PROYECTO 2: Digital Intelligence, S. A.

202000166 – Gerson Rubén Quiroa del Cid

Resumen

El proyecto presente es un programa que simula una máquina capaz de ensamblar un producto por componentes, dicha máquina consta de n línea de ensamblaje con m cantidad de componentes.

Cada línea consta con un brazo robótico quien es el que se encarga de ensamblar cada componente de cualquier producto a realizar.

Los brazos robóticos pueden moverse al mismo tiempo para llegar al componente que tienen que ensamblar en su línea, la única condición es que tiene que ensamblar una línea a la vez, ya que el producto tiene que ser ensamblado en un orden específico y no aleatoriamente.

Entre otras opciones generales que tiene esta aplicación, es la carga de archivos tipo xml, la sección de reportes, la sección de ayuda, ensamblado individual y una simulación gráfica por medio de una tabla.

Palabras clave

Simulación, xml, estructura de datos, líneas, algoritmo.

Abstract

The present project is a program that simulates a machine capable of assembling a product by components, said machine consists of n assembly line with m quantity of components.

Each line consists of a robotic arm who is in charge of assembling each component of any product to be made.

The robotic arms can move at the same time to reach the component that it has to assemble on its line, the only condition is that it has to assemble one line at a time, since the product has to be assembled in a specific order and not randomly.

Among other general options that this application has, it is the loading of xml files, the reports section, the help section, individual assembly and a graphic simulation by means of a table.

Keywords

Simulation, xml, data structure, lines, algorithm.

Introducción

El presente proyecto tiene como principal objetivo simular el comportamiento de una máquina que ensambla diferentes productos.

La máquina esta compuesta por n línea de ensamblaje y m cantidad de componentes, cada línea con su respectivo brazo robótico, quien será el que se encargue de ensamblar los productos requeridos.

El programa tiene como uno de los objetivos también el estimar un tiempo t en el que se tardará la máquina en ensamblar un producto.

Para la elaboración de este programa se utilizó estructuras de datos hecha en su totalidad por el estudiante, tanto las clases de las listas como la clase nodo principal.

Se utilizó programación orientada a objetos como se puede deducir en el anterior párrafo. También cuenta con una interfaz gráfica sencilla e intuitiva para que el usuario navegue a través de todo el programa.

Desarrollo del proyecto

El análisis para este proyecto fue el siguiente:

Primero se procedió a leer el enunciado y a entenderlo. Luego se procedió a hacer un boceto con los puntos más importantes del proyecto.

Uno de los puntos más importantes, por no decir el más importante, es el algoritmo con el que el programa generará los pasos que la máquina seguirá para realizar el ensamblado del producto.

Se analizó, se estudió, se entendió y se procedió hacer un algoritmo en pseudocódigo. Luego, se procedió a pasar este algoritmo a código en lenguaje Python.

Después de lograr realizar este algoritmo en código, y hacer las pruebas pertinentes para saber que el algoritmo funciona de manera óptima, se pasó hacer la interfaz gráfica.

En realidad, el reto de este proyecto fue hacer el algoritmo anteriormente descrito, luego se podría decir que fue más sencillo. Se empezó con la interfaz gráfica, utilizando una herramienta externa llamada PyQt, que se especializa en hacer interfaces gráficas de manera sencilla y con un aspecto muy bueno.

Luego de esto, se hizo la carga de archivos de tipo xml, para poder leer la información necesaria sobre la máquina y las simulaciones que se necesitar hacer sobre los productos a ensamblar.

Terminado eso, se pasó a realizar la sección de reportes, donde se generan htmls con la información de cada producto y cómo se ensambló dicho producto, por medio de una tabla donde se especifica el tiempo y las líneas donde se realizaron alguna acción.

También en la sección de reportes, existe un apartado donde genera un grafo con los pasos para elaborar un producto en forma de cola.

Se creó también un apartado de ayuda, donde se podrá visualizar los datos del estudiante e información acerca del programa.

Por último, cabe mencionar que en este proyecto se utilizaron estructuras de datos elaboradas por el estudiante, en este caso se utilizaron dos estructuras, una lista enlazada y una lista en forma de cola.

Pasaremos a describir brevemente las clases que se utilizaron:

Clases de la interfaz gráfica:

En total se utilizaron 5 clases para la interfaz gráfica que equivale a decir 5 ventanas diferentes para el programa que se detallará a continuación.

