



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Modelo para la simulación de la producción de calzado masculino hecho a mano en Bogotá, Colombia

Luis Gabriel Diaz Fonseca

Universidad Nacional de Colombia
Facultad de Minas,
Departamento de ciencia de la computación y la decisión
Medellín, Colombia
2023

Modelo para la simulación de la producción de calzado masculino hecho a mano en Bogotá, Colombia

Luis Gabriel Diaz Fonseca

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título
de:

Magister en Ingeniería Analítica

Director (a):

Ph.D Fernán Alonso Villa Garzón

Línea de Investigación:

Analítica

Universidad Nacional de Colombia

Facultad de Minas,

Departamento de ciencia de la computación y la decisión

Medellín, Colombia

2023

"El éxito es la capacidad de ir de un fracaso a otro sin perder tu entusiasmo."

Winston Churchill

Declaración de obra original

Yo declaro lo siguiente:

He leído el Acuerdo 035 de 2003 del Consejo Académico de la Universidad Nacional. «Reglamento sobre propiedad intelectual» y la Normatividad Nacional relacionada al respeto de los derechos de autor. Esta disertación representa mi trabajo original, excepto donde he reconocido las ideas, las palabras, o materiales de otros autores.

Cuando se han presentado ideas o palabras de otros autores en esta disertación, he realizado su respectivo reconocimiento aplicando correctamente los esquemas de citas y referencias bibliográficas en el estilo requerido.

He obtenido el permiso del autor o editor para incluir cualquier material con derechos de autor (por ejemplo, tablas, figuras, instrumentos de encuesta o grandes porciones de texto).

Por último, he sometido esta disertación a la herramienta de integridad académica, definida por la universidad.



Nombre Luis Gabriel Diaz Fonseca

Fecha 31/01/2023

Agradecimientos

A mí familia, por cada momento en el que me han brindado su amor incondicional y su fe en mí. Gracias por ser mis cimientos en los momentos difíciles y celebrar conmigo los buenos momentos.

A mis amigos, quiero darles las gracias por ser una segunda familia, por su amistad y por estar presentes cuando más los he necesitado.

A mí tutor Fernan, le agradezco infinitamente por su guía, por su conocimiento y por su paciencia en todo este proceso.

Y a la universidad pública, estaré eternamente agradecido por brindarme la oportunidad de adquirir conocimientos y formarme como una persona y como un profesional.

Resumen

Modelo para la simulación de la producción de calzado masculino hecho a mano en Bogotá, Colombia

La producción de calzado masculino hecho a mano es una industria de tradición en Bogotá, donde existen numerosas fábricas y talleres que ofrecen productos hechos a mano con un enfoque en la calidad y la atención al detalle. Esta industria es conocida por su variedad de diseños y estilos, se distingue por su autenticidad y la habilidad de los artesanos.

Esta industria enfrenta varios desafíos, como la competencia de productos de producción en masa, la escasez de materiales y los riesgos asociados a que sea un proceso manual que en muchas ocasiones no está estandarizado.

Considerando principalmente el último desafío, se realiza este trabajo que busca implementar políticas que optimicen la producción del calzado.

De este trabajo, se puede concluir que es posible optimizar la producción reduciendo los tiempos de espera, con la implementación de modelos de aprendizaje por refuerzo entrenados desde una simulación por eventos descritos.

Palabras clave: (Simulación, aprendizaje por refuerzo, optimización, calzado).

Abstract

Model for the simulation of handmade men's shoe production in Bogotá, Colombia

Handmade men's shoe production is a traditional industry in Bogotá, where there are numerous factories and workshops offering handmade products with a focus on quality and attention to detail. This industry is known for its variety of designs and styles, and is distinguished by its authenticity and the skills of the artisans.

This industry faces various challenges, such as competition from mass-produced products, scarcity of materials, and risks associated with being a manual process that is often not standardized.

Considering mainly the last challenge, this work aims to implement policies to optimize shoe production.

From this work, it can be concluded that it is possible to optimize production by reducing wait times, with the implementation of reinforcement learning models trained from a simulation based on described events.

Keywords: (Simulation, reinforcement learning, optimization, footwear).

Contenido

	Pág.
Resumen	IX
Lista de figuras	XIII
Lista de tablas	15
1. Formulación del problema	16
1.1 Introducción	16
1.2 Motivación.....	16
1.3 Revisión de la literatura	16
1.4 Planteamiento del problema	20
1.5 Objetivos.....	21
1.5.1 Objetivo general	21
1.5.2 Objetivo específico.....	21
1.6 Metodología	21
1.6.1 Entendimiento del negocio.....	23
1.6.2 Entendimiento de los datos.....	23
1.6.3 Preparación de los datos	23
1.6.4 Modelado	24
1.6.5 Evaluación.....	24
1.6.6 Despliegue	24
1.7 Alcance	24
2. Análisis descriptivo del flujo de los datos	25
2.1 Introducción	25
2.2 Flujo de datos	25
3. Implementación del modelo	29
3.1 Introducción	29
3.2 Descripción general del flujo del proceso de manufacturación	30
3.3 Implementación del ambiente de simulación.....	31
3.4 Evaluación de la simulación	40
3.5 Ejecución del modelo.....	41
3.5.1 Convergencia.....	41
3.5.2 Validación del modelo	42
3.6 Ambiente personalizado openIA gym	42
4. Optimización	46
4.1 Entrenamiento	46
4.2 Análisis de resultados.....	48

5. Conclusiones y recomendaciones.....	51
5.1 Conclusiones.....	51
5.2 Recomendaciones.....	52
Bibliografía	53

Lista de figuras

	Pág.
Figura 1-1: Integración de las dos técnicas.....	22
Figura 2-1: Esquema general flujo de datos.	26
Figura 2-2: Orden del cliente.....	26
Figura 2-3: Tareas agrupadas por prioridad.	27
Figura 3-1: Flujo del proceso de manufacturación del calzado	30
Figura 3-2: Distribución normal para la actividad de corte.....	32
Figura 3-3: Distribución normal para la actividad de guarnición.	33
Figura 3-4: Distribución normal para el área necesaria de cuero.	34
Figura 3-5: Método generador actividad para corte.	35
Figura 3-6: Método generador actividad para guarnición.	36
Figura 3-7: Distribución normal para la actividad de suela.	37
Figura 3-8: Distribución normal para la actividad de plantilla.....	37
Figura 3-9: Método generador actividad suela o plantilla.	38
Figura 3-10: Distribución normal para la actividad de zapato.....	39
Figura 3-11: Método generador actividad para calzado.	40
Figura 3-12: Método get índice	41
Figura 3-13: Diagrama integración.....	43
Figura 3-14: Método reset.....	44
Figura 3-15: Método step.....	45
Figura 4-1: Entrenamiento de los agentes.....	47

Lista de tablas

	Pág.
Tabla 1-1: Estudios de simulación.....	19
Tabla 3-1: Actividades que componen la simulación	31
Tabla 3-2: Distribución corte.....	32
Tabla 3-3: Distribución guarnición	33
Tabla 3-4: Distribución cueros	34
Tabla 3-5 : Distribución suelas	37
Tabla 3-6: Distribución suelas	37
Tabla 3-7 : Distribución zapatos	39
Tabla 3-8: Parámetros de simulación.....	41
Tabla 4-1 : Tasas de aprendizaje a cada modelo	48
Tabla 4-2 : Métrica de recompensa	48
Tabla 4-3: Resultados indicie promedio para diferentes algoritmos	49

1. Formulación del problema

1.1 Introducción

En el sector de la fabricación del calzado, se ha detectado una carencia en la gestión de los riesgos debidos a la elevada incertidumbre en el proceso de producción. Este trabajo propone una solución a este problema implementado estrategias que reduzcan estos riesgos.

De acuerdo con la literatura existente, se constata que hay un amplio número de estudios que se enfocan en la optimización de procesos de fabricación, pero una cantidad menor de investigaciones que se concentran específicamente en la optimización de procesos de producción de calzado.

Considerando lo anterior se define como objetivo principal de este trabajo proponer un modelo para la simulación de la producción manual de calzado masculino en Bogotá, Colombia. Se logrará mediante la caracterización de las variables para el modelo de simulación de la producción, implementando dicho modelo para su posterior validación y optimización.

1.2 Motivación

En la industria de producción de calzado se ha identificado una falencia para manejar los riesgos debidos a la alta variabilidad en el proceso de producción. Esta investigación tiene como motivación implementar políticas que ayuden a mitigar este tipo de riesgos y de este modo optimizar la producción.

1.3 Revisión de la literatura

En el año 2003 (Rabelo, y otros, 2003) proponen integrar un modelo de eventos discretos con un modelo de sistema dinámico para la simulación de la producción en una fábrica de semiconductores en Estados Unidos.

Dentro del paper se menciona que la simulación mediante eventos discretos es la metodología más utilizada para estudiar el diseño y las operaciones de los sistemas de fabricación, el cual está basado en los siguientes tres argumentos:

- Tiene la capacidad de describir los sistemas de fabricación más complejos e incluir elementos estocásticos, que no pueden describirse fácilmente mediante modelos matemáticos o analíticos
- Permite rastrear el estado de entidades y recursos individuales en la instalación y estimar numerosas medidas de desempeño asociadas con esas entidades bajo una amplia gama de condiciones operativas proyectadas.
- Se pueden comparar diseños de instalaciones o políticas de operación alternativas para una instalación a través de simulación para ver cuál cumple mejor con un objetivo de rendimiento específico.

