

Diseño de una herramienta software en ambiente web como propuesta para la solución
del problema de distribución de planta.

Autor:

Reinaldo Pabón Fuentes

Trabajo de Grado para Optar al Título de Ingeniero Industrial e Ingeniero de Sistemas e
Informática.

Director:

MSc. Edwin Alberto Garavito Hernández

Codirector:

PhD. Gabriel Rodrigo Pedraza Ferreira

Universidad Industrial de Santander

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2023

AGRADECIMIENTOS

A Dios, primeramente, quien cada día me enseña y recuerda el valor que tengo en esta vida, porque a pesar de todo nunca me ha dejado atrás, porque me ha sostenido y ha hecho de mí la persona que soy en este momento.

A mi papá quien desde muy niño me ha enseñado a amar, a ser un CAMPEÓN, a comprender que sin importar cuántas veces caiga, siempre podré levantarme, aprender y ser mejor; sin su apoyo nunca hubiese podido haber cumplido este logro.

A mis hermanas Paola y Nubia, por todo el amor, apoyo incondicional, los consejos y las palabras de apoyo y ánimo que siempre me han dado.

Al profe Gabriel, agradezco que Dios y la vida me hayan permitido ser su estudiante, gracias por permitirme ser su Auxiliar, su Tesista y su Amigo, por compartir su conocimiento incontables veces y por ser una guía académica, profesional y personal durante todo este camino.

Al profesor Edwin Garavito, por darme la oportunidad de ser su tesista, por la confianza depositada, la paciencia y el apoyo en cada decisión que se iba tomando.

A My Boss, Amigo y Hermano Leonardo Talero, agradezco que durante muchas etapas de este proyecto me orientó, aconsejó y enseñó el arte de la investigación, gracias por haberme levantado, cargado y lanzado nuevamente al ruedo para culminar esta etapa de mi vida, gracias por esa amistad genuina e incondicional.

A Ángela Vargas, por creer en mí, por acompañarme, por recordarme quién soy y de lo que soy capaz. Gracias por ser un instrumento de bendición y aprendizaje para mi vida, por ser un apoyo incondicional en todo momento y por su forma de ser.

A todos mis amigos que de una forma u otra estuvieron durante toda etapa, quienes, con una palabra, una guía, una orientación o una simple sonrisa apoyaron la realización de este proyecto.

Al Grupo de Investigación OPALO y todas las personas que conocí allí, a Laura Escobar, Julio Camacho, Lina y Fabián y a nuestro gran Amigo David Puentes, los llevo en mi corazón, las enseñanzas, risas y gratos momentos que pasé con ustedes hicieron de este proyecto una experiencia aún más enriquecedora.

A las escuelas de Estudios Industriales y Empresariales y la Escuela de Ingeniería de Sistemas e Informática, por alojarme en sus aulas y permitirme cumplir este sueño.

A la Universidad Industrial de Santander porque en ella pasé una temporada de aprendizaje y alegría.

DEDICATORIA

A Dios por todo lo que me ha permitido hacer

A mí mismo, quien con la culminación de esta etapa me demuestro que soy capaz de hacer lo que me propongo, de cumplir mis sueños, de luchar por lo que quiero y soy más que suficiente.

A mi papá y mis hermanas por todo el apoyo y amor que me han dado.

A mis amigos y las personas que han estado a mi lado en todo este tiempo.

Tabla de contenido

Introducción	19
1. Planteamiento del Problema	21
2. Justificación del proyecto	23
3. Objetivos	29
3.1 Objetivo General	29
3.2 Objetivos Específicos.....	29
4. Revisión bibliométrica de literatura	32
4.1 Primera etapa: Evolución del FLP	33
4.2 Segunda etapa: Revisión del problema a trabajar	38
4.3 Selección del método de solución.....	43
5. Fundamentación teórica y conceptual.....	45
5.1 Marco teórico	45
5.1.1 Planeación de la distribución de instalaciones.....	45

5.2 Jerarquía de la planeación de la distribución de instalaciones.....	46
5.2.1 Planificación sistemática de la distribución (SLP)	47
5.2.2 Fase 1. Localización.....	47
5.2.3 Fase 2. Distribución general global	47
5.2.4 Fase 3. Planes detallados de la distribución.....	47
5.2.5 Fase 4. Instalación.....	48
5.2.6 El patrón de procedimientos de SLP.....	48
5.3 Metaheurística.....	51
5.3.1 Metaheurísticas inspiradas versus no inspiradas en la naturaleza	51
5.3.2 Metaheurísticas determinísticas versus estocásticas	52
5.3.3 Metaheurísticas de búsquedas basadas en una población	52
5.4 Marco Tecnológico	52
5.4.1 Computación en la nube.....	52
5.4.2 Software	53
5.4.3 Ingeniería	53
5.4.4 Ingeniería de Software	53
6. Metodología	54
6.1 Fase 1. Revisión de la literatura e identificación de las necesidades.....	56
6.1.1 Actividades de la fase 1	56
6.2 Fase 2: Plan y diseño rápido	56

6.2.1 Actividades de la fase 2	57
6.3 Fase 3: construcción del prototipo	57
6.3.1 Actividades de la fase 3	57
6.4 Fase 4: Validación, implementación y evaluación del prototipo	57
6.4.1 Actividades de la fase 4	58
7. Resultados	59
7.1 Análisis preliminar de la literatura	59
7.1.1 Formulación del FLP	60
7.1.2 Tipo de modelo	68
7.1.3 Solución óptima	70
7.1.4 Método de solución	71
7.2 Documento de requisitos	79
7.2.1 Funciones del producto	80
7.2.2 Características del usuario	81
7.2.3 Requisitos funcionales	81
7.2.4 Requisitos no funcionales	88
7.3 Tecnologías para desarrollo web	89
7.3.1 HTML y CSS	89
7.3.2 JavaScript	90
7.3.3 PHP	90

7.3.4 Ruby on Rails.....	91
7.3.5 Angular	91
7.3.6 Django.....	91
7.3.7 Python	92
7.4 Herramienta para la generación de soluciones	92
7.4.1 Prototipo de herramienta.....	92
7.4.2 Arquitectura Django.....	96
7.4.3 Arquitectura de la herramienta.....	97
7.4.4 Diseño de datos	98
7.4.5 Diagrama de casos de uso	98
7.4.6 Diagrama de estado.....	100
7.4.7 Diagramas de actividades	101
7.4.8 Diagramas de clases.....	104
7.4.9 Modelo de optimización multiobjetivo.....	108
7.4.10 Heurística	113
7.4.11 Recocido simulado.....	116
7.5 Desarrollo y evaluación del prototipo.....	126
7.5.1 Desarrollo del prototipo 1: Algoritmo heurístico y recocido simulado	126
7.5.2 Desarrollo del prototipo 2: Backend	128
7.5.3 Desarrollo del prototipo 3: Frontend, conexión con servidor y guía de usuario....	130

7.5.4 Evaluación estadística del algoritmo de recocido simulado	137
7.5.5 Análisis del algoritmo de recocido simulado para la solución del QAP.....	144
7.5.6 Análisis de instancias de referencia probadas en la literatura.....	147
8. Conclusiones	151
9. Recomendaciones	154
Referencias Bibliográficas	156

Lista de Tablas

Tabla 1 Cumplimiento de Objetivos Específicos.....	30
Tabla 2 Cantidad de documentos por base de datos para el FLP.....	34
Tabla 3 Artículos por fuente para el problema a trabajar	41
Tabla 4 Convención información consolidada revisión de literatura	73
Tabla 5 Convención información consolidada revisión de literatura Parte 2	75
Tabla 6 Información consolidada revisión de literatura	76
Tabla 7 Ficha para los requisitos	80
Tabla 8 Requisito Funcional 1	81
Tabla 9 Requisito Funcional 2	82
Tabla 10 Requisito Funcional 3	82
Tabla 11 Requisito Funcional 4	83
Tabla 12 Requisito Funcional 5	83
Tabla 13 Requisito Funcional 6	84
Tabla 14 Requisito Funcional 7	84
Tabla 15 Requisito Funcional 8	85
Tabla 16 Requisito Funcional 9	85
Tabla 17 Requisito Funcional 10	86

Tabla 18 Requisito Funcional 11	86
Tabla 19 Requisito Funcional 12	87
Tabla 20 Requisito Funcional 13	87
Tabla 21 Herramientas de desarrollo utilizadas	95
Tabla 22 Niveles del diseño factorial.....	138
Tabla 23 Tabla ANOVA.....	144
Tabla 24 Tabla de resultados para las 15 instancias QAP	146
Tabla 25 Tabla de resultados resumen para instancias halladas en la literatura.....	148

Lista de Figuras

Figura 1 Ecuación de Búsqueda en la Primera Etapa	33
Figura 2 Filtros de área de conocimiento, tipo de documento e idioma para el FLP	34
Figura 3 Documentos por año para el FLP	35
Figura 4 Documentos publicados por país para el FLP	36
Figura 5 Documentos por autor para el FLP.....	36
Figura 6 Documentos por fuentes de información para el FLP	37
Figura 7 Nube de palabras clave para el FLP	37
Figura 8 Ecuación de búsqueda para la revisión.....	38
Figura 9 Artículos por año para el problema a trabajar	39
Figura 10 Artículos por país para el problema a trabajar.....	39
Figura 11 Aduna de países para el problema a trabajar	40
Figura 12 Artículos por afiliación del autor del problema a trabajar.....	41
Figura 13 Artículos por autor para el problema a trabajar	42
Figura 14 Aduna de autores para el problema a trabajar	43
Figura 15 Ecuación de búsqueda para la selección del método de solución	44
Figura 16 Cantidad de repeticiones por método de solución	44
Figura 17 División de la planeación de instalaciones	46

Figura 18 Fases de la planeación de instalaciones.....	48
Figura 19 Patrón de Procedimientos de SLP	49
Figura 20 Fases del proyecto	58
Figura 21 Resumen Metaheurísticas utilizadas en la revisión de literatura.....	79
Figura 22 Diagrama de flujo sobre las generalidades de la herramienta	94
Figura 23 Diagrama MVT Django.....	96
Figura 24 Arquitectura herramienta FLPWeb	97
Figura 25 Diagrama de Base de Datos de FLPWeb	99
Figura 26 Diagrama de casos de uso.....	100
Figura 27 Diagrama de estados.....	101
Figura 28 Diagrama de Actividades Crear Usuario - Iniciar Sesión	102
Figura 29 Diagrama de Actividades Simulación	103
Figura 30 Diagrama de Actividades Consulta de soluciones.....	104
Figura 31 Diagrama de clases General de FLPWeb	105
Figura 32 Diagrama de clases Paquete externals	106
Figura 33 Diagrama de clases del paquete flpwebAPP	108
Figura 34 Representación gráfica de la distribución a trabajar.....	110
Figura 35 Pseudocódigo del recocido simulado	117
Figura 36 Ejemplo para seis instalaciones en seis locaciones	120
Figura 37 Asignación inicial.....	122
Figura 38 Cinco interambios para la instalación 1 y su vecindad.....	122
Figura 39 Cuatro intercambios para la instalación 2 y su vecindad	123
Figura 40 Tres intercambios para la instalación 3 y su vecindad	123

Figura 41 Dos intercambios para la instalación 4 y su vecindad	124
Figura 42 Un intercambio para la instalación 5 y su vecindad	124
Figura 43 Diagrama de flujo del algoritmo de recocido simulado modificado	125
Figura 44 Paquete externals	127
Figura 45 Muestra código primer prototipo	127
Figura 46 Muestra código segundo prototipo	129
Figura 47 Interfaz página inicio segundo prototipo	129
Figura 48 Validación simulación segundo prototipo	130
Figura 49 Página inicio prototipo	132
Figura 50 Vista crear usuarios	132
Figura 51 Vista de inicio con usuario logueado.....	133
Figura 52 Vista iniciar sesión	133
Figura 53 Vista historial de simulaciones.....	134
Figura 54 Vista registro de simulación	134
Figura 55 Vista ingresar datos de las matrices.....	135
Figura 56 Vista validar datos simulación.....	135
Figura 57 Vista simulación con pesos objetivos.....	136
Figura 58 Vista mostrar resultados	136
Figura 59 Gráfica resumen de las corridas	140
Figura 60 Gráfica resumen de las corridas dividida por niveles del factor Q.....	141
Figura 61 Gráfica resumen de las corridas dividida por niveles del factor Lambda1	141
Figura 62 Gráfica resumen de las corridas dividida por niveles del factor Lambda2	142
Figura 63 Gráfica resumen de los tiempos de cómputo en horas para la solución.....	149

Figura 64 Modelo de regresión para la estimación de tiempo de cómputo 150

Lista de Apéndices

(Ver apéndices adjuntos y pueden ser consultados en la base de datos de la Biblioteca
UIS)

Apéndice A. Artículo Carácter Publicable RCC.....	
Apéndice B. Listado de artículos consolidados para la revisión de literatura	73
Apéndice C. Guía de Usuario	136
Apéndice D. Matrices de Prueba.zip.....	137
Apéndice E. Resultados y Tabulación Calibración Parámetros RS.zip.....	139
Apéndice F. Resultados y Tabulación QAPlib.zip	145
Apéndice G. Resultados y Tabulación Instancias Literatura.zip	147

Resumen

Título del proyecto: Diseño de una herramienta software en ambiente web como propuesta para la solución del problema de distribución de planta.*.

Autor: Reinaldo Pabón Fuentes**

Palabras clave: Desarrollo Web, FLP, Metaheurística, QAP, Recocido Simulado

Descripción: El problema de distribución de planta es un tema de gran importancia en la ingeniería industrial y en la toma de decisiones en el campo de la investigación de operaciones. Una variante importante de este problema es el problema de asignación cuadrática, que se utiliza para representar la asignación de instalaciones a locaciones cuando existe un mismo número de elementos de ambos conjuntos. Dado que el QAP es un problema NP-hard, es necesario utilizar algoritmos metaheurísticos para encontrar soluciones aceptables. Una opción ampliamente implementada es el algoritmo de recocido simulado. Sin embargo, la implementación de herramientas de fácil acceso que resuelvan el QAP es escasa. En este trabajo, se presenta una herramienta en entorno WEB (FLPWeb) que utiliza recocido simulado para resolver el problema QAP multiobjetivo mediante ponderación de objetivos haciendo uso de cuatro estrategias Manual, GMWM, SDWM y CRITICM. La herramienta logra resolver en tiempos razonables problemas clásicos de la literatura científica con instancias que van desde las seis instalaciones hasta las treinta instalaciones, logrando así una herramienta de eficiente y eficaz para resolver problemas QAP.

*Proyecto de pregrado

** Facultad de Físico-mecánicas, Escuela de Estudios Industriales y Empresariales. Director: MSc. Edwin Alberto Garavito Hernández, Codirector: PhD. Gabriel Rodrigo Pedraza Ferreira

Abstract

Title: Design of software in a web environment for solving the Facility Layout Problem.*.

Author: Reinaldo Pabón Fuentes**

Keywords: FLP, Metaheuristic, QAP, Simulated Annealing, Web environment.

Description: The facility layout problem is critical for the decision-making process in industrial engineering and operations research. The quadratic assignment problem is a significant subtype of this problem that depicts the assignment of facilities to places with the same number of members in both sets. It is an essential variety of this problem. Because it is an NP-hard issue, adequate solutions to the QAP can only be found using metaheuristic algorithms. A widely implemented option is the simulated annealing algorithm. However, implementing easily accessible tools that solve the QAP is scarce. This work presents a WEB environment tool (FLPWeb) that uses simulated annealing to solve the multi-objective QAP problem using weighting strategies through four strategies: Manual, GMWM, SDWM, and CRITICM. The tool solves classic problems from scientific literature with instances ranging from six to thirty facilities, achieving an efficient and effective tool for solving QAP problems.

*Bachelor Project

** Faculty of Physicomechanical Engineering and Business School. Advisor 1: MSc. Edwin Alberto Garavito Hernández, Advisor 2: PhD. Gabriel Rodrigo Pedraza Ferreira

Introducción

La planificación de la distribución de planta es importante para tener éxito en la competencia global actual. Esto incluye un enfoque multidisciplinario y habilidades para mejorar la ubicación de la maquinaria y equipo en una planta de producción, lo que puede reducir los costos de operación hasta en un 30% y apoyar la producción rentable. El problema de diseño de instalaciones o FLP (Facility Layout Planning por sus siglas en inglés) se refiere a la ubicación de instalaciones en un área específica, donde se destacan enfoques algorítmicos y procedimentales (Yang Y Kuo, 2003). Debido a la diversidad de empresas de manufactura, se han propuesto muchas soluciones para el FLP. Una opción destacada es la aplicación de técnicas metaheurísticas, que suelen proporcionar soluciones de mayor calidad que los métodos heurísticos y tienen un tiempo de cómputo menor que los métodos exactos. Algunas técnicas metaheurísticas comunes incluyen algoritmos genéticos, recocido simulado, optimización por enjambre de partículas y búsqueda tabú (Paper, 2014).