Clase GUI:

Esta es la clase principal de todas, y el main del programa, contiene 6 botones, el botón de cargar archivos, el botón de reportes, el botón de ayuda, el botón de actualizar, el botón de ensamblar y el botón de salir.

Aparte de los botones contiene diferentes funciones, como la función actualizar, que actualiza el combo box con todos los productos que puede ensamblar la máquina.

La función ensamblar es la que ensambla un producto individualmente y muestra el proceso en una tabla que se actualizar al pulsar este botón.

Clase CargaWin:

Esta clase es la ventana que se encarga de cargar los archivos de tipo xml, cuando se ejecuta esta clase crea una ventana sencilla con 2 botones y 1 botón de aceptar.

El botón de ConfigMaq es la que carga la configuración de la máquina y los productos que es capaz de ensamblar y el botón simulación que es el que carga el archivo de simulación con los productos que se desean ensamblar.

Las funciones de esta clase son 2, ambas realizan el mismo trabajo, pero para diferente tipo de archivo, llamada browsefiles.

Clase RepWin:

Esta clase es la encargada de generar los reportes tanto html como los grafos de graphviz, al igual que la anterior clase, es una ventana sencilla con 2 botones, la primera para generar los reportes html y la segunda para generar los grafos en graphviz.

Cuentan con 2 funciones sencillas que llaman a las funciones para generar dichos reportes, con la condición de que, si no se han cargado los archivos con la información correspondiente, abra una ventana con el mensaje de que no se puede generar las gráficas hasta que cargue dichos archivos.

Clase HTMLWin:

Esta clase abre otra ventana cuando se pulsa el botón de reportes html en la anterior clase descrita. Muestra 2 botones más, el primero para que genere un reporte general de la simulación que se realizó y otra para mostrar el reporte del ensamblado general, cuando se creen los reportes saldrá un mensaje diciendo que se generó correctamente el reporte.

Cuenta con 2 funciones simples que llaman a los métodos que generan los reportes y que muestre el mensaje de éxito.

Clase AyudaWin:

Esta clase es una ventana sencilla con un par de labels, con información acerca del estudiante y de la aplicación, cuenta con único botón que es el de aceptar para que cierre la ventana.

Clase ErrorDialog:

Esta es la última clase de la interfaz gráfica y es una ventana sencilla con un mensaje de error por si en algún caso no se pudiera leer los archivos o no se pudiera generar los reportes.

Las clases Nodo, Cola y LinkedList:

Estas clases se utilizaron para crear unas estructuras de datos muy conocidas como lo son las listas enlazadas y las colas, en resumidas cuentas, se creo

una lista dinámica donde se puede agregar, eliminar y buscar cualquier dato que haya sido ingresada en dicha lista.

Al igual que las colas, contienen las mismas funciones, con la diferencia que, a la hora de eliminar algún dato, este elimina el primer dato de la fila, y si se quiere agregar algún dato, se agregará al final de la lista, como una cola.

En este caso se ingresaron los productos que fueron analizados, así como también otros muchos datos, se usó una estructura u otra dependiendo de la necesidad de cada dato para ser tratado y usado.

Clase HTML:

En esta clase se crean los diferentes reportes que general el programa, consta de 2 métodos. El primero llamado repSimulacion, que es el que genera cuando apretamos el botón de reporte de simulación, genera n tablas, detallando el proceso del ensamblado de n productos.

Por otro lado, se tiene la función repIndividual, que genera el reporte de los productos que se ensamblaron individualmente, al igual que el reporte de simulación genera n tablas para los n productos que se ensamblan individualmente.

Clase Grafica:

La clase gráfica es muy sencilla, cuenta con una sola función que es la que se encarga de generar los grafos por medio de la herramienta Graphviz. Genera n grafos para las n productos a ensamblar.

Clases de Objetos que se usaron en el programa:

Se usaron 6 clases para diferentes objetos que se usaron en el proyecto que se detallarán a continuación.

