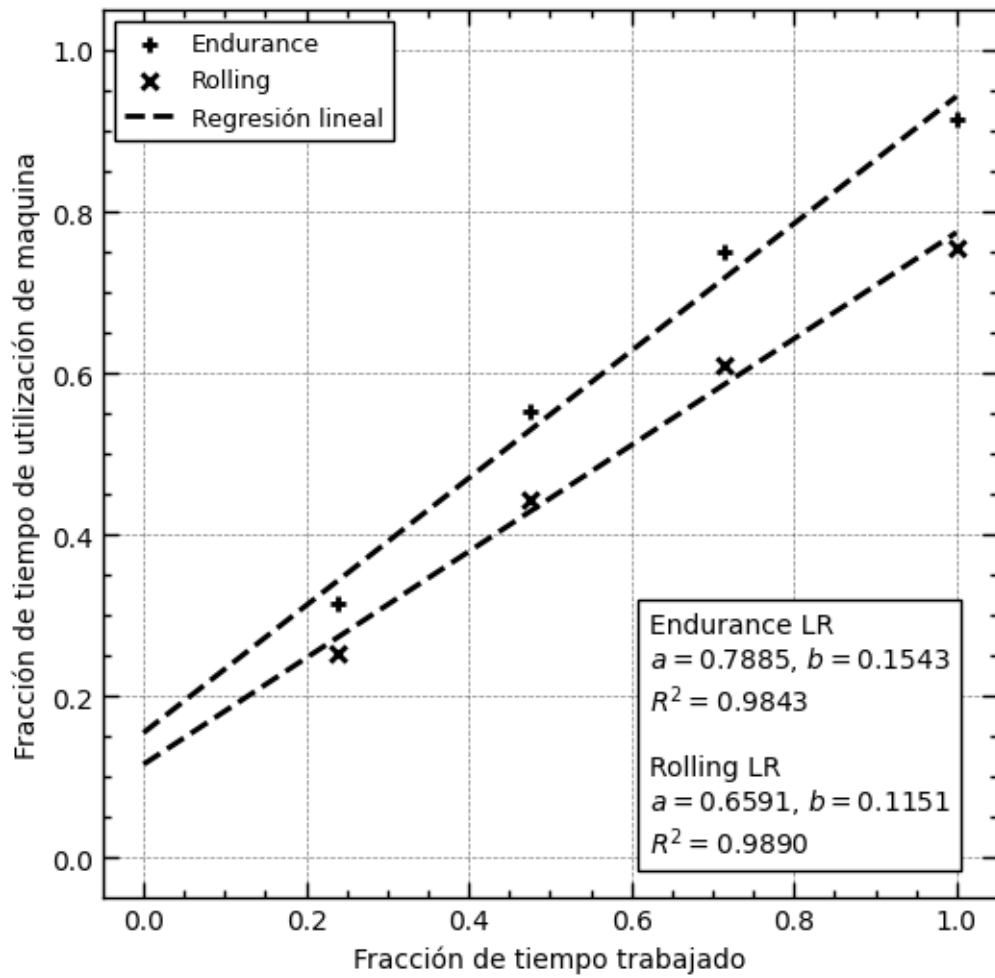


Figura 15: Tendencia proporcional del aumento del numero de ensayos al añadir turnos de trabajo.

Cuadro 10: Resultados del test estadístico formulado a partir de los resultados de la Figura 17

Resultados	Endurance	Rolling
valor-t	-2.279	-5.179
valor-p	0.023	0.000

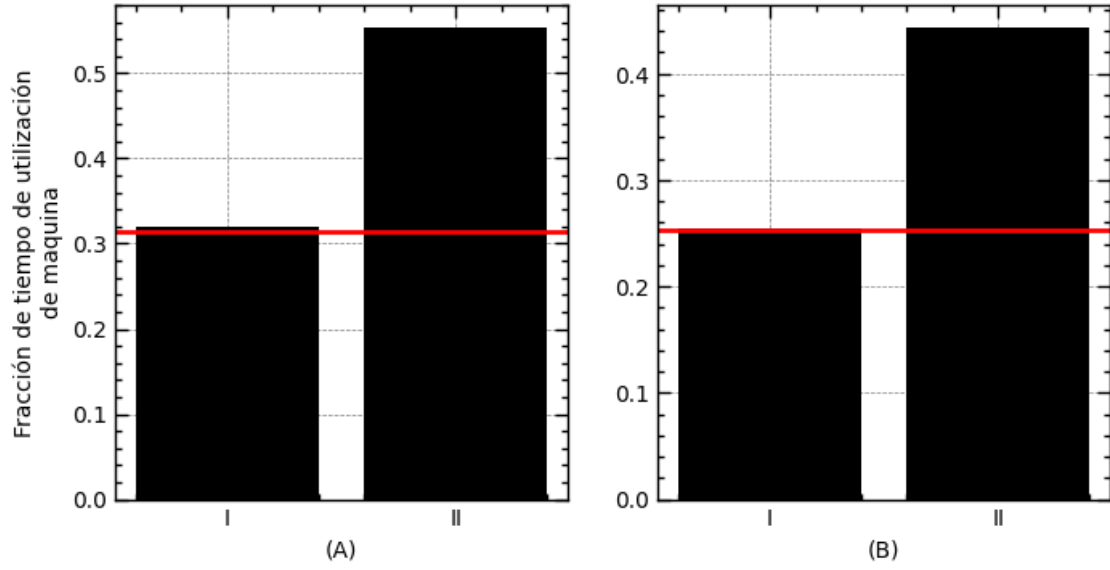


Figura 16: Comparación de el nivel de saturación en tiempo entre: [I] 2 técnicos trabajando simultáneamente, [II] 2 técnicos en distintos turnos. (A) Ensayos endurance. (B) Ensayos rolling.

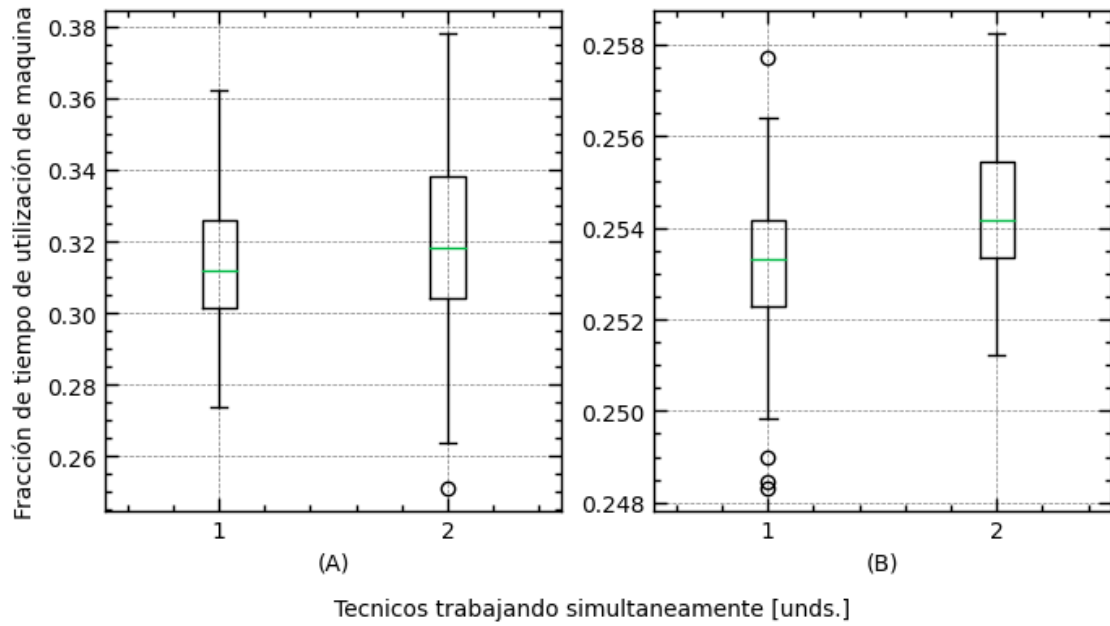


Figura 17: Comparación de el nivel de saturación en tiempo entre 1 técnico y 2 técnicos trabajando simultáneamente. (A) Ensayos endurance. (B) Ensayos rolling.

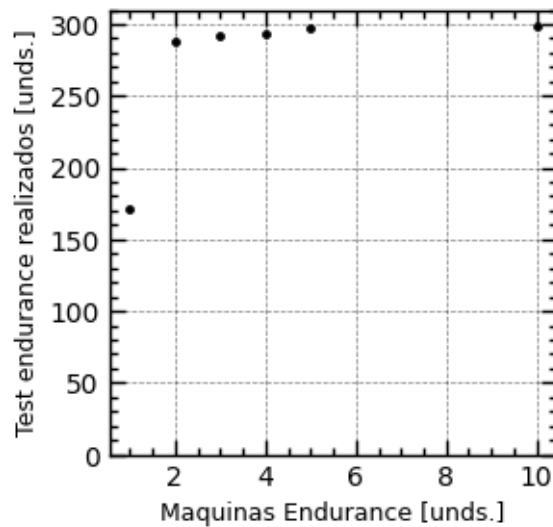


Figura 18: Aumento de los ensayos realizados en función de las maquinas endurance instaladas.

observa como añadir maquinas no aumenta de manera significativa la cantidad de ensayos realizados, ya que el técnico queda saturado de trabajo con la configuración actual.

Este análisis concluye, que en caso de necesitar capacidad adicional, se opte por reposicionar a uno de los técnicos en un turno adicional durante el periodo que sea necesario. Ya que, es con diferencia la opción mas optima.

## 5. CONCLUSIONES

A modo de cierre de este trabajo, se puede señalar que se llegó a lograr el objetivo planteado al comienzo del proyecto. Dentro de las posibilidades tomadas en cuenta, el escenario mas óptimo ha sido hallado. Mantener un enfoque hacia los hitos marcados al principio, ha ayudado en el extenso proceso de la elaboración de la simulación.

Entre los resultados más relevantes obtenidos, se encuentran la gran capacidad de mejora que posee el laboratorio en su estado actual. El descubrimiento de una tendencia proporcional entre la cantidad de horas trabajadas y el numero de ensayos realizados, ha sido particularmente útil. Mediante los resultados de

la regresión obtenida, se podrá facilitar la futura organización de los turnos de trabajo, y anticipar las desviaciones respecto a los objetivos de manera más precisa. El descarte de la ampliación de la maquinaria del laboratorio, supondrá un ahorro tanto en espacio como en inversión de la fabrica.

Respecto al proceso, se han encontrado numerosas dificultades. Primeramente, la familiarización con las DES, ha requerido un extenso trabajo de investigación acerca de sus métodos e implementaciones. Se ha requerido ajustar este método de simulación a los objetivos del trabajo, lo que ha supuesto un periodo previo al desarrollo, en el que se han valorado múltiples enfoques. La tarea más complicada, ha resultado ser el proceso de traducción del modelo desde los diagramas, a el código fuente. Este periodo, ha durado semanas, en las cuales se ha aprendido cero a usar la librería Simpy. El proyecto, al involucrar tantas lineas de código, ha supuesto un proceso complicado de detección y corrección de errores de funcionamiento, que ha limitado lo que podría haber sido el análisis final. Aún así, el análisis de los resultados ha resultado satisfactorio, para un Trabajo de Fin de Grado.

Este trabajo ha causado, de manera transversal, un entendimiento aún mas detallado del proceso de lo que se tenía en un inicio. Además, los conocimientos obtenidos en este TFG, podrán ser útiles en futuros procesos de optimización en producción.