El modelo descrito genera como salidas la tasa de producción, proyección de la capacidad, como también datos de configuración como el número necesario de máquinas y de trabajadores.

En el año 2008 (Paternina-Arboleda, Montoya-Torres, & Fábregas-Ariza, 2008) sugieren el uso del algoritmo de aprendizaje por refuerzo y las redes neuronales con el objetivo de encontrar el óptimo global. En particular, se destaca la importancia de seleccionar cuidadosamente los valores de entrada en la optimización de una simulación, de modo que se logre optimizar la función de costo.

En este caso, se ha optado por utilizar una modificación del algoritmo SMART (Semi-Markov Average Reward Technique) para el entrenamiento del agente. Durante dicho entrenamiento, se lleva a cabo la optimización de diversos parámetros, como la cantidad de agentes, la tasa de aprendizaje, la tasa de explotación (que determina la probabilidad de seleccionar una acción aleatoria en cada iteración), el parámetro de búsqueda y convergencia (expresado como un porcentaje del máximo número de iteraciones) y el número inicial del paso (representado como un porcentaje del rango o longitud del dominio de búsqueda de las variables de decisión).

En el año 2011 (Daaboul, Da Cunha, Bernard, & Laroche, 2011) explora la variabilidad de los productos y la variabilidad en el proceso de los productos en producción en masa y cómo estos dos componentes interactúan. Define una métrica de valor basado en la perspectiva del cliente que consiste en tres criterios: el costo, la calidad y el tiempo de espera.

Expone un caso de estudio para la industria del calzado y propone una metodología para la implementación de la simulación, basado en la cadena de valor.

En la simulación se considera que existen diferentes departamentos encargados del proceso de manufactura: diseño técnico, el departamento de corte, el departamento de guarnición, el departamento de confección y el departamento de acabados.

Para poder realizar el modelado a cada uno de los procesos los describe por: número de trabajadores, tiempo de instalación, tiempo de espera, tiempo del proceso y tiempo total.

Adicionalmente para la implementación del modelo de simulación utiliza la siguiente metodología:

- Formulación del problema
- Recolección de los datos y desarrollo del modelo
- Computarización del modelo
- Verificación
- Validación
- Diseño del experimento
- Ejecución de la simulación
- Análisis de la simulación

Para poder analizar el desempeño de la simulación se utilizaron cinco criterios: producción diaria, análisis de tiempo en el flujo, análisis del tiempo en colas, utilización de los recursos, análisis de los reprocesos.

En el año 2013 (Diaz-Elsayed, Jondral, Greinacher, Dornfeld, & Lanza, 2013) describe una metodología de tres pasos para poder evaluar implementaciones de estrategias LEAN Y verdes, en el sistema de manufactura.

El primer paso es la simulación mediante eventos discretos realizando una modelación del estado actual. El segundo paso es evaluar las diferentes estrategias LEAN y verdes enunciadas en el artículo y al final como tercer paso es realizar una evaluación económica de cada una de las alternativas.

Se propone minimizar la función de costo descrita como la sumatoria entre cinco componentes los cuales son: el costo de oportunidad, el costo de los materiales, el costo de las máquinas, el costo del inventario y el costo del personal.

En el año 2019 (Hernández-González, Ramírez-Tapia, & Jiménez-García, 2019) realizaron una simulación mediante eventos discretos para la producción de calzado hecho a mano, con el fin de hacer un diagnóstico y determinar cuáles operaciones necesitan ser redimensionadas para que puedan soportar un demanda más alta. Como medidas de desempeño utilizaron la producción diaria, el análisis de tiempo en el flujo y la utilización de los recursos.

En el año 2020 (Calderón Andrade, Hernández Gress, & Montufar Benítez, 2020) utilizaron la simulación mediante eventos discretos en una fábrica de calzado en Hidalgo (México)

para poder evaluar varias propuestas de optimización de la línea de producción utilizando la técnica de la reingeniería. Como medidas de desempeño toma el porcentaje de utilización de las maquinas, promedio diario de producción y la cantidad de inventario que está siendo procesado.

En el año 2021 (Kuhnle, Káiser, Theiß, Stricker, & Lanza, 2021) utilizaron la modelación mediante eventos discretos para poder entrenar un agente mediante aprendizaje por refuerzo. Implementaron el agente en varios módulos:

El módulo de acción, encargado de seleccionar la acción que debe tomar el agente en cada una de las iteraciones del entrenamiento.

- El módulo del estado, encargado de sustraer la información relevante del estado del ambiente con el objetivo que el agente tome decisiones que maximicen la recompensa.
- El módulo de recompensa, encargado de evaluar permanentemente el rendimiento del agente, con el objetivo que el agente lo tome como retroalimentación y pueda mejorar su desempeño.
- El módulo del agente implementa los algoritmos de aprendizaje por refuerzo.
-

En el año 2022 (Samsonov, Hicham, & Meisen, 2022) implementaron la optimización en el proceso de manufactura mediante aprendizaje por refuerzo. Dividen la metodología en tres niveles. El nivel de percepción donde se define el estado actual del ambiente. El nivel de interacción donde se diseña el espacio de acción y recompensa y El nivel de cognición que se compone de diferentes algoritmos de aprendizaje por refuerzo.

A continuación una tabla resumen con los estudios de simulación:

Tabla 1-1:Estudios de simulación

autores	Objetivo del estudio	medidas de desempeño	Técnica de simulación
Rabelo, y otros, 2003	Integrar la simulación de eventos discretos con modelo de sistema dinámico para mejorar la interpretación del impacto de las decisiones tomadas en la simulación.	Tasa de producción Proyección de la capacidad	Simulación por eventos discretos Sistema dinámico
Daaboul, Da Cunha, Bernard, & Laroche, 2011	En el caso de estudio expuesto en el paper, se describen los pasos para simular la red de valor en la industria del calzado	costo calidad tiempo de espera	Simulación por eventos discretos

autores	Objetivo del estudio	medidas de desempeño	Técnica de simulación
selman erilmaz	Medir el efecto de la variabilidad de los estilos en la línea de producción	producción diaria análisis de tiempo en el flujo análisis del tiempo en colas utilización de los recursos	Simulación por eventos discretos
Hernández-González, Ramírez-Tapia, & Jiménez-García, 2019	En el caso de estudio expuesto en el paper, se realiza una simulación para determinar si las capacidades de la línea de producción suportaban un aumento en la demanda	Producción total de zapatos Tamaño medio de la cola tiempo de espera en la cola	Simulación por eventos discretos
Calderón Andrade, Hernández Gress, & Montufar Benítez, 2020	Realizar una simulación para evaluar diferentes propuestas utilizando la técnica de la reingeniería	Utilización promedio de las máquinas Promedio diario de producción Inventario en proceso por operación	Simulación por eventos discretos

Se puede observar que existen una gran cantidad de investigaciones que se centran en la optimización de procesos de manufactura, hay una cantidad más reducida de artículos que se especializan en la optimización de procesos de producción de calzado.

1.4 Planteamiento del problema

Con base en la revisión de la literatura, hay temas que se deben enriquecer ya que se puede evidenciar que existe una tendencia a utilizar combinaciones de simulación discreta con alguna metodología que agregue valor al proceso de producción basado en la experiencia, pero no se ha explorado la posibilidad de utilizar metodologías con base en aprendizaje automático para proponer las políticas de optimización.

1.5 Objetivos

1.5.1 Objetivo general

Proponer un modelo para la simulación de la producción manual de calzado masculino en Bogotá, Colombia.

1.5.2 Objetivo específico

- Caracterizar las variables para el modelo de simulación de la producción manual de calzado masculino.
- Implementar un modelo para la simulación del proceso de producción manual de calzado masculino.
- Validar el modelo implementado de producción manual de calzado masculino.

1.6 Metodología

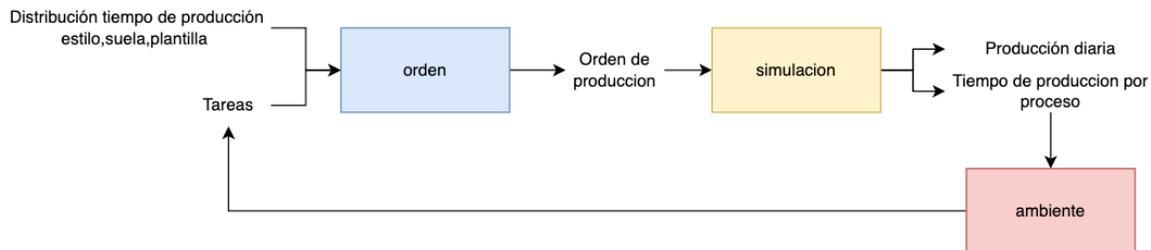
Con el fin de cumplir con el objetivo mencionado anteriormente, se implementarán dos técnicas: una simulación para representar el comportamiento del proceso de producción de calzado y utilizar esta representación para entrenar un agente utilizando aprendizaje por refuerzo.

La simulación se encargará de crear un modelo simplificado del proceso de producción de calzado, que capturará las características esenciales y las relaciones clave del sistema. Este modelo permitirá estudiar y analizar el comportamiento del proceso en diferentes escenarios, realizar experimentos virtuales y predecir resultados futuros.