Clase Máquina:

Esta clase contiene 3 atributos los cuales son: la cantidad de líneas, la lista de líneas y la lista de productos. Aquí se guardará toda la información que contenga el archivo de entrada maquina.xml

Clase LineaProduccion:

Esta clase contiene 6 atributos los cuales son los siguientes: número de línea, cantidad de componentes, tiempo de ensamblaje, si está usado o no, una lista de componentes a ensamblar y un contador para saber en dónde se encuentra el brazo robótico.

Esta información será parte de clase máquina.

Clase Producto:

Esta clase contiene 4 atributos, los cuales son: el nombre del producto, el orden de elaboración, los tiempos en cada segundo, y el tiempo total de ensamblado.

Esta información también será parte de clase máquina.

Clase Elaboración:

Esta clase cuenta con 2 atributos que es el número de línea y el número del componente a ensamblar. Esta información es parte de clase Producto.

Clase Simulación:

Esta clase contiene 2 atributos que son el nombre de la simulación y lista de los nombres de los productos a ensamblar.

Aquí será guardada la información del archivo de entrada de simulación_N.xml

Clase Tiempo:

Esta clase tiene 3 atributos que son los siguientes: El segundo en el que sucede una acción, la línea de ensamblaje en la que sucede la acción y la descripción de la acción.

Esta información es parte de la clase Producto.

Clase archivosXML:

Esta es la última clase del programa y probablemente la clase más importante de este proyecto, ya que es donde reside el algoritmo que calcula los movimientos por cada segundo que pasa.

Consta de 3 atributos que son un objeto de tipo Maquina, un objeto de tipo Simulación y una lista enlazada; y 7 funciones las cuales son:

1. analizarConfig: esta función lee el archivo de entrada con la configuración de la máquina, como el archivo de entrada es de tipo xml, las lee por etiquetas y las guarda en el objeto de tipo Maquina.
2. Pasos: Esta función crea una cola con objetos de tipo Elaboración donde están los pasos a seguir para ensamblar un producto correctamente.
3. Simulación: esta función es la encargada de procesar la información del archivo de entrada de la simulación, como el tipo de archivo es xml, se procesa la información por medio de etiquetas y las guarda en una lista de objetos de tipo Simulación.
4. ejecSimulacion: esta función se encarga de ejecutar la simulación para cada producto que esté dentro de la lista simulación y manda a llamar a la función elaboración descrita en breves momentos.
5. ensambladoP: esta función ejecuta la elaboración de un producto en específico por si el usuario desea simular un solo producto y ver como gráficamente se ejecuta el proceso de ensamblado.
6. Elaboración: esta función es la más importante, ya que dentro se encuentra el algoritmo que crea los movimientos que hará la máquina para ensamblar cada producto en el menor tiempo posible. Por cada segundo que pasa, se crea un objeto de tipo Tiempo que se guarda en un atributo de la clase Producto para su posterior uso.
7. crearArchivoSimulacion: esta función es la encargada de crear el archivo de salida para el archivo de simulación y para cada simulación individual que el usuario haga. El archivo de salida será de igual manera de tipo xml y con un etiquetado específico, este proceso se hará automáticamente cada vez que se ejecute el archivo de entrada de simulación y cada vez que se ensamble un producto individual.

Estas son todas las clases del programa, el usuario no tendrá problemas usar este programa ya que es muy sencillo e intuitivo para que lo use cualquier persona.

Y con esto se termina la explicación del proyecto. Gracias.

Conclusiones

Se aprendió a usar de forma correcta y a crear las diferentes estructuras de datos, en este caso, creando la clase Nodo, Cola y ListaEnlazada, se analizó y comprendió dichas clases.

Se determinó un algoritmo para ensamblar productos diferentes, determinar la acción en cada segundo y que encuentre el menor tiempo posible, luego se implementó en código Python.

Se aprendió a procesar archivos tipo xml, a guardar la información del archivo en objetos, en colas y en listas enlazadas.
Así como a escribir archivos del mismo tipo xml, con su estructura característica.

Se comprendió el uso de la herramienta Graphviz para crear grafos gráficamente y así tener una visualización mejor del problema y del algoritmo.

