Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 1er. Semestre 2021.

PROYECTO 2

202010904 – Eduardo Alexander Reyes Gonzalez

Resumen

En el proyecto 2 de laboratorio de introducción a la programación y computación 2 se presenta la resolución del problema de calcular las instrucciones mas optimas para dirigir unas líneas de construcción hacia los componentes para posteriormente construirlos. La aplicación cuenta con el botón cargar, el cual carga archivos de máquina, cuenta con el botón cargar simulación que carga los archivos en donde se encuentran los datos de los productos a construir, esta el botón de analizar que analiza y crea las instrucciones para que las líneas de producción las sigan, y el botón de ayuda el cual muestra una pequeña ayuda al usuario para guiarse en el programa. Este programa es capaz de mostrar reportes en el tiempo de análisis y crea un xml con los datos finales.

Palabras clave
Optimo, producción, Tiempo, Reporte, Python

Abstract

In project 2 of the introductory laboratory to programming and computing 2, the resolution of the problem of calculating the most optimal instructions to direct a construction line to the components is presented in order to later build them. The application has the load button, which loads machine files, it has the load simulation button that loads the files where the data of the products to be built is located, there is the analyze button that analyzes and creates the instructions so that the production lines follow them, and the help button which shows a small help to the user to guide them through the program. This program is capable of displaying reports at the time of analysis and creates an xml with the final data.

Keywords
Optimal, Production, Time, Report, Python

Introducción

El problema al que se da solución es el de optimizar el las instrucciones que las líneas de construcción deben seguir, la empresa posee n cantidad de líneas de construcción, y n componentes los cuales son utilizados para construir diferentes productos que son solicitados mediante un archivo con extensión xml. Los datos de la maquina son enviados a travez de un archivo xml el cual posee el numero de líneas, el nombre de componentes, y los procesos los cuales la línea debe seguir para armar correctamente el componente.

Desarrollo del tema

El problema presentado en el proyecto 2 es dar solución al problema de generar las instrucciones necesarias para el correcto funcionamiento de una maquina que posee una cierta cantidad de líneas de producción, y es capaz de construir diversos productos. El usuario cuando inicia el programa, se le es presentado una interfaz grafica simple pero conforme va activando las funcionalidades se van agregando automáticamente a la pantalla.
Los archivos utilizados en la solución del problema son:

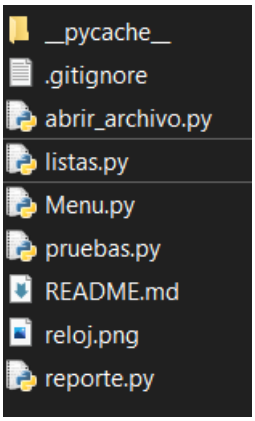

Figura 1. Archivos necesarios de programa

Fuente: extraída de la solución en Python

Las funcionalidades son:

1) Cargar Archivo
2) Cargar Simulación
3) Analizar
4) Reportes
5) Ayuda
6) Salir

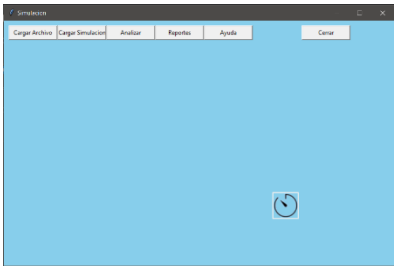

Figura 2. Interfaz grafica inicial

Fuente: extraída de la solución en Python

Cargar Archivo

En esta opción del menú el usuario podrá seleccionar un archivo con formato xml que contenga los datos de la maquina desea analizar la estructura del archivo que debe de ser analizado es el siguiente:

```
<?xml version="1.0"?>
- <Maquina>
    <CantidadLineasProduccion>2</CantidadLineasProduccion>
  - <ListadoLineasProduccion>
    - <LineaProduccion>
        <Numero>1</Numero>
        <CantidadComponentes>4</CantidadComponentes>
        <TiempoEnsamblaje>1</TiempoEnsamblaje>
      </LineaProduccion>
    - <LineaProduccion>
        <Numero>2</Numero>
        <CantidadComponentes>4</CantidadComponentes>
        <TiempoEnsamblaje>1</TiempoEnsamblaje>
      </LineaProduccion>
    </ListadoLineasProduccion>
  - <ListadoProductos>
    - <Producto>
        <nombre>Smartwatch</nombre>
        <elaboracion>L1C2 L2C1 L2C2 L1C4</elaboracion>
      </Producto>
    - <Producto>
        <nombre>Camara</nombre>
        <elaboracion>L1C2 L2C3 L1C4 L2C2</elaboracion>
      </Producto>
    - <Producto>
        <nombre>USBStick</nombre>
        <elaboracion>L1C1 L2C3</elaboracion>
      </Producto>
    </ListadoProductos>
</Maquina>
```

