
SOLUCIÓN SIMULACIÓN DE MANEJO A LA MÁQUINA CREADA POR DIGITAL INTELLIGENCE S.A PARA OPTIMIZAR EL TIEMPO DE ENSAMBLAJE MEDIANTE EL USO DE GUI Y TDA'S

Carnet 1 – Nombre completo del estudiante

Resumen

Se desarrolló la solución en un algoritmo capaz de simular el manejo de la máquina con la mayor eficiencia, permitiendo esto ensamblar los productos en el menor tiempo posible, gracias al archivo de configuración con formato xml, se obtiene la cantidad de líneas de ensamblaje, los componentes disponibles en cada línea y el tiempo de ensamblaje de estas. Mediante el modelo de FIFO, se grafica el proceso de ensamblaje de cada producto. La solución fue desarrollada en el lenguaje de programación Python, con el uso de varias librerías, TDA's y QT Designer para el desarrollo de la GUI. Tras ser cargado el archivo, los productos están disponibles para que se realice su ensamble al presionar un botón, por otra parte, para hacer un ensamblaje masivo se debe cargar otro archivo xml, con los artículos a producir. Esta solución fue posible gracias al manejo de los TDA's y la POO que permitió realizar un manejo más eficiente de los datos.

Palabras clave

- TDA
- FIFO
- GUI

Abstract

We've developed a solution through an algorithm that can simulate the function of the machine in a way that makes it the most efficient, leading this to assemble the products in the shortest time possible. Thanks to the configuration file in xml format, we're able to obtain the quantity of assembling lines, the components for each line and the assembly time. We also used the FIFO model for graphing the process of assembly for each product. The solution was developed in the Python programming language, with the help from some libraries, ADTs, and QT designer for the GUI design. After loading the files, the products will become available for assembling by pressing a button, also you can make a mass assembly you got to load another xml file with the products to assemble. This solution was possible with the correct use of the ADTs and de OOP that allowed a more efficient handling of the data.

Keywords:

- ADT
- FIFO
- GUI

Introducción

En este informe será planteada la solución al problema presentado en el proyecto, el cual fue solucionado mediante el uso de una interfaz gráfica realizada con QT designer, el lenguaje de programación Python, la aplicación de programación orientada a objetos y el manejo de los TDAs también conocidos como “Tipos de datos abstractos”

El paradigma de programación POO nos permitió crear varias clases, entre las cuales se manejaron los TDAs, dándonos esto un código eficiente en el manejo de los objetos y el uso de funciones específicas de cada clase.

Las clases que fueron implementadas son, una clase principal, tres nodos para el manejo de los distintos datos y tres listas simples para mover los datos.

Desarrollo del tema

Se nos presenta la siguiente problemática:

“La empresa Digital Intelligence, S. A. ha desarrollado una máquina capaz de ensamblar las partes de cualquier producto. La máquina creada por Digital Intelligence, S.A. puede construir cualquier producto ensamblando automáticamente los componentes (partes) que lo conforman. Para esto, la máquina desarrollada consta de “n” líneas de ensamble y un brazo robótico para cada una de éstas, además, cada línea de ensamble posee un mecanismo que le permite acceder a “m” componentes distintos. El brazo robótico demora 1 segundo en colocarse sobre el recipiente que contiene el 1er componente, 2 segundos para colocarse en el recipiente que contiene el 2do componente y así sucesivamente hasta requerir “m” segundos para colocarse en el “m-ésimo” componente. Adicionalmente, para ensamblar el componente en el

producto que se construye, el brazo robótico utilizará “xm” segundos para el “m-ésimo” componente. Usted ha sido contratado para desarrollar un software que simule el funcionamiento de esta máquina con “n” líneas de ensamble y cada línea de ensamble con “m” posibles componentes a seleccionar de forma que pueda predecir el tiempo “óptimo” para elaborar cualquier producto que pueda ser ensamblado en la máquina.” La figura 1 muestra una línea de ensamble.

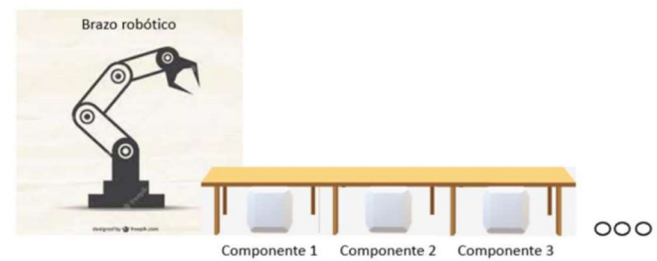


Figura 1. línea de ensamble

Fuente: Proporcionada por la empresa.

Para la solución de este problema iniciamos diseñando una interfaz gráfica con ayuda del software QT Designer

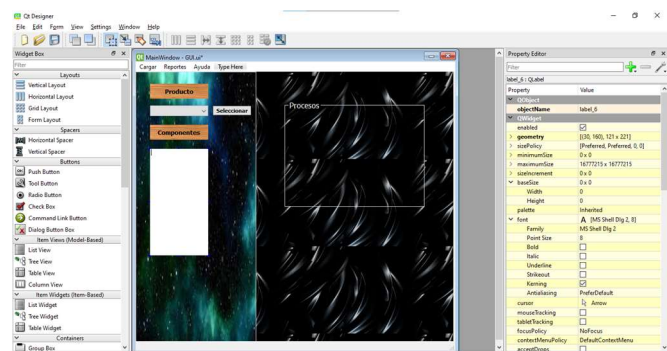


Figura 2. Interfaz gráfica.