## A. ANEXO I: CÓDIGO FUENTE

### A.1. EXPOSICIÓN DE UN CASO FICTICIO DE UNA DES

```
# -*- coding: utf-8 -*-
"""
Created on Thu Oct 6 17:37:23 2022

@author: andoi
"""

import simpy
import random
import pandas as pd
from numpy import random as nprng
import csv

rng = nprng.default_rng()

# Global class cantaining the system common variables

class g:
    DURATION = 960 # Duration of the simulation in minutes (16 hours)
    RUNS = 20

    GAS_STATION_QTY = 4 # units
    PARKING_CAPACITY = 100 # units
    P_REFILL = 0.5 # Chance a car might refill it's tank at the gas station

    DRIVE_LONG = (8, 3) # (Shape 'k', Scale 'theta') min. Gamma distribution.
    DRIVE_SHORT = (2, 3, 4) # (a, m, b) min. Triangular distribution.
    SEARCH_PARKING = (10, 1) # (k, theta) min. Gamma distribution.
    REFILL = (4, 5, 8) # (a, m, b) min. Triangular distribution.
    INTER_ARRIVAL = (0.5, 5) # exponential interarrival times for (A, B) pop.

    PARKING_A = (560, 60) # (Mean, S.Dev) in minutes. Normal distribution.
    PARKING_B = (5, 0.7) # (Mean, Sigma) in minutes. Lognormal distribution.
```



```

df = pd.DataFrame(columns=['RUN',
                           'CAR_TYPE',
                           'HAS_PARKED',
                           'HAS_REFILLED',
                           'A', 'B', 'C',
                           'D', 'E', 'F',
                           'G', 'H', 'I',
                           'J', 'K',
                           'T_DRIVE_LONG',
                           'Q_REFILL',
                           'T_REFILL',
                           'T_DRIVE_SHORT',
                           'T_SEARCH_PARKING',
                           'T_PARKED'])

# Defines attributes of car class, the dynamic entity moving through the model.

class car:

    def __init__(self, car_id, driver_type, has_parked, has_refilled):
        self.id = car_id
        self.type = driver_type # Either worker (A) or casual (B).
        self.has_parked = has_parked
        self.has_refilled = has_refilled

# Contains the simulation model itself.

class example_model:
    def __init__(self, number_of_runs):
        self.env = simpy.Environment()

        self.car_counter = 0
        self.parked_cars = 0

        self.number_of_runs = number_of_runs

        # Defining the resources
        self.res_gs = simpy.Resource(self.env, capacity=g.GAS_STATION_QTY)

```

```

# Defining data structures
self.tmp_lst = []

def car_generator_A(self):
    while self.env.now < 120:
        car_gen = car(self.car_counter, 'A', True, False)
        self.car_counter += 1

        self.env.process(self.main_proc(car_gen))

        sampled_interarrival = random.expovariate(1.0/g.INTER_ARRIVAL[0])
        yield self.env.timeout(sampled_interarrival)

def car_generator_B(self):
    while True:
        car_gen = car(self.car_counter, 'B', True, False)
        self.car_counter += 1

        self.env.process(self.main_proc(car_gen))

        sampled_interarrival = random.expovariate(1.0/g.INTER_ARRIVAL[1])
        yield self.env.timeout(sampled_interarrival)

def main_proc(self, car_ent):
    tmp_dict = {"RUN": self.number_of_runs}
    tmp_dict["CAR_TYPE"] = car_ent.type
    tmp_dict["A"] = self.env.now

    drive_long_time = rng.gamma(g.DRIVE_LONG[0], g.DRIVE_LONG[1])
    yield self.env.timeout(drive_long_time)

    tmp_dict["B"] = self.env.now

    if random.uniform(0, 1) > 0.5:
        car_ent.has_refilled = True
        refill_time = rng.triangular(g.REFILL[0], g.REFILL[1], g.REFILL[2])
        with self.res_gs.request() as req_gs:
            yield req_gs
            tmp_dict["C"] = self.env.now

```

```

        yield self.env.timeout(refill_time)
    else:
        tmp_dict["C"] = self.env.now

    tmp_dict["D"] = self.env.now

    drive_short_time = rng.triangular(g.DRIVE_SHORT[0], g.DRIVE_SHORT[1],
                                       g.DRIVE_SHORT[2])
    yield self.env.timeout(drive_short_time)

    tmp_dict["E"] = self.env.now

    if self.parked_cars >= g.PARKING_CAPACITY:
        car_ent.has_parked = False
        search_parking_time = rng.gamma(g.SEARCH_PARKING[0], g.SEARCH_PARKING[1])
        yield self.env.timeout(search_parking_time)
    else:
        self.parked_cars += 1

    tmp_dict["F"] = self.env.now

    if car_ent.type == "A":
        parking_time = rng.normal(g.PARKING_A[0], g.PARKING_A[1])
    else:
        parking_time = rng.lognormal(g.PARKING_B[0], g.PARKING_B[1])

    yield self.env.timeout(parking_time)

    if car_ent.has_parked:
        self.parked_cars -= 1

    tmp_dict["G"] = self.env.now

    drive_short_time = rng.triangular(g.DRIVE_SHORT[0], g.DRIVE_SHORT[1],
                                       g.DRIVE_SHORT[2])
    yield self.env.timeout(drive_short_time)

    tmp_dict["H"] = self.env.now

    if random.uniform(0,1) > 0.5:

```

```

        car_ent.has_refilled = True
        refill_time = rng.triangular(g.REFILL[0], g.REFILL[1], g.REFILL[2])
        with self.res_gs.request() as req_gs:
            yield req_gs
            tmp_dict["I"] = self.env.now
            yield self.env.timeout(refill_time)
    else:
        tmp_dict["I"] = self.env.now

    tmp_dict["J"] = self.env.now

    yield self.env.timeout(drive_long_time)

    tmp_dict["K"] = self.env.now
    tmp_dict["HAS_PARKED"] = car_ent.has_parked
    tmp_dict["HAS_REFILLED"] = car_ent.has_refilled

    self.tmp_lst.append(tmp_dict)

def run(self):
    self.env.process(self.car_generator_A())
    self.env.process(self.car_generator_B())

    self.env.run(until=g.DURATION)
    tmp_df = pd.DataFrame(self.tmp_lst)
    g.df = pd.concat([g.df, tmp_df], ignore_index=True)
    g.df['T_DRIVE_LONG'] = g.df['B'] - g.df['A'] + g.df['K'] - g.df['J']
    g.df['Q_REFILL'] = g.df['C'] - g.df['B'] + g.df['I'] - g.df['H']
    g.df['T_REFILL'] = g.df['D'] - g.df['C'] + g.df['J'] - g.df['I']
    g.df['T_DRIVE_SHORT'] = g.df['E'] - g.df['D'] + g.df['H'] - g.df['G']
    g.df['T_SEARCH_PARKING'] = g.df['F'] - g.df['E']
    g.df['T_PARKED'] = g.df['G'] - g.df['F']
    g.df.to_csv('out.csv', index=False)

for run in range(g.RUNS):
    print("Run ", run+1, "of ", g.RUNS, sep=" ")
    my_example = example_model(run)
    my_example.run()