Utilizando la simulación como entorno de entrenamiento, se aplicará el enfoque de aprendizaje por refuerzo para entrenar a un agente. El agente aprenderá a tomar decisiones óptimas a través de la interacción con el entorno simulado, recibiendo retroalimentación en forma de recompensas positivas o negativas según su desempeño. A medida que el agente realiza acciones y experimenta las consecuencias en la simulación, ajustará su comportamiento para maximizar las recompensas y mejorar su rendimiento en la producción de calzado.

A continuación un diagrama de alto nivel de la integración de las dos técnicas:

Figura 1-1: Integración de las dos técnicas



Como se evidencia en el diagrama la implementación se dividirá en tres módulos:

Orden : Generar las órdenes de producción que se utilizarán como insumo en la simulación. Este módulo tiene como entradas las distribuciones de tiempo de producción para cada estilo de zapato que se desea simular, así como también la distribución de producción de las suelas y plantillas. Además, requiere de una lista de tareas que se desean simular.

Simulación : Tiene como objetivo modelar el comportamiento de la producción de calzado y calcular el tiempo de producción por proceso, así como también la producción diaria de calzado.

Ambiente : Se encarga de entrenar una agente mediante aprendizaje por refuerzo. El agente de aprendizaje interactúa con el entorno, recibiendo como entrada la orden de producción y los tiempos de producción generados por la simulación. El agente realiza acciones, modificando las tareas y a medida que avanza la simulación, recibe retroalimentación en forma de recompensas.

Para poder realizar esta implementación se utilizará la metodología **CRoss Industry Standard Process for Data Mining (CRISP-DM)**, que consiste en los siguientes pasos:

1. Entendimiento del negocio

2. Entendimiento de los datos
3. Preparación de los datos
4. Modelado
5. Evaluación
6. Despliegue

A continuación, se explicarán cada una de las fases y cómo se aplicarán en el proyecto.

1.6.1 Entendimiento del negocio

Esta fase se enfoca en determinar los objetivos y requerimientos del proyecto, en conjunto con el negocio y la revisión de literatura que implementará el objetivo y los criterios de éxito propuesto en esta tesis.

Por lo tanto, como objetivo se plantea proponer un modelo para la simulación de la producción manual de calzado masculino para explorar diferentes políticas que optimicen la producción y como criterio de éxito se propone utilizar los criterios que (Selman ERYILMAZ, Osman KUŞAKCI, & GAVRANOVIC, 2012) describen: tasa de producción diaria, el tiempo requerido en flujo del proceso y el análisis de las colas.

1.6.2 Entendimiento de los datos

Se debe identificar los agentes que participan en el sistema, como también los eventos y acciones. Después, se recolectarán los datos de estos elementos, se analizarán para validar la calidad y veracidad de los datos.

1.6.3 Preparación de los datos

Se preparan los datos para su modelado con ayuda de las dos fases anteriores y se determina que datos se van a utilizar. Luego se realiza una limpieza de los datos, dependiendo de los requerimientos se generan nuevos atributos que sean de ayuda para la investigación o se integran varios grupos de datos y como última tarea se debe realizar un formateo de los datos.

1.6.4 Modelado

Se utilizará SimPy, un paquete de desarrollo en Python, para modelar la producción de calzado mediante una simulación de eventos discretos. Posteriormente, se aplicará esta simulación al aprendizaje por refuerzo utilizando el paquete Stable Baselines3 para entrenar agentes y optimizar el proceso de producción.

1.6.5 Evaluación

Esta fase consta de dos fases: la evaluación de la simulación de eventos discretos y el entrenamiento mediante aprendizaje por refuerzo.

En la primera fase, se evaluarán las métricas de desempeño de la simulación y se realizarán ajustes iterativos hasta lograr un comportamiento deseado.

En la segunda fase, se utilizará la simulación optimizada como insumo para entrenar a los agentes mediante aprendizaje por refuerzo, buscando reducir los tiempos de producción.

La segunda fase depende de los resultados obtenidos en la primera, asegurando que la simulación sea representativa y proporcione un entorno adecuado para el entrenamiento de los agentes.

1.6.6 Despliegue

En esta fase, se implementa el modelo que ha sido evaluado en la fase anterior. Este modelo desplegado estará disponible para su uso cada vez que se desee generar una orden de producción.

1.7 Alcance

Se propone utilizar Python, con las librerías SimPy y Stable Baselines3, para implementar un modelo de simulación de la producción manual de calzado masculino en Bogotá, Colombia. Esto permitirá caracterizar variables, modelar el proceso de producción y validar el modelo. Además, el uso de software libre brinda flexibilidad y adaptabilidad al proyecto.

2. Análisis descriptivo del flujo de los datos

2.1 Introducción

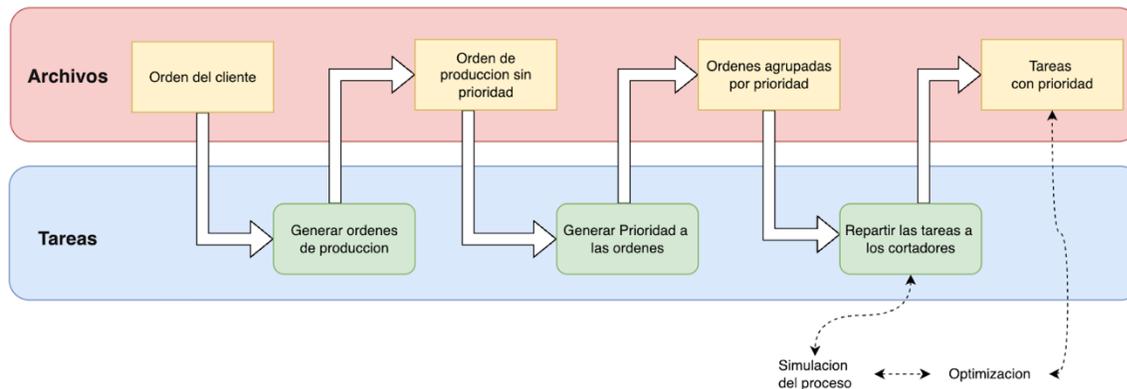
Se ha estado realizando mejoras continuas en el método de trabajo, siempre apostando a una transformación digital, por tal motivo el proceso de generación de órdenes de producción está altamente tecnificado.

Aunque existen algunos pasos que siguen siendo netamente empíricos y dependen en su totalidad de la experiencia de las personas que lo ejecutan. En estos pasos se realizar una intervención, de forma tal que se garantice en todo el flujo del trabajo un estándar.

2.2 Flujo de datos

En la **figura 2-1**, se visualiza el esquema general del flujo de los datos desde que el cliente realiza la orden hasta el inicio de la manufactura del calzado, donde se evidencia a pesar de ser un proceso ordenado se presentan desperdicios debido a que sigue siendo heurístico.

Figura 2-1: Esquema general flujo de datos.



Fuente: Elaboración propia

A continuación, una descripción de cada uno de los elementos que componen este diagrama de flujo:

- **Orden del Cliente (Archivo) :** Este es el primer insumo de todo el flujo, consta de un archivo de Excel donde se encuentra divididos los pedidos en estilos con sus respectiva cantidad de pares por número.

En la **figura 2-2** se puede evidenciar el esquema para cada uno de los estilos que tiene una orden de producción, como se puede observar están bien esquematizado para que un humano los pueda trabajar, pero es necesario realizar procesos adicionales para que un programa pueda realizar actividades con la información del archivo de Excel.

Figura 2-2: Orden del cliente.

Horma Bristol Cuero Negro/Suela cuero bajita negra		SKU	Estilo	Talla US	Cantidad
		OXCMCL1007A4	Dean Oxford - Cuero negro	7	2
		OXCMCL1008A4	Dean Oxford - Cuero negro	8	3
		OXCMCL1008SA4	Dean Oxford - Cuero negro	8,5	2
		OXCMCL1009A4	Dean Oxford - Cuero negro	9	8
		OXCMCL1009SA4	Dean Oxford - Cuero negro	9,5	10
		OXCMCL10010A4	Dean Oxford - Cuero negro	10	14
		OXCMCL10010SA4	Dean Oxford - Cuero negro	10,5	11
		OXCMCL10011A4	Dean Oxford - Cuero negro	11	7
		OXCMCL10011SA4	Dean Oxford - Cuero negro	11,5	2
		OXCMCL10012A4	Dean Oxford - Cuero negro	12	8
		OXCMCL10013A4	Dean Oxford - Cuero negro	13	5
		OXCMCL10014A4	Dean Oxford - Cuero negro	14	1

Fuente : Calzado Tauro

- **Generación orden de producción (Actividad):** Previo a este trabajo se realizó un desarrollo que lee el archivo con la orden del cliente, toma cada uno de los estilos y los divide en tareas, procurando uniformidad en la cantidad de pares.
- **Orden de producción sin prioridad (Archivo):** Como output de la actividad anterior se generan dos archivos, el primero es un PDF con las tareas que se deben realizar y el segundo es un archivo de Excel donde se encuentran todas las actividades en una misma tabla.
- **Generar prioridad a las tareas (Actividad):** Mediante un compendio de reglas básicas, que tienen como objetivo darle mayor prioridad a las tareas que se demoran más, se agrupan las tareas.
- **Tareas agrupadas por prioridad (Archivo):** Como output de la actividad anterior se genera un archivo en formato Excel que tiene todas las tareas con su respectiva prioridad, donde prioridad 1 serán para las tareas de mayor prioridad y entre más alto el número de prioridad más baja será.

En la figura 2-3, se pueden visualizar como están agrupadas las ordenes de producción por orden, como se puede observar la horma y la suela van a jugar un papel importante dentro del orden de las tareas.