Figura 3. Estructura Archivo xml. maquina

Fuente: extraída de archivo de prueba

Los datos que el archivo xml debe poseer es el siguiente:

Cantidad líneas de producción: Muestra la cantidad exacta de líneas de producción que la maquina debe tener.

Línea de producción: En este apartado debe mostrar la información necesaria de las líneas de producción tales como los componentes y el tiempo de ensamblaje.

Listado Productos: En este apartado debe mostrar la información de todos los productos que la maquina es capaz de construir, en este apartado debe mostrar los pasos de la elaboración del producto, y los datos exactos de la línea de producción que construye el componente.

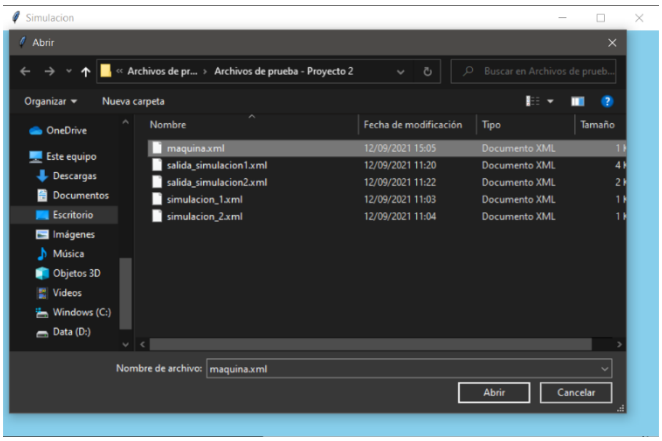

Figura 4. Abrir archivo botón cargar Archivo

Fuente: extraída de la solución en Python

Cargar Simulación

En esta opción el usuario es capaz de cargar el archivo con extensión xml que contenga los datos de los productos que desea construir.

La estructura del archivo xml es el siguiente:

```
<?xml version="1.0"?>
- <Simulacion>
    <Nombre>prueba1</Nombre>
  - <ListadoProductos>
      <Producto>Smartwatch</Producto>
      <Producto>Camara</Producto>
    </ListadoProductos>
</Simulacion>
```

Figura 5. Estructura archivo xml simulación

Fuente: extraída de la solución en Python

Los datos que este archivo debe poseer son los siguientes:

Nombre: En este apartado del archivo se debe especificar el nombre del proceso de construcción.

Listado productos: En este apartado se debe especificar el nombre de los productos que el usuario desea construir.

Analizar

Este botón es la acción mas importante de todo el programa ya que sin esta acción el programa solo leería información y no realizaría ni una acción.

La opción analizar crea las listas necesarias y adjunta toda la información sobre los productos que el usuario desea construir. Luego activa hilos los cuales van sucediendo cada segundo lo cual se muestra en la pantalla, Se puede observar una tabla que va mostrando lo que pasa cada segundo y mientras se esta ejecutando se podrá realizar otras acciones tales como la generación de reportes.

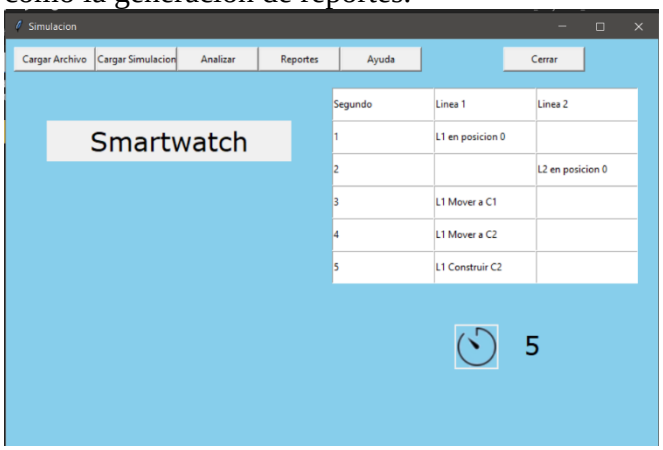

Figura 6. Interfaz ejecutando acción analisis

Fuente: extraída de la solución en Python

Si finaliza el proceso de analizar entonces el programa mostrará en la interfaz que no esta construyendo ningún producto y se moverá automáticamente a la posición 0 en cada línea de producción.