Referencias bibliográficas

Auxiliares de IPC2, (2021). *Proyecto 1*. Disponible en: https://bit.ly/2XgRSD0

Anónimo, (2021). *Cola (informática)*. Disponible en: https://bit.ly/2VC90SU

Anónimo, (sin fecha). Estructura Lista Enlazada. PDF. Disponible en: https://bit.ly/3sJyt9a

Anónimo, (sin fecha). Estructuras de datos: listas enlazadas, pilas y colas. Calcifer.org. Disponible en: https://bit.ly/3j6CscB

Apéndices

Gerson Rubén Quiroa del Cid
Carné: 202000166

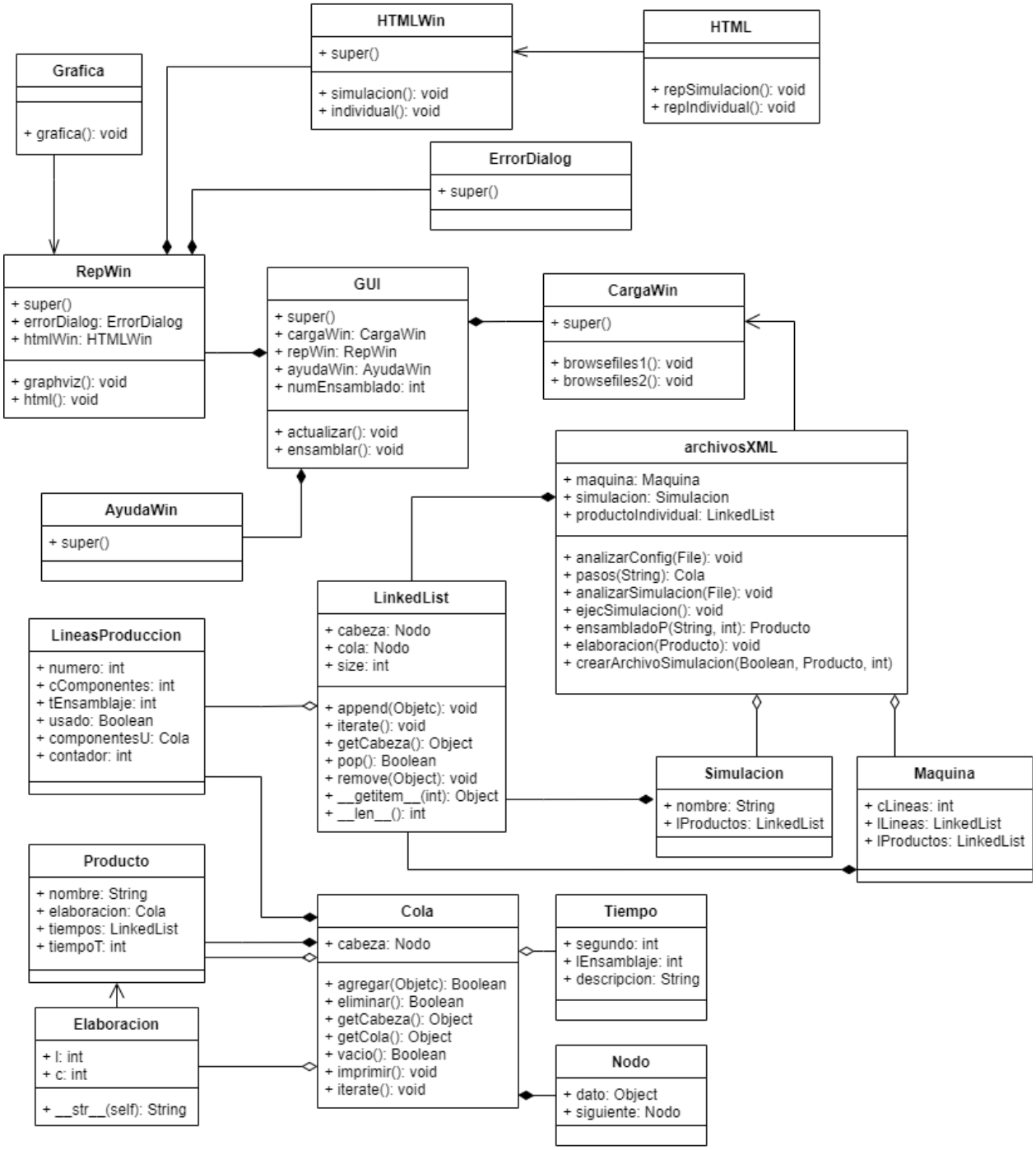

Figura 1. Diagrama de clases proyecto 2
Fuente: elaboración propia.