Figura 2-3: Tareas agrupadas por prioridad.

horma	suela	vira	entreSuela	tapa	7.0	8.0	8.5	9.0	9.5	10.0	10.5	11.0	11.5	12.0	13.0	14.0	Total	Prioridad
Waterproof	Suela Eva Alta Ondas Cafe		n/a	café	0	1	2	3	0	0	0	1	0	1	0	1	9	1
Waterproof	Suela Eva Alta Ondas Negra		n/a	negro	0	0	0	0	0	3	0	0	0	0	0	0	3	
	Total WATERPROOF				0	1	2	3	0	3	0	1	0	1	0	1	12	
Tubular	Suela Cuero Bajita Vira Chafan Cafe	bajita chafan café	n/a	café	0	0	0	2	2	3	5	1	0	1	0	0	14	2 A
Tubular	Suela Cuero Bajita Vira Chafan Caoba	bajita chafan caoba	n/a	café	1	1	0	2	4	2	4	1	1	1	1	0	18	
Tubular	Suela Cuero Bajita Vira Chafan Negra	bajita chafan negra	n/a	negro	0	0	3	4	3	2	3	5	2	0	0	1	23	
Tubular	Plantisuela café		n/a	café	0	4	2	6	6	4	10	6	8	3	3	0	52	2 B
Tubular	plantisuela caoba		n/a	café	1	2	1	17	10	11	7	10	5	15	3	3	85	
Tubular	Plantisuela negra		n/a	negro	0	7	2	9	12	12	4	6	11	10	9	6	88	

Fuente: Calzado Tauro

- **Asignación de tareas (Actividad):** Para cada uno de los grupos de prioridades se ordenen las tareas.

En la actualidad cómo se asignan las tareas se basa netamente en métodos heurísticos, confinado en la experiencia de la persona que realiza esta actividad.

Es en esta actividad donde se propone realizar una intervención (simulación y optimización) ya que del orden que se seleccione en este paso del flujo tendrá un efecto directo en el tiempo de producción.

- **Tareas ordenadas para la manufactura (Archivo):** Con las tareas priorizadas, se iniciará el proceso de manufactura del calzado, realizando los cortes del cuero, las suelas y las plantillas necesarias.

3. Implementación del modelo

3.1 Introducción

En este capítulo se describirá la implementación del modelo de simulación, que es una simplificación del proceso de producción de calzado, capturará las características esenciales y las relaciones clave del sistema.

Este modelo tiene como objetivo estudiar y analizar el comportamiento del proceso en diferentes escenarios, realizando experimentos virtuales que proporcionarán los datos necesarios para entrenar al agente de aprendizaje por refuerzo.

Teniendo en cuenta que es una simplificación de un proceso, se asumieron las siguientes consideraciones:

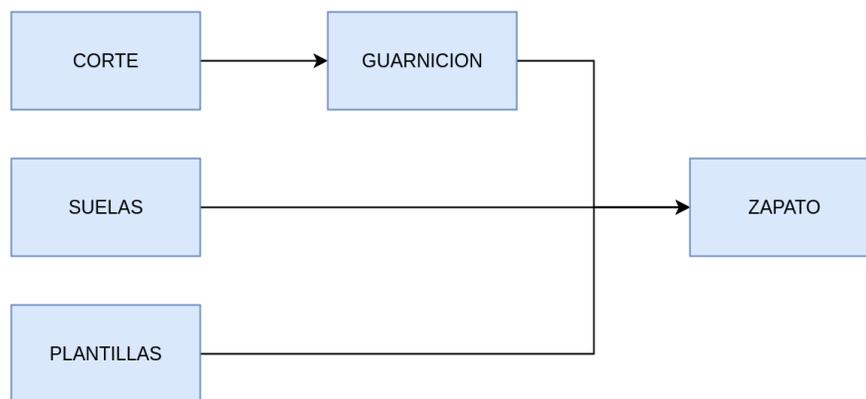
- Se agrupa una gran cantidad de procesos en cinco actividades principales: corte, guarnición, plantilla, suela y zapato. Estas actividades representan las etapas clave en la fabricación de calzado.
- Dado que la empresa cuenta con una amplia variedad de estilos de calzado, se seleccionaron los más representativos para este análisis. Es importante destacar que estos estilos son solicitados de manera constante en las órdenes de producción.
- Es importante destacar que, en el proceso de fabricación de calzado, a veces pueden surgir reprocesos o retrabajos en las diferentes actividades. Sin embargo, en esta simulación, no se tendrán en cuenta estos reprocesos.
- Las distribuciones de los procesos se refieren al tiempo requerido, en minutos, para llevar a cabo un proceso específico en la fabricación de un par de calzado en particular.

3.2 Descripción general del flujo del proceso de manufacturación

En un constante diálogo con el negocio se llega a la conclusión que la simulación del proceso debe ser por cada una de las macro actividades, corte, guarnición, suelas, plantillas, zapatos.

En la figura 3-1 se grafican las actividades que se deben realizar para la manufactura del zapato, las actividades de corte, suela y plantilla se pueden hacer en paralelo, en cambio las actividades de guarnición y zapato dependen de actividades previas.

Figura 3-1: Flujo del proceso de manufacturación del calzado



Fuente: Elaboración propia

A continuación, una descripción de cada una de las actividades:

- **Corte:** En esta actividad se toman las láminas de cuero y se producen las piezas que componen la capellada del zapato.

En la actualidad cómo se organizan las actividades de la orden de producción es dándole prioridad a los estilos que requieren más tiempo y cuidado para su elaboración, adicionalmente para optimizar el consumo de la materia prima (cuero) se ordenan las actividades de forma tal que en la misma lámina de cuero se puedan sacar cortes tanto para zapatos como para botas.

- **Guarnición:** Esta actividad depende de la actividad del corte, ya que se toman las piezas para formar la capellada.
- **Suelas y plantillas:** Al nivel que se plantea realizar la simulación estas dos actividades son identificadas. Se reciben las actividades, se agrupan por el tipo de horma y se realiza su producción como un solo bloque.
- **Zapato:** En esta actividad se toma las capelladas generadas en guarnición, las suelas y las plantilla para producir el zapato.

A continuación una tabla resumen con las actividades que componen la simulación:

Tabla 3-1:Actividades que componen la simulación

Proceso	Entradas	medidas de desempeño
Corte	Distribución de producción corte Distribución de cuero Capacidad del recurso corte	Tiempo en espera Tiempo de ejecución
Guarnición	Distribución de producción guarnición Capacidad del recurso guarnición	Tiempo en espera Tiempo de ejecución
Plantilla	Distribución de producción plantilla Capacidad del recurso línea de producción plantilla	Tiempo en espera Tiempo de ejecución
Suela	Distribución de producción suela Capacidad del recurso línea de producción suela	Tiempo en espera Tiempo de ejecución
Zapato	Distribución de producción zapato Capacidad del recurso zapato	Tiempo en espera Tiempo de ejecución

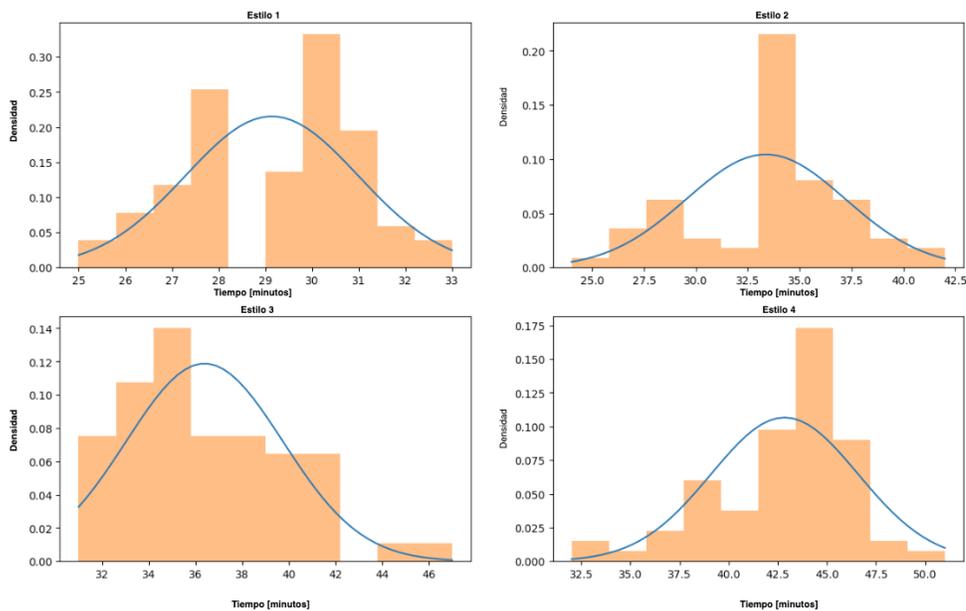
3.3 Implementación del ambiente de simulación

Mediante datos de la producción del calzado suministrados por el negocio, se logra caracterizar las variables que definen cada actividad. En concreto se logra determinar funciones de distribución normal con los tiempos estimados para la ejecución de cada

actividad, como también cuantas personas intervienen en cada proceso, esto servirá como insumo para la simulación.

Para las actividades de corte y guarnición (figura 3-2 y 3-3), aunque existan múltiples estilos con diferentes colores de cuero, se agrupan en cuatro estilos. Dependen de la dificultad del corte o por la presencia de adornos o cantidad de piezas.