```

```
print()
```

## A.2. CÓDIGO FUENTE DE LA SIMULACIÓN PRINCIPAL

```
# *****#
#
#
#          :::::::::: ::::::::::#
#  lep_sim_rewrited.py          :+      :+      :+      #
#
#          ++      ++      ++      #
#  By: andoitzcp <andoitzcp@gmail.com>      ++      ++      ++      #
#
#          ++      ++      #
#
#  Created: 2023/02/01 01:09:17 by andoitzcp      ##      ##      ##      #
#  Updated: 2023/02/01 01:09:31 by acampo-p      ###      #####.fr      #
#
# *****#
```

```
import simpy
import pandas as pd
from numpy import random as nprng
from functools import reduce
```

```
rng = nprng.default_rng()
```

```
class G:
    SIM_DET = {
        'SIM_DUR': 365 * 24 * 60,
        'SHIFT_DUR': 8 * 60,
        'SHIFT_N': 1,
        'RUN_EXEC': 100,
    }
    P_QCED = {
        'INTR_RVAL': 20 * 60 / 10,
        'R_PRIO': 0,
        'M_MEAN': 45,
        'M_SDEV': 10,
        'M_PRIO': -1,
        'C_DUR': 3 * 60,
        'C_PRIO': -3,
        'P_DUR': 5,
        'P_PRIO': -4,
```

```

        'T_MEAN': 72 * 60,
        'T_SDEV': 6 * 60,
    }
P_RR = {
    'INTR_RVAL': 8 * 60 / 10,
    'R_PRIO': -1,
    'M_MEAN': 30,
    'M_SDEV': 5,
    'M_PRIO': -2,
    'C_DUR': 6 * 60,
    'C_PRIO': -2,
    'T_POS': [150, 170, 200],
    'T_PROB': [0.1, 0.7, 0.2],
}
RSRCS = {
    'INDOOR': 4,
    'RRM': 1,
    'BUTLR': 1,
    'TECHN': 1,
    'MANT': 1,
    'OPR': 1,
    'Q_RIMS': 14,
    'R_RIMS': 20,
}
P_RIM = {
    'PRIO': 0,
    'A': 12,
    'M': 15,
    'B': 20,
}

```

```

class data:
    df = pd.DataFrame()

    def calc_intervals(df):
        df['Q_GEN'] = df['RIM1'] - df['GEN']
        df['Q_RIM'] = df['RIM2'] - df['RIM1']
        df['D_RIM'] = df['RIM3'] - df['RIM2']
        df['Q_MOUNT'] = df['MOUNT1'] - df['RIM3']

```

```

df['T_MOUNT'] = df['MOUNT2'] - df['MOUNT1']
df['D_MOUNT'] = df['MOUNT3'] - df['MOUNT2']
df['Q_COND'] = df['COND1'] - df['MOUNT3']
df['T_COND'] = df['COND2'] - df['COND1']
df['D_COND'] = df['COND3'] - df['COND2']
df['Q_PMEAS'] = df['PMEAS1'] - df['COND3']
df['T_PMEAS'] = df['PMEAS2'] - df['PMEAS1']
df['D_PMEAS'] = df['PMEAS3'] - df['PMEAS2']
df['Q_QCED'] = df['QCED1'] - df['PMEAS3']
df['D_QCED'] = df['QCED2'] - df['QCED1']

df['Q_GEN_h'] = df['Q_GEN'] / 60
df['Q_RIM_h'] = df['Q_RIM'] / 60
df['D_RIM_h'] = df['D_RIM'] / 60
df['Q_MOUNT_h'] = df['Q_MOUNT'] / 60
df['T_MOUNT_h'] = df['T_MOUNT'] / 60
df['D_MOUNT_h'] = df['D_MOUNT'] / 60
df['Q_COND_h'] = df['Q_COND'] / 60
df['T_COND_h'] = df['T_COND'] / 60
df['D_COND_h'] = df['D_COND'] / 60
df['Q_PMEAS_h'] = df['Q_PMEAS'] / 60
df['T_PMEAS_h'] = df['T_PMEAS'] / 60
df['D_PMEAS_h'] = df['D_PMEAS'] / 60
df['Q_QCED_h'] = df['Q_QCED'] / 60
df['D_QCED_h'] = df['D_QCED'] / 60

def stat_calc(df):
    filtered_df = df.loc[df['D_QCED_h'] > 0]
    df2 = filtered_df.groupby(['TEST'])['TEST'].count()
    print(df2)

class tire:

    def __init__(self, t_id, t_test, t_client):
        self.id = t_id
        self.test = t_test
        self.client = t_client

class lab_model:

```



```

def __init__(self, run_nbr):
    self.env = simpy.Environment()
    self.run_nbr = run_nbr
    self.tire_cntr = 0
    self.tire_bufrr_indoor = []
    self.tire_bufrr_rr = []

    # Resources
    self.rsrc_indoor = \
        simpy.Resource(self.env, capacity=G.RSRCS['INDOOR'])
    self.rsrc_rrm = \
        simpy.Resource(self.env, capacity=G.RSRCS['RRM'])
    self.rsrc_butlr = \
        simpy.Resource(self.env, capacity=G.RSRCS['BUTLR'])
    self.rsrc_techn = \
        simpy.PreemptiveResource(self.env, capacity=G.RSRCS['TECHN'])
    self.rsrc_mant = \
        simpy.Resource(self.env, capacity=G.RSRCS['MANT'])
    self.rsrc_opr = \
        simpy.Resource(self.env, capacity=G.RSRCS['OPR'])
    self.rsrc_qrims = \
        simpy.Resource(self.env, capacity=G.RSRCS['Q_RIMS'])
    self.rsrc_rrims = \
        simpy.Resource(self.env, capacity=G.RSRCS['R_RIMS'])

    # Temporal storage for data
    self.tmp_lst_gen = []
    self.tmp_lst_rim = []
    self.tmp_lst_mount = []
    self.tmp_lst_cond = []
    self.tmp_lst_pmeasure = []
    self.tmp_lst_test = []

def gen_qced_rp(self):
    self.env.timeout(32.59)
    while True:
        tmp_dict = {'RUN': self.run_nbr,
                    'TEST': 'QCED',
                    'CLIENT': 'RP',

