UNIVERSIDAD
Panamericana

Alumno:

Eduardo Ulises Martinez Vaca

Brandon Mitzrael Magaña Avalos

Fernanda Mena Santos

Trabajo: Project Midterm II

Materia: Simulación Gráfica

Profesor: Gabriel Castillo Cortés

Fecha de entrega: 11/04/2025

Introduction

In a modern industrial world, the ability to monitor and understand production dynamics in real-time is crucial for ensuring efficiency, maintaining its quality and having a rapid response to disruptions. The traditional reporting systems have now fallen short limiting the capacity of managers and engineers to make good decisions with the given data.

This paper seeks to detail the analysis, data preparation and design decisions that went into the development of an interactive visualisation tool that focuses on industrial production. The dashboard created allows us to explore key metrics such as production, accidents, rejects and efficiency at different time periods (daily, weekly, monthly, quarterly and yearly).

The simulation is implemented in the EUMV_FMS.py file using the SimPy discrete-event simulation library. Each product moves through a sequence of six stations. Stations 0 to 3 are processed sequentially, while stations 4 and 5 operate in parallel.

To reflect the complexity of real-world production systems, the simulation captures a wide array of performance indicators. These indicators include:

- Busy time (occupancy)
- Downtime (failures and maintenance)
- Waiting times in queue
- Repairs (including fixing time)
- Rejected products (quality control failures)
- Randomly occurring facility-level accidents

By simulating 364 full production cycles, each one representing an operational day, the model emulates an entire year of activity, providing a simulated dataset for analysis.

Project Objectives

The core objectives of this project are:

- To simulate a realistic industrial production process using discrete-event modeling
- To track and analyze station-level data to identify bottlenecks and operational inefficiencies
- To present the results through an interactive and visually accessible dashboard
- To support managerial decision–making by offering multi-scale metrics and visual storytelling.

Enhancements Implemented During Laboratory Sessions (Lab D3)

The original simulator only provided global metrics. To support more granular reporting and visualization, station-level metrics were added. This includes the following fields:

- good_products: units successfully processed and accepted
- rejected_products: units discarded due to quality issues
- accident_count: number of accidents per station

These enhancements allow for detailed performance analysis by station, capturing both output and operational risk.

Exporting Simulation Results to JSON

To facilitate integration with a dashboard, results from the 365 simulation runs are consolidated into a structured JSON format. The export logic is implemented in the plotting.py file through the generate_station_json function.

The function was modified to:

- Use each of the 365 daily simulation results individually, avoiding statistical averaging.
- Aggregate data by period using daily values:
  - Daily: day 0
  - Weekly: days 0–6
  - Monthly: days 0–29
  - Quarterly: days 0–89
  - Yearly: days 0–364

For each period, the following metrics are calculated:

- Total production (good products)
- Total rejected units
- Occupancy hours (converted from station occupancy rates)
- Average fixing time
- Rejection percentage
- Average downtime or delay
- Number of recorded accidents

These values are grouped by station and serialized to JSON for use in the dashboard.

Interactive Dashboard

An HTML-based interactive dashboard was created to visualize the results of the production simulation results. The dashboard is designed to facilitate an exploratory analysis of key performance indicators

Architecture and Data Flow

The dashboard operates by reading a JSON file that was generated at the end of the simulation code. This file contains the performance data grouped by time period (daily, weekly, monthly, quarterly, yearly) and the station. This data is then loaded dynamically using JavaScript and it is rendered into visual components by using D3.js and Chart.js

- **HTML + Bootstrap:** For layout and design
- **D3.js + Chart.js:** For interactive charts
- **JavaScript:** For loading data, updates and the user interaction

Functionalities

The dashboard includes the following interactive components:

- **Period-based filtering:** Users can check different time resolutions such as day, week, month, quarter and year of work using navigation buttons.
- **Per-station metric:** All charts support analysis of individual stations (Downtime, rejections and incidents)
- **Dynamic updates:** The dashboard reflects the data automatically (Supporting new data as well) without requiring a manual refresh.

Layout

The interface is divided into two primary sections:

- **Main content area**: Includes the title, period navigation controls, main production graph, and secondary metric buttons with chart containers.

- **Sidebar (KPI cards)**: Displays critical summary metrics.

Each secondary chart is hidden or revealed based on user interaction, ensuring the interface remains clean and focused on one metric at a time.

Graphs Visualization

Main graph - Production

- Type: Horizontal bars
- Justification: Bars allow for easy comparison between the stations
- Key Features: Custom colours, labels, and rounding for a modern aesthetic

Secondary graphs

Metrics	Type of graph	Justification
Rejection	Pie Graph	To show distribution by station in a clear way.
Delay	Bar Graph	Simple comparison of numerical values.
Accidents	Line Graph	Ideal for viewing time trends.
Occupation Hours	Area Graph	Visually reinforces volume and continuity.
Rejection %	Pie Graph	Proportional relationship between stations.

Graphs considered but discarded

- Scatter Plot : It was evaluated for correlations between production and efficiency. Even though it was a good way to represent the information, it was discarded due to lack of granularity in the data and it is a visually complicated interface.

Preparación y Estructura de los Datos

Initial Development (First Midterm)

- Implemented the discrete-event simulator SimPy.
- Modeled six workstations with distinct processing and failure characteristics.
- Integrated global tracking for production and rejected items.

Enhancements using D3 Labs

- Refactored the simulator to include station-level data:
  - Added counters for accepted, rejected, and failed units per station.
  - Tracked accident events and assigned them to specific stations.
- Revised data export:
  - Moved from average-based aggregation to true daily-recorded values.
  - Grouped daily data to calculate weekly, monthly, and yearly metrics.
- Improved accuracy in reporting:
  - Downtime is now reported correctly as avg_delay_minutes.
  - Accident counts are preserved even when a run ends prematurely due to failure.

Visualization Phase

- Built an interactive dashboard connected to the generated JSON.
- Enabled automatic refresh and period-based filtering.
- Designed plots for occupancy, downtime, quality metrics, and incidents.

Final Validation

- Ran the simulation for 365 days to simulate one full year.
- Validated that:
  - All downtime and accidents were correctly tracked per station.
  - The rejection rate and production output matched expectations.
  - The dashboard displayed accurate, disaggregated data dynamically.

Conclusions

This project successfully integrates simulation, data preparation, and interactive visualization to deliver insights on manufacturing efficiency. The dashboard serves as a powerful decision-making tool, visually representing complex system dynamics. Although some fields like accidents remain placeholders, the framework is ready for more advanced logic.

Overall, this solution transitions the simulation from a back-end technical tool into a managerial decision support system aligned with data storytelling best practices.