En el presente proyecto de grado se propone el desarrollo de un prototipo software en entorno web enfocado en solucionar el FLP (*FLPWeb*) derivado de un análisis de flujo de trabajo e implementando un algoritmo de recocido simulado para la solución de un problema de optimización multiobjetivo mediante ponderación de funciones, donde el entorno web facilitan flexibilidad ya que: “los servicios web no están atados a ningún sistema operativo o lenguaje de programación específico...” (Laudon Y Laudon, 2012, p. 193). Es fundamental mencionar que

esta herramienta permitirá a quienes estén interesados en el problema de diseño de instalaciones o FLP poner en práctica los conocimientos teóricos adquiridos y mejorar su rendimiento profesional. Los resultados estadísticos indican que la herramienta logra dar soluciones de manera eficiente y eficaz a problemas FLP multiobjetivos modelados como problemas de asignación cuadrática

1. Planteamiento del Problema

El problema de distribución de planta busca encontrar un arreglo de instalaciones o departamentos en el área de una planta cumpliendo con uno o varios objetivos. Este problema ha sido abarcado desde diferentes puntos de vista y soluciones, se han creado algoritmos para encontrar la solución al mismo, a pesar de los esfuerzos realizados por obtener soluciones óptimas, se ha demostrado que el problema no tiene una solución en un tiempo computacional razonable. Debido a esto, se han generado diversas investigaciones para crear otros tipos de algoritmos llamados heurísticos y metaheurísticos para poder encontrar una buena solución, dando así una posible solución a todas las empresas que tienen que afrontar este problema, que según Muther (1973), a pesar de no ser una actividad operativa en una empresa, se debe realizar desde un plano estratégico para poder reducir costos y triunfar en el mercado. En la actualidad muy pocas empresas pueden conservar sus instalaciones antiguas, ya que se han presentado cambios importantes en las técnicas y el equipo de producción de las fábricas, dejando como única opción materializarse con la misma rapidez con la que estos cambios aparecen (Tompkins et al., 2011).

En la literatura, se encuentran pocos trabajos que han utilizado herramientas software para encontrar o mejorar la solución del FLP, entre ellos se encuentra a Tasadduq *et al.* (2015) y a Banduka *et al.* (2017) quienes utilizan los softwares VIP PLANOPT 10 y visTABLE, respectivamente. Otros pocos, desarrollan pequeñas interfaces para que el diseñador de planta

pueda interactuar con el algoritmo: García-Hernández *et al.* (2014) y Sikaroudi y Shahanaghi (2016). Solo Garcia-Hernandez *et al.* (2013) crearon una herramienta web para integrar el algoritmo tratado en su investigación. Revisando la escasa actividad con respecto a crear herramientas web para tratar el problema de distribución de planta, se propone la creación de una herramienta web para darle solución al FLP, aportando una nueva forma de estudiar y resolver este problema; además, esta herramienta será de acceso libre para cualquier usuario que quiera trabajar con ella, generando así mayores oportunidades para aquellos que estén interesados en aprender, conocer, mejorar o darle solución a una distribución de planta.

2. Justificación del proyecto

La planeación de la distribución de planta es una tarea multidisciplinaria que requiere de un conocimiento intensivo, amplia experiencia, información, conocimiento y habilidades, la cual es de vital importancia para sobrevivir en el ambiente competitivo global actual (Jiang Y Nee, 2013; Shariatzadeh et al., 2012). El FLP se refiere a la planeación de la distribución de la planta como el diseño de los planes de ubicación de las máquinas o equipos en un piso de manufactura, los cuales pueden mejorar la forma en que la planta apoye la producción de la empresa, además, en algunos casos el estudio y la aplicación efectiva de una buena distribución de planta, permite reducir los costos de operación de una fábrica hasta en un 30% y brinda los fundamentos para una producción rentable (Tompkins et al., 2011). Así mismo, Kulkarni, Bhatwadekar, y Thakur (2015) definen la ubicación de instalaciones en un área específica como el problema de diseño de instalaciones o FLP (Facility Layout Problem) por sus siglas en inglés.

En la actualidad muy pocas empresas pueden conservar sus instalaciones antiguas, ya que se han presentado cambios importantes en las técnicas y el equipo de producción de las fábricas, dejándole como única opción a las empresas materializarse con la misma rapidez con la que estos cambios aparecen (Tompkins et al., 2011). Por su parte, Tong, Van Luttervelt, Wedman, y Bikker (2003) señalan que la industria de la manufactura se han encontrado con grandes retos a medida que pasa el tiempo, debido a la competencia intensa a la que se enfrentará; por esto, para que una

empresa de manufactura se pueda mantener competitiva, está en la obligación de permitir el diseño o rediseño de sus sistemas productivos de acuerdo a los drásticos, rápidos y frecuentes cambios en el ambiente en que la compañía se encuentra inmersa, sin mencionar que la industria de la manufactura necesitará rediseñar sus fábricas existentes o diseñar nuevas fábricas con mucha más frecuencia.

Castillo, Westerlund, Emet, y Westerlund (2005), definen el objetivo principal del diseño de la distribución de planta en bloques como la minimización del costo asociado con las interacciones que se tendrán entre departamentos. Estas interacciones pueden reflejar dos cosas, o bien el costo de la intensidad del flujo de materiales (el total de la cantidad de materiales que se transportan entre departamentos) o la preferencia de la cercanía (adyacencia) entre departamentos; este costo se puede calcular como la distancia lineal entre departamentos multiplicada por el flujo de material o una calificación de cercanía entre los centros de cada par de departamentos. Este problema está principalmente sujeto a dos restricciones, la primera son los requerimientos de área del piso del departamento y la segunda son todas las restricciones de localización de los departamentos. De hecho, las restricciones de localización pueden referirse a que los departamentos: no se pueden colocar uno encima de otro, deben ser ubicados dentro de la instalación y algunos deben estar fijos a una ubicación o no pueden ser ubicados en una región específica (Castillo Y Westerlund, 2005). Algo semejante ocurre con Tompkins et al (2011), quienes señalan que las distribuciones de planta se pueden tomar desde dos puntos de vista, el primero es la distribución en bloques, cuyo objetivo es mostrar las ubicaciones y los tamaños relativos de los departamentos, por otra parte, la distribución detallada indica la ubicación exacta de todo el equipo, las mesas de trabajo y las áreas de almacenamiento de cada departamento. Además, destacan la importancia y la principal preocupación de estos dos tipos de distribuciones,

la distribución en bloques se enfoca en los macro flujos en la planta, mientras que la distribución detallada se encarga de los micro flujos. Cabe destacar que una distribución de planta completa y efectiva tiene estos dos tipos de distribución desarrollados y evaluados cuidadosamente.

Muther (1973) responde a la pregunta: ¿Por qué planear la distribución de planta?; menciona que, en algunos casos, generar una distribución puede ser tan sencillo como trasladar de un área a otra los equipos, maquinarias, bancos de trabajo, etc., para luego tomarse el tiempo de reubicar repetidamente estos equipos hasta quedar satisfecho. Sin embargo, Muther hace hincapié en que para las industrias el solo hecho de hacer una reubicación trae como resultado pérdidas en tiempo, tener personal improductivo e incluso daños a los equipos por el traslado. Por esto, es importante invertir tiempo en la planeación de la distribución antes de ser instalada para prevenir pérdidas para la empresa (Hales Y Muther, 2015)

Yang y Kuo (2003) señalan que la mayoría de la literatura del problema del diseño de la distribución de planta se divide en dos enfoques: el enfoque algorítmico y el procedimental. El enfoque algoritmo simplifica las restricciones y los objetivos de diseño para alcanzar una función objetivo-sustituta, cuya solución puede luego ser obtenida. Por otra parte, Muther demuestra que el enfoque procedimental puede incluir objetivos tanto cualitativos como cuantitativos en el proceso de diseño (Hales Y Muther, 2015). Sobre el enfoque procedimental, Yang y Kuo (2003) mencionan que el proceso de diseño es dividido en pasos que se completan secuencialmente. En cuanto a este enfoque, se pueden mencionar los trabajos realizados por Apple y Reed citados por Tompkins et al. (2011), quienes desarrollaron secuencias de pasos detalladas y un plan de ataque sistemático respectivamente; esto con el fin de planificar, preparar y producir una distribución de planta (Tompkins et al., 2011). Otro trabajo relevante fue el realizado por Muther, “quien desarrolló un procedimiento para la distribución que denominó planificación sistemática de la

distribución o SLP (Systematic Layout Planning” (Tompkins et al., 2011). Es importante destacar que a pesar del tiempo desde que Muther propuso su procedimiento, en la actualidad es ampliamente utilizado y reconocido para planificar el diseño de la distribución de plantas (Kulkarni et al., 2015).

El enfoque algorítmico se encargó principalmente de desarrollar “procedimientos o algoritmos formales que ayudan al analista a desarrollar o mejorar una distribución, y al mismo tiempo le brindan ciertos criterios objetivos para facilitar la evaluación de diversas alternativas de distribución que surgen en el proceso” (Tompkins et al., 2011) Existen numerosos ejemplos de investigaciones de esta naturaleza. El primer ejemplo es el desarrollado por Armour y Buffa (1963) quienes propusieron uno de los primeros algoritmos de distribución publicados, CRAFT (Computerized relative allocation of facilities technique, Técnica computarizada de asignación relativa de plantas). Otro ejemplo es LOGIC (Layout optimization with guillotine induced cuts: Optimización de la distribución con cortes tipo guillotina), desarrollado por Tam citado por Tompkins et al. (Tompkins et al., 2011), entre otros muchos que se pueden encontrar en la literatura.

Muchas soluciones han sido propuestas para el FLP, pero éste es un problema de alta optimización combinatoria, por esto, diferentes metodologías han sido propuestas para lograr soluciones trabajables. En los últimos años, los investigadores se han interesado por las metaheurísticas, éstas obtienen mejor calidad en la solución que los métodos heurísticos, además, su tiempo de cómputo es menor que los métodos exactos. En la literatura es posible encontrar métodos metaheurísticos tales como los algoritmos genéticos (GA: Genetic Algorithm), el recocido simulado (SA: simulate annealing), optimización por enjambre de partículas (PSO: particles swarm optimization), búsqueda tabú (TS: tabu search) (Paper, 2014), ejemplos de estos

trabajos se pueden revisar en Bozorgi et al., (2015); Hosseini-Nasab Y Emami, (2013); J. Liu et al., (2021); Masrom et al., (2017); McKendall Y Liu, (2012); Peng et al., (2018); Pourvaziri Y Naderi, (2014); Ramkumar et al., (2009a); Uddin, (2015); Zha et al., (2020) respectivamente.

Según Muther, el plan de diseño es la entrega que realiza la persona responsable de planear una distribución, además, esta persona es el diseñador de un producto y su plan se convierte en la especificación del mismo, el producto final es el diseño físicamente instalado (Muther, 1973). Por esto, se han creado paquetes de diseño de distribuciones comercialmente disponibles, los cuales permiten evaluar diferentes opciones de solución a partir del ingreso de datos de forma textual y utilizando ambientes CAD (CAD: Computer Aided Design, Diseño Asistido por Computadoras). Como ejemplo de estas, se pueden mencionar: Flow Planner, Layout-iQ, VIP-PLANOPT (Tompkins et al., 2011), entre otras, que ayudan a mejorar la entrega del producto final del diseñador de plantas. Sin embargo, se menciona al respecto:

“Existen varios paquetes comerciales para la distribución de una planta. No obstante, con ciertas excepciones, tales paquetes sólo están destinados para presentaciones (es decir, son herramientas electrónicas de dibujo que facilitan la presentación o el mantenimiento de una distribución existente) o están diseñadas principalmente como una herramienta para evaluar una distribución...” (Tompkins et al., 2011, p. 303)

Además de la consideración de Tompkins et al. (2011), éstas tienen un alto costo comercial, ninguna de ellas proporciona una plataforma basada en la web imposibilitando la distribución y uso de la misma, presentan gran cantidad de funcionalidades que en muchos casos no son utilizadas al máximo de su capacidad y ofrecen un número limitado de algoritmos o metaheurísticas para evaluar la distribución planteada.

Teniendo en cuenta lo anterior, sin olvidar la importancia y la posibilidad de darle solución a este problema mediante un nuevo enfoque de flujo de trabajo, así mismo, cumplir el objetivo de minimizar el costo asociado con las interacciones entre departamentos, se propone diseñar y desarrollar una herramienta software, para facilitar el proceso de planeación de la distribución de planta mediante la generación de distribuciones en bloques o áreas de trabajo que minimicen el costo asociado de las interacciones entre las mismas, sin la necesidad de generar un diseño en detalle del plano final de planta. Esta herramienta se desarrollará en un entorno web con el fin de minimizar el tiempo del proceso y resaltando que: “los servicios web no están atados a ningún sistema operativo o lenguaje de programación específico...” (Laudon Y Laudon, 2012, p. 193). Además, esta herramienta será asequible a las personas interesadas en este tema, permitiendo que la herramienta sea extensible a otros casos que se presenten durante el proceso de la planeación de la distribución de la planta. Finalmente, es de gran importancia resaltar que esta herramienta, permitirá a las personas interesadas en el FLP, aplicar conocimientos teóricos aprendidos y mejorar el desempeño profesional de las mismas.

3. Objetivos

3.1 Objetivo General

Diseñar y desarrollar una herramienta software en ambiente web para la generación y evaluación de alternativas de distribución de plantas mediante metaheurísticas.

3.2 Objetivos Específicos

- Identificar métodos de solución desarrollados a partir de la literatura relacionada con el problema de distribución de planta en bloques y algoritmos metaheurísticos usados para la solución del mismo.
- Identificar las necesidades funcionales, tecnológicas y los parámetros de entrada de la herramienta a desarrollar.
- Diseñar la arquitectura de la herramienta para la generación y evaluación de soluciones alternativas al problema de distribución de espacios a partir del análisis de relaciones de actividad.
- Desarrollar y evaluar un prototipo de la herramienta software con al menos un algoritmo metaheurístico, que permita generar soluciones y evaluar el desempeño de la distribución obtenida desde el punto de vista de la función objetivo.

- Realizar un artículo de carácter publicable en base a los resultados obtenidos del trabajo realizado

Tabla 1*Cumplimiento de Objetivos Específicos*

Objetivo	Sección o Documento
Identificar métodos de solución desarrollados a partir de la literatura relacionada con el problema de distribución de planta en bloques y algoritmos metaheurísticos usados para la solución del mismo.	Capítulo 4 y Subcapítulo 7.1
Identificar las necesidades funcionales, tecnológicas y los parámetros de entrada de la herramienta a desarrollar	Subcapítulo 7.2 y Subcapítulo 7.3
Diseñar la arquitectura de la herramienta para la generación y evaluación de soluciones alternativas al problema de distribución de espacios a partir del análisis de relaciones de actividad.	Subcapítulo 7.4
Desarrollar y evaluar un prototipo de la herramienta software con al menos un algoritmo metaheurístico, que permita generar soluciones y evaluar el	Subcapítulo 7.5

desempeño de la distribución obtenida desde el punto de vista de la función objetivo

Realizar un artículo de carácter publicable en base a los resultados obtenidos del trabajo realizado.

Apéndice A. Artículo de
Carácter Publicable

4. Revisión bibliométrica de literatura

Durante siglos el ser humano ha estado diseñando las instalaciones físicas que lo rodean. Ejemplos de esto son los imperios de Grecia y Roma, quienes diseñaron la forma en que sus pueblos debían estar ubicados, prueba de esto está registrado en la historia de los mismos. (Moore, 1974). En las últimas décadas, investigadores han trabajado el problema de distribución de planta (FLP: Facility Layout Problem), los cuales han desarrollado diferentes métodos de solución incluyendo las variaciones de este.

En este análisis se pretende mostrar un estudio de la evolución investigativa que ha tenido el FLP, por otra parte, se pretende también realizar una búsqueda de las variaciones y propuestas que los investigadores han hecho en sus desarrollos. Para lograr esto, se realizaron tres etapas de búsqueda: la primera Evolución del FLP, tiene que ver con la búsqueda de investigaciones hechas entorno al FLP, de esta etapa se obtiene una mejora en la ecuación de búsqueda al incluir palabras clave obtenidas del análisis de esta. En la segunda etapa Revisión del problema a trabajar, se incluyen las palabras clave y se hacen filtros para obtener resultados representativos para el FLP estático, el cual es el problema a trabajar en el presente proyecto. Finalmente, en la tercera etapa Selección del método de solución, que es independiente a las otras dos, se realiza un análisis para revisar cuál es la metaheurística más utilizada en el FLP.

4.1 Primera etapa: Evolución del FLP

En esta etapa se incluyen las investigaciones sobre el FLP, para esto se utiliza la frase “Facility Layout Problem” en el título, resumen y palabras clave en las bases de datos Scopus, Web of Science y Science Direct, además, se utilizan las comillas para que la frase completa esté incluida en el mismo orden en la búsqueda; estas ecuaciones de búsqueda iniciales se pueden ver en la Figura 1.

