

---

## Desarrollo de Algoritmos para la Resolución de Problemas en Maquetas Mecatrónicas

---

202200196 – Gerardo Leonel Ortiz Tobar

### Resumen

El proyecto se centra en el diseño e implementación de un algoritmo en Python para resolver un desafío planteado por el Ministerio de Educación de Guatemala en el ámbito de la mecatrónica. El desafío consiste en desarrollar un dispositivo capaz de recolectar objetivos en una maqueta compleja con caminos y paredes, siguiendo un camino lineal sin repetir segmentos de camino ya recorridos. Se aplican conceptos de programación estructurada y orientada a objetos, manipulación de archivos XML, generación de reportes visuales con Graphviz, y análisis de complejidad algorítmica. El ensayo aborda el análisis detallado del problema, el diseño y la implementación del algoritmo, las pruebas realizadas y las conclusiones obtenidas. Se destacan los aportes del proyecto en el contexto de la mecatrónica y se sugieren áreas de investigación futuras.

El desafío presentado por el Ministerio de Educación de Guatemala representa un escenario real donde convergen múltiples aspectos de la mecatrónica y su aplicación práctica. A través de este proyecto, se busca no solo resolver el desafío propuesto, sino también explorar nuevas estrategias y técnicas que puedan mejorar la eficiencia y versatilidad del dispositivo recolector.

### Palabras clave

1. Python
2. mecatrónica,
3. Algoritmo
4. Graphviz
5. programación estructurada

### Abstract

*The project focuses on designing and implementing an algorithm in Python to solve a challenge posed by the Ministry of Education of Guatemala in the field of mechatronics. The challenge involves developing a device capable of collecting objectives in a complex model with paths and walls, following a linear path without repeating segments of the path already traveled. Concepts of structured and object-oriented programming, XML file manipulation, visual report generation with Graphviz, and algorithmic complexity analysis are applied. The essay addresses the detailed analysis of the problem, the design and implementation of the algorithm, the tests performed, and the conclusions obtained. The contributions of the project in the context of mechatronics are highlighted, and areas for future research are suggested.*

*The challenge presented by the Ministry of Education of Guatemala represents a real scenario where multiple aspects of mechatronics*

*and its practical application converge. Through this project, the goal is not only to solve the proposed challenge but also to explore new strategies and techniques that can improve the efficiency and versatility of the collecting device.*

### **Keywords**

1. *Python,*
2. *mechatronics*
3. *Algorithm*
4. *Graphviz*
5. *Structure programming*

## **Introducción**

### **Análisis del problema**

El problema planteado por el Ministerio de Educación presenta varios desafíos técnicos y algorítmicos. En primer lugar, se debe diseñar un algoritmo que pueda mapear la maqueta con sus caminos y objetivos de manera eficiente y precisa. Además, el algoritmo debe garantizar que el dispositivo recolector siga un camino lineal sin repetir segmentos de camino, lo que implica un análisis exhaustivo de las posibles rutas y la optimización del recorrido. Otro aspecto importante del análisis es la consideración de posibles variaciones en la maqueta y la capacidad del algoritmo para adaptarse a estos cambios sin perder eficacia en la recolección de objetivos.

### **Diseño del algoritmo**

El diseño del algoritmo se basa en una combinación de técnicas de programación estructurada y orientada a objetos. Se utiliza una estructura de datos adecuada para representar la maqueta y los objetivos, lo que facilita el mapeo y la planificación de rutas. El algoritmo se divide en etapas, desde la inicialización y carga de datos hasta la generación del camino

óptimo para recolectar los objetivos. Se aplican conceptos de complejidad algorítmica para garantizar la eficiencia del algoritmo en diferentes escenarios y tamaños de maqueta.

### **Implementación en Python**

La implementación del algoritmo se realiza en Python, aprovechando las capacidades del lenguaje para manipular estructuras de datos complejas y realizar cálculos eficientes. Se utilizan bibliotecas y herramientas externas, como lxml para la manipulación de archivos XML y Graphviz para la generación de reportes visuales. El código se organiza de manera modular y se documenta adecuadamente para facilitar su comprensión y mantenimiento. Además, se realizan pruebas exhaustivas para validar el funcionamiento del algoritmo en diferentes escenarios y se ajustan parámetros según sea necesario para optimizar el rendimiento.

### **Evaluación y pruebas**

La evaluación del algoritmo se realiza mediante la definición de casos de prueba que abarcan diferentes configuraciones de maquetas y objetivos. Se analizan los resultados obtenidos en términos de tiempo de ejecución, eficiencia en la recolección de objetivos y adaptabilidad a cambios en la maqueta. Se discuten los resultados y se proponen posibles mejoras o ajustes para optimizar el rendimiento del algoritmo en situaciones específicas. La evaluación también incluye la comparación con otros enfoques y algoritmos existentes en la literatura para resaltar las fortalezas y debilidades de la solución propuesta.

### **1. Clase Entrada:**

- Descripción de la clase **Entrada** que representa la entrada en la maqueta.
- Métodos para obtener y establecer la fila y columna de la entrada.
- Método para desplegar la información de la entrada.

2. **Clase Lista Doble:**

- Descripción de la clase **lista\_doble** que implementa una lista doblemente enlazada.
- Métodos para insertar, buscar, eliminar, desplegar y ordenar elementos en la lista.
- Método para verificar si la lista está vacía.
- Método para buscar una entrada o objetivo en la lista.

3. **Clase Maqueta:**

- Descripción de la clase **Maqueta** que representa la maqueta con sus características como nombre, filas, columnas y estructura.
- Métodos para agregar objetivos y entradas, obtener información de la maqueta y desplegar su contenido.
- Métodos para generar código DOT y crear una imagen de la maqueta.

4. **Clase Nodo:**

- Descripción de la clase **Nodo** utilizada en la implementación de la lista doblemente enlazada.

5. **Clase Objetivos:**

- Descripción de la clase **Objetivos** que representa los objetivos en la maqueta.
- Métodos para obtener y establecer el nombre, fila y columna de los objetivos.
- Método para desplegar la información de los objetivos.

6. **Clase Printer:**

- Descripción de la clase **Printer** utilizada para generar texto.

Tabla I.  
*Variables globales.*

CATEGORÍA	CATEGORÍA
Ruta	String
Lista_maquetas	ListaDoble
maquetaSeleccionada matriz	ListaDoble
matrizResolucion	ListaDoble

Fuente: elaboración propia, 2024 y 3.

Conclusiones

El desarrollo del algoritmo para el dispositivo recolector de objetivos en la maqueta representa un avance significativo en la aplicación de la mecatrónica en problemas prácticos.

Se han aplicado técnicas avanzadas de programación y análisis algorítmico para proporcionar una solución eficiente y adaptable a diferentes escenarios.

Las conclusiones destacan la importancia de la innovación tecnológica en la mecatrónica y sugieren áreas de investigación futuras para mejorar la funcionalidad y la eficiencia del dispositivo recolector.

Referencias bibliográficas

Máximo 5 referencias en orden alfabético.

1. C. J. Date, (1991). An introduction to Database Systems. Addison-Wesley Publishing Company, Inc.

2. Autor, A. (Año). Título del libro. Editorial.

3. Autor, B. (Año). Título del artículo. Revista, volumen (número), páginas.

4. Autor, C. (Año). Título de la tesis. Universidad.

Apéndice