Figura 3-2: Distribución normal para la actividad de corte.

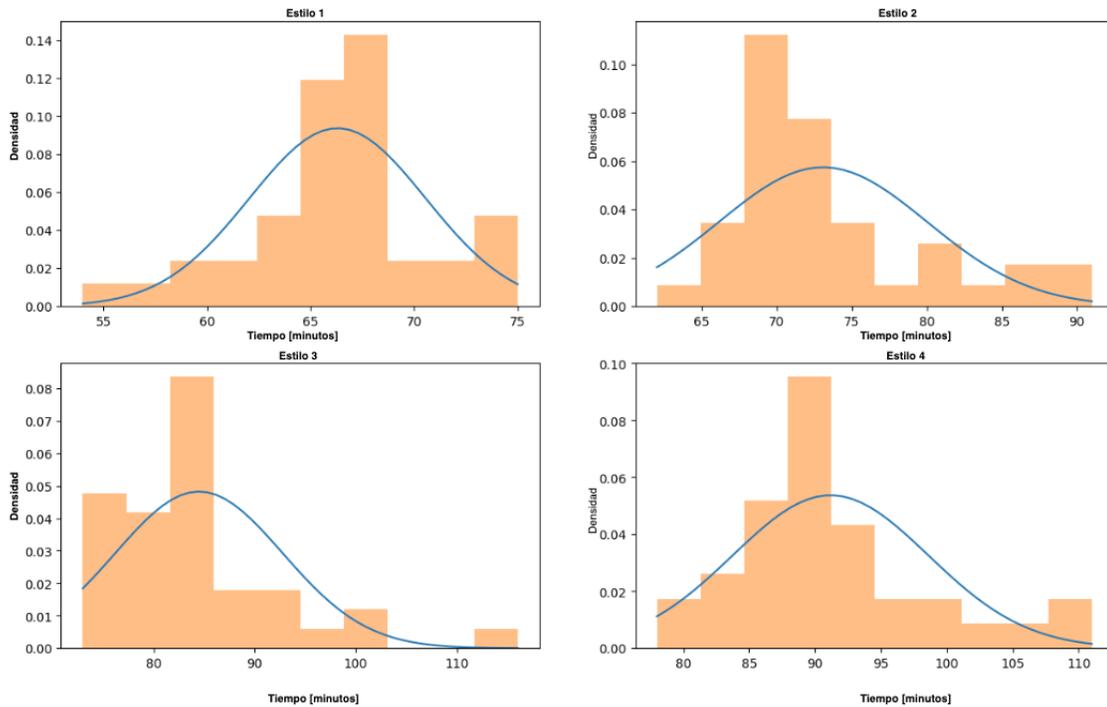


Fuente: Calzado Tauro

Tabla 3-2: Distribución corte

Distribución Corte						
nombre	cantidad	media	std	mínimo	máximo	p_value
estilo 1	64	29,14062	1,85293	25,0	33,0	0,029452
estilo 2	62	33,37097	3,82367	24,0	42,0	0,022482
estilo 3	58	36,39655	3,35765	31,0	47,0	0,043477
estilo 4	70	42,84286	3,74027	32,0	51,0	0,040397

Figura 3-3: Distribución normal para la actividad de guarnición.



Fuente: Calzado Tauro

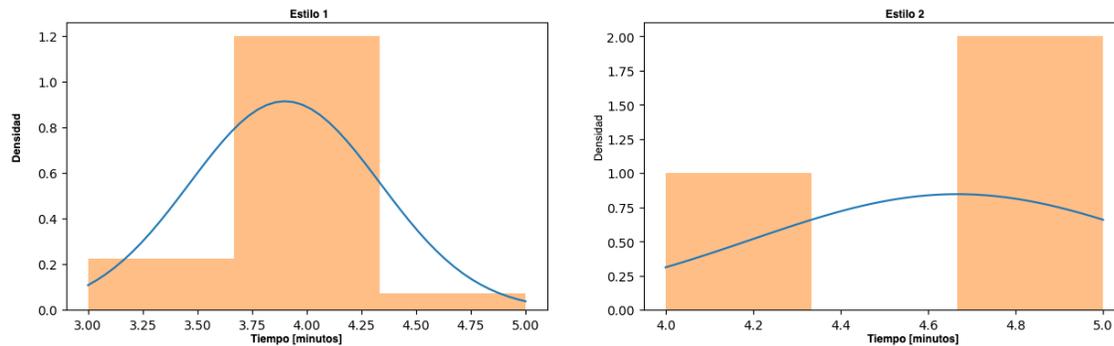
Tabla 3-3: Distribución guarnición

Distribución Guarnición						
nombre	cantidad	media	std	mínimo	máximo	p_value
estilo 1	40	66,30000	4,25558	54,0	75,0	0,032792
estilo 2	40	73,07500	6,94762	62,0	91,0	0,047669
estilo 3	39	84,48718	8,27685	73,0	116,0	0,040331
estilo 4	35	91,17143	7,43154	78,0	111,0	0,045619

Como se mencionó en la sección anterior, también se quiere considerar la utilización del cuero como parámetro dentro de la simulación, por este motivo se determinó una distribución del área de cuero necesaria por par de zapatos (figura 3-4).

Cabe aclarar que esta distribución no depende de la talla de los zapatos, si no del estilo, esto debido al hecho que existen estilos que requieren muchas más piezas de cuero para su elaboración.

Figura 3-4: Distribución normal para el área necesaria de cuero.



Fuente: Calzado Tauro

Tabla 3-4: Distribución cueros

Distribución Cueros						
nombre	cantidad	media	std	mínimo	máximo	p_value
estilo 1	20	3,90000	0,43589	3,0	5,0	0,000469
estilo 2	15	4,66667	0,47140	4,0	5,0	0,005356

Con los tres parámetros cuero, guarnición y corte es posible implementar las actividades de corte y guarnición dentro del ambiente de simulación, para este trabajo se utiliza el framework Simpy, que es capaz de realizar simulaciones por eventos discretos a base de python estándar, a continuación, un fragmento del código con los métodos que generan las dos actividades:

En la figura 3-5, se observa la clase **Corte_Guarnicion** se implementa el método **generador_cortador**, que calcula el tiempo total de la actividad y lo pasa al ambiente de desarrollo mediante la palabra reservada yield, posteriormente se agregan métricas de la actividad a un dataframe para análisis y se genera la actividad de guarnición.

Figura 3-5: Método generador actividad para corte.

```
class Corte_Guarnicion():

    def generador_cortador(self, corte):

        inicio_tiempoCola = self.env.now

        with self.cortadores.request() as req:
            yield req
            inicio_proceso = self.env.now
            self.tiempo_colas += (inicio_proceso - inicio_tiempoCola)
            id_cortador = self.get_id_cortador()
            estilo = corte["estilo"]
            cuero = corte["cuero"]
            cantidad = corte["cantidad"]

            area_total = sum([ random.gauss(corte["area_media"], corte["area_desv"]) for i in range(cantidad)])
            tiempo_cambio = self.get_tiempo_cambio_cuero(
                id_cortador, cuero, corte["area_media"], corte["area_desv"]
            )
            numero_laminas = self.get_laminas(self.datos[id_cortador]["area"], area_total)

            tiempo_tarea = sum(
                [random.gauss(corte["corte_media"], corte["corte_desv"]) for i in range(cantidad)]
            )
            tiempo_lamina = sum(
                [
                    random.expovariate(1.0 / Hiperparametros.intervalo_cambio_laminas)
                    for i in range(numero_laminas)
                ]
            )

            tiempo_total = tiempo_cambio + tiempo_tarea + tiempo_lamina

            yield self.env.timeout(tiempo_total)

            self.id_cortadores.append(id_cortador)

            fin_proceso = self.env.now
            self.tiempo_proceso += (fin_proceso - inicio_proceso)

            df_tmp = pd.DataFrame({
                "tipo": [1],
                "id": [corte["id"]],
                "inicio": [inicio_proceso],
                "fin": [fin_proceso]
            })

            self.df_metricas = self.df_metricas.append(df_tmp)

        g = self.generador_guarnicion(corte["guarnicion_media"], corte["guarnicion_desv"], cantidad, estilo, corte["id"])
        self.env.process(g)
```

Fuente: Elaboración propia

En la misma clase **Corte_Guarnicion**, está implementado el método **generador_guarnicion**, encargado de generar la actividad de guarnición.

Como se puede evidenciar en la figura 3-6, mediante la palabra reservada `yield` se agrega el tiempo de esta actividad al ambiente de simulación, después se agregan métricas al dataframe para análisis y por último se agrega la información de la actividad al objeto pipe, encargado de conectar la información en las diferentes actividades.