Reportes

En esta opción el usuario es capaz de generar los reportes de cada producto que desea construir, genera un archivo con extensión html el cual posee una interfaz simple pero agradable a la vista, el cual posee la información detallada de la construcción de ese componente analizado.

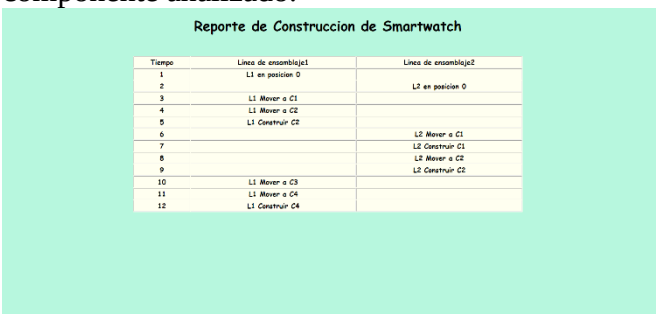

Figura 7. Vista grafica de reportes.

Fuente: extraída de la solución en Python

Ayuda

Esta opción muestra unas ventanas de alerta informando al usuario sobre el programa, la información que muestra es la siguiente:
Datos del que programo la aplicación.
Información sobre cada botón de la interfaz y su funcionamiento.

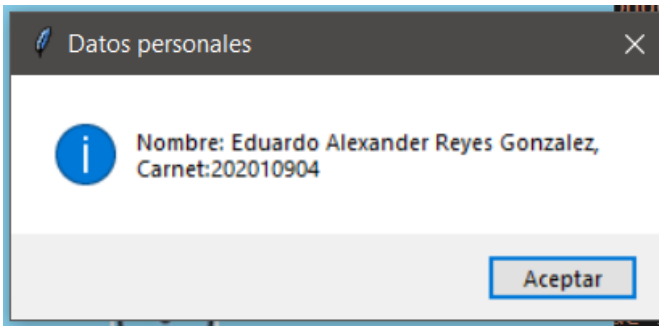

Figura 8. Información de botón ayuda

Fuente: extraída de la solución en Python

Figura 9. Información de botón ayuda

Fuente: extraída de la solución en Python

Salir

En esta opción el programa finaliza todo lo que esta realizando y termina la ejecución del mismo.

Clases Utilizadas

Las clases se que se utilizaron fueron de ayuda para la resolución del problema presentado las clases son las siguientes:

Clase Interfaz

Esta clase almacena la todo lo relacionado a la interfaz visual del programa, también posee las funciones del de la misma las cuales son:

Función carga: Esta función llama a otro documento con extensión .py el cual muestra la ventana emergente al usuario para que pueda seleccionar el archivo.

Función Cargar Simulación: Esta función llama a otro archivo el cual es capaz de mostrar la ventana emergente para seleccionar el archivo de la simulación.

Función hilos: Esta función es el proceso de análisis de los archivos y ejecuta hilos para mostrar al usuario la información en cada segundo que transcurre.

Función work: Esta función es la encargada de ejecutar la acción que realiza los hilos.

Función reportes: En esta función el programa llama a otro documento y ejecuta la función de generar reportes y los muestra al usuario cuando presiona el botón.

Función ayuda: Esta función muestra al usuario las ventanas emergentes que poseen la información mas importante del programa.

Clase Nodo
Esta clase posee la información necesaria de las listas y colas utilizadas en el programa, es necesario para crear un apuntador en las listas y colas.

Clase Lista
Esta clase es capaz de crear listas personalizadas en las que se puede guardar la información que queramos.

Clase Cola
Esta clase es capaz de crear colas creadas por el programador y se utilizan para poder llevar un orden en el que el primer dato que entra es el primero en salir.

Clase Reporte
En esta clase se puede ejecutar la acción que realiza el botón reportes que esta ubicado en la interfaz. Genera los reportes necesarios sobre cada producto de la lista de simulación.

Listas y Colas Utilizadas
Para dar solución al problema de generar las instrucciones necesaria se tuvieron que utilizar diferentes listas y colas que guardan diferentes valores que se utilizaron dentro del proyecto, las cuales son:

Lista Listado producto
En esta lista se guardó toda la información del documento. El cual son los productos que el usuario puede construir.