```

```

    }

    self.tire_cntr += 1
    obj = tire(self.tire_cntr, "QCED", "RP")
    self.tire_bufr_indoor.append(obj)
    intr_rval = rng.exponential(G.P_QCED['INTR_RVAL'])

    tmp_dict['ID'] = self.tire_cntr
    tmp_dict['GEN'] = self.env.now
    self.tmp_lst_gen.append(tmp_dict)
    self.env.process(self.proc_master_indoor())
    yield self.env.timeout(intr_rval)

def gen_rr_rp(self):
    while True:
        tmp_dict = {'RUN': self.run_nbr,
                    'TEST': 'RR',
                    'CLIENT': 'RP',
                    }

        self.tire_cntr += 1
        obj = tire(self.tire_cntr, "RR", "RP")
        self.tire_bufr_rr.append(obj)
        intr_rval = rng.exponential(G.P_RR['INTR_RVAL'])

        tmp_dict['ID'] = self.tire_cntr
        tmp_dict['GEN'] = self.env.now
        self.tmp_lst_gen.append(tmp_dict)
        self.env.process(self.proc_master_rr())
        yield self.env.timeout(intr_rval)

def obstruct_techn(self):
    day = 0
    week = ['M', 'T', 'X', 'R', 'F', 'S', 'U']
    t_up = G.SIM_DET['SHIFT_N'] * G.SIM_DET['SHIFT_DUR']
    while True:
        t_down = (24 * 60) - t_up
        if week[day % 7] == 'F':
            day += 3
            t_down += 48 * 60

```

```

        else:
            day += 1
            yield self.env.timeout(t_up)
            with self.rsrc_techn.request(priority=-10, preempt=True) \
                as req_techn_A:
                yield req_techn_A
            yield self.env.timeout(t_down)

def obstruct_indoor(self):
    while True:
        t_tofailure = rng.exponential(G.F_DET['INTR'])
        yield self.env.timeout(t_tofailure)
        if not self.broken:
            self.proc_qced_A.interruptp()

def t_conditioning(self, t_left):
    t_toshiftend = (G.SIM_DET['SHIFT_DUR'] * G.SIM_DET['SHIFT_N']) - self.env.r
    if G.SIM_DET['SHIFT_N'] == 3:
        return t_left
    elif t_left > t_toshiftend:
        t_left = G.SIM_DET['SHIFT_DUR'] * (3 - G.SIM_DET['SHIFT_N'])
        t_left += t_toshiftend
    return t_left

def proc_master_indoor(self):
    yield (self.env.timeout(2*24*60))
    if len(self.tire_buf indoor) < 2:
        pass
    else:
        qced_sample_A = self.tire_buf indoor.pop(0)
        qced_sample_B = self.tire_buf indoor.pop(0)
        with self.rsrc_qgrims.request() as req_qgrims:
            yield req_qgrims
            yield self.env.process(self.proc_rim(qced_sample_A))
            yield self.env.process(self.proc_rim(qced_sample_B))
        with self.rsrc_indoor.request() as req_indoor:
            yield req_indoor
            yield self.env.process(self.proc_mount(qced_sample_A))
            yield self.env.process(self.proc_mount(qced_sample_B))
            yield self.env.process(self.proc_cond_qced(qced_sample_A,

```

```

                                                                 qced_sample_B))
        yield self.env.process(
            self.proc_pmeasure(qced_sample_A,
                               qced_sample_B))
        yield self.env.process(
            self.proc_qced_A(qced_sample_A)) & \
            self.env.process(
                self.proc_qced_B(qced_sample_B))

def proc_master_rr(self):
    if len(self.tire_bufrr) > 0:
        rr_sample = self.tire_bufrr.pop(0)
        with self.rsrc_qgrims.request() as req_qgrims:
            yield req_qgrims
            yield self.env.process(self.proc_rim(rr_sample))
            yield self.env.process(self.proc_cond_rr(rr_sample))
        with self.rsrc_rrm.request() as req_rrm:
            yield req_rrm
            yield self.env.process(self.proc_mount(rr_sample))
            yield self.env.process(self.proc_rr(rr_sample))

def proc_rim(self, tire_ent):
    tmp_dict = {'ID': tire_ent.id}

    aux = [self.env.now]
    prio = G.P_RIM['PRIO']
    t_left = rng.triangular(G.P_RIM['A'], G.P_RIM['M'], G.P_RIM['B'])
    while t_left > 0:
        with self.rsrc_techn.request(priority=prio,
                                     preempt=False) as req_techn_rim, \
            self.rsrc_butlr.request() as req_butlr:
            try:
                yield req_techn_rim & req_butlr
                aux.append(self.env.now)
                yield self.env.timeout(t_left)
                t_left = 0
            except simpy.Interrupt as interrupt:
                usage = self.env.now - interrupt.cause.usage_since
                t_left -= usage
                prio -= 0.1

```

```

tmp_dict['RIM1'] = aux[0]
tmp_dict['RIM2'] = aux[1]
tmp_dict['RIM3'] = self.env.now
self.tmp_lst_rim.append(tmp_dict)

def proc_mount(self, tire_ent):
    tmp_dict = {'ID': tire_ent.id}

    start = [self.env.now]
    if tire_ent.test == 'QCED':
        t_left = rng.normal(G.P_QCED['M_MEAN'], G.P_QCED['M_SDEV'])
        prio = G.P_QCED['M_PRIO']
    elif tire_ent.test == 'RR':
        t_left = rng.normal(G.P_RR['M_MEAN'], G.P_RR['M_SDEV'])
        prio = G.P_RR['M_PRIO']
    while t_left > 0:
        with self.rsrc_techn.request(priority=prio, preempt=False) \
            as req_techn_mount:
            yield req_techn_mount
            start.append(self.env.now)
        try:
            yield self.env.timeout(t_left)
            t_left = 0
        except simpy.Interrupt as interrupt:
            usage = self.env.now - interrupt.cause.usage_since
            t_left -= usage
            prio -= 0.1

    tmp_dict['MOUNT1'] = start[0]
    tmp_dict['MOUNT2'] = start[1]
    tmp_dict['MOUNT3'] = self.env.now
    self.tmp_lst_mount.append(tmp_dict)

def proc_cond_qced(self, tire_ent_A, tire_ent_B):
    tmp_dict_A = {'ID': tire_ent_A.id}
    tmp_dict_B = {'ID': tire_ent_B.id}

    prio = G.P_QCED['C_PRIO']
    start = [self.env.now]