Figura 1

Ecuación de Búsqueda en la Primera Etapa para las Diferentes Bases de Datos

Scopus: TITLE-ABS-KEY ("facility layout problem")	Web of Science: TEMA: ("facility layout problem")	Science Direct: Title, abstract, keywords: "facility layout problem"
---	---	--

A pesar de que la cantidad de documentos encontrados en Scopus y Science Direct son bastante aproximados (Ver Tabla 2), se escoge la base de datos Scopus para realizar el análisis y proseguir con las siguientes etapas. Esta decisión se toma debido a que es la base de datos que tiene más documentos sobre el FLP, además, se evidencia que gran cantidad de los documentos que se encuentran en las otras dos bases de datos, también están presentes en Scopus.

Tabla 2*Cantidad de documentos por base de datos para el FLP*

Base de datos	Cantidad de documentos
Scopus	742
Web of Science	288
Science Direct	615

Se realizan filtros (Ver Figura 2) en las áreas de conocimiento, tipo de documento e idioma, para trabajar sobre los campos que tienen validez en el análisis, validar con documentos publicados y eliminar idiomas que no son relevantes en el estudio, respectivamente. De estos filtros se obtienen 667 documentos para realizar el análisis de esta primera etapa.

Figura 2*Filtros de área de conocimiento, tipo de documento e idioma para el FLP*

Para realizar el análisis se utilizó el software The Vantage Point (Search Technology, 2018), en el cual se agregan los resultados de la búsqueda realizada, esto es los registros de los 667 documentos que se seleccionaron anteriormente. Después de esto se realiza una limpieza en el campo nombre de autores debido a que pueden existir nombres duplicados o pueden aparecer con

múltiples variaciones de un documento a otro y terminar siendo el mismo nombre. Finalmente se procede a realizar el análisis con las gráficas presentadas a continuación.

Figura 3

Documentos por año para el FLP

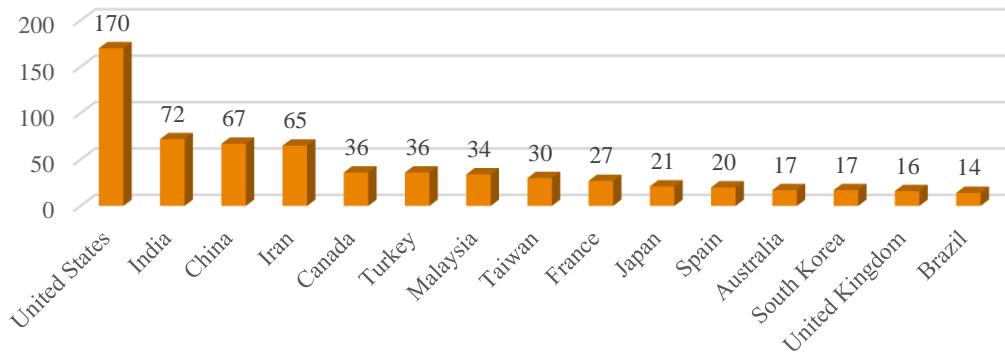


En primer lugar, en la Figura 3 se puede visualizar la cantidad de documentos publicados por año para el FLP; desde mediados de los setenta empezó el estudio sobre este, pero no fue hasta la década de los noventa que se inicia con una tendencia positiva en la investigación de este tema. Sin embargo, durante el final del siglo XX e inicios del XXI, se presentó una caída en la publicación de documentos; a pesar de esto, se puede observar que en los últimos diez años se han publicado artículos sin seguir una tendencia claramente definida.

En cuanto a la cantidad de documentos publicados por cada país, en la Figura 4, se puede observar que Estados Unidos, India, China e Irán han contribuido con más del 50% de documentos publicados, esto es 170, 72, 67 y 65 respectivamente; en la gráfica se evidencia además que el único país latinoamericano que está entre los 15 países que más han publicado sobre el FLP es Brasil con 14 documentos. Por otra parte, Singh, S. P. ha sido el autor que más ha publicado sobre el FLP (ver Figura 5), con 20 artículos; se nota en la gráfica que hay parejas o tríos de autores que tienen la misma cantidad de publicaciones, podría existir una alta probabilidad de que estos autores hayan trabajado juntos en las publicaciones.

Figura 4

Documentos publicados por país para el FLP



Cabe destacar también, que en la Figura 6 se evidencia que la International Journal of Production Research y la European Journal of Operational Research, son las fuentes de información con más participación en la publicación de documentos sobre el FLP, cada una con 73 y 43 documentos respectivamente.

Figura 5

Documentos por autor para el FLP

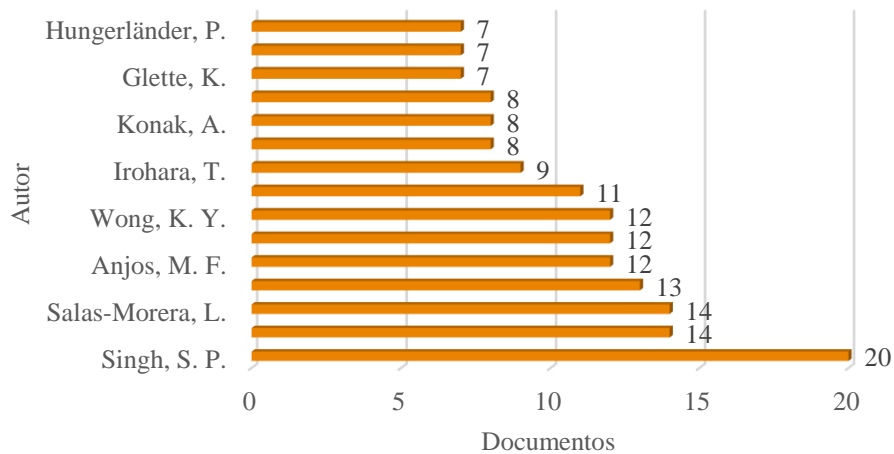


Figura 6

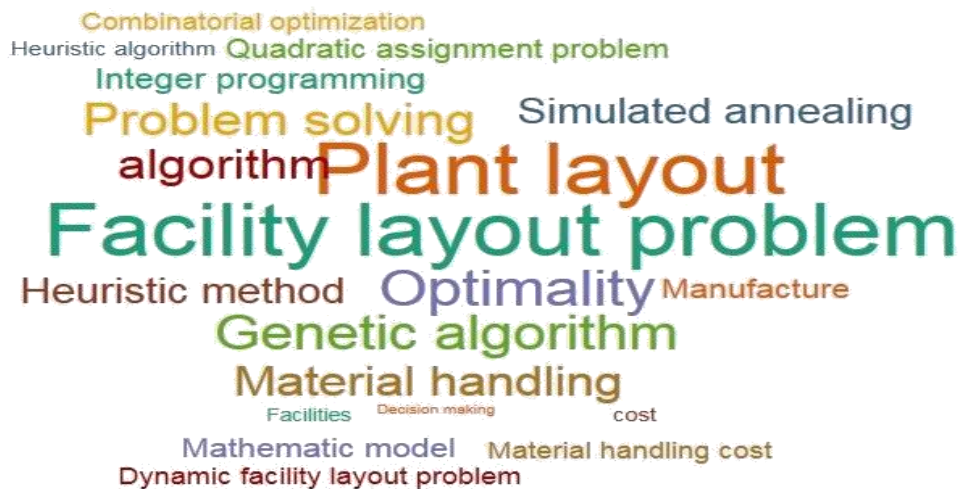
Documentos por fuentes de información para el FLP



Finalmente, se decide incluir la palabra clave “*Plant Layout*” en la ecuación de búsqueda de la segunda etapa, esto se da porque es la palabra de mayor tamaño en la nube de palabras de la Figura 7, representando su mayor utilización en los documentos analizados (437 veces). Se puede señalar también que es tema de estudio en el FLP los métodos y algoritmos heurísticos y metaheurísticos que se plantean como una solución para el presente proyecto.

Figura 7

Nube de palabras clave para el FLP



4.2 Segunda etapa: Revisión del problema a trabajar

En esta etapa se acota la búsqueda añadiendo las siguientes palabras clave a la ecuación de búsqueda: “Plant Layout”, “metaheuristics” y “heuristic” (junto a sus variaciones), “block layout” y “Systematic Layout Planning”, estas cuatro últimas se incluyen debido a que el proyecto se enfocará en estas metodologías de solución y subtemas del FLP. Además, se limita el período de tiempo desde el 2007 hasta el 2018 y finalmente se limita la búsqueda a artículos. Ver Figura 8.

Figura 8

Ecuación de búsqueda para la revisión

```
( TITLE-ABS-KEY ( "facility layout proble*" ) OR TITLE-ABS-KEY ( "plant layout" )
OR TITLE-ABS-KEY ( "systemat* layout plan*" ) OR TITLE-ABS-KEY ( "block layout"
) ) AND ( TITLE-ABS-KEY ( "heuristic" ) OR TITLE-ABS-KEY ( "metaheuristic" ) OR
TITLE-ABS-KEY ( "meta-heuristic" ) OR TITLE-ABS-KEY ( "meta heuristic" ) ) AND (
LIMIT-TO ( SUBJAREA , "ENGI " ) OR LIMIT-TO ( SUBJAREA , " COMP " ) OR
LIMIT-TO ( SUBJAREA , " DECI " ) OR LIMIT-TO ( SUBJAREA , " MATH " ) OR
LIMIT-TO ( SUBJAREA , " BUSI " ) ) AND ( LIMIT-TO ( LANGUAGE , "English " ) )
AND ( LIMIT-TO ( DOCTYPE , "ar" ) ) PUBYEAR > 2006 AND PUBYEAR < 2019
```

De la anterior ecuación de búsqueda se obtienen 152 artículos, de estos se hace una revisión rigurosa eliminando artículos duplicados y artículos que no tengan relación con la problemática a trabajar. Asimismo, se eliminan artículos que incluyan el tema Dynamic Facility Layout Problem (DFLP), debido a que el proyecto se realizará con un período de tiempo estático, es decir, la distribución de planta no cambiará de acuerdo con varios períodos de tiempo con demandas, costos variables y otras características propias del DFLP. Del filtro anterior, se obtienen 54 artículos de los cuales se presenta el siguiente análisis.

En primer lugar, en la Figura 9 se puede visualizar que existe un comportamiento parecido a la etapa anterior, esto en cuanto a los artículos publicados por año. Cabe destacar que a partir del año 2008 se ha mantenido una constante de publicar por lo menos 2 artículos o más por año. A pesar de que, en el análisis de la evolución del FLP, Estados Unidos era el país que tenía más publicaciones (170 para ser exactos), en esta etapa, India e Irán son los países que más han hecho publicaciones, 14 y 9 respectivamente, dejando a Estados Unidos en tercer lugar con 8 artículos (ver Figura 10).

Figura 9

Artículos por año para el problema a trabajar

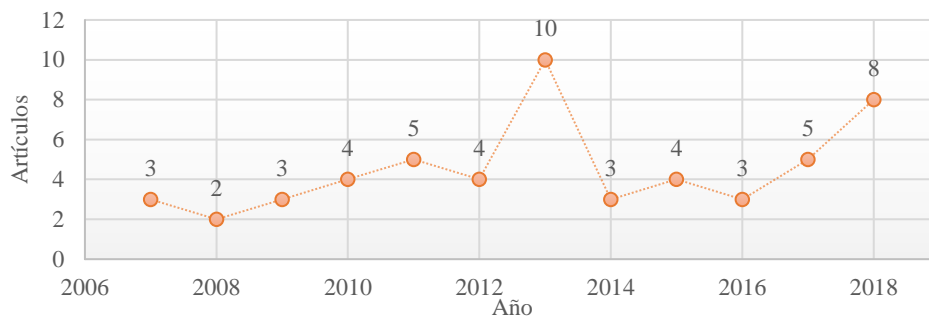
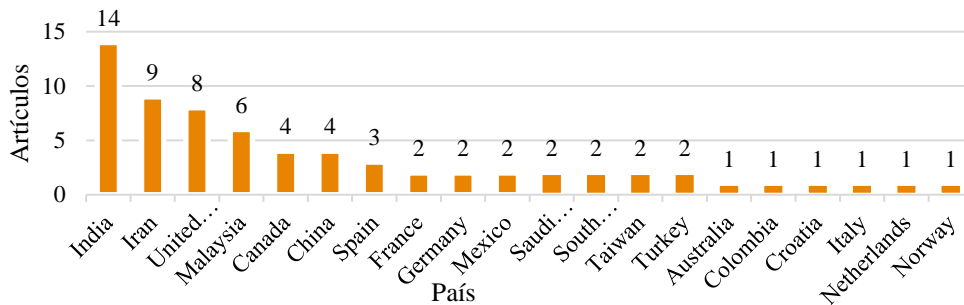


Figura 10

Artículos por país para el problema a trabajar

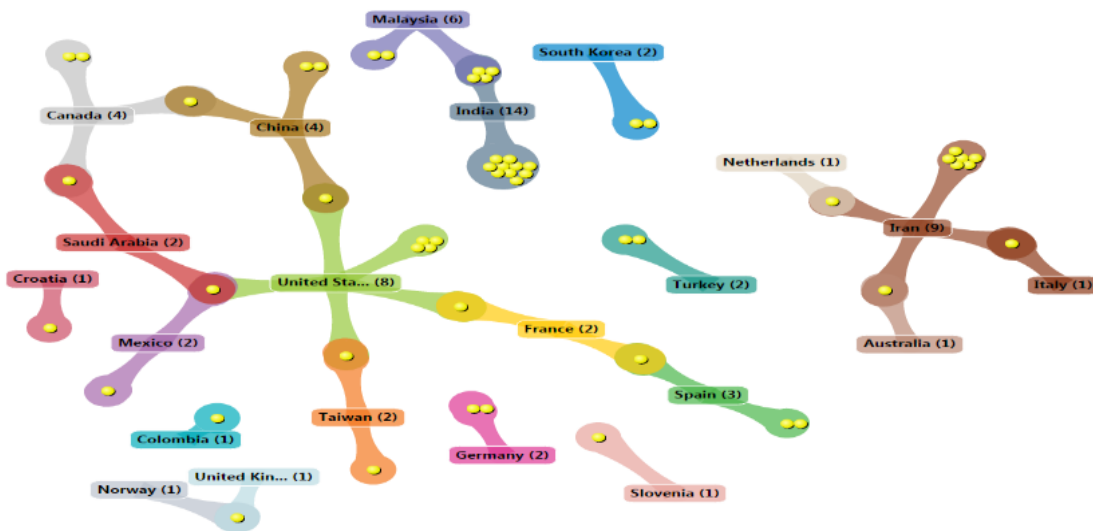


Sin embargo, Estados Unidos es uno de los países que más ha realizado publicaciones conjuntas con otros países, esto se muestra en que el 50% de sus artículos publicados han sido con otros países, en cambio India del total de sus artículos, el 28% ha sido publicado junto a otros países (ver Figura 11). En esta etapa es de suma importancia resaltar que hay un aporte por parte de Colombia, donde realizan una aplicación real utilizando un algoritmo genético para solucionar el problema de distribución de planta (Atencio Y Rodado, 2016).

Es importante agregar que el Birla Institute of Technology and Science y el Indian Institute of Technology Delhi son las universidades que tienen más publicaciones en el problema a trabajar, con 4 artículos cada una de ellas (ver Figura 12).

Figura 11

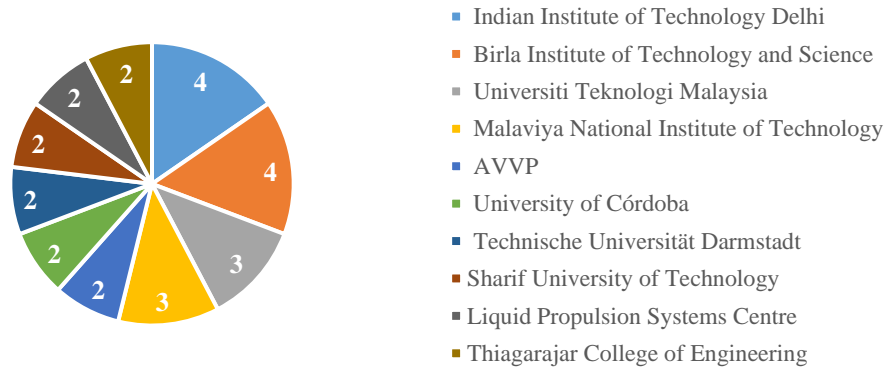
Aduna de países para el problema a trabajar



Adaptado de: Software Vantage Point, 2018.

Figura 12

Artículos por afiliación del autor del problema a trabajar



Ahora, si se menciona sobre las fuentes de información con más artículos publicados, en la Tabla 3 se puede visualizar que son la International Journal of Production Research, la European Journal of Operational Research y la Expert Systems with Applications, cada una con 10 y 5 artículos respectivamente para las dos últimas.