Lista Líneas de producción
Esta lista guarda el número de las líneas de producción en el orden en el que se construye los productos.

Cola lista construcción
En esta cola se guarda la información de los productos que se desean construir y se almacena en el orden en el que van apareciendo en el documento.

Cola Pasos
Esta cola guarda los pasos separados de las letras, en esta cola se almacena los números correspondientes a las líneas de producción y a los componentes que se deben llegar.

Lista Pasos intermedios
Esta lista almacena la información separada de las líneas de producción, las cuales se consideran pasos intermedios porque se utilizan para llegar a la solución del problema.

Lista líneas intermedias
En esta lista se almacenan los números de las líneas de construcción los cuales coinciden con las líneas que se utilizan para lista de pasos intermedios.

Cola reporte final
Esta cola es capaz de almacenar las instrucciones que se utilizaron al final del análisis los cuales sirven para la generación de los archivos de salida y los reportes respectivos.

Cola líneas reporte final

Esta cola es capaz de almacenar los números pertenecientes a cada dato de la cola de reporte final, el cual es utilizado para agregarlos al reporte de cada producto.

Conclusiones

1) El programa desarrollado en Python muestra los pasos necesarios para construir productos.
2) El uso de listas facilito la solución del problema presentado
3) Se genero reportes que facilitan la visualización de los pasos de construcción de cada producto
4) El uso de listas y colas ayudo a comprender la estructura y funcionamiento de las mismas

Diagrama de clases

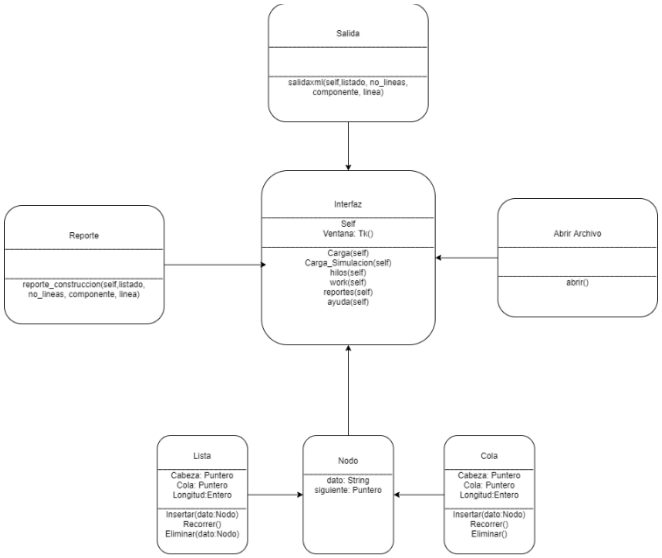

Figura 10. Diagrama de clases

Fuente: Elaboración Propia 2021

Diagrama de Actividades

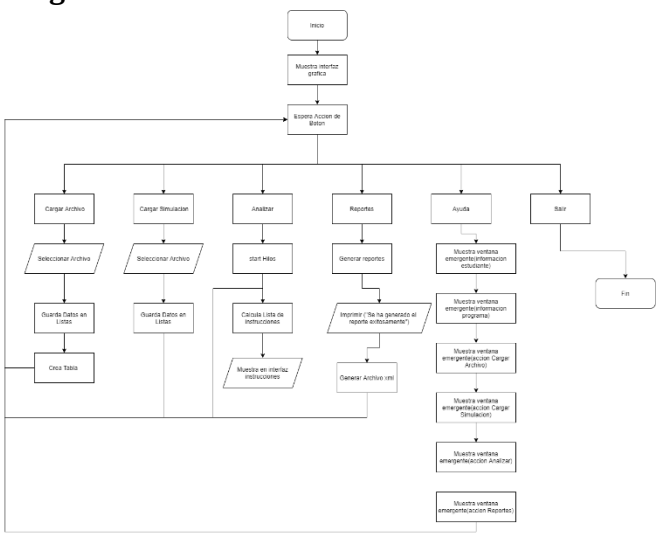

Figura 11. Diagrama de actividades

Fuente: Elaboración Propia 2021

Referencias bibliográficas

Uniwebsidad, (2011). *Algoritmos de programacion en python*. https://uniwebsidad.com/libros/algoritmos-python/capitulo-17/colas