

# Analisis y Diseño de Algoritmos 2

## Proyecto 2

Alejandro Sierra Betancourt - 2259559 ,  
Juan Pablo Castaño Arango - 2258487 ,  
Juan Manuel Ramirez Agudelo - 2258482 ,  
Kevin Stiven Ramirez Torres - 2259371

Escuelas de Ingeniería de Sistemas  
Universidad del Valle, Tuluá, Colombia

Email: {alejandro.sierra, juan.castano, juan.ramirez, kevin.ramirez}@correounivalle.edu

**Abstract**—El presente informe documenta el desarrollo e implementación de una solución computacional para optimizar la ubicación de nuevos programas de ingeniería de sistemas en un plano cartesiano bidimensional. Este proyecto busca maximizar los beneficios poblacionales y empresariales bajo restricciones específicas, tales como evitar ubicaciones contiguas y garantizar umbrales mínimos de segmento poblacional y entorno empresarial.

La solución integra MiniZinc para modelar el problema de optimización combinatoria, PyQt5 para construir una interfaz gráfica que facilite la interacción con el usuario y Python como lenguaje de control y orquestación. El desarrollo cumple con las especificaciones planteadas en las rúbricas del proyecto, incluyendo la validación del modelo mediante pruebas exhaustivas.

**Index Terms**—Optimización combinatoria, MiniZinc, PyQt5, Ingeniería de Sistemas, Usabilidad

### I. INTRODUCCIÓN

El presente informe documenta el desarrollo e implementación de una solución computacional para optimizar la ubicación de nuevos programas de ingeniería de sistemas en un plano cartesiano bidimensional. Este proyecto busca maximizar los beneficios poblacionales y empresariales bajo restricciones específicas, tales como evitar ubicaciones contiguas y garantizar umbrales mínimos de segmento poblacional y entorno empresarial.

La solución integra MiniZinc para modelar el problema de optimización combinatoria, PyQt5 para construir una interfaz gráfica que facilite la interacción con el usuario y Python como lenguaje de control y orquestación. El desarrollo cumple con las especificaciones planteadas en las rúbricas del proyecto, incluyendo la validación del modelo mediante pruebas exhaustivas.

### II. DESCRIPCIÓN DE LAS ETAPAS DEL PROYECTO

#### A. Interfaz Gráfica

La interfaz fue desarrollada utilizando PyQt5. Incluye las siguientes funcionalidades:

- Selección de archivos con datos de entrada (.dzn).
- Configuración del solver (gecode, chuffed o coin-bc).
- Visualización gráfica de los resultados en un plano cartesiano bidimensional.

- Despliegue textual de la solución, incluyendo las ubicaciones base, nuevas ubicaciones propuestas y ganancias asociadas.

Se cuidó el diseño estético y la usabilidad mediante hojas de estilo CSS, garantizando control y libertad del usuario para configurar y analizar resultados.

#### B. Modelo Matemático

El modelo se construyó en MiniZinc, definiendo una función objetivo para maximizar la ganancia total. Se establecieron restricciones que aseguran:

- Que las nuevas ubicaciones no sean contiguas.
- Umbrales mínimos para los segmentos de población y el entorno empresarial.
- Ubicaciones predefinidas y un número fijo de programas.

Estas reglas se implementaron utilizando funciones y estructuras como matrices y restricciones lógicas.

### III. PRUEBAS

Se realizaron cinco pruebas para validar la funcionalidad y precisión del modelo:

- **Prueba 1:** Pruebas iniciales del proyecto

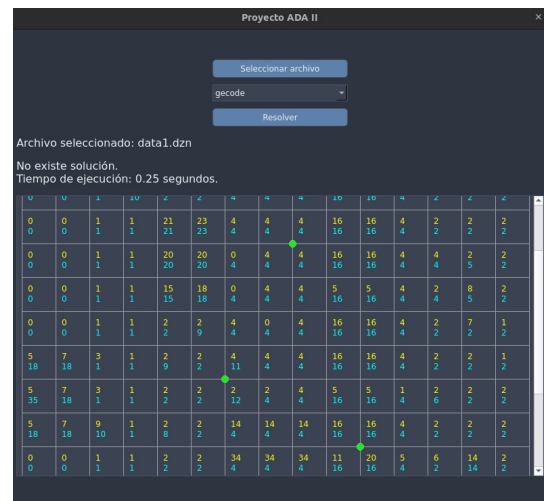


Fig. 1. Resultados de la Prueba 1

- **Prueba 2:** Prueba de 4 puntos muy cercanos entre si

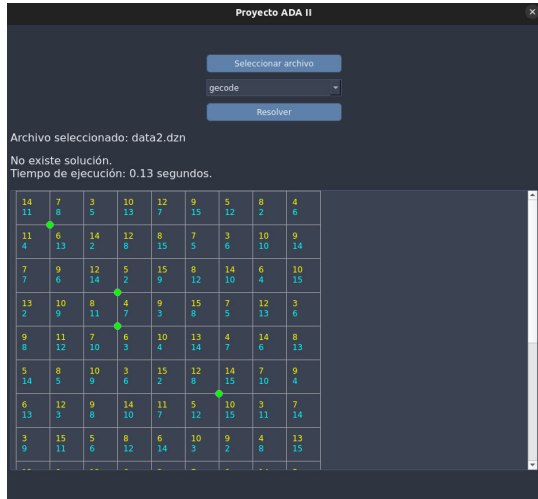


Fig. 2. Resultados de la Prueba 2

- **Prueba 3:** Prueba de dos puntos con gran posibilidad a sus alrededores

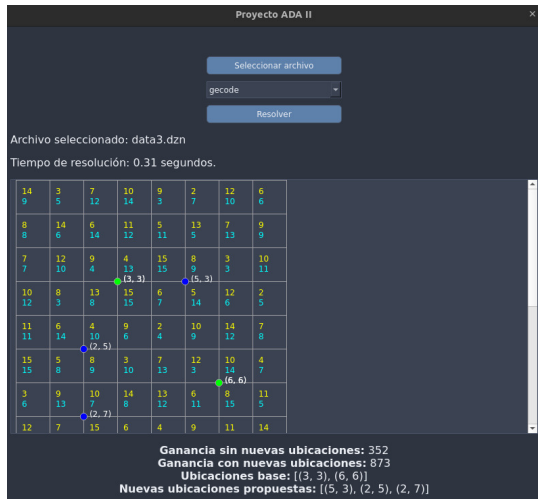


Fig. 3. Resultados de la Prueba 3

- **Prueba 4:** Prueba de 3 puntos con ganancia en sus cercanías

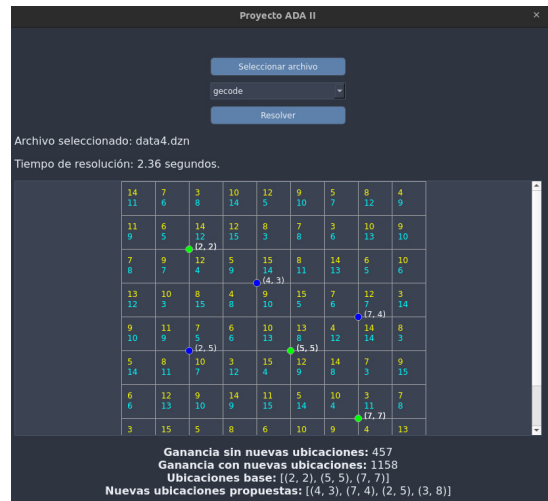


Fig. 4. Resultados de la Prueba 4

- **Prueba 5:** Prueba de 3 puntos con ganancia en sus lejanía

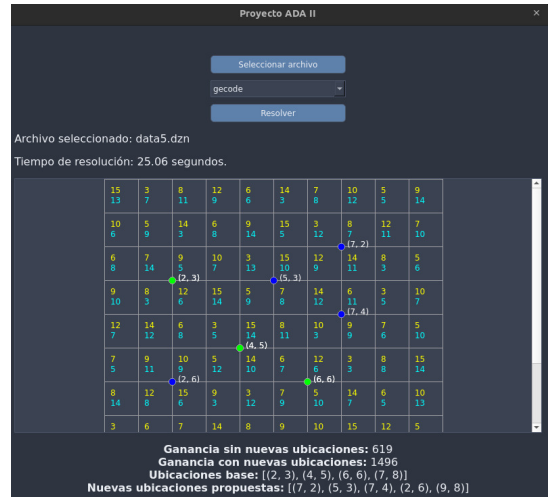


Fig. 5. Resultados de la Prueba 5

## IV. CONCLUSIONES

Se resumió el impacto de la solución propuesta para la ubicación de nuevos programas de ingeniería de sistemas. El modelo implementado logró satisfacer las restricciones y optimizar la ganancia total en todos los casos probados. Sin embargo, se identificaron limitaciones en el tiempo de ejecución para matrices muy grandes y se proponen mejoras futuras, como la integración de heurísticas para reducir la complejidad computacional.

## REFERENCES

- [1] MiniZinc. *Modelamiento básico en MiniZinc*. Disponible en: <https://docs.minizinc.dev/en/stable/modelling.html>. Accedido el 15 de diciembre de 2024.
- [2] MiniZinc. *Modelos más complejos en MiniZinc*. Disponible en: <https://docs.minizinc.dev/en/stable/modelling2.html>. Accedido el 15 de diciembre de 2024.

- [3] MiniZinc. *Integración de MiniZinc con Python*. Disponible en: <https://docs.minizinc.dev/en/stable/python.html>. Accedido el 15 de diciembre de 2024.
- [4] Aucaruri, C. *Modelado de vertederos en MiniZinc*. Disponible en: [https://github.com/Carlos-Andres-Aucaruri/modelling\\_and\\_fill](https://github.com/Carlos-Andres-Aucaruri/modelling_and_fill). Accedido el 12 de diciembre de 2024.