Tabla 3

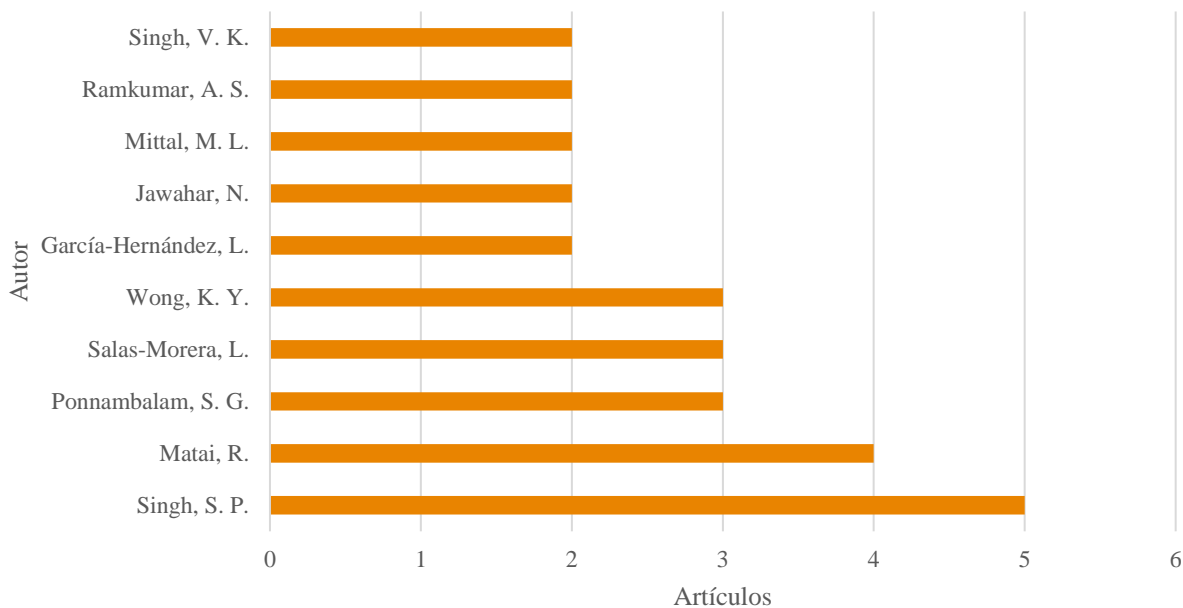
Artículos por fuente para el problema a trabajar

Fuente	Artículos
International Journal of Production Research	10
European Journal of Operational Research	5
Expert Systems with Applications	5
Computers and Industrial Engineering	3
International Journal of Advanced Manufacturing Technology	3
International Journal of Operational Research	2
International Journal of Simulation Modelling	2
Journal of Intelligent Manufacturing	2
Applied Mathematics and Computation	1
Applied Soft Computing Journal	1

Con respecto a los autores que más han publicado sobre el problema a trabajar, se observa en la Figura 13, que Singh, S. P. sigue siendo el autor que más publica con 5 artículos, esto con relación a la etapa en la que se analizó la evolución del FLP, le sigue Matai, R con 4 artículos. Además, al revisar la cantidad de artículos publicados en conjunto con otros autores, se evidencia que Singh, S. P. comparte sus 5 publicaciones con otros autores y que la mayoría de los autores no escriben sus artículos solos sino en conjunto con otros investigadores (ver Figura 14), confirmando lo que se había planteado en la etapa anterior con respecto al trabajo conjunto de autores.

Figura 13

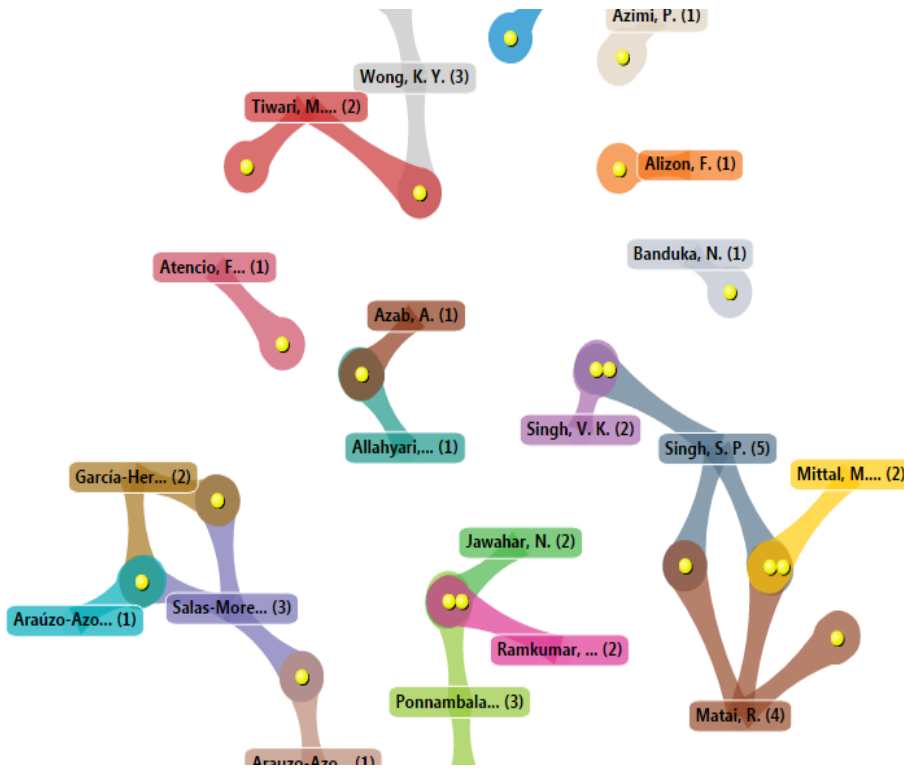
Artículos por autor para el problema a trabajar



Como una breve conclusión al análisis bibliométrico desarrollado en las dos etapas anteriores, se puede afirmar que el problema del FLP acotado a las características del presente proyecto se encuentra vigente y, además, existe un alto potencial de investigación para proponer herramientas que ayuden con la solución del FLP

Figura 14

Aduna de autores para el problema a trabajar



Adaptada de Software Vantage Point, 2018.

4.3 Selección del método de solución

Se realiza una nueva búsqueda para encontrar todos los trabajos hechos entorno al FLP utilizando como método de solución metaheurísticas; para esto, se crea una ecuación de búsqueda donde se utilizan como palabras clave “Facility Layout Problem”, “facilities design”, “Facility Layout design”, “facilities Planning”, “heuristic” y “metaheuristics”, cada una de estas con sus variaciones en plural o diferentes formas de escritura, además, en esta búsqueda no se tuvieron en cuenta límites de tiempo (ver Figura 15).

Figura 15

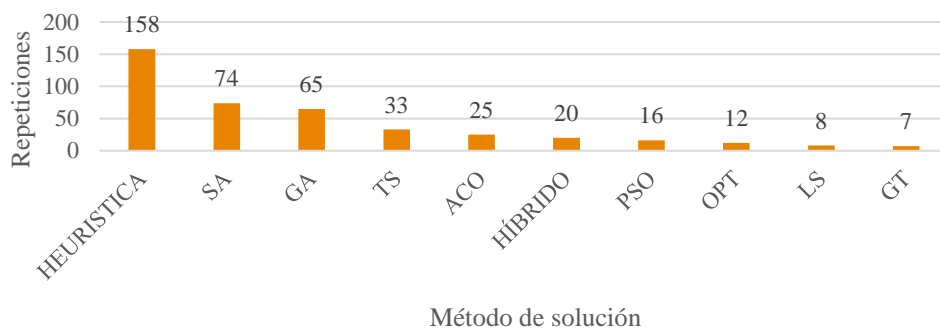
Ecuación de búsqueda para la selección del método de solución

```
TITLE-ABS-KEY ( "Facilit* layout* proble*" ) OR TITLE-ABS-KEY ( "Facilit* layout*" ) OR
TITLE-ABS-KEY ( "facilit* design" ) OR TITLE-ABS-KEY ( "facilit* layout* design" ) OR
TITLE-ABS-KEY ( "facilit* plan*" ) OR TITLE-ABS-KEY ( "FLP" ) OR TITLE-ABS-KEY (
"Layout* proble*" ) AND ( TITLE-ABS-KEY ( "heuristic" ) OR TITLE-ABS-KEY (
"metaheuristic*" ) OR TITLE-ABS-KEY ( "meta-heuristic" ) OR TITLE-ABS-KEY ( "meta
heuristic" ) ) AND ( LIMIT-TO ( SUBJAREA , "ENGI" ) OR LIMIT-TO ( SUBJAREA , "COMP"
) OR LIMIT-TO ( SUBJAREA , "DECI" ) OR LIMIT-TO ( SUBJAREA , "MATH" ) OR LIMIT-
TO ( SUBJAREA , "BUSI" ) ) AND ( LIMIT-TO ( DOCTYPE , "ar" ) ) AND ( LIMIT-TO (
LANGUAGE , "English" ) )
```

De esta búsqueda, se encuentran 576 artículos, de los cuales se revisó el título, resumen y palabras claves, para poder tabular la cantidad de veces (repeticiones) que se ha utilizado cada metaheurística en la literatura, dando como resultado la Figura 16, que representa los 10 métodos de solución más utilizados en la literatura.

Figura 16

Cantidad de repeticiones por método de solución



La Figura 16 muestra que el método de solución más utilizado para el FLP son las heurísticas, seguido de ellas están las metaheurísticas Recocido Simulado, Algoritmos Genéticos (GA: Genetic Algorithm por sus siglas en inglés) y Búsqueda Tabú (TS: Tabu Search por sus siglas en inglés). Se decide entonces utilizar la metaheurística recocido simulado debido a que es la más

utilizada en la literatura para resolver el FLP; esto la convierte en una metaheurística con bastante soporte, gran documentación, un gran número de ejemplos y sin mencionar que obtiene una muy buena solución, cercana al óptimo en un tiempo de cómputo bastante razonable.

5. Fundamentación teórica y conceptual

5.1 Marco teórico

5.1.1 Planeación de la distribución de instalaciones

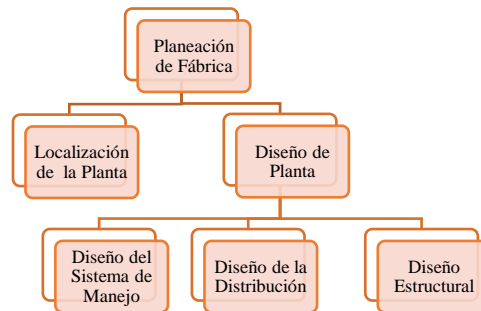
La ubicación de las instalaciones necesarias en un área específica de una planta es llamado el problema de distribución de instalaciones (Kulkarni et al., 2015). Además, el FLP se ha definido como: encontrar un arreglo factible de departamentos que no se superpongan teniendo unos requisitos dados de área en una instalación minimizando el costo de la interacción entre los mismos departamentos. Este costo de interacción entre dos departamentos es calculado u expresado como el flujo multiplicado por la distancia entre los mismos (Meller Y Bozer, 1997).

5.2 Jerarquía de la planeación de la distribución de instalaciones

Kulkarni et al (2015) proponen una división de la planeación de instalaciones (Ver Figura 17), señalando que la misma tiene un gran campo de trabajo, sin mencionar que muchas actividades se encuentran en este trabajo.

Figura 17

División de la planeación de instalaciones



Adaptado de: (Kulkarni et al., 2015). *A literature review of facility planning and plant layouts.*

International Journal of Engineering Sciences y Research Technology, 4(3), 35–42

La localización de la planta es la ubicación de una instalación con respecto a sus clientes, proveedores y otras instalaciones con las que interactúa. La estructura de la planta consiste de los edificios y los servicios públicos (ejemplo: gas, agua, energía, calor, luz, aire, aguas residuales), que se deben tener en cuenta. Asimismo, la distribución consiste de las localizaciones relativas de todos los equipos, maquinaria y mobiliario dentro de la estructura. Finalmente, el sistema de manejo trata del mecanismo por el cual todas las interacciones requeridas por la distribución son satisfechas (ejemplo: materiales, personal, información, y equipo del sistema de manejo) (Kulkarni et al., 2015).

5.2.1 Planificación sistemática de la distribución (SLP)

Muther desarrolló un procedimiento para la distribución de planta que lo define como: “un modo organizado para dirigir la planeación de la distribución. Éste consiste en un marco de trabajo de fases (Ver Figura 18), un patrón de procedimientos y un conjunto de convenciones para identificar, valorar y visualizar los elementos y áreas involucradas en planear una distribución de planta.”(Muther, 1973, pp. 1–2).

5.2.2 Fase 1. Localización

En esta fase, se debe determinar la localización del área a ser distribuida. Muther menciona que éste no es necesariamente un problema de una nueva ubicación. A menudo este problema es el de determinar si la nueva distribución o re-distribución, será en el mismo lugar en el que está ahora, en un área de almacenamiento que puede desocuparse para tal propósito, en un edificio nuevo adquirido, o en algún otro espacio disponible.

5.2.3 Fase 2. Distribución general global

En esta etapa, es necesario obtener el patrón de flujo de materiales, recursos, entre áreas para las actividades que se vayan a desarrollar, además, es importante especificar la superficie, las relaciones entre las áreas y la configuración de cada actividad.

5.2.4 Fase 3. Planes detallados de la distribución

La fase 3 se encarga de ubicar cada parte específica de maquinaria y equipo. En la planeación detallada, la ubicación actual de cada característica física específica del área a ser distribuido es establecida. Esto incluye los servicios públicos también. El plan detallado de la

distribución es comúnmente una hoja o cuadro con réplicas de las máquinas o equipos ubicado o dibujado en el mismo.

5.2.5 Fase 4. Instalación

Una vez que la fase 3 está completa, los detalles considerables de los dibujos de instalación y la planeación de movimientos deben estar funcionando. Los fondos por la instalación deben ser apropiados y los movimientos actuales para instalar la maquinaria, equipo y servicios deben ser hechos tal como se planearon (Muther, 1973, pp. 1–8). Según Muther, las cuatro fases se deben desarrollar secuencialmente, pero para obtener mejores resultados, estas fases se deben traslapar, tal como se ve en la Figura 18.

Figura 18

Fases de la planeación de instalaciones



Adaptado de: (Muther, 1973). *Systematic Layout Planning (2 ed.)*. Boston, Massachusetts: Cahners Books.

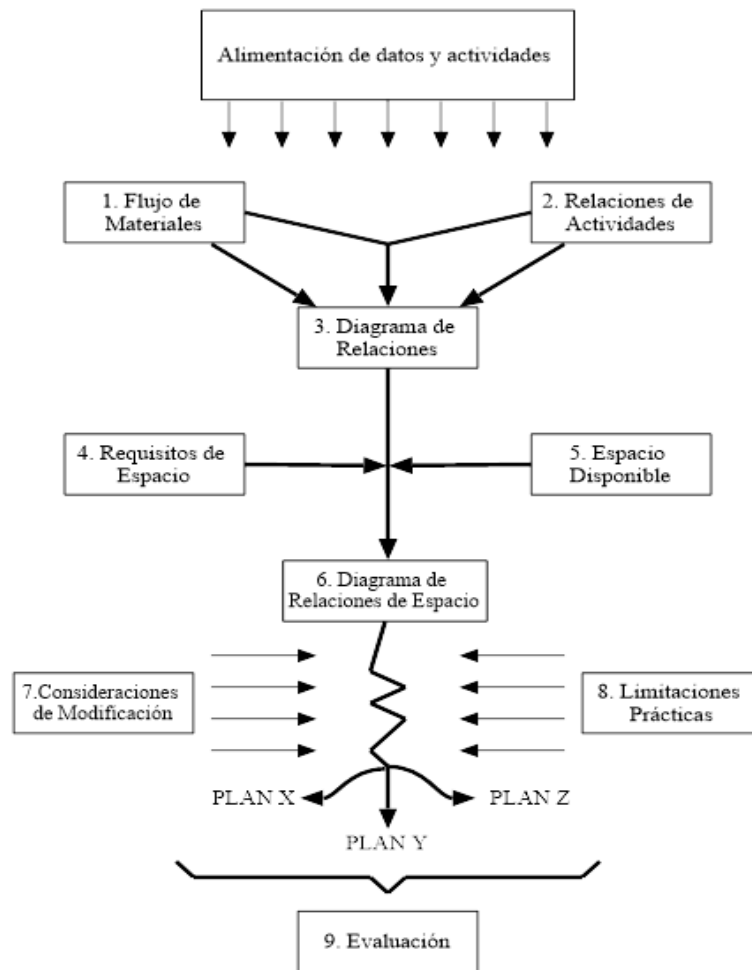
5.2.6 El patrón de procedimientos de SLP.