```

```

with self.rsrc_techn.request(priority=prio, preempt=False) \
    as req_techn_cond_qced:
    yield req_techn_cond_qced
start.append(self.env.now)
t_left = self.t_conditioning(G.P_QCED['C_DUR'])
yield self.env.timeout(t_left)

tmp_dict_A['COND1'] = start[0]
tmp_dict_B['COND1'] = start[0]
tmp_dict_A['COND2'] = start[1]
tmp_dict_B['COND2'] = start[1]
tmp_dict_A['COND3'] = self.env.now
tmp_dict_B['COND3'] = self.env.now
self.tmp_lst_cond.append(tmp_dict_A)
self.tmp_lst_cond.append(tmp_dict_B)

def proc_cond_rr(self, tire_ent):
    tmp_dict = {'ID': tire_ent.id}

    prio = G.P_RR['C_PRIO']
    aux = [self.env.now]
    with self.rsrc_techn.request(priority=prio, preempt=False) \
        as req_techn_cond_rr:
        yield req_techn_cond_rr
    aux.append(self.env.now)
    t_left = G.P_RR['C_DUR']
    yield self.env.timeout(t_left)

    tmp_dict['COND1'] = aux[0]
    tmp_dict['COND2'] = aux[1]
    tmp_dict['COND3'] = self.env.now
    self.tmp_lst_cond.append(tmp_dict)

def proc_pmeasure(self, tire_ent_A, tire_ent_B):
    tmp_dict_A = {'ID': tire_ent_A.id}
    tmp_dict_B = {'ID': tire_ent_B.id}

    start = [self.env.now]
    start.append(self.env.now)
    yield self.env.timeout(G.P_QCED['P_DUR'])

```

```

tmp_dict_A['PMEAS1'] = start[0]
tmp_dict_A['PMEAS2'] = start[1]
tmp_dict_A['PMEAS3'] = self.env.now
tmp_dict_B['PMEAS1'] = start[0]
tmp_dict_B['PMEAS2'] = start[1]
tmp_dict_B['PMEAS3'] = self.env.now
self.tmp_lst_pmeasure.append(tmp_dict_A)
self.tmp_lst_pmeasure.append(tmp_dict_B)

def proc_qced_A(self, tire_ent):
    tmp_dict = {'ID': tire_ent.id}
    tmp_dict['QCED1'] = self.env.now

    test_failure = False
    t_left = rng.normal(G.P_QCED['T_MEAN'], G.P_QCED['T_SDEV'])
    try:
        yield self.env.timeout(t_left)
    except simpy.Interrupt:
        test_failure = True

    tmp_dict['TEST_FAIL'] = test_failure
    tmp_dict['QCED2'] = self.env.now
    self.tmp_lst_test.append(tmp_dict)

def proc_qced_B(self, tire_ent):
    tmp_dict = {'ID': tire_ent.id}
    tmp_dict['QCED1'] = self.env.now

    test_failure = False
    t_left = rng.normal(G.P_QCED['T_MEAN'], G.P_QCED['T_SDEV'])
    try:
        yield self.env.timeout(t_left)
    except simpy.Interrupt:
        test_failure = True

    tmp_dict['TEST_FAIL'] = test_failure
    tmp_dict['QCED2'] = self.env.now
    self.tmp_lst_test.append(tmp_dict)

```

```

def proc_rr(self, tire_ent):
    tmp_dict = {'ID': tire_ent.id}
    tmp_dict['QCED1'] = self.env.now

    test_failure = False
    t_left = rng.normal(180, 20)
    try:
        yield self.env.timeout(t_left)
    except simpy.Interrupt:
        test_failure = True
    tmp_dict['TEST_FAIL'] = test_failure
    tmp_dict['QCED2'] = self.env.now
    self.tmp_lst_test.append(tmp_dict)

def df_gen(self):
    df_tmp_gen = pd.DataFrame(self.tmp_lst_gen)
    df_tmp_rim = pd.DataFrame(self.tmp_lst_rim)
    df_tmp_mount = pd.DataFrame(self.tmp_lst_mount)
    df_tmp_cond = pd.DataFrame(self.tmp_lst_cond)
    df_tmp_pmeasure = pd.DataFrame(self.tmp_lst_pmeasure)
    df_tmp_test = pd.DataFrame(self.tmp_lst_test)
    print(df_tmp_gen)

    df_tmp_lst = [df_tmp_gen,
                  df_tmp_rim,
                  df_tmp_mount,
                  df_tmp_cond,
                  df_tmp_pmeasure,
                  df_tmp_test]
    df_tmp_master = reduce(lambda left, right:
                           pd.merge(left, right, on=['ID'], how='outer'),
                           df_tmp_lst)
    data.df = pd.concat([data.df, df_tmp_master], ignore_index=True)

def run(self):
    self.env.process(self.obstruct_techn())
    self.env.process(self.gen_qced_rp())
    self.env.process(self.gen_rr_rp())
    self.env.run(until=G.SIM_DET['SIM_DUR'])

```



```

        self.df_gen()

for run in range(G.SIM_DET['RUN_EXEC']):
    print("Run ", run+1, "of ", G.SIM_DET['RUN_EXEC'], sep="")
    print(G.SIM_DET['RUN_EXEC'])
    my_test_model = lab_model(run)
    my_test_model.run()
    print()

data.calc_intervals(data.df)
data.df.to_csv('lep.csv', index=False)
df = data.df
print(df)
print(data.stat_calc(df))