Figura 3-6: Método generador actividad para guarnición.

```
class Corte_Guarnicion():

    def generador_guarnicion(self,media_g,desv_g,cantidad,estilo,id_tarea):

        inicio_tiempoCola = self.env.now

        with self.guarnecedores.request() as req:
            yield req
            fin_tiempoCola = self.env.now
            self.tiempo_colas += (fin_tiempoCola - inicio_tiempoCola)
            inicio = self.env.now
            t = sum([random.gauss(media_g,desv_g) for i in range(cantidad)])

            yield self.env.timeout(t)

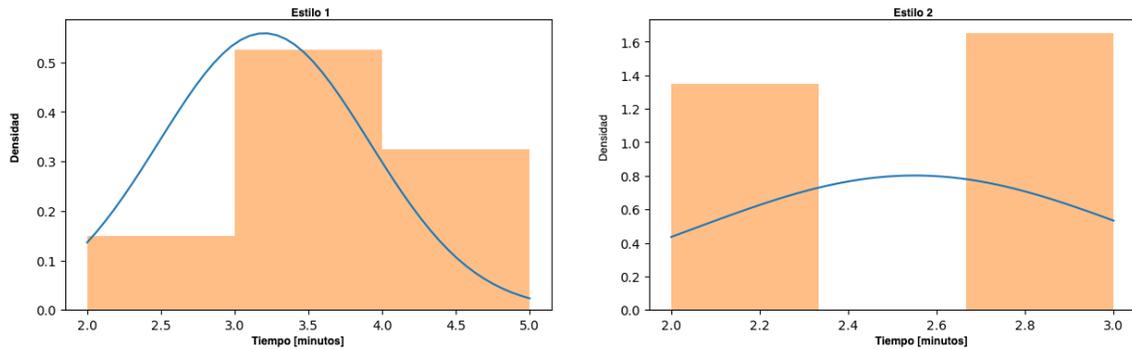
            fin = self.env.now
            self.tiempo_proceso += (fin-fin_tiempoCola)

        df_tmp = pd.DataFrame({"tipo" : [4], "id" :[id_tarea], "inicio" : [inicio], "fin" : [fin] })
        self.df_metricas = self.df_metricas.append(df_tmp)
        if self.pipe != {}:
            data = (1,estilo,cantidad,id_tarea)
            self.pipe.put(data)
```

Fuente: Elaboración propia

Así mismo, para las actividades de suela y plantilla se determinaron las funciones de distribución normal del tiempo de producción para un par (figura 3-7 y 3-8), en este caso para ambas actividades se hizo distinción entre el tipo de horma.

Figura 3-7: Distribución normal para la actividad de suela.

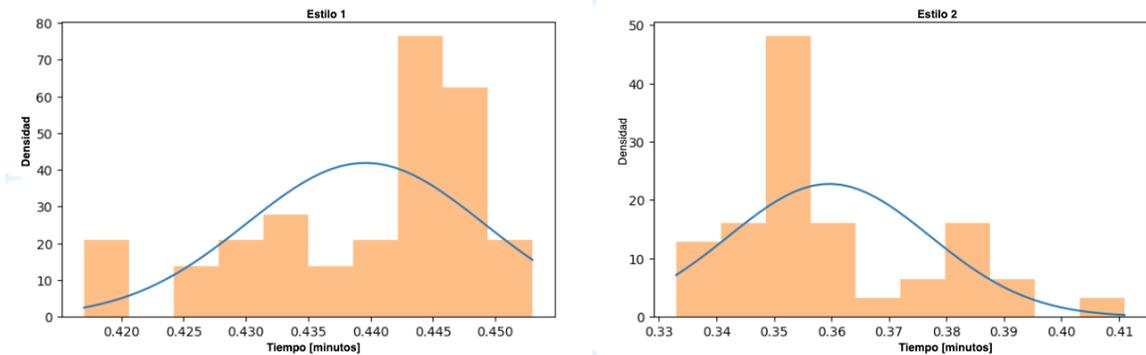


Fuente: elaboración propia

Tabla 3-5 : Distribución suelas

Distribución Suelas						
nombre	cantidad	media	std	mínimo	máximo	p_value
estilo 1	40	3,20	0,71414	2,0	5,0	0,002248
estilo 2	40	2,55	0,49749	2,0	3,0	0,000024

Figura 3-8: Distribución normal para la actividad de plantilla.



Fuente: elaboración propia

Tabla 3-6: Distribución plantillas

Distribución Plantillas						
nombre	cantidad	media	std	mínimo	máximo	p_value
estilo 1	40	0,43957	0,00952	0,417	0,453	0,04108
estilo 2	40	0,35965	0,01752	0,333	0,411	0,04597

Como se mencionó anteriormente estas dos actividades son similares, por lo tanto, como se ve en la figura 3-9, en la clase **Suela_Plantilla** se implementa únicamente el método **generador_actividad**, para ambas actividades, donde según el tipo (2 suela y 3 plantilla) se calcula el tiempo de la actividad y se agrega al ambiente de simulación, se adicionan las métricas y por último se pasa los datos de la actividad al objeto pipe.

Figura 3-9: Método generador actividad suela o plantilla.

```
class Suela_Plantilla():

    def generador_actividad(self,orden):

        inicio_tiempoCola = self.env.now

        with self.capacidad.request() as req:

            yield req
            fin_tiempoCola = self.env.now

            self.tiempoColas += (fin_tiempoCola - inicio_tiempoCola)
            inicio = self.env.now
            id_actividad = orden["id"]
            cantidad = orden["cantidad"]

            id_producto = 0
            if self.tipo == 2 :

                tiempo_actividad = sum(
                    [random.gauss(orden["suela_media"],orden["suela_desv"]) for i in range(cantidad)]
                )

                id_producto = orden["id_suela"]

            elif self.tipo == 3 :
                tiempo_actividad = sum(
                    [random.gauss(orden["plantilla_media"],orden["plantilla_desv"]) for i in range(cantidad)]
                )
                id_producto = orden["id_plantilla"]

            yield self.env.timeout(tiempo_actividad)

            fin = self.env.now

            self.tiempo_proceso += (fin-fin_tiempoCola)

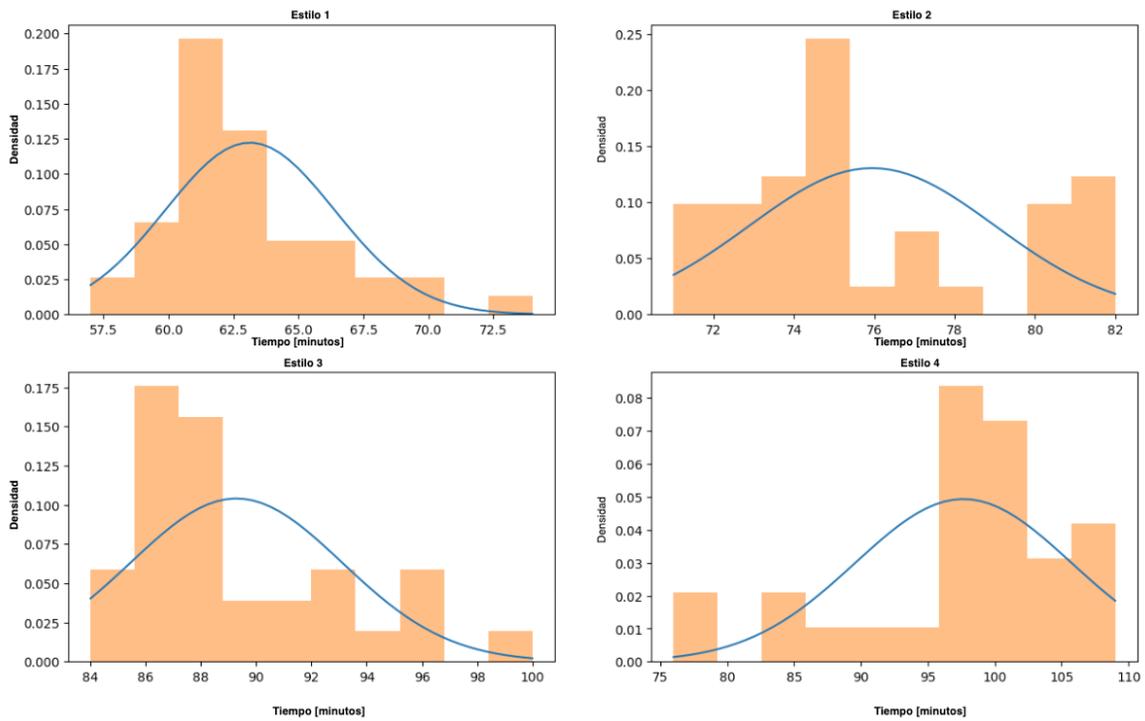
            df_tmp = pd.DataFrame(
                {"tipo" : [self.tipo] ,
                 "id" :[id_actividad],
                 "inicio" : [inicio],
                 "fin" : [fin] })
            self.df_metricas = self.df_metricas.append(df_tmp)

            if self.pipe != {}:
                data = (self.tipo,id_producto,cantidad,id_actividad)
                self.pipe.put(data)
```

Fuente: elaboración propia

Para finalizar con la simulación, se implementó la actividad zapato, que depende de las actividades de suela, plantilla y guarnición, ya que consiste en el ensamble del zapato, para esta actividad también se generó la función de distribución normal del tiempo necesario para armar un par de zapatos (figura 3-10) . En este caso se agrupan los estilos de zapatos por el color del cuero y por el tipo de calzado (zapato, bota) ya que esto infliere en el proceso y por lo tanto en el tiempo.

Figura 3-10: Distribución normal para la actividad de zapato.



Fuente: elaboración propia

Tabla 3-7 : Distribución zapatos

Distribución Zapatos						
nombre	cantidad	media	std	mínimo	máximo	p_value
estilo 1	45	63,13333	3,26326	57,0	74,0	0,015826
estilo 2	37	75,94595	3,05752	71,0	82,0	0,020711
estilo 3	32	89,28125	3,83434	84,0	100,0	0,024642
estilo 4	29	97,65517	8,09119	76,0	109,0	0,036956

En la imagen 3-11, para la clase `Calzado`, se implementa el método `iniciar_calzado`, donde el objeto `pipe` espera hasta que alguna actividad agregue información, en ese momento se almacena la información en el objeto `almacen`, donde se validará si se puede iniciar la actividad de ensamble del calzado, generando una actividad con el método `generador_calzado`.