El patrón de procedimientos propuesto por Muther se visualiza en la Figura 19. Este patrón se alimenta de los datos recolectados de un análisis individual y conjunto de los productos (ejemplo: los bienes que se producen en el área en cuestión, los materiales iniciales, las partes en

proceso, los bienes terminados, etc.) y la cantidad de bienes que se van a producir, proveer o usar en la nueva distribución. Información adicional como el ruteo del proceso (el proceso como tal, su equipo, sus operaciones y su secuencia), los servicios de apoyo de los procesos (tales como: mantenimiento, reparación de máquinas, cuartos de herramientas, baños, primeros auxilios, oficinas, etc.) y los tiempos de operación son también incluidos.

Figura 19

Patrón de Procedimientos de SLP



Adaptado de: (Muther, 1973) *Systematic Layout Planning (2 ed.)*. Boston, Massachusetts: Cahners Books.

Finalmente, es necesario identificar las actividades o áreas incluidas en la distribución. El siguiente paso en el procedimiento es obtener el *Flujo de Materiales*. Al planear la distribución alrededor de la secuencia y la intensidad del movimiento de materiales, se logra un flujo progresivo a través de todas las áreas involucradas. Todos los servicios de soporte deben ser integrados y planeados, como resultado de esto, se desarrolla el diagrama de las *Relaciones de Actividades*, que son las relaciones a través de las actividades o funciones de servicio y soporte.

El diagrama de relaciones de espacio es esencialmente una distribución. Pero, casi siempre, no es una distribución efectiva hasta que es ajustada y manipulada para integrarlas con su espacio y cualquiera de las 7. Las *Consideraciones de Modificación* incluyen consideraciones tan básicas como el método de manejo, las prácticas de operación, la programación de almacenamientos, y similares. Tan pronto como cada consideración o idea potencialmente buena con respecto a estas características es pensada, debe ser probada contra las *Limitaciones Prácticas*, tales como el costo, la seguridad industrial y las preferencias de los empleados.

A medida que la integración y ajuste de las varias consideraciones de modificación y sus limitaciones son desarrolladas, una idea tras otra es probada y examinada. Las ideas que tienen valor práctico son conservadas y aquellas que no pasan la evaluación son eliminadas. Finalmente, después de abandonar aquellos planes que no parecen valiosos, se termina con dos, tres, cuatro o cinco propuestas alternativas de distribución. Cada una de ellas funcionará, cada una tiene valor. El problema yace en decidir cuál de estos planes debe ser escogido. Estos planes alternativos podrían ser temporalmente *Plan X*, *Plan Y*, y *Plan Z*.

En este punto, algún análisis de costo de algún tipo debería ser hecho para propósitos de comparación y justificación. Además, alguna evaluación de factores intangibles debería también ser hecha. Esto es llamado una evaluación de distribuciones alternativas o una *Evaluación de*

Costos e Intangibles. Como resultado de esta evaluación, una de estas alternativas es escogida, sin embargo, frecuentemente una modificación o combinación de dos o más distribuciones pueden resultar del proceso de evaluación (Muther, 1973, pp. 2-1,2-3).

5.3 Metaheurística

Según Talbi (2009), los métodos de búsqueda metaheurística pueden ser definidos como: “unas metodologías o plantillas generales de alto nivel que pueden ser usadas como guías estratégicas en el diseño de heurísticas fundamentales para resolver problemas específicos de optimización.” (2009, p. 1). Una de las características principales de las metaheurísticas, es que permiten atacar problemas de gran tamaño de instancias entregando buenas soluciones en un tiempo razonable, además, no existe ninguna garantía que se encuentren soluciones globales óptimas (Talbi, 2009). Por otra parte, las metaheurísticas crean un balance entre la diversificación e intensificación del espacio de búsqueda, y también pueden ser usadas para resolver eficientemente problemas con gran cantidad de variables y funciones objetivo no lineales (Bass et al., 2012). Según Talbi (2009), muchos criterios de clasificación pueden ser usados para las metaheurísticas:

5.3.1 Metaheurísticas inspiradas versus no inspiradas en la naturaleza

Muchas metaheurísticas son inspiradas por procesos naturales, ejemplo de ellas son los algoritmos evolutivos y los sistemas inmunes artificiales de la biología. Además, los basados en la inteligencia de los enjambres, tales como: las hormigas, colonia de abejas y optimización de enjambre de partículas. También procesos físicos como el recocido simulado.

5.3.2 Metaheurísticas determinísticas versus estocásticas

Una metaheurística determinística soluciona un problema de optimización realizando decisiones determinísticas, por ejemplo: búsqueda local, búsqueda tabú. En las metaheurísticas estocásticas, algunas reglas aleatorias son aplicadas durante la búsqueda, ejemplo de estas son los algoritmos evolutivos y el recocido simulado. En algoritmos deterministas, usar la misma solución inicial guiará a la misma solución final, mientras en las heurísticas estocásticas, se obtienen diferentes soluciones finales desde la misma solución inicial.

5.3.3 Metaheurísticas de búsquedas basadas en una población versus en una única solución

Algoritmos como la búsqueda local y el recocido simulado son algoritmos basados en una única solución, los cuales manipulan durante la búsqueda; mientras algoritmos como el enjambre de partículas y los algoritmos evolutivos, que son algoritmos de búsquedas basadas en población, una población entera evoluciona mientras se desarrolla el algoritmo.

5.4 Marco Tecnológico

5.4.1 Computación en la nube

Una definición de computación en la nube que ofrece el NIST (Instituto Nacional de Estándares y Tecnología) en los Estados Unidos es la siguiente: un modelo para habilitar acceso de red ubicuo (siempre presente) conveniente a pedido a un conjunto de recursos informáticos configurables (como como redes, servidores, almacenamiento y servicios) que se pueden aprovisionar rápidamente pero que, en cambio, se liberan con un esfuerzo de gestión mínimo o interacción con el proveedor de servicios. (Mell Y Grance, 2011, p. 2).

5.4.2 Software

El software es un sistema formal que incluye componentes lógicos necesarios para realizar tareas específicas. No se refiere a componentes físicos. El software está escrito en lenguajes de programación de alto nivel, que son más fáciles y eficaces para los programadores debido a su cercanía al lenguaje natural. Los lenguajes de alto nivel se traducen en lenguaje de máquina con un compilador, un intérprete o una combinación de ambos. No obstante, el software puede estar escrito en lenguaje ensamblador, que es un lenguaje de bajo nivel y se traduce en lenguaje de máquina con un ensamblador, este suele ser más complejo pero eficiente que los lenguajes de alto nivel (Dooley, 2017).

5.4.3 Ingeniería

La ingeniería es el diseño y la construcción de máquinas, estructuras y otros objetos mediante el uso de principios científicos y tecnología de innovación. Los ingenieros utilizan conocimientos y herramientas para desarrollar y mejorar técnicas que satisfagan las necesidades y resuelvan problemas técnicos. Durante el proceso de desarrollo tecnológico, los ingenieros utilizan ciencias básicas como la física, la química, la biología, las matemáticas, las ciencias económicas y administrativas, las ciencias de la ingeniería y la ingeniería aplicada para manejar de manera productiva y eficiente los recursos y las fuerzas de la naturaleza. La ingeniería se considera una actividad que convierte el conocimiento en algo práctico para beneficio de la sociedad (Garrison, 2018).

5.4.4 Ingeniería de Software

Según la definición IEEE 610.12 la ingeniería de software “hace referencia a la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento

de software, es decir la aplicación de la Ingeniería de Software y el estudio de dichos enfoques” (IEEE, 1990). Por tanto, la ingeniería de software es el proceso sistemático y disciplinado de desarrollo, operación y mantenimiento de software. No se trata solo de programación, sino también de definir los requisitos del producto de software y crear diseños para satisfacerlos. El proyecto debe tener un plan y ser entregado en un plazo y presupuesto determinados. Los requisitos deben describir con precisión el problema que se busca resolver y deben ser revisados exhaustivamente para garantizar que claramente se cumplan las necesidades del cliente. Una vez definidos los requisitos y creado el diseño, se escribe el código y se verifica y valida la corrección del software mediante revisiones de pares y pruebas.

6. Metodología

De manera general, en el marco de desarrollo de software se resalta la implementación de metodologías ágiles, estas se caracterizan por la entrega iterativa e incremental de software funcional, de acuerdo con una sinergia entre actores donde resalta principalmente el cliente. A diferencia de métodos como en cascada (Sherrell, 2013), las metodologías ágiles no siguen un plan detallado de principio a fin, sino se centran en la entrega continua de pequeñas porciones de software y en la capacidad de adaptarse rápidamente a los cambios en los requisitos y en el entorno. Las metodologías ágiles incluyen Scrum (Bibik, 2018), Extreme Programming (XP) (Beck, 1999), Crystal (Cockburn, 2004), Lean Development (Janes Y Succi, 2014), entre otras. Si bien en este

proyecto no se cuenta con un cliente ya que el director de proyecto es a la vez actor de diseño de requerimientos, y el equipo de desarrollo es unipersonal, dada la naturaleza del proyecto (investigativo), se opta por implementar principios de desarrollo de software iterativo, lo cual permite aplicar cambios a los requerimientos del producto conforme avanza el desarrollo del proyecto (Pressman Y Troya, 2007)

En esta primera fase se realiza una investigación sobre el problema de distribución de planta, mediante la realización de una revisión bibliométrica utilizando el software Vantage Point (Search Technology, 2018) y una revisión bibliográfica. Para esto se plantean tres etapas con ecuaciones de búsqueda; la primera busca todas las publicaciones realizadas con el tema de interés, la segunda acota el problema a trabajar y, además, limita la ventana de tiempo para seleccionar los artículos a revisar teniendo en cuenta otros criterios; finalmente, la tercera etapa busca demostrar la selección de metaheurística a trabajar en el proyecto. Por otra parte, se realiza una entrevista con el director del proyecto para identificar las necesidades y requisitos de la herramienta a desarrollar, también, se realiza una revisión de las tecnologías utilizadas actualmente para el desarrollo de herramientas web con las características identificadas de la herramienta a desarrollar. Por otro lado, se capacita sobre el uso de la tecnología escogida en la actividad anterior para iniciar con el desarrollo de la herramienta. En esta etapa también se diseña y formula el modelo que se va a implementar en la herramienta software.

Durante el transcurso y para el desarrollo del proyecto propuesto se realizarán las actividades que se muestran a continuación en cada una de las fases, además, se utilizará el modelo de proceso por prototipos evolutivos en las fases 2, 3, y 4, el cual permite aplicar cambios a los requerimientos del producto conforme avanza el desarrollo del proyecto (Pressman y Troya, 2007).

6.1 Fase 1. Revisión de la literatura e identificación de las necesidades de la herramienta

6.1.1 Actividades de la fase 1

A1.1 Revisión de la literatura del problema de distribución de planta junto a los algoritmos metaheurísticos utilizados para darle solución a este problema.

Entregable: Reporte donde se recopile la información investigada y analizada. Además, en este reporte, se identificará el algoritmo metaheurístico a implementar durante el desarrollo del prototipo de la herramienta.

A.1.2 Entrevista con el codirector del proyecto para identificar las necesidades y requisitos de la herramienta a desarrollar.

Entregable: Documento de requisitos.

A.1.3 Revisión de las tecnologías utilizadas actualmente para el desarrollo de herramientas web con las características de la herramienta a desarrollar.

Entregable: Reporte de una revisión de tecnologías junto a la selección de la misma, además de los principales requisitos de la herramienta.

A.1.4 Capacitación sobre el uso de la tecnología escogida en la actividad anterior para el desarrollo de la herramienta.

Entregable: Autor capacitado para el desarrollo de la herramienta.

6.2 Fase 2: Plan y diseño rápido

En esta etapa se busca crear el diseño arquitectural de la herramienta software, además, se plantea realizar la definición de la interfaz de usuario de la herramienta, junto a los casos de uso y diagramas necesarios.

6.2.1 Actividades de la fase 2

A.2.1 Realizar el diseño arquitectural de la herramienta software.

Entregable: Diseño arquitectural de la herramienta.

A.2.2 Realizar la definición de la interfaz de usuario de la herramienta.

Entregable: Diseño de la interfaz de usuario.

6.3 Fase 3: construcción del prototipo

Codificar el prototipo de la herramienta software que se va a desarrollar es la actividad más importante que se ejecuta en esta fase. Además, se implementa el algoritmo metaheurístico seleccionado anteriormente. Esta fase será desarrollada con la tecnología web seleccionada de la fase 1, la cual debe ser de uso libre para poder hacerla extensible a cualquier usuario.

6.3.1 Actividades de la fase 3

A.3.1 Codificar el prototipo de la herramienta software que se va a desarrollar

Entregable: Prototipo funcional

A.3.2 Implementar el algoritmo metaheurístico seleccionado anteriormente.

Entregable: Prototipo funcional

6.4 Fase 4: Validación, implementación y evaluación del prototipo

En esta fase se validan las soluciones del problema de la distribución obtenidas con la herramienta desarrollada mediante un análisis estadístico de los resultados esperados y obtenidos. Por otro lado, la herramienta se implementa en un servidor para poder ser accesible por cualquier persona.

Entregable: Prototipo funcional con al menos un algoritmo metaheurístico implementado

6.4.1 Actividades de la fase 4

A.4.1 Validar las soluciones del problema de la distribución obtenidas con la herramienta desarrollada.

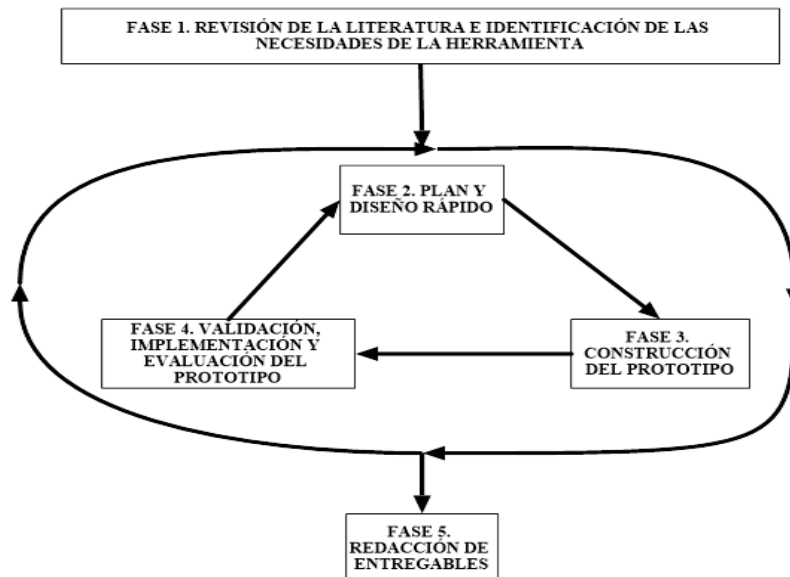
Entregable: Análisis con la comparación de los resultados esperados y obtenidos.

A.4.2 Implementar la herramienta en un servidor.

Entregable: Prototipo de la herramienta implementado.

Figura 20

Fases del proyecto



7. Resultados

7.1 Análisis preliminar de la literatura.

Según Drira, Pierreval, y Hajri-Gabouj (2007) el problema de distribución de planta se refiere a la ubicación de instalaciones en el área de una planta; ésta tiene un gran impacto en los costos de manufactura, trabajo en proceso y tiempos. Este problema ha sido estudiado desde mediados del siglo XIX, pero fueron Beckmann y Koopmans (1957) quienes modelaron por primera vez este problema ubicando plantas en ubicaciones, quienes a su vez modelaron el FLP como un problema de asignación cuadrática (PAC), debido a que en su función objetivo contiene un término de segundo grado, eliminando la linealidad de la función. Este problema también ha sido tratado teniendo en cuenta las áreas de los departamentos o instalaciones como desiguales; Armour y Buffa (1963), fueron los primeros en darle solución a este problema, quienes presentaron una heurística para determinar ubicaciones relativas para las instalaciones físicas, mediante el uso de la simulación. Por otra parte, Chwif, Pereira, y Moscato (1998) fueron pioneros en el desarrollo de un algoritmo de recocido simulado para tratar con el problema de la distribución de instalaciones por bloques, en este se tuvieron en cuenta instalaciones con diferentes áreas y orientaciones obteniendo resultados favorables en un buen tiempo de cómputo.