```

## B. ANEXO II: DIAGRAMAS DE FLUJO

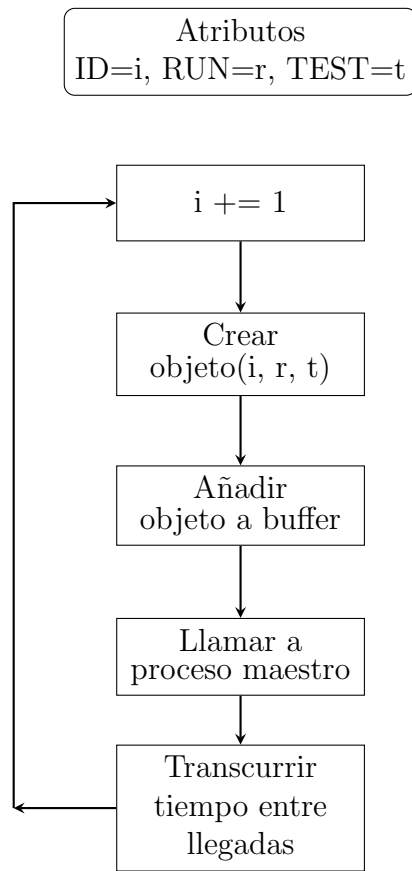


Figura 19: Modelo de generación de entidades

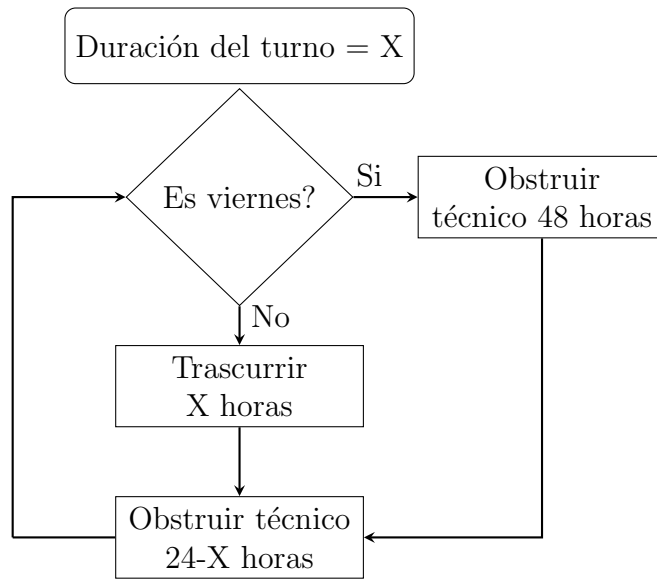


Figura 20: Modelo de obstrucción del técnico durante horas no activas

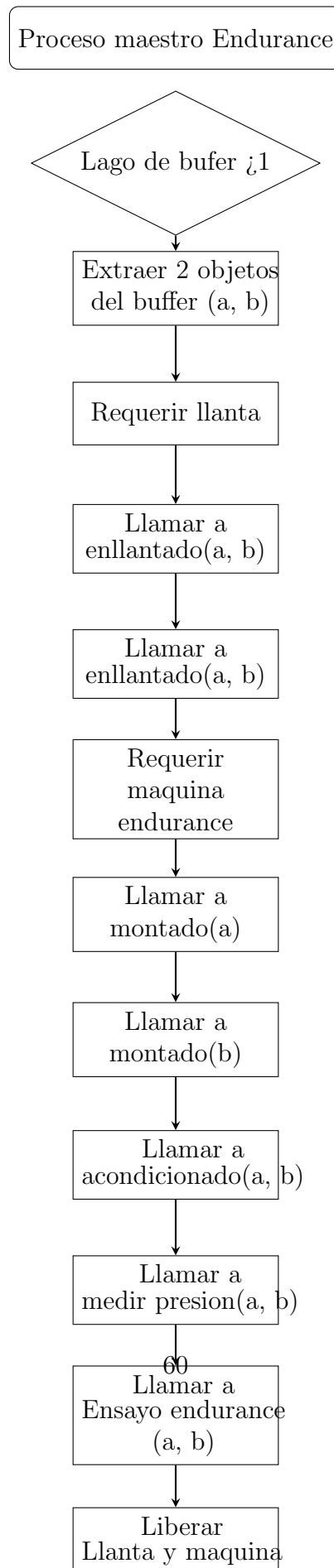


Figura 21: Modelo de flujo para el proceso maestro Endurance

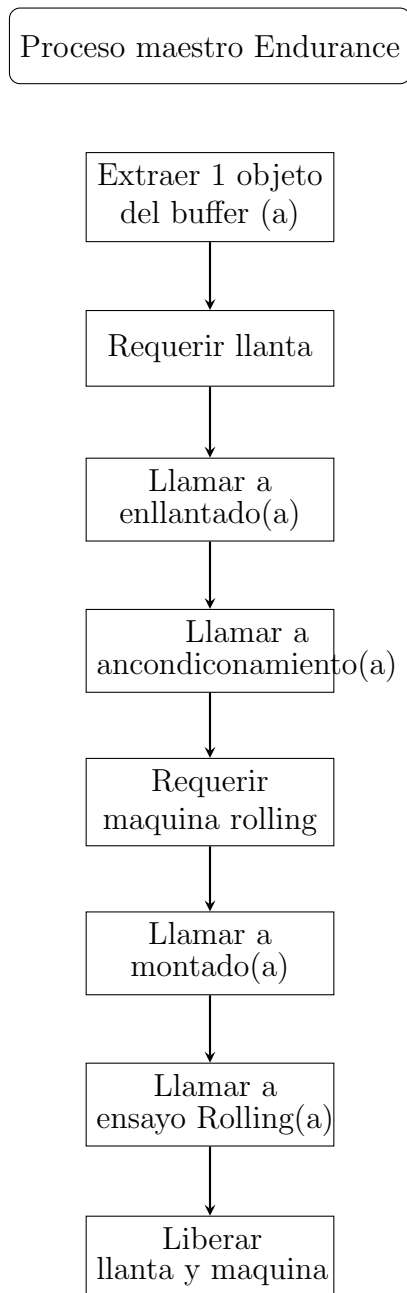


Figura 22: Modelo del proceso maestro de ensayos Endurance

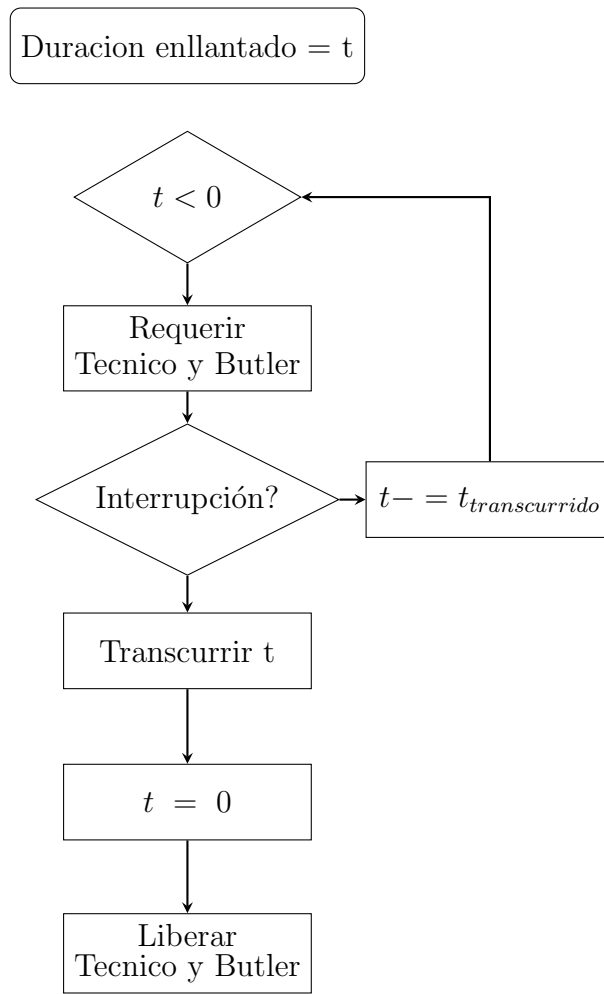


Figura 23: Modelo del subproceso de enlanchado

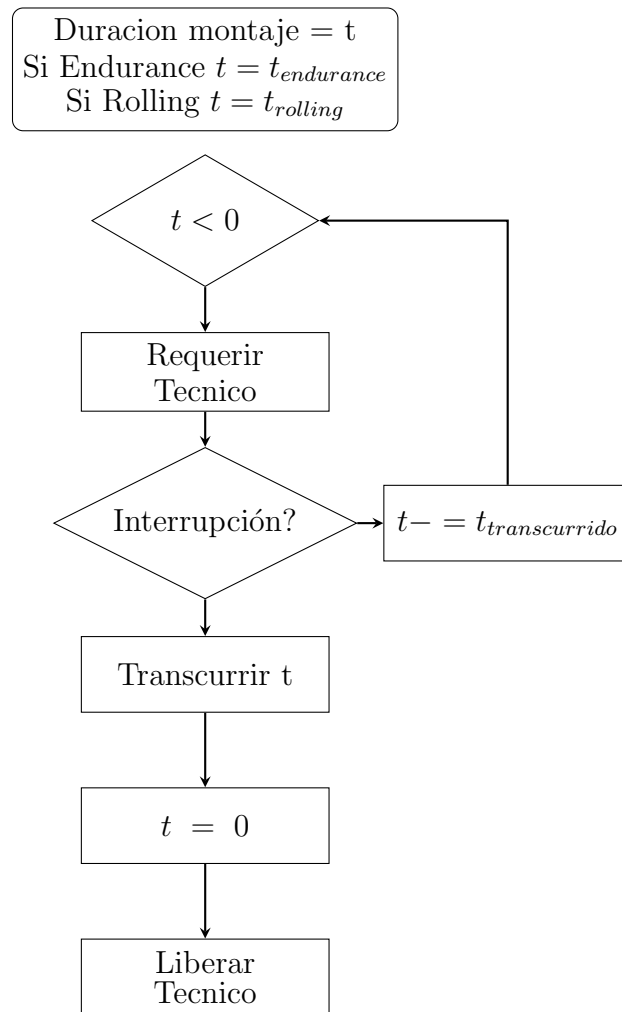


Figura 24: Modelo del subproceso de montado de máquina

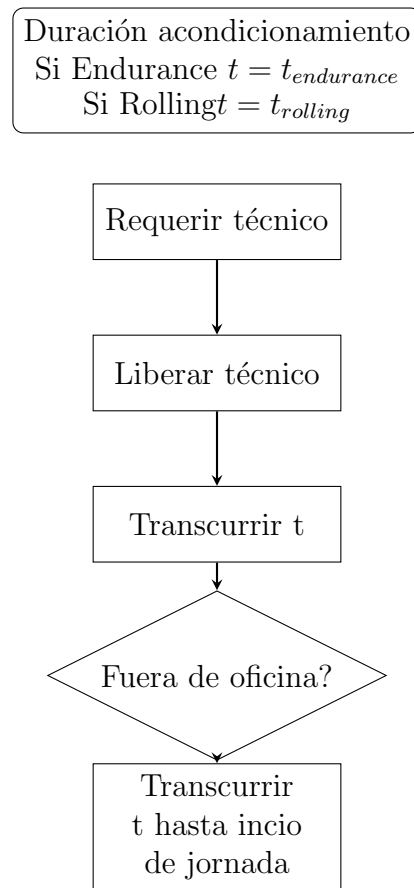


Figura 25: Modelo del subproceso de acondicionado de sala



Duración ajuste de presión = t

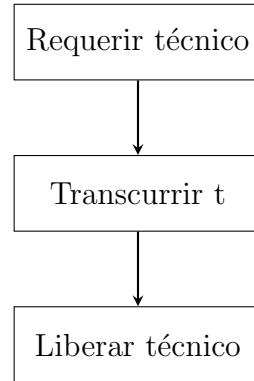


Figura 26: Modelo del subproceso de ajuste de presión de la cubierta

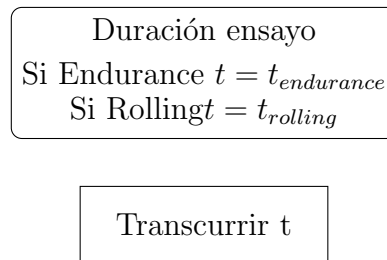


Figura 27: Modelo del subproceso de ensayo de la cubierta

## Referencias

- Abu-Taieh, E.M., El Sheikh, A.A.R., Abu-Tayeh, J.M., Al Abdallat, H., 2009. History of simulation, in: Encyclopedia of Information Science and Technology, Second Edition. IGI Global, pp. 1769–1776.
- Allen, T.T., 2011. Introduction to discrete event simulation and agent-based modeling: voting systems, health care, military, and manufacturing. Springer Science & Business Media.