El método `generador_calzado` agrega el tiempo de la actividad al ambiente de simulación y agrega las métricas al dataframe de métricas.

Figura 3-11: Método generador actividad para calzado.

```
class Calzado():

    def generador_actividad(self, tiempo, id, tiempoCola):

        inicio = self.env.now
        with self.zapateros.request() as req:
            yield req
            yield self.env.timeout(tiempo)
            fin = self.env.now
            df_tmp = pd.DataFrame({"tipo" : [5], "id" :id, "inicio" : [inicio], "fin" : [fin] })
            self.df_metricas = self.df_metricas.append(df_tmp)

    def iniciar_calzado(self):

        almacen = Almacen(self.df_estilo, self.env)

        while True:
            data = yield self.pipe.get()
            almacen.almacenar(data[0], data[1], data[2])
            proceso, tiempoCola = almacen.validar_completo()
            self.tiempo_proceso += proceso
            self.tiempo_colas += tiempoCola

            if proceso != 0:
                c = self.generador_actividad(proceso, data[3], tiempoCola)
                self.env.process(c)
```

Fuente: elaboración propia

3.4 Evaluación de la simulación

Antes de realizar el entrenamiento es necesario establecer una métrica o índice que nos permita determinar qué tan favorable es la distribución de la orden, en este caso en particular en todas las actividades se calculó el tiempo utilizado para el proceso como también el tiempo en colas.

Por tal motivo se define el índice como la relación entre el tiempo en cola sobre el tiempo total (tiempo en proceso + tiempo en cola).

Como se puede observar en la figura 3-12, se implementa el cálculo del índice en la clase **Simulacion_calzado** con el método **get_indice** :

Figura 3-12: Método get índice

```
class Simulacion_calzado():
    def get_indice(self):
        indice = self.tiempo_colas / (self.tiempo_colas + self.tiempo_proceso )
        return indice
```

Fuente: elaboración propia

3.5 Ejecución del modelo

3.5.1 Convergencia

Se ejecuto la simulación de 1840 pares con los siguientes parámetros:

Tabla 3-8: Parámetros de simulación

Parámetro	Descripción	Valor
Capacidad del recurso corte	Cantidad ordenes que se pueden hacer en paralelo en el proceso de corte	20
Capacidad del recurso guarnición	Cantidad ordenes que se pueden hacer en paralelo en el proceso de guarnición	20
Capacidad del recurso línea de producción suela	Cantidad ordenes que se pueden hacer en paralelo en el proceso de suela	1
Capacidad del recurso línea de producción plantilla	Cantidad ordenes que se pueden hacer en paralelo en el proceso de plantilla	1
Capacidad del recurso zapato	Cantidad ordenes que se pueden hacer en paralelo en el proceso de zapato	15

En cada iteración de la simulación, se determina la cantidad de pares que se deben producir diariamente, así como el parámetro de índice que representa la relación entre los tiempos de espera y los tiempos de proceso. Mediante ejecuciones exploratorias, se observó que el número de días calculados se mantiene constante.

Por lo tanto, se establece el parámetro de convergencia como el índice, con el fin de determinar cuántas iteraciones se deben realizar antes de detenerse.

Para calcular el criterio de convergencia, se obtiene la diferencia absoluta entre el índice actual y el índice de la iteración anterior, y se divide entre el índice de la iteración actual. Este cálculo permite evaluar la estabilidad del índice a lo largo de las iteraciones y determinar si se ha alcanzado la convergencia.

Considerando lo anterior se define un límite de 0.04%, al realizar múltiple ejecuciones se evidencia que la simulación converge en 9 iteraciones.

3.5.2 Validación del modelo

Para validar los resultados de la simulación, se realiza un t-test comparando la distribución de pares diarios simulada con la distribución real. Este análisis permite determinar si hay una diferencia significativa entre ambas distribuciones, lo que proporciona una medida de la precisión de la simulación en relación al fenómeno real.

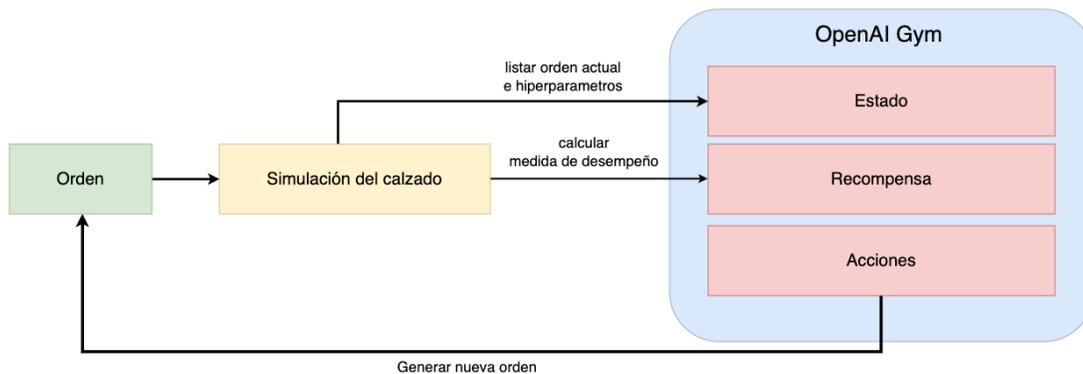
Distribución producción de pares por día																				
Real	65	92	94	82	92	86	91	93	90	90	95	94	81	91	82	94	86	92	80	170
Simulación	73	110	101	106	103	107	103	104	105	104	106	103	107	105	103	103	104	93	-	-
Resultado p_value = 0.02669																				

Debido a que el resultado obtenido es menor al 5%, se puede afirmar que la simulación es válida y proporciona una estimación confiable del fenómeno real.

3.6 Ambiente personalizado openIA gym

Para implementar las políticas utilizando un agente entrenado con algoritmos de aprendizaje por refuerzo, es necesario convertir el ambiente de simulación en un entorno personalizado compatible con OpenAI Gym. A continuación el diagrama de la integración.

Figura 3-13: Diagrama integración



La integración implica definir tres elementos principales: el estado, las acciones y la recompensa. Aquí se presenta un resumen de cada uno de ellos:

- **Definición del estado:** El estado se representará preferentemente como un vector unidimensional que contenga la información necesaria para que el agente pueda aprender. En este caso particular, el estado estará compuesto por la cantidad de pares para cada estilo de calzado, donde la posición en el vector representa el estilo específico. Además, se incluirán los hiperparámetros relacionados con la cantidad de recursos disponibles para cada proceso.
- **Definición de las acciones:** Es importante considerar las restricciones de las acciones en el contexto de la simulación. Por defecto, las acciones se representarán como números enteros. Con esto en mente, se puede crear un diccionario donde las claves corresponden a los números de las acciones y los valores son arreglos que representan la distribución de la orden de producción.

Al seleccionar una acción, se actualizará la orden de producción y se utilizará en la siguiente iteración de la simulación. Esto significa que la acción elegida afectará directamente el proceso de producción y tendrá un impacto en los resultados de la simulación.

- **Definición de la recompensa:** La recompensa es fundamental para que el agente aprenda, en este caso se quiere minimizar los tiempos muertos (en cola), por tal motivo la recompensa será el inverso del índice, de este modo entre más grande sea el índice menor será la recompensa.

Se crea la clase `Calzado_env` a partir de la clase `gym.Env`, se deben implementar los métodos de `reset` y `step`.

Desde la imagen 3-13, el método `reset` será el encargado de reiniciar el ambiente, cada vez que se ejecute se generará una orden nueva y se retornará como una observación.

Figura 3-14: Método `reset`.

```
class Calzado_env(gym.Env):  
  
    def reset(self):  
  
        self.orden.orden = self.orden.generar_datos()  
        self.orden.run()  
        self.pasos = 0  
        self.observation = np.array([dato[1] for dato in self.orden.orden])  
        return self.observation
```

Fuente: Elaboración propia

El método **step**, implementado como en la figura 3-14, recibe la acción que debe ejecutar el agente, esa acción es un número entero, por lo tanto se debe transformar con la ayuda de un diccionario que tiene como llave la acción y como valor un arreglo con la distribución de la orden, luego con la nueva distribución de la orden se realiza la simulación y se calcula el índice y la recompensa que será el inverso del índice, para finalizar se determina si se debe terminar el entrenamiento del agente.

Figura 3-15: Método step.

```
class Calzado_env(gym.Env):

    def step(self, action):
        self.cambiar_orden(action)

        self.simulacion = Simulacion_calzado(
            self.orden.orden_corte,
            self.orden.orden_suela,
            self.orden.orden_plantilla,
            self.orden.df_estilo)

        self.simulacion.generar_simulacion()
        info = {"paso" : self.pasos , "accion" : action}
        self.reward = (1 / self.simulacion.get_indice())
        self.observation = np.array([dato[1] for dato in self.orden.orden])
        if self.simulacion.get_indice() > parametros.indice_minimo :
            self.done = True
        else :
            self.done = False
        self.orden.orden = self.orden.generar_datos()
        return self.observation, self.reward, self.done, info
```

Fuente: Elaboración propia

4. Optimización

4.1 Entrenamiento

El objetivo principal de la optimización es reducir el tiempo de espera en las colas de los procesos mediante la implementación de políticas aprendidas a partir de simulaciones.

Para lograr esto, se utilizará el enfoque de aprendizaje por refuerzo, que permitirá generar un modelo capaz de predecir el orden de las tareas.

La optimización de la simulación se llevará a cabo utilizando el algoritmo de aprendizaje por refuerzo de la librería Stable Baselines 3. Esta librería simplifica el proceso de entrenamiento de estos algoritmos, proporcionando una interfaz más comprensible para trabajar.

El procedimiento para poder entrenar un agente en esta librería inicia seleccionando el ambiente de simulación. En este caso se utilizará el ambiente descrito en el capítulo anterior, luego se debe escoger el algoritmo que se va a utilizar para el entrenamiento.

Posteriormente empieza la etapa de entrenamiento, donde el agente empieza a interactuar con el ambiente mediante acciones. En cada paso, el agente observa el estado actual del entorno, elige una acción utilizando su política de toma de decisiones y la ejecuta en el entorno. Luego, el agente recibe una recompensa del entorno y actualiza su conocimiento interno y su política de acciones.

Este ciclo de interacción, observación, toma de decisiones, ejecución de acciones, recepción de recompensas y actualización se repite iterativamente durante el

entrenamiento. A medida que el agente acumula experiencia y recibe retroalimentación del entorno, ajusta sus acciones y políticas para maximizar las recompensas obtenidas.

En este caso, se sigue el procedimiento descrito en la Figura 4-1. Para entrenar el modelo, se inicia creando un objeto de tipo "orden". Luego, se crea un ambiente personalizado utilizando la clase "Calzado_env". Finalmente, se importa desde el módulo Stable Baselines 3 el modelo correspondiente al algoritmo seleccionado.

Para iniciar el entrenamiento, se proporciona al modelo la política a seguir, el ambiente de simulación y otros parámetros adicionales. Entre estos parámetros se encuentra "verbose", que permite mostrar mensajes informativos durante la ejecución del entrenamiento, y "tensorboard_log", que permite recopilar métricas y registros del entrenamiento para su posterior análisis o total_timesteps que corresponde al número total de pasos que se van a ejecutar en el ambiente de simulación.

Figura 4-1: Entrenamiento de los agentes.

```
orden = Generador_ordenes(path_estilo,path_suela,path_plantilla)
env=Calzado_env(orden)
env.reset()

if tipo=="ppo" :
    model = PPO('MlpPolicy', env, verbose=1,tensorboard_log=path_logs)
else :
    model = A2C('MlpPolicy', env, verbose=1,tensorboard_log=path_logs)

TIMESTEPS = 10000

for i in range(30):
    model.learn(total_timesteps=TIMESTEPS, reset_num_timesteps=False, tb_log_name=nombre_modelo)
    model.save(f"{path_modelos}/{TIMESTEPS*i}")

env.close()
```

Fuente: Elaboración propia

Se procede al entrenamiento de cuatro modelos, dos utilizando el algoritmo PPO y dos utilizando el algoritmo A2C.

Cada modelo se configura con una tasa de aprendizaje diferente, mientras que los demás parámetros se mantienen en sus valores por defecto. A continuación, se detallan las tasas de aprendizaje asignadas a cada modelo:

Tabla 4-1 :Tasas de aprendizaje a cada modelo

Modelo	Algoritmo	tasa de aprendizaje
ppo_0	PPO	0,001
ppo_06_a_0	PPO	0,0001
A2C_0_0_0	A2C	0,0002
a2c_0	A2C	0.0007

Después de 54500 iteraciones se tiene la métrica de la recompensa para cada uno de los modelos, a continuación, una tabla:

Tabla 4-2 : Métrica de recompensa

Modelo	Recompensa máxima
ppo_0	13,17
ppo_06_a_0	13,44
A2C_0_0_0	8,78
a2c_0	9,96

4.2 Análisis de resultados

Se utilizaron dos algoritmos diferentes para el entrenamiento de los modelos: PPO (Proximal Policy Optimization) y A2C (Advantage Actor-Critic).

PPO es un algoritmo de optimización de políticas que utiliza una aproximación ascendente del gradiente para actualizar y mejorar la política del agente. Su enfoque principal es mantener una actualización "proximal" de la política, lo que significa que los cambios en la política se mantienen dentro de ciertos límites para asegurar una mayor estabilidad en el proceso de entrenamiento.

Por otro lado, A2C es un algoritmo que combina elementos del método Actor-Critic. El "actor" representa la política del agente, que selecciona las acciones, mientras que el "critic" estima el valor esperado de las acciones tomadas por el actor. A2C utiliza una

estructura en la que el actor y el crítico comparten una red neuronal, lo que permite una actualización simultánea de ambos componentes. Esto mejora la eficiencia del entrenamiento y facilita la convergencia del modelo.

Para el análisis de resultados se procede a organizar órdenes generadas aleatoriamente desde la clase **Generador_Orden**, posteriormente se organizaron mediante 3 metodologías, de manera manual, y mediante los dos algoritmos de PPO

Tabla 4-3: Resultados indicie promedio para diferentes algoritmos

Algoritmo	Recompensa
manual	7.2
ppo_06_a_0	7.19
ppo_0	7.5

5. Conclusiones y recomendaciones

5.1 Conclusiones

- Se logró implementar un modelo para la simulación de la producción manual del calzado masculino en Bogotá, Colombia. considerando las necesidades del negocio se realizó a nivel de macro actividades (corte, guarnición, suela, plantilla, zapato).
- Se logró caracterizar las variables para la simulación de la producción manual de calzado masculino, en dos fases.

La primera fase consistió en determinar el alcance y el nivel de detalle de la simulación y la segunda fase consistió en determinar los componentes que intervienen en cada una de las actividades.

- Se logró adaptar el ambiente de simulación para el entrenamiento de los agentes por aprendizaje por refuerzo, definiendo las observaciones como la distribución de la orden en un arreglo y la recompensa como el porcentaje de tiempo en cola.
- Después de realizar varias simulaciones (1000) y validar con diferentes algoritmos, se llega a la conclusión que el modelo que se debe implementar para que realice la organización de las órdenes de producción es el agente entrenado con el algoritmo PPO con una tasa de aprendizaje de 0,001 .
- La mejor forma para administrar el código de la simulación y la optimización es mediante un repositorio de git (<https://github.com/lgdiazf-unal/Simulacion-calzado>) y utilizar el paradigma de la programación orientada a objetos.

5.2 Recomendaciones

Este trabajo sirve como base en investigaciones futuras que tengan como objetivo optimizar la producción del calzado, integrando internet de las cosas en la línea de producción.

Por otro lado, se podría fortalecer la definición de la recompensa dentro del ambiente de simulación, ya que se implementó pensando específicamente en los estilos que trabaja el negocio, se recomienda explorar la implementación de un extractor de características.

Bibliografía

Selman ERYILMAZ, M., Osman KUŞAKCI, A., & GAVRANOVIC, H. (2012). Analysis Of Shoe Manufacturing Factory By Simulation Of Production Processes. Southeast Europe Journal of Soft Computing.

Qin, J., Hu, F., Liu, Y., Witherell, P., Wang, C., Rosen, D., . . . Tang, Q. (2022). Research and application of machine learning for additive manufacturing. Additive Manufacturing.

Heilala, J., Vatanen, S., Tonteri, H., Montonen, J., Lind, S., Johansson, B., & Stahre, J. (2008). SIMULATION-BASED SUSTAINABLE MANUFACTURING SYSTEM DESIGN. Winter Simulation Conference.

Paternina-Arboleda, C., Montoya-Torres, J., & Fábregas-Ariza, A. (2008). SIMULATION-OPTIMIZATION USING A REINFORCEMENT LEARNING APPROACH. Winter Simulation Conference.

Samsonov, V., Hicham, K., & Meisen, T. (2022). Reinforcement Learning in Manufacturing Control: Baselines, challenges and ways forward. Engineering Applications of Artificial Intelligence.

Calderón Andrade, R., Hernández Gress, E. S., & Montufar Benítez, M. (2020). Productivity Improvement through Reengineering and Simulation: A Case Study in a Footwear-Industry. Hidalgo: applied science.

Kuhnle, A., Kaiser, J.-P., Theiß, F., Stricker, N., & Lanza, G. (2021). Designing an adaptive production control system using reinforcement learning. Journal of Intelligent Manufacturing, 855–876.

Daaboul, J., Da Cunha, C., Bernard, A., & Laroche, F. (2011). Design for mass customization: Product variety vs. process variety. *CIRP Annals - Manufacturing Technology*, 169-174.

Diaz-Elsayed, N., Jondral, A., Greinacher, S., Dornfeld, D., & Lanza, G. (2013). Assessment of lean and green strategies by simulation of manufacturing systems in discrete production environments. *CIRP Annals - Manufacturing Technology*, 475-478.

Rabelo, L., Held, M., Son, Y.-J., Jones, A., Mim, J., & Deshmukh, A. (2003). A HYBRID APPROACH TO MANUFACTURING ENTERPRISE SIMULATION. *Winter Simulation Conference*.

Hernández-González, S., Ramírez-Tapia, R., & Jiménez-García, J. (2019). Analysis of the Productivity of a Shoe Production Line—Application of Queueing Theory and Lean Manufacturing. *Best Practices in Manufacturing Processes*, 367-388.

Hotz, N. (2022, Abril 16). What is CRISP DM? Retrieved from Data Science Process Alliance: <https://www.datascience-pm.com/crisp-dm-2/>