En la literatura del problema de distribución de planta se encuentran diferentes características, formas de modelar el problema, entre otros; para esta revisión de la literatura se

tiene en cuenta el aporte hecho por Drira *et al.* (2007), quienes en su trabajo de revisión clasificaron la formulación del FLP en seis categorías: *continua, discreta, multi-objetivo, mono-objetivo, datos estocásticos o difusos y problemas simultáneos*. Por otro lado, diferentes objetivos han sido planteados por los investigadores, el más común es *el costo del flujo de materiales*, pero también existen otros objetivos tales como *la relación de cercanía entre las instalaciones, el flujo de materiales*, entre otros. En el trabajo de Kusiak y Heragu (1987) se muestran los diferentes tipos de modelo que se han utilizado para el FLP; algunos de ellos son: *el problema de asignación cuadrática (PAC o QAP: Quadratic Assignment Problem por sus siglas en inglés), la programación lineal entera mixta (PLEM), la teoría de grafos*, entre otros más que se han utilizado a lo largo de los años de investigación. El FLP se considera *np-completo* según Sahni y Gonzalez (1976), a pesar de esto, se han creado algoritmos para encontrar *soluciones óptimas*, pero a medida que la cantidad de instalaciones aumenta, el problema se vuelve inmanejable para los computadores de hoy en día, por esto, se han creado métodos aproximados tales como *las heurísticas, meta-heurísticas, la simulación* y otros más para poder encontrar buenas soluciones en tiempo de cómputo menores. Las características anteriormente mencionadas son estudiadas, revisadas y tabuladas en la presente revisión de literatura, esto con el fin de obtener un conocimiento más profundo en el problema de distribución de planta; es importante mencionar que esta revisión de literatura fue enfocada en la búsqueda de soluciones para el FLP, por eso en todos los trabajos que se mencionan a continuación, la metodología de solución es enunciada o reseñada.

7.1.1 Formulación del FLP

Drira *et al.* (2007) clasificaron el FLP en seis enfoques, a continuación, se realiza una revisión de los artículos publicados con cada uno de estos enfoques.

7.1.1.1 Enfoque mono-objetivo. En este enfoque los investigadores resuelven el FLP utilizando un solo objetivo. Un objetivo comúnmente tratado en la literatura es el de minimizar el costo del manejo de materiales a través de la planta; Xiao, Seo, y Seo (2013), proponen un modelo de programación lineal entera (PLE) con este objetivo, en este, utilizan una heurística de dos pasos para poder encontrar una solución buena al problema y poder comparar su método propuesto con otros que se encuentran en la literatura.

En su modelo. Q. Liu y Meller, (2007), utilizando la aproximación par de secuencia que es utilizada en el diseño de microchips, logran eliminar soluciones no factibles del problema, haciéndolo más manejable computacional y finalmente minimizan el flujo de materiales. Con este mismo objetivo Nourelfath, Nahas y Montreuil (2007) obtuvieron buenos resultados, al formular el FLP como un problema de asignación cuadrática (PAC o *QAP: Quadratic Assignment Problem*) y dándole solución a través de un algoritmo híbrido entre las metaheurísticas optimizaciones por colonia de hormigas y el gran diluvio extendido.

En el mismo año, Chang y Ku (2013) proponen un algoritmo basado en la metaheurística búsqueda armónica y es codificado a través de la estructura de árboles de corte para lograr el mismo objetivo; en este mismo demuestran que su algoritmo es robusto, y obtiene muy buenos resultados en tiempos de cómputo aceptables. Por otro lado, Tasadduq, Imam y Ahmad (2015), proponen un algoritmo para resolver el FLP con módulos rígidos de dimensiones desiguales en espacio abierto. El problema es formulado con un modelo de programación no lineal entera mixta (PNLEM) sin restricciones de los límites del área. Además, proponen un algoritmo que es una combinación de la heurística búsqueda por agrupación en los límites en la fase de construcción y la búsqueda analítica del descenso más empinado en la fase de mejora. Recientemente, Friedrich, Klausnitzer, y Lasch (2018) proponen un estudio cuyo objetivo es minimizar la distancia total de

transporte a través de los perímetros de los departamentos. En este caso, su modelo incluye la ubicación de puntos de entrada y salida para el manejo de materiales de cada departamento, además, utilizaron la metaheurística templado en paralelo para darle solución al problema que abordaron.

7.1.1.2 Enfoque multi-objetivo. Algunos investigadores han intentado dar más realismo al problema de distribución de planta, por esto, consideran más de un objetivo en sus trabajos (Drira et al., 2007). El MOFLP (FLP con múltiples objetivos), busca encontrar una distribución eficiente aplicando objetivos tanto cualitativos como cuantitativos. Singh y Singh (2010a) buscaron en su investigación, minimizar el flujo de trabajo y el tiempo de manejo de materiales, maximizar la tasa de cercanía y los movimientos riesgosos en su trabajo; para darle solución al problema, proponen una heurística de tres etapas, en la primera etapa normalizan la matriz objetivo, luego computan los pesos de cada objetivo y finalmente obtienen una solución.

Años más tarde, Matai, Singh y Mittal (2012), proponen una nueva forma para combinar objetivos ya sean cuantitativos o cualitativos basándose en el procedimiento de normalización de Harmonosky y Totho (1992), en este caso en particular utilizaron como objetivos maximizar la tasa de cercanía y minimizar el costo del manejo de materiales; para poder darle solución a este caso crearon una heurística basada en una búsqueda sistemática en un vecindario de soluciones.

Al año siguiente, los mismos autores; R. Matai, Singh, y Mittal (2013b), propusieron un algoritmo basado en el recocido simulado para el MOFLP; en este. los objetivos que utilizaron fueron la tasa de cercanía entre las instalaciones y el flujo del trabajo entre las instalaciones, en este trabajo también utilizaron el procedimiento de normalización de Harmonosky y Totho (1992) para convertir el MOFLP en un FLP con un solo objetivo. Kapoor y Tak (2015), proponen un enfoque basado en una heurística de tres fases que utiliza la agrupación jerárquica, en este se

tiene como objetivos el flujo de los materiales y la tasa de cercanía entre instalaciones, formando una agrupación de manera aglomerada a partir del par de instalaciones que tiene mayor flujo y la que tiene el requisito de proximidad más grande. Más adelante, en el mismo año, Mohamadi, Ebrahimnejad, y Tavakkoli-Moghaddam (2018) y Turgay (2018), hicieron sus aportes a este problema; el primero tuvo en cuenta como objetivos la tasa de cercanía y el espacio muerto para determinar la ubicación de las instalaciones utilizando bloques rectangulares de área desigual; en este caso, se utilizó una heurística de dos etapas, en la primera se determina el orden de entrada y las ubicaciones de los departamentos, en la segunda se programan tres metaheurísticas y se comparan los resultados entre ellas para ver cuál obtiene una mejor solución. El segundo, toma como objetivos el costo del manejo del material, la función de proximidad, el uso del área y el costo de reconfiguración; en este se implementa un modelo de programación no lineal entera mixta, después se convierte el modelo en uno lineal y finalmente, se le da solución mediante un algoritmo basado en el recocido simulado.

7.1.1.3 Representación continua. En este tipo de representación las instalaciones o departamentos se pueden ubicar en cualquier lugar de la planta y no se deben sobreponer entre ellas (Drira et al., 2007). Khilwani, Shankar y Tiwari (2008) utilizan dos enfoques, uno cualitativo en el cual utilizan un grupo de tomadores de decisión, quienes son expertos sobre el tema, luego a cada uno le asignan su rol específico: un diseñador, un tomador de decisión y un planeador. De estos tres se obtiene la matriz de relación de actividades, la cual es usada en la función objetivo, después proceden a utilizar la técnica de optimización psicoclinal que usa conceptos derivados de la psicología humana y el sistema inmune artificial, para resolver el problema resultante.

Scholz, Petrick y Domschke (2009), realizan un trabajo importante debido a que utilizan la representación de árboles de corte para almacenar la distribución de planta, además, utilizan el

concepto de curvas delimitadas para representar las características geométricas de las instalaciones; con esto se puede tener en cuenta instalaciones con un área conocida, con dimensiones fijas o libres, entre otras; finalmente, utilizan la búsqueda tabú para buscar buenas distribuciones en las que se generan y se guardan en los árboles de corte.

Komarudin y Wong (2010), proponen un algoritmo basado en un sistema de hormigas artificiales para construir nuevas soluciones a través de la representación de árboles de corte, en este pretenden minimizar el flujo de materiales mediante tres fases, la construcción de las soluciones que son las hormigas, la búsqueda local y la actualización de las feromonas.

En el trabajo de Chan y Malmberg (2013), se propone el método de Monte Carlo para darle solución al problema de distribución de planta por bloques de gran tamaño, además, se demuestra gracias a una amplia experimentación que las búsquedas simples aleatorias, pueden generar soluciones de calidad y eficiencia comparable con el recocido simulado. Kulturel-Konak y Konak (2013), proponen un algoritmo híbrido entre un algoritmo genético y la programación lineal; el algoritmo genético establece las variables binarias de decisión del problema, mediante la representación de parejas ubicación/forma, mientras la programación lineal le da solución al FLP. En el mismo período de tiempo, Garcia-Hernandez, Pierreval, Salas-Morera, y Arauzo-Azofra (2013) resuelven el FLP con áreas desiguales utilizando un algoritmo genético interactivo, éste necesita de la intervención de un tomador de decisión (el diseñador de planta), para determinar una solución satisfactoria; en este trabajo el algoritmo propuesto es integrado en una aplicación web, la cual le permite al diseñador tomar decisiones desde cualquier computador con acceso a internet.

Un enfoque diferente es utilizado por Chen, Jiang, Wahab y Long (2015) quienes presentan una solución a un problema de la vida real, el problema de distribución de planta para parques

logísticos, en estas áreas funcionales (instalaciones), tienen formas irregulares tales como un polígono o una curva cualquiera. Para solucionar este modelo, se propone una heurística híbrida entre un algoritmo genético mejorado y una búsqueda dispersa. Más adelante, Latifi, Mohammadi, y Khakzad (2017) proponen un modelo matemático para generar la distribución de planta para una refinería en Irán. Este problema se modela de forma en que cada instalación es un bloque de tres dimensiones, además, se considera el riesgo de derrame, diferentes escenarios de explosión e incendio y el efecto dominó; en este trabajo se implementa la metaheurística basada en la ecolocalización de los murciélagos.

7.1.1.4 Representación discreta. En esta representación, la planta se divide en bloques iguales y en cada uno se asigna una instalación o un departamento. Usualmente, esta formulación se representa mediante el problema de asignación cuadrática QAP (Quadratic Assignment Problem). Ramkumar, Ponnambalam, y Jawaha, realizan dos aportes en el mismo año, en el primero (Ramkumar et al., 2009a), proponen un modelo basado en el PAC, en este buscan minimizar el costo total del transporte de materiales, lo logran utilizando la metaheurística búsqueda local rápida iterativa. En su segundo trabajo, (Ramkumar et al., 2009b), proponen un algoritmo híbrido basado en un sistema de colonia de hormigas y una búsqueda local. para darle solución al FLP discreto modelándolo mediante el PAC. Más adelante, Navidi, Bashiri y Bidgoli (2012) realizan un modelo basado en el PAC con todos los parámetros del costo de manejo de materiales dependiendo juntos, finalmente el problema se modela como un juego estratégico de dos personas, quienes son el diseñador de la distribución de planta y el diseñador del sistema de manejo de materiales; para poderle dar solución a este modelo la metaheurística recocido simulado es implementada.

Atencio y Rodado (2016) al igual que sus predecesores, modelan el FLP discreto como un PAC, y utilizan una combinación entre el método de Sule y un algoritmo genético para darle

solución; además, utilizan una aplicación de la vida real, obtienen la distribución de planta para una fábrica que produce productos para las infraestructuras eléctricas y de telecomunicaciones de edificios.

Al igual que en trabajos pasados, para la formulación discreta también existe el enfoque multiobjetivo. En el trabajo de Singh y Singh (2011), se pretende minimizar el flujo de trabajo y maximizar la tasa de cercanía entre departamentos, el modelo es un problema de asignación cuadrática multiobjetivo (PAC-MO), que mezcla los objetivos cualitativos y cuantitativos mencionados; para darle solución al problema, se propone una heurística basada en el proceso de análisis jerárquico, el cual se utiliza para cuantificar el criterio de todos los objetivos cualitativos en uno solo. Asimismo, R. Matai (2015) propone un modelo basado en el PAC-MO cuyos objetivos son el flujo del trabajo, la tasa de cercanía, tiempo de manejo de materiales y los movimientos peligrosos, en este se propone un algoritmo de recocido simulado modificado que está basado en la heurística de tres etapas de Singh y Singh (2010).

7.1.1.5 Datos estocásticos o difusos y problemas simultáneos. En muchos casos, los investigadores buscan solucionar varios problemas a la vez, un ejemplo es resolver el problema de distribución planta y el problema de asignación de máquinas. Por otro lado, existen ocasiones en que los datos son tratados en forma difusa o de forma estocástica.

Existen numerosos trabajos en el que se incluyen datos estocásticos o modelos manejados con lógica difusa, Sangwan es uno de ellos, en su investigación utiliza una combinación del proceso analítico jerárquico con lógica difusa para asignar diferentes pesos a los factores cualitativos del FLP-MO (Facility Layout Problem Multi Objective, Problema de Distribución de Planta Multi Objetivo), la cual permite dar una aproximación con múltiples criterios al FLP, en

donde los factores cualitativos y cuantitativos para el diseño de una distribución son considerados simultáneamente (Sangwan, 2010).

Por otro lado, Derakhshan, Wong y Tiwari (2015), proponen un algoritmo heurístico que permite resolver el problema de distribución de planta con áreas desiguales estocásticas, donde las formas de los departamentos son fijas durante la iteración del algoritmo y las demandas de los productos son estocásticas con varianza y valor esperado conocido. Asimismo, Azimi y Soofi (2017) proponen un enfoque nuevo a la solución del FLP-MO, el cual se modela como un problema de programación entera no lineal; en este enfoque se considera como objetivos minimizar el costo del manejo de materiales y también la terminación de los tiempos de trabajo (makespan). Se utiliza la simulación computacional integrada junto a una red neuronal artificial para estimar este makespan durante el proceso de búsqueda, el tiempo de manejo de los transportadores es estocástico con una distribución de probabilidad conocida; luego para realizar la búsqueda en el espacio de solución, se desarrolla un algoritmo híbrido basado en algoritmo genético junto a una búsqueda local adaptativa.

En cuanto al tema de resolver dos problemas al mismo tiempo, se puede enunciar a Taghavi y Murat (2011) que encontraron una configuración para un conjunto de máquinas y asignaron productos a estas máquinas, todo esto con el objetivo de minimizar el costo total del movimiento de materiales; para esto se divide el problema en dos subproblemas, el FLP y el modelo de asignación del flujo de los productos, además, mediante un procedimiento heurístico integrado basado en la heurística alternativa de Cooper (1963), un algoritmo de perturbación y una heurística de ubicación secuencial le dan solución al problema. En el caso de Forghani, Khamseh y Mohammadi (2012a), se desarrolla una heurística para abordar la distribución de instalaciones en los departamentos y la distribución de los departamentos en la planta; se formula inicialmente

como un PAC (*QAP: Quadratic Assignment Problem*) modificado quien, debido a su notación representada a través de una malla, reduce considerablemente el número de variables binarias del problema. Zhao y Wallace (2014), trabajan el FLP con máquinas duplicadas en un ambiente estocástico, esto es, el problema de distribución de planta y asignación de flujo integrado, el problema se formula como uno estocástico de dos fases, la primera decide la ubicación de las máquinas y en la segunda, se asignan flujos entre las máquinas para un volumen y trabajo dado; finalmente, la heurística propuesta reduce las dos fases anteriores en un PAC (*QAP: Quadratic Assignment Problem*) estándar y determinístico. En un trabajo reciente, Samanta, Philip y Chakraborty (2018a) abordan el problema de optimización de la distribución de instalaciones bi-objetivo, en el cual el espacio de solución de la segunda función objetivo depende y cambia a través de las posibles soluciones factibles de la primera función objetivo. En este estudio, se utiliza un problema de la vida real, esto es, una fábrica semiautomática de buses; de la cual se busca solucionar dos problemas simultáneamente: la localización de las estaciones de trabajo (estaciones de ensamble de buses) y la ubicación de subestaciones de trabajo. Para esto se utilizan tres métodos para unir estos dos objetivos: una optimización de dos niveles, una optimización de suma de pesos y una optimización bi-objetivo. Para darle solución al problema, se utiliza un algoritmo de enjambre de abejas modificado en el cual se utilizan operadores de búsqueda de adopción genética y un operador de búsqueda de vecindario.

7.1.2 Tipo de modelo

La teoría de grafos ha sido ampliamente utilizada en la optimización combinatoria, es así como Jokar y Sangchooli (2011) proponen modelar el FLP con la teoría de grafos para aplicar el concepto de área de la cara (AC) y obtener una distribución por bloques. AC consiste en que, dado un grafo con 4 vértices iniciales, se añaden iterativamente nuevos vértices teniendo en cuenta las

áreas formadas por 3 vértices. Del mismo modo, Kaveh y Sharafi (2011) proponen la metaheurística búsqueda en sistema cargado (BSS) para el problema de la ubicación de instalaciones sobre una red. Este problema es tratado con una aproximación mini-sum, donde la suma de los costos de servir clientes es minimizada. Para esto modelan el problema como una programación lineal entera y a su vez utilizan la teoría de grafos para poder darle solución con la BSS. Este método utiliza las leyes vigentes de Coulomb y Gauss de la física eléctrica y las leyes de movimiento de la mecánica Newtoniana.

La programación lineal entera mixta ha estado presente al igual que la teoría de grafos desde el inicio de la investigación del FLP. En el trabajo de Li, Chen y Zhao (2013) se obtiene una solución para el FLP generando un modelo de programación lineal entera mixta; en este se analizan las características de múltiples especies y modo de producción de lotes variables que pueden ser definidos como: diversificación de objetos, máquinas, cambios rápidos del mercado, entre otros. Para esto se utiliza un algoritmo genético mejorado, el cual tiene un método óptimo de conservación y una estrategia codiciosa en su operador de selección. Por otra parte, en la investigación de Kang y Chae (2017), la configuración de la distribución por bloques se representa a través de la representación de árboles de corte, con una mejora para diversificar el rango de posibles cortes y se trabaja con un modelo de programación lineal entera mixta. En este trabajo se utiliza el algoritmo búsqueda armónica como algoritmo de solución propuesta, esta es inspirada por el proceso de improvisación usado por los músicos.

El problema de asignación cuadrática (PAC o *QAP: Quadratic Assignment Problem*) fue el primer modelo propuesto para el FLP y en toda la literatura se encuentran trabajos utilizando este modelo. Matai, Singh y Mittal (2013a) proponen una heurística con un intercambio sistemático por pares de dos instalaciones en el vecindario de ubicaciones, además, esta heurística

está basada en una estrategia no codiciosa para resolver el FLP modelado matemáticamente como un PAC. Un año después, Tuzkaya, Gülsün, Tuzkaya, Onut y Bildik (2013) proponen algoritmos para el FLP de la vida real, para lo cual desarrollan un algoritmo genético, un algoritmo híbrido que tiene un enfoque de aplicación secuencial de un algoritmo genético y luego el recocido simulado, y un algoritmo de recocido simulado para obtener distribuciones óptimas de su modelo de PAC (*QAP: Quadratic Assignment Problem*). Por otra parte, se realiza un caso de estudio en una empresa que fabrica ascensores, la cual tiene problemas en su planta, la cual tiene rutas complicadas para sus productos.

La simulación también ha sido aplicada en el FLP, Sikaroudi y Shahanaghi (2016) presentan el algoritmo heurístico de distribución de planta por detección de colisiones (DPDC). Este algoritmo mediante simulación ubica los puntos centrales de los departamentos para luego trabajar en dos fases: en la primera, la solución inicial es construida y la salida de esta fase es mejorada en la segunda; la segunda fase es comprimida en dos tipos diferentes de algoritmos, los algoritmos heurísticos en formato de malla (FM) y sin malla (SFM) respectivamente, tal que la construcción de las distribuciones finales con respecto a la forma de los departamentos es habilitada. Del mismo modo, Banduka, Mladineo y Eric (2017) en su trabajo presentan una solución a un FLP de la vida real con 15 productos diferentes que se producen en 17 estaciones, combinando la heurística método de Schmigalla junto a una simulación con el software visTABLE para modelar y optimizar la solución.

7.1.3 Solución óptima

A pesar de que la tendencia hoy en día es encontrar soluciones aproximadas con buenos tiempos de cómputo, aún se desarrollan trabajos para encontrar la solución óptima. Alizon, Dallery, Feillet y Michelon (2007) utilizan la programación de restricciones, la cual se basa en

restricciones y variables, más que en la función objetivo. Utilizan una aplicación de la vida real, la cual es ubicar puestos de trabajo para una fábrica de autos. En este contexto, tienen que lidiar con el problema logístico de entrega de productos de los almacenes a la planta para poder fabricar los autos, al darle solución al problema obtienen una solución óptima gracias a la heurística que aplican. Otro estudio relevante fue el conducido por Solimanpur y Jafari (2008), estos priorizan y le dan una gran importancia a la linealización del modelo para poder ser aplicado el algoritmo ramificación y corte; Gracias a esto pueden aplicar el algoritmo en tres instancias diferentes, esto es, una de pequeña, otro de mediano y finalmente una de gran tamaño; encontrando la solución óptima para las dos primeras, pero en la tercera no logran encontrar la solución óptima.

Finalmente es importante citar el trabajo realizado por Martinez-Gomez, Nápoles-Rivera, Ponce-Ortega, Serna-González y El-Halwagi (2014a), en el que se propone una nueva metodología para la distribución óptima de planta, donde el costo y el riesgo son minimizados simultáneamente para compensar ambos objetivos opuestos. Este problema es formulado como un problema multiobjetivo de programación lineal entera mixta, luego el método de restricciones es utilizado para poder convertir el problema en uno con un solo objetivo, en este caso, el costo es seleccionado como la función objetivo y el riesgo se impone como una restricción. Este modelo se analiza a través de un caso de estudio para el proceso de destilación del hexano, luego se soluciona en GAMS para obtener el óptimo.

7.1.4 Método de solución.

En el presente proyecto se hace indispensable revisar la literatura en cuanto al método de solución de los diferentes trabajos realizados, por eso se enuncian algunos de los aportes que se han hecho mediante metaheurísticas. El primero que se debe nombrar es el realizado por Fon y Wong (2010), en el cual el FLP es formulado como un PAC (*QAP: Quadratic Assignment*

Problem), y utiliza el Algoritmo Enjambre de Abejas (AEA). El AEA se basa en el comportamiento que tienen las abejas para ir a recolectar comida, esto es, primero envían un grupo de exploradoras a encontrar una flor con néctar, luego las exploradoras vuelven al panal y mediante un movimiento de meneo (baile), les informan a las recolectoras donde se encuentra la comida. El segundo ejemplo, es el trabajo de Hernández Gress, Mora-Vargas, Herrera Del Canto y Díaz-Santillán (2011), quienes utilizan el modelo de Castillo y Westerlund (2005), para resolver el FLP continuo. Además, proponen un algoritmo genético (AG) para obtener las variables binarias de este modelo, con esto logran reducir la complejidad del problema. Más tarde, Bozer y Wang (2012) proponen una heurística, la cual se basa en la representación par de grafos y el recocido simulado para darle solución al FLP. Con su modelo de programación lineal, se obtiene una distribución inicial factible, luego con el recocido simulado realizan una búsqueda para identificar una distribución con un costo menor. El trabajo de García-Hernández, Pérez-Ortiz, Araúzo-Azofra, Salas-Morera y Hervás-Martínez (2014), proponen un sistema que permite incorporar el conocimiento de un experto en la decisión final. Se utiliza un sistema que incluye un AG y redes neuronales artificiales (RNA); el AG es usado para generar un número de soluciones teniendo en cuenta el flujo de material, los requisitos de adyacencia y la relación de aspecto; estas soluciones se obtienen usando un proceso de optimización.

Ficko y Palcic (2013) utilizan la heurística del método del triángulo modificada propuesto por Schmigalla, para representar el espacio de la distribución y las correlaciones (grados de flujos) entre las instalaciones, Con esto ingresan los parámetros iniciales al algoritmo genético, quien calcula el costo de la función objetivo de las soluciones factibles y finalmente, obtienen una solución aproximada. Asimismo, Palomo-Romero, Salas-Morera y García-Hernández (2017), proponen el uso del Algoritmo Genético con el Modelo de Isla (AGMI) para resolver el FLP, el

cual se basa en algoritmos genéticos paralelos para mejorar la diversificación de la búsqueda y demorar la convergencia prematura del algoritmo. Más tarde, Li y Smith (2018) buscan darle solución al problema de las industrias basadas en departamentos de entretenimiento, tales como los parques temáticos, los de diversiones, entre otros. Para lograrlo, una búsqueda tabú es diseñada para buscar el mejor diseño por bloques posible.

Un aporte valioso es el algoritmo híbrido entre las metaheurísticas AG y RS propuesto por Leno, Sankar y Ponnambalam (2018), quienes utilizan una técnica de representación de pares de secuencia para codificar la distribución de planta. Por otro lado, Allahyari y Azab (2018) generan un modelo de programación entera mixta para darle solución al problema. Como método de solución, se utiliza un SA con una búsqueda de múltiple inicio para mejorar la capacidad de exploración y, además, se desarrollan trampas para evitar caer en óptimos locales. Otro aporte es el desarrollado por Liu, Zhang, He y Jiang (2018), quienes utilizan un algoritmo multiobjetivo modificado de enjambre de partículas para resolver el FLP-MO. En este caso, los objetivos que se buscan cumplir son optimizar: el costo total del manejo de materiales, el valor de la adyacencia entre departamentos y la tasa de utilización del piso de trabajo simultáneamente.

Finalmente se consolida toda la información de las características estudiadas (Ver Tabla 4 y Tabla 5) del FLP, esta es presentada en la Tabla 6 por otro lado, la lista de los artículos que se utilizaron en la revisión de literatura se encuentra en el A

Tabla 4

Convención información consolidada revisión de literatura

Objetivos		Formulación	
Nombre	Convención	Nombre	Convención
Flujo de materiales	A	Continua	A
Relación de cercanía de las instalaciones	B	Discreta	B
Distancia recorrida entre instalaciones	C	Mono-Objetivo	C
Costo de manejo de materiales	D	Multi-Objetivo	D
Tiempo de manejo de materiales	E	Estocástico o problemas simultáneos	E
Movimientos riesgosos	F		
Costo de servir al cliente	G		
Costo total anual	H		
Riesgo	I		
Uso del espacio	J		
Costo de reconfiguración	K		

Continuación Tabla 4

Convención información consolidada en la revisión de literatura Parte 1

Método de solución	
Nombre	Convención
Ramificación y Corte	RC
Recocido Simulado	RS
Redes Neuronales Artificiales	RNA
Simulación	SIM
Simulación de Montecarlo	SM
Software	S
SOLVER	SOL
Técnica de Optimización Psicoclonal	TOP
Teoría de Grafos	TG
Recocido Simulado	RS
Redes Neuronales Artificiales	RNA
Simulación	SIM
Simulación de Montecarlo	SM
Software	S
SOLVER	SOL
Técnica de Optimización Psicoclonal	TOP
Teoría de Grafos	TG

Tabla 5*Convención información consolidada revisión de literatura Parte 2*

Método de solución		Tipo de Modelo	
Nombre	Convención	Nombre	Convención
Algoritmo de Enjambre de Abejas	AEB	Problema de Asignación Cuadrática	PAC/QAP
Algoritmo Genético	AG	PAC-Multiobjetivo	PAC-MO
Búsqueda Armónica	BA	Programación Lineal Entera Mixta	PLEM
Búsqueda de Sistema Cargado	BSC	PLEM-Multiobjetivo	PLEM-MO
Búsqueda Local	BL	Programación de Restricciones	PR
Búsqueda Tabú	BT	Programación Entera Mixta	PEM
Heurística	H	Programación Lineal	PL
Metaheurística de Murciélago	MM	Simulación	SIM
Optimización por Colonia de Hormigas	OCH	Teoría de Grafos	TG
Optimización por Enjambre de Partículas	OEP		
Proceso Jerárquico Analítico	PJA		

Tabla 6

Información consolidada revisión de literatura

Referencia	Objetivos											Formulación					Tipo de Modelo	Método de Solución
	A	B	C	D	E	F	G	H	I	J	K	A	B	C	D	E		
(Q. Liu Y Meller, 2007)	√											√		√			PEM	AG
(Nourelfath et al., 2007)	√													√	√		PAC	OCH
(Alizon et al., 2007)	√													√	√		PR	SOL
(Khilwani et al., 2008)		√										√		√			PL	TOP
(Solimanpur Y Jafari, 2008)			√									√		√			PLEM	RC
(Ramkumar et al., 2009a)				√									√				PAC	BL
(Scholz et al., 2009)				√								√					PLEM	BT
(Ramkumar et al., 2009b)	√												√	√			PAC	BL, OCH
(Sangwan, 2010)		√											√	√		√	PLE	PJA
(Singh Y Singh, 2010)	√	√			√	√							√		√		PAC-MO	H
(Komarudin Y Wong, 2010)	√											√		√			PNL	OCH
(Fon Y Wong, 2010)				√									√	√			PAC	H, AEB
(Hernández Gress et al., 2011)	√											√		√			PLEM	AG
(Singh Y Singh, 2011)	√	√											√		√		PAC-MO	PJA
(Kaveh Y Sharafi, 2011)							√					√		√			TDG	BSC
(Taghavi Y Murat, 2011)				√								√			√		PLEM	H
(Jokar Y Sangchooli, 2011)		√										√		√			TDG	H
(Forghani et al., 2012a)				√								√		√		√	PAC y PEM	H
(Navidi et al., 2012)				√									√	√			PAC Y PLEM	RS
(Matai et al., 2012)		√		√									√		√		PAC-MO	H
(Bozer Y Wang, 2012)				√								√		√			PEM	RS, TG
(D. Li et al., 2013)				√								√		√			PLEM	AG

Continuación Tabla 6

Información consolidada de la revisión de literatura

Referencia	Objetivos											Formulación					Tipo de Modelo	Método de Solución
	A	B	C	D	E	F	G	H	I	J	K	A	B	C	D	E		
(Matai et al., 2013b)		√		√									√		√		PAC-MO	RS
(Chan Y Malmborg, 2013)				√								√		√			PLEM	SC
(Matai et al., 2013a)				√									√	√			PAC	H
(Kulturel-Konak Y Konak, 2013)				√								√		√			PLEM	AG
(Xiao et al., 2013)				√								√		√			PEM	SOL
(García-Hernandez et al., 2013)	√	√	√	√	√	√	√					√			√		PL	AG
(Ficko Y Palcic, 2013)				√								√		√			PNL	AG, H
(Tuzkaya et al., 2013)				√									√	√			PAC	AG, RS
(Chang Y Ku, 2013)				√								√		√			PLEM	H, BA
(García-Hernández et al., 2014)				√								√		√			PNL	AG, RNA
(Zhao Y Wallace, 2014)				√									√	√		√	PAC	H
(Martinez-Gomez et al., 2014b)								√	√				√		√		PLEM-MO	SOL
(Kapoor Y Tak, 2015)	√	√											√		√		SIM	H
(Y. Chen et al., 2015)	√											√		√			PLEM	AG, H
(Matai, 2015)	√	√			√	√							√		√		PAC-MO	RS
(Tasadduq et al., 2015)				√								√		√			PLEM	H, S
(Derakhshan Asl et al., 2015)				√								√		√		√	PLEM	AG, OEP
(Sikaroudi Y Shahanaghi, 2016)	√											√		√			SIM	H
(Latifi et al., 2017)								√				√			√		PLEM	MM
(Azimi Y Soofi, 2017)	√				√								√	√		√	PLEM	AG, BL, RNA, SIM

Continuación Tabla 6

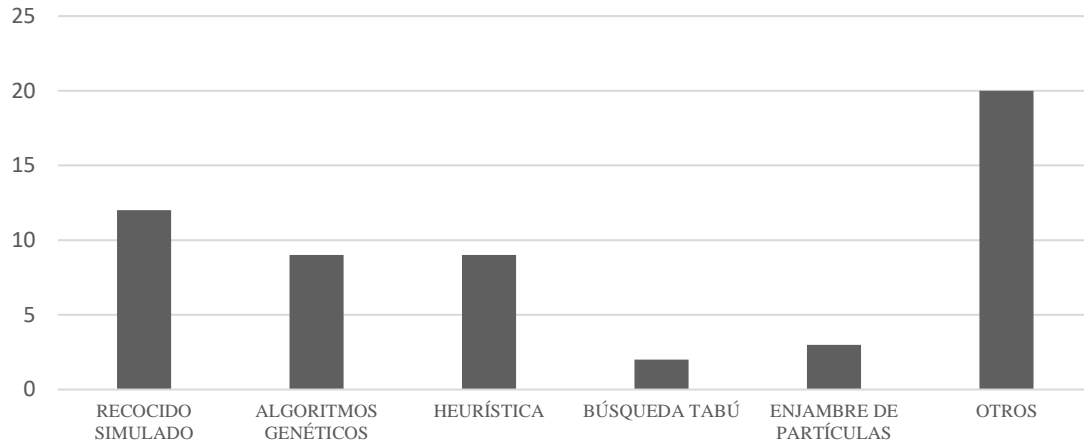
Información consolidada de la revisión de literatura

Referencia	Objetivos											Formulación					Tipo de Modelo	Método de Solución
	A	B	C	D	E	F	G	H	I	J	K	A	B	C	D	E		
(Kang Y Chae, 2017)				√								√		√			PLEM	BA
(Banduka et al., 2017)			√									√		√			SIM	H, S
(Palomo-Romero et al., 2017)				√								√		√			PNL	AG
(Mohamadi et al., 2018)		√								√		√			√		PLEM	AG, H, RS, OEP
(J. Li Y Smith, 2018)		√										√		√			PLEM	BT
(Allahyari Y Azab, 2018)				√								√		√			PLEM	RS
(Leno et al., 2018)				√								√		√			PEM	AG, RS
(2018a)				√								√			√	√	PAC-MO	AEB
(Turgay, 2018)		√		√						√	√	√			√		PLEM	RS
(Friedrich et al., 2018)			√									√		√			PLE	H
(J. Liu et al., 2018)		√		√						√		√		√			PLEM	BL, OEP

Como resultado de la revisión realizada en el capítulo Revisión bibliométrica de literatura subsección 4.3 se determina el uso del algoritmo de recocido simulado para un problema de asignación cuadrática de instalaciones y que considere la integración de múltiples objetivos mediante una estrategia de ponderación de funciones objetivo. Además, en la Figura 21 se representa la cantidad de veces que se utilizan diferentes metaheurísticas en los artículos de la Revisión de literatura realizada, donde se evidencia que el Recocido Simulado es uno de los más usados seguido por los algoritmos genéticos y heurísticas propias desarrolladas por cada autor.