- Banks, J., 1998. Handbook of simulation: principles, methodology, advances, applications, and practice. John Wiley & Sons.
- Banks, J., 1999. Introduction to simulation, in: Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1, pp. 7–13.
- Chicu, N., Prioteasa, A.L., Deaconu, A., 2020. Current trends and perspectives in tyre industry. Studia Universitatis Vasile Goldiş Arad, Seria Ştiinţe Economice 30, 36–56.
- Garzia, R.F., Garzia, M.R., Zeigler, B.P., 1986. Discrete-event simulation: When prototyping of large, complex systems is impossible, this technique, based on known behaviors of interacting elements, is useful. IEEE Spectrum 23, 32–36.
- Martin, A.J., 2022. Introduction to hypothesis testing .
- Meester, R., 2008. A natural introduction to probability theory. Springer Science & Business Media.
- Pritsker, A.A.B., 1974. The Gasp Iv Simulation Language [By] A. Alan B. Pritsker. Wiley.
- Rodgers, B., 2020. Tire Engineering: An Introduction. CRC Press.
- Shannon, R., Johannes, J.D., 1976. Systems simulation: the art and science. IEEE transactions on systems, man, and cybernetics , 723–724.
- Simon, M.K., 2002. Probability distributions involving Gaussian random variables: A handbook for engineers and scientists. Springer.
- Varga, A., 2001. Discrete event simulation system, in: Proc. of the European Simulation Multiconference (ESM'2001), pp. 1–7.