Fuente: Elaboración propia.

Tras esto iniciamos a trabajar en VSCode creando nuestras clases, iniciando con el main, para luego “Quemar nuestra interfaz gráfica” en una clase aparte. La primera función que se crea es la lectura

del archivo, esto con la ayuda de la librería “Element tree” y realizando la anidación de varios ciclos for guardamos los siguientes datos en listas simples:

- Cantidad de líneas de producción.
- Número de línea.
- Cantidad de componentes de la línea.
- Tiempo de ensamblaje de la línea.
- Nombre del producto.
- Elaboración del producto.

Estos datos fueron guardados en distintas clases, para posteriormente ser manejados en el main. Las clases que fueron creadas son las siguientes:

- Proyecto2
- Simplesim.
- Nodosim.
- Simplelineas.
- Simpleproce.
- Simple
- Nodo.
- Nodosim.
- NodoProces.
- Nodolineas.

El menú del programa se despliega al momento de iniciar la ejecución, mostrándonos una interfaz amigable, en la cual podemos ingresar el archivo.

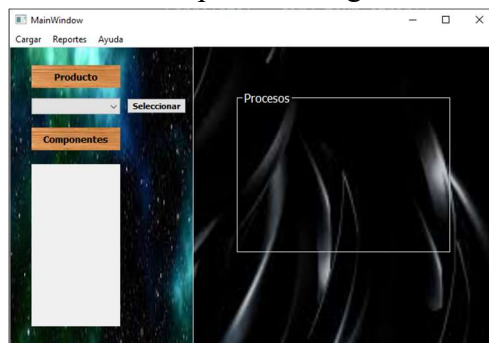


Figura 3. Menú

Fuente: Elaboración propia.

El archivo será leído mediante varios ciclos for y luego fue guardando los valores de los distintos atributos en variables que luego son enviadas a los TDAs.

Usaremos los datos del producto, para realizar la gráfica de la salida de estos tras el ensamble, esto graficación fue posible gracias al lenguaje “.dot” el cual permite la creación de archivos e imágenes en graphviz.



Figura 4. Gráfica de salida.

Fuente: Elaboración propia.

Como todo programa, cuenta con una sección de ayuda, en esta se encontrarán los datos del estudiante y una descripción breve sobre la aplicación.

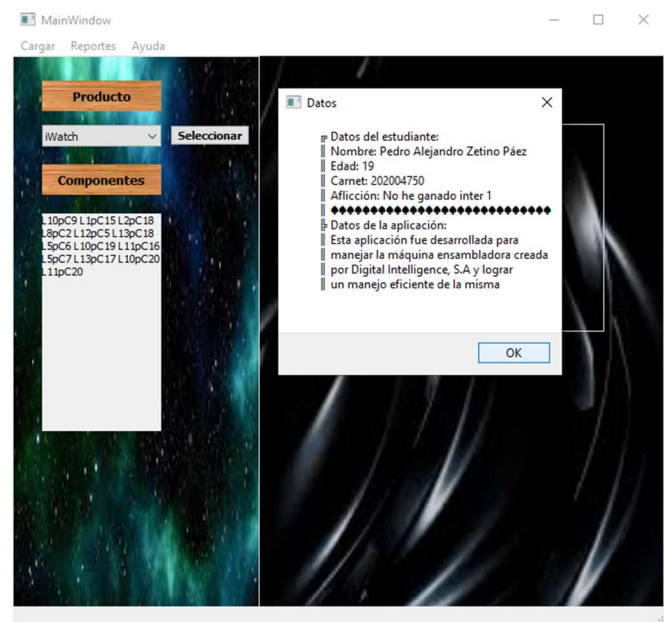


Figura 5. Menú de ayuda

Fuente: Elaboración propia.

Conclusiones

El correcto uso del lenguaje de programación Python, junto al paradigma de la POO y los TDAs permite la solución de una infinidad de problemas, junto a la optimización de una gran cantidad de procesos. En este caso específico se realizó un algoritmo de que optimiza el ensamble de varios productos.

Un ejemplo más cercano a la realidad de este programa serían los programas que manejan las líneas de ensamblaje de carros.

Las ventajas presentadas por estas aplicaciones de la programación es la capacidad de llevar a la realidad algo que solamente imaginabas.

Algo importante a resaltar, es que para el manejo de los TDAs debemos comprender primero el concepto, de lo contrario no podremos utilizarlos de forma eficiente.

Referencias bibliográficas

Gansner, E. R. (2009). Drawing graphs with Graphviz. Technical Report, Technical Report.

Guttag, J. (1977). Abstract data types and the development of data structures. Communications of the ACM, 20(6), 396-404.

Willman, J. M. (2020). Creating GUIs with Qt Designer. In Beginning PyQt (pp. 165-203). Apress, Berkeley, CA.

Anexos

Diagrama de clases IPC2 Proyecto

PEDRO ALEJANDRO ZETINO PÁEZ | September
 26, 2021

