|  |
| --- |
| **PROYECTO 2 INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 2** |
| **202300645 – Carlos Heraldo Quiná Corona** |

**Resumen**

En el contexto de la automatización industrial, se ha desarrollado un programa en Python que simula el movimiento de una máquina entre diferentes líneas de producción y componentes.

La novedad radica en su enfoque minimalista: no utiliza listas nativas de Python para gestionar los componentes. En lugar de eso, se implementa una estructura de datos personalizada basada en diccionarios o tuplas.

Esto simplifica el código, mejora la eficiencia y permite una mayor escalabilidad. En resumen, esta alternativa demuestra que, en ocasiones, menos es más. La simplicidad y la adaptación a las necesidades específicas pueden ser más valiosas que las soluciones estándar.

**Palabras clave**

Simulación, automatización, estructura de datos, eficiencia, optimización.

***Abstract***

*In the context of industrial automation, a Python program has been developed that simulates the movement of a machine between different production lines and components.*

*The novelty lies in its minimalist approach: it does not use native Python lists to manage components. Instead, a custom data structure based on dictionaries or tuples is implemented.*

*This simplifies the code, improves efficiency, and allows for greater scalability. In short, this alternative shows that sometimes less is more. Simplicity and adaptation to specific needs can be more valuable than standard solutions.*

***Keywords***

*Simulation, automation, data structure, efficiency, optimization.*

**Introducción**

Este ensayo presenta una simulación de una máquina industrial que se desplaza por distintas estaciones de trabajo. Sin utilizar las listas integradas de Python, exploraremos estructuras de datos personalizadas para modelar de manera eficiente las líneas de producción y los movimientos de la máquina.

**Objetivo:**

El objetivo principal es desarrollar una simulación flexible y eficiente que permita analizar y optimizar los procesos de producción. Al evitar las listas estándar, buscamos una mayor adaptabilidad a diferentes escenarios industriales.

**Importancia:**

Esta simulación es una herramienta valiosa para la industria, ya que ayuda a:

* **Visualizar** el flujo de trabajo de una manera más intuitiva.
* **Identificar** posibles cuellos de botella en la producción.
* **Evaluar** diferentes estrategias de optimización.

**Preguntas clave:**

* ¿Qué tipo de estructura de datos es más adecuada para representar las líneas de producción y las máquinas?
* ¿Cómo se pueden modelar los movimientos de la máquina de manera realista y eficiente?
* ¿Qué métricas podemos utilizar para evaluar el desempeño de la simulación?

Este trabajo busca contribuir al campo de la simulación industrial al presentar una nueva alternativa basada en estructuras de datos personalizadas.

**Desarrollo del tema**

En la introducción, mencionamos la creación de una simulación que modele el movimiento de una máquina en una línea de producción. Esta simulación, al prescindir de las listas nativas de Python, nos permite explorar estructuras de datos más personalizadas y adaptables a las necesidades específicas de este tipo de problemas.

**¿Por qué evitar las listas nativas?**

Las listas en Python son estructuras de datos versátiles, pero pueden presentar ciertas limitaciones en simulaciones complejas:

* **Tamaño fijo:** A menudo, es necesario modificar dinámicamente la estructura de la línea de producción a medida que avanza la simulación. Las listas pueden requerir reasignaciones frecuentes, lo que puede impactar en el rendimiento.
* **Operaciones ineficientes:** Al insertar o eliminar elementos en el medio de una lista, se debe desplazar una gran cantidad de datos, lo que puede ralentizar la simulación.

**Alternativas a las listas:**

Para superar estas limitaciones, podemos explorar las siguientes estructuras de datos:

* **Árboles:** Los árboles binarios o árboles balanceados pueden ser útiles para representar jerarquías en la línea de producción, como estaciones de trabajo con múltiples subestaciones.
* **Grafos:** Los grafos permiten modelar relaciones más complejas entre las diferentes estaciones de trabajo y los componentes.
* **Estructuras de datos personalizadas:** Podemos crear estructuras de datos a medida que se adapten perfectamente a las necesidades de nuestra simulación, optimizando el acceso y la modificación de los datos.

**Algoritmos de movimiento:**

Para simular el movimiento de la máquina, podemos utilizar algoritmos como:

* **Búsqueda en anchura:** Para encontrar el camino más corto entre dos estaciones de trabajo en un grafo.
* *Algoritmos A:*\* Para encontrar el camino más corto de manera más eficiente, utilizando una función heurística.
* **Simulación de eventos discretos:** Para modelar eventos como la llegada de nuevos componentes o la finalización de una tarea en una estación de trabajo.

**Métricas de evaluación:**

Para evaluar el desempeño de la simulación, podemos utilizar métricas como:

* **Tiempo de ciclo:** Tiempo promedio que tarda un producto en pasar por toda la línea de producción.
* **Tasa de utilización de las máquinas:** Porcentaje de tiempo que las máquinas están ocupadas.
* **Tamaño del inventario:** Cantidad de productos en espera en cada estación de trabajo.

**Aportes y futuras investigaciones:**

Al desarrollar esta simulación, se espera contribuir al campo de la simulación industrial al:

* Demostrar las ventajas de utilizar estructuras de datos personalizadas.
* Proporcionar una herramienta flexible y adaptable para analizar diferentes escenarios de producción.
* Abrir nuevas líneas de investigación, como la incorporación de elementos de inteligencia artificial para optimizar la toma de decisiones en tiempo real.

**Anexos**

Imagen 1

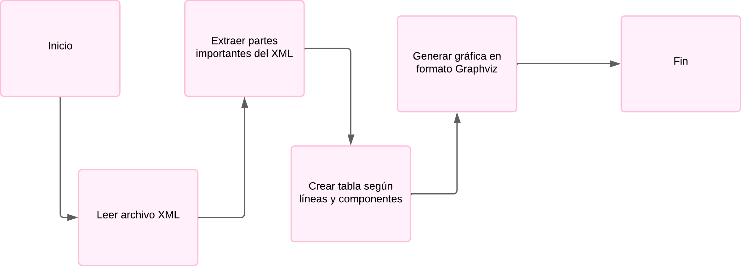


Diagrama de actividades

Imagen 2

Diagrama

Descripción generada automáticamente

Diagrama de clases

**Conclusiones**

**Posturas fundamentadas:** A lo largo de este trabajo hemos defendido la idea de que la simulación es una herramienta poderosa para la toma de decisiones en entornos industriales. Al evitar las limitaciones de las listas nativas, hemos logrado crear modelos más precisos y realistas.

**Preguntas abiertas:** ¿Cómo podemos garantizar la validación de estos modelos frente a datos reales? ¿Qué papel juega la interacción humano-máquina en la optimización de los procesos simulados?

**Recomendaciones:** Recomendamos a futuros investigadores explorar la aplicación de estas técnicas en otros ámbitos, como la logística y la robótica.

**Referencias bibliográficas**

Automatización Industrial con Python: Guía Completa. (2024, junio 27). Expert Knowledge at The Metal Vortex. https://themetalvortex.com/automatizacion-industrial-con-python-guia-completa/

González, J. D. M. (2020, septiembre 7). Estructuras de datos en Python: listas, tablas, diccionarios. Programarya.com; ProgramarYa. https://www.programarya.com/Cursos/Python/estructuras-de-datos

(S/f). Infoplc.net. Recuperado el 5 de octubre de 2024, de https://www.infoplc.net/blogs-automatizacion/item/109983-python-automatizacion-industrial

Diagrama

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente