
PROYECTO 2: DIGITAL INTELLIGENCE ASSEMBLING MACHINE

201903342 – Eduardo Abraham Barillas del Aguila

Resumen

El proyecto desarrollado en el lenguaje de programación de Python está orientado a la optimización del tiempo de la máquina de ensamblaje de los distintos componentes necesarios para la creación de distintos productos.

Para el desarrollo de la lógica de dicha máquina se realiza la implementación de distintos Tipos de Datos Abstractos (TDA's) para un mejor representación del funcionamiento de la maquina se a desarrollado una interfaz gráfica capaz de permitir al usuario realizar la carga de los archivos de entrada.

Para la interfaz se ha utilizado una librería nativa de Python para el desarrollo de interfaces gráficas, así como el uso de la librería de Graphviz para realizar imágenes con la respectiva información de los productos.

Palabras clave

- 1)TDA – Tipo de Dato Abstracto
- 2) Algoritmo – Serie de paso para encontrar una solución.
- 3) Función 4) XML

Abstract

This project was developed using the python programming language to optimize the time the machine requires to assemble the components that certain product requires to be fully assembled

For the development of the logic of the machine que require to implement Abstract Data Types (ADT's) in order to create a better graphic representation of the machine procedures and to provide a better way to handle the information to show the information in a a graphic environment.

For the user interface I used a python native library and also the graphviz library to generate images from the queues to show in a realistic way the information handled in the project.

Keywords

- 1) ADT – Abstract Data Type
- 2) Algorithm
- 3) Function
- 4) XML

Introducción

La temática principal de este programa es la elaboración de un algoritmo capaz de encontrar la ruta óptima para recorrer un terreno en específico consumiendo la cantidad mínima de combustible.

Para ello se debieron implementar diferentes recursos y practicas entre los cuales podemos mencionar los tipos de datos abstractos y practicas tales como el uso de herramientas de versionamiento las cuales serán tratadas más a profundidad a lo largo de este documento.

Desarrollo del tema

¿Qué es Python?

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código.

Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

¿Qué es una librería en Python?

Una librería o biblioteca es el conjunto de implementos funcionales que te ayudan a codificar lenguajes de programación para crear una interfaz independiente.

Las librerías tienen la libertad de ser utilizadas por otros programas independientes y simultáneamente. A su vez, se pueden apoyar mutuamente entre librerías para definir o refinar el comportamiento de la biblioteca original. También se la coloca en disposición de otra tecnología o lenguaje de programación.

¿Qué es un IDE?

Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Para el desarrollo de este programa fue utilizado el IDE de nombre PyCharm el cual tiene una orientación específica para python lo que hace que el desarrollo de programas basados en este lenguaje sean más fáciles de trabajar.

¿Qué librerías fueron utilizadas en el proyecto?

- Graphviz: Una herramienta en línea de comandos para producir imágenes en capas de grafo dirigido en una variedad de formatos de salida
- ElementTree: El modulo *xml.etree.ElementTree* implementa una API simple y eficiente para analizar y crear datos XML

¿Qué son los archivos XML?

XML son las siglas de Extensible Markup Language, que podemos traducir como Lenguaje de Marcas Extensibles.

Normalmente, los archivos XML pueden contener información de cualquier tipo, ya sea información sensible o información de cualquier ámbito.

Los archivos XML se componen de etiquetas que nos aportan datos e información que queremos procesar. Estas etiquetas pueden estar de forma individual o anidada.

¿Cómo se trabajan los archivos XML?

Para trabajar con el archivo XML primero debemos de importar una librería llamada ElementTree. Con el siguiente código:

```
import xml.etree.ElementTree as ET
```

Para acceder a la raíz del árbol generado por la anterior línea de código podemos hacer uso de las siguientes líneas de código:

```
tree = ET.parse("Ruta absoluta del archivo")  
root = tree.getroot()
```

Luego haciendo uso de diferentes métodos tales como *findall()* y *find()* entre otros para poder acceder a cada uno de los nodos internos del archivo.

¿Qué es Graphviz?

Graphviz es un programa de visualización gráfica de fuente abierta. La visualización de gráficos es una forma de representar información estructural como diagramas de gráficos y redes abstractos.

Tiene importantes aplicaciones en redes, bioinformática, ingeniería de software, diseño de bases de datos y web, aprendizaje automático y en interfaces visuales para otros dominios técnicos.

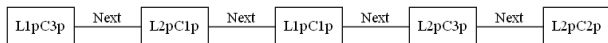


Figura 1. Cola de componentes de una simulación generada por nuestra aplicación

Fuente: elaboración propia, o citar al autor, año y página.

¿Qué es el versionamiento?

El versionado de software es el proceso de asignación de un nombre, código o número único, a un software para indicar su nivel de desarrollo.

¿Cómo se aplicó el versionamiento en este proyecto?

Para este proyecto el versionamiento fue aplicado por medio de una rama principal la cual tiene el código principal y completo que irá siendo

liberado a medida que el programa sea completado.

De la misma forma existe una rama Development que contiene el código en desarrollo que aun cuenta con errores y por ultimo hay ramas nombradas features/nombre de la nueva funcionalidad el cual será una implementación a agregar.

¿Qué son los TDA's?

En ciencias de la computación un tipo de dato abstracto (TDA) o tipo abstracto de datos (TAD) es un modelo matemático compuesto por una colección de operaciones definidas sobre un conjunto de datos para el modelo.

¿Qué TDA's fueron implementados en este proyecto?

1. Listas Simplemente Enlazadas
2. Colas
3. Listas Doblemente Enlazadas

Listas Simplemente Enlazadas: es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. El principal beneficio de las listas enlazadas respecto a los vectores convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en la memoria o el disco, permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.

Colas: Es una estructura de datos, caracterizada por ser una secuencia de elementos en la que la

operación de inserción push se realiza por un extremo y la operación de extracción pull por el otro. También se le llama estructura FIFO (del inglés First In First Out), debido a que el primer elemento en entrar será también el primero en salir.

Tabla I.

Nombre de los TDA's implementados y su tipo.

CATEGORÍA	CATEGORÍA
Queue	Cola
LinkedList	Lista
	Simplemente Enlazada
BasicLinkedList	Lista
	Simplemente Enlazada
DoubleLinkedList	Lista
	Simplemente Enlazada

Fuente: elaboración propia, o citar al autor, año y página.

En la parte de anexos se encuentra la imagen correspondiente al diagrama de clases utilizado para este proyecto.

Conclusiones

Al concluir la realización del proyecto se han adquirido los suficientes conocimientos para manipular los archivos de tipo XML.

El uso de versionamiento permitió tener un mejor control sobre los cambios y la forma de implementar nuevas funciones dentro del proyecto.

Se comprendió el uso e implementación de estructuras de datos abstractos como lo son las listas y las colas utilizadas anteriormente.

Anexos

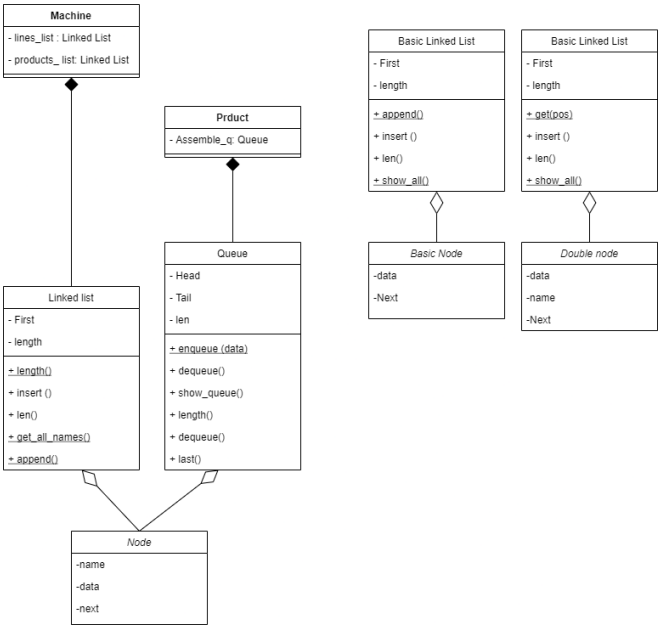


Imagen 1. Diagrama de clases utilizada para el proyecto

Fuente: elaboración propia, o citar al autor, año y página.

Referencias bibliográficas

Luis, J. A., & Aguilar, L. J. (2008). *Fundamentos de programación*. McGraw-Hill Education.