Figura 21

Resumen Metaheurísticas utilizadas en la revisión de literatura



7.2 Documento de requisitos

Con la especificación y documentación de los requisitos de la herramienta, se pretende generar una base para permitir el diseño y desarrollo de la misma. Por otro lado, este documento está dirigido al encargado del diseño de la herramienta, con este documento se pretende validar y justificar las características y funcionalidades que tiene la herramienta.

El documento de especificaciones que se presenta a continuación contiene la descripción, los requisitos de la herramienta, diagramas y cualquier otra información que sea pertinente para comprender y llevar a cabo el desarrollo de la herramienta. Se describe inicialmente cómo están organizados los requisitos funcionales de la herramienta y las funciones del producto, las características del usuario entre otros temas relevantes. Por último, se abarcan los requisitos

funcionales junto a su descripción, así como los requisitos no funcionales que tendrá la herramienta.

La herramienta es la primera versión de una aplicación web enmarcada dentro de las herramientas software didácticas, interactivas y de experimentación para darle solución a problemas de optimización tratados en la investigación científica, específicamente centrado en el problema de distribución de planta estático.

Tabla 7

Ficha para los requisitos

Campo	Información		
Código de requisito			
Nombre de requisito			
Entradas			
Salidas			
Interacción del usuario	Alta	Media	Nula
Prioridad del requisito	Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción			

7.2.1 Funciones del producto

- La herramienta permitirá ingresar las matrices con los datos numéricos de los objetivos a trabajar por parte del usuario.
- La herramienta permitirá crear casos de estudio (simulaciones) y cargar archivos de configuración inicial.
- Permitirá escoger cuáles métodos de cálculo de pesos objetivos utilizar en la solución (opcional).
- La herramienta permitirá al usuario configurar los parámetros de inicio de esta.

- Al terminar de correr la metaheurística, la herramienta entregará la mejor solución escogida entre los métodos de cómputo de pesos objetivos; sin embargo, permitirá visualizar las otras soluciones obtenidas a partir de los métodos ya mencionados.

7.2.2 Características del usuario

La aplicación web a desarrollar está orientada a estudiantes, profesores e investigadores del problema de distribución de planta estático y del uso de metaheurísticas, por esto, es necesario contar con un conocimiento mínimo sobre el problema a desarrollar para que la herramienta sea utilizada al máximo de su capacidad. Para la interacción entre hombre computador, no se necesita ninguna formación, experiencia o capacitación debido a que es una herramienta intuitiva.

7.2.3 Requisitos funcionales

En la Tabla 7 se muestra la representación gráfica para los requisitos funcionales que se utilizará a continuación, donde se muestran las características del requisito y la descripción del mismo.

Tabla 8

Requisito Funcional 1

Campo	Información		
Código de requisito	RF-1		
Nombre de requisito	Selección de objetivo(s)		
Entradas	Opciones: tasa de cercanía, flujo de materiales, tiempo de manejo de materiales, movimientos peligrosos o añadir nuevos objetivos		
Salidas	Lista de opciones seleccionadas		
Interacción del usuario	X Alta	Media	Nula
Prioridad del requisito	X Alta/Eencial	Media/Deseado	Baja/Opcional
Descripción	La herramienta debe permitir al usuario escoger uno o varios de los objetivos que se aplicaran en la solución de la instancia.		

Tabla 9*Requisito Funcional 2*

Campo	Información		
Código de requisito	RF-2		
Nombre de requisito	Entrada de datos de la matriz de cada objetivo		
Entradas	Valores de la matriz de cada objetivo		
Salidas	Matriz de cada objetivo llena		
Interacción del usuario	X Alta	Media	Nula
Prioridad del requisito	X Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	Cuando el usuario escoja el objetivo a ingresar, la herramienta permite digitar los valores de la matriz objetivo de acuerdo con la definición de cada una.		

Tabla 10*Requisito Funcional 3*

Campo	Información		
Código de requisito	RF-3		
Nombre de requisito	Entrada de datos de la matriz de distancias		
Entradas	Valores de la matriz de distancias		
Salidas	Matriz de distancias llena		
Interacción del usuario	X Alta	Media	Nula
Prioridad del requisito	X Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	La herramienta permite al usuario ingresar los valores de la matriz de distancias para la instancia a resolver.		

Tabla 11*Requisito Funcional 4*

Campo	Información		
Código de requisito	RF-4		
Nombre de requisito	Entrada de datos mediante un archivo de configuración		
Entradas	Archivos de configuración, uno para cada matriz.		
Salidas	Herramienta cargada con los parámetros iniciales.		
Interacción del usuario	X Alta	Media	Nula
Prioridad del requisito	X Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	La herramienta debe permitir ingresar los datos de entrada de las matrices y los parámetros de la meta heurísticas, mediante el cargue de un archivo de configuración para cada uno.		

Tabla 12*Requisito Funcional 5*

Campo	Información		
Código de requisito	RF-5		
Nombre de requisito	Selección de metodologías de cálculo de objetivos ponderados		
Entradas	Opciones: Manual, GMWM, SDWM, CRITICM		
Salidas	Lista de opciones seleccionadas		
Interacción del usuario	X Alta	Media	Nula
Prioridad del requisito	X Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	La herramienta debe permitir al usuario escoger al menos una de las cuatro opciones de las metodologías de cálculo de pesos objetivos ponderados.		

Las metodologías de cálculo de objetivos ponderados que se mencionan en la subsección 7.4.10. permiten unificar los objetivos en tantos objetivos ponderados como objetivos se hayan seleccionado inicialmente. La herramienta en cada metodología debe normalizar las matrices de los objetivos y luego utilizar cada uno de los pasos definidos en cada método para encontrar los pesos objetivos ponderados.

Tabla 13*Requisito Funcional 6*

Campo	Información			
Código de requisito	RF-6			
Nombre de requisito	Crear simulaciones			
Entradas	Todos los datos, matrices, parámetros de entrada			
Salidas	Simulación creada			
Interacción del usuario	X	Alta	Media	Nula
Prioridad del requisito	X	Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	El usuario puede crear simulaciones dependiendo de toda la información ingresada como datos de entrada.			

Tabla 14*Requisito Funcional 7*

Campo	Información			
Código de requisito	RF-7			
Nombre de requisito	Normalización de objetivos			
Entradas	Matriz de cada objetivo			
Salidas	Matriz normalizada de cada objetivo			
Interacción del usuario		Alta	Media	X Nula
Prioridad del requisito	X	Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	Internamente y sin que el usuario deba hacer algún cálculo, introducir valores, etc., la herramienta debe normalizar cada matriz de cada objetivo utilizando la metodología de Singh y Singh (2010)			

Para el requisito funcional 7, se utiliza el método de normalización de Singh y Singh (Singh Y Singh, 2010a) el cual comprende dos casos particulares: el primero se presenta cuando la matriz

objetivo es simétrica, esto es, los elementos de la matriz triangular superior son los mismos de la matriz triangular inferior. El segundo caso se da cuando no se tiene una matriz objetivo simétrica.

Tabla 15*Requisito Funcional 8*

Campo	Información			
Código de requisito	RF-8			
Nombre de requisito	Cálculo de cada metodología de cómputo de objetivos			
Entradas	Matriz normalizada de cada uno de los objetivos seleccionados.			
Salidas	Pesos relativos para cada objetivo seleccionado			
Interacción del usuario	Alta	Media	X	Nula
Prioridad del requisito	X Alta/Eencial	Media/Deseado	Baja/Opcional	
Descripción	Con las matrices normalizadas de cada objetivo, la herramienta debe computarlas con la metodología seleccionada y convertirlas en tantos pesos relativos como objetivos se hayan seleccionado.			

Tabla 16*Requisito Funcional 9*

Campo	Información			
Código de requisito	RF-9			
Nombre de requisito	Metaheurística Recocido Simulado (RS) como solución predefinida.			
Entradas	Matriz de distancias, matriz objetivo de cada método de cómputo seleccionado, peso relativo de cada objetivo.			
Salidas	Solución obtenida mediante el recocido simulado			
Interacción del usuario	Alta	Media	X	Nula
Prioridad del requisito	X Alta/Eencial	Media/Deseado	Baja/Opcional	
Descripción	Con los datos anteriores, la herramienta debe solucionar el problema utilizando la metaheurística RS y con los parámetros predefinidos de la misma			

Tabla 17*Requisito Funcional 10*

Campo	Información			
Código de requisito	RF-10			
Nombre de requisito	Entrada de parámetros metaheurística recocido simulado			
Entradas	Datos de los parámetros de arranque del recocido simulado			
Salidas	Metaheurística recocido simulado con los parámetros de arranque definidos por el usuario			
Interacción del usuario	X	Alta	Media	Nula
Prioridad del requisito	X	Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	La herramienta debe permitir al usuario modificar los parámetros de arranque del recocido simulado.			

Tabla 18*Requisito Funcional 11*

Campo	Información			
Código de requisito	RF-11			
Nombre de requisito	Representación de soluciones			
Entradas	Solución obtenida de cada método de cómputo de objetivos			
Salidas	Entrega de resultados al usuario mostrando la configuración de departamentos, instalaciones, etc., junto al valor de la solución objetivo de cada método de cómputo de objetivos			
Interacción del usuario		Alta	Media	X Nula
Prioridad del requisito	X	Alta/Esencial	Media/Deseado	Baja/Opcional
Descripción	Al finalizar, la herramienta debe mostrar la solución con cada método de cómputo.			

Tabla 19*Requisito Funcional 12*

Campo	Información		
Código de requisito	RF-12		
Nombre de requisito	Creación de usuarios y mantenimiento de este		
Entradas	Datos de usuario: correo, contraseña, simulación con sus matrices		
Salidas	Usuario creado con sus datos almacenados en la base de datos		
Interacción del usuario	X Alta	Media	Nula
Prioridad del requisito	X Alta/Eencial	Media/Deseado	Baja/Opcional
Descripción	La herramienta debe permitir crear usuarios a los cuales se les asociará las simulaciones que trabajen y poder revisar en todo tiempo los datos almacenados.		

Tabla 20*Requisito Funcional 13*

Campo	Información		
Código de requisito	RF-13		
Nombre de requisito	Almacenamiento de datos de interés		
Entradas	Información del usuario, simulaciones en las que trabaja el usuario, matrices, parámetros de entrada de la metaheurística		
Salidas	Base de datos alimentada con los datos de los diferentes usuarios que interactúan con la herramienta		
Interacción del usuario	X Alta	Media	Nula
Prioridad del requisito	X Alta/Eencial	Media/Deseado	Baja/Opcional
Descripción	Se debe crear una base de datos donde se almacenen confiablemente todos los datos de interés que los usuarios necesiten para utilizar la herramienta.		

7.2.4 Requisitos no funcionales

7.2.4.1 Desempeño. Algunos de los requisitos de desempeño son: *Primero*, la herramienta no se debe detener de manera automática mientras el usuario la esté utilizando. *Segundo*, la herramienta debe permitir a cada usuario ejecutar el algoritmo el 95% de las veces que intenten correrlo. *Tercero*, la aplicación debe permitir que el usuario modifique los parámetros iniciales, los datos de entrada, los métodos de cómputo a trabajar y los objetivos a utilizar. *Cuarto*, la herramienta debe permitir cargar sin problemas los archivos de configuración de cada caso de estudio.

7.2.4.2 Usabilidad. La herramienta debe permitir la fácil interacción hombre máquina, para lo cual debe implementar reglas y características de usabilidad dando como resultado que los usuarios puedan trabajar de una forma intuitiva, amena y presentando a lo sumo 2 errores de interacción cada 60 minutos.

7.2.4.3 Fiabilidad. Los requisitos son los siguientes: *Primero*, En caso de que exista un error en la ejecución de la herramienta, esta deberá detenerse de forma automática y mostrará un mensaje de error en la pantalla. *Segundo*, la herramienta debe permitir reiniciarse el 90% de las veces después de haber presentado una falla.

7.2.4.4 Extensibilidad. La herramienta debe permitir la solución del problema de optimización para un número indeterminado de objetivos.

7.2.4.5 Disponibilidad. Los requisitos son: *Primero*, la aplicación debe poder utilizarse por al menos 40 usuarios en un tiempo continuo de dos horas o más. *Segundo*, la herramienta debe estar disponible el 100% del tiempo. *Tercero*, la aplicación debe funcionar en cualquier navegador de cualquier sistema operativo para computadora personal

7.2.4.6 Mantenibilidad. El lenguaje en el que se desarrollará la herramienta principalmente es Python 3, además, se utilizará toda la suite de desarrollo Web: HTML5, CSS3 y JavaScript. Por otro lado, se utilizará el framework Django para integrar el código en Python con la parte web de la herramienta.

7.2.4.7 Persistencia. Después de trabajar durante por lo menos una sesión, la herramienta guardará un registro con todos los datos y la actividad realizada. Permitiendo al usuario volver a cargar en una próxima sesión el caso de estudio trabajado con anterioridad, con un plazo máximo de 8 días de almacenamiento en el servidor.

7.2.4.8 Portabilidad. Los requisitos son: Primero, la aplicación debe funcionar en cualquier navegador web. Segundo, la herramienta podrá ser utilizada desde cualquier sistema operativo ya que se accederá a él a través de algún navegador web

7.3 Tecnologías para desarrollo web

Una vez se conoce la lista de requisitos, tanto de la herramienta como del usuario, se procede con la elección de la tecnología de desarrollo para la herramienta web que permitirá solucionar problemas de distribución de planta. La creación de la herramienta web es una tarea esencial y existe una variedad de herramientas y tecnologías con las que se puede realizar el desarrollo, a continuación, se mencionan las principales.

7.3.1 HTML y CSS

Lenguajes de marcado y estilo utilizados para estructurar y dar formato al contenido de un sitio web. El funcionamiento de HTML consiste en un conjunto de códigos cortos, estos códigos son clasificados como archivos de textos en las etiquetas, es decir, se crean archivos llamados HTML y en estos se guardan los textos, estos archivos se pueden encontrar mediante los

buscadores. HTML permitirá describir todo el contenido de un sitio web, ya sea la información estructurada de párrafos, las imágenes, entre otros. CSS sirve para indicar la representación visual de las estructuras HTML. De esta manera, el lenguaje CSS sirve para acotar y trazar el aspecto visual de las etiquetas generadas por el HTML.(Niederst, 2012).

7.3.2 JavaScript

Lenguaje de programación o de secuencia de comandos utilizado para añadir interactividad y funcionalidades dinámicas a un sitio web. Este lenguaje programa el comportamiento de los elementos que contiene la interfaz y de esta forma genera el dinamismo de esta, se pueden crear animaciones, botones, objetos, cookies, mapas interactivos, desplazamientos, reproductor de videos, etc. Una de sus principales virtudes es la funcionalidad construida sobre este lenguaje, las interfaces de programación de aplicaciones (API) son conjuntos de bloques de código listos para usar, de esta forma resulta más fácil la programación.(mdn web docs, 2022)

7.3.3 PHP

Lenguaje de programación utilizado principalmente en el lado del servidor para generar contenido dinámico y procesar formularios y bases de datos. Una de las características principales de PHP es que puede ser usado para la comunicación entre el sitio web y el servidor de datos, lo cual hace posible la creación de contenido dinámico que permite trabajar con bases de datos como MySQL y HTML. Utilizando las herramientas de PHP también es posible recopilar datos de formularios, así como, modificar bases de datos o administrar archivos en el servidor (php.net, 2023).

7.3.4 Ruby on Rails

Marco de desarrollo web basado en Ruby que facilita la creación de aplicaciones web, también es conocido como RoR o Rails, está escrito en el lenguaje de programación Ruby siguiendo el paradigma del patrón Modelo Vista Controlador (MVC). Este lenguaje de programación permite programar escribiendo menos código que los demás haciéndolo más simple y con un mínimo de configuración. Debido a que está escrito con Ruby permite el uso de la meta programación, esto hace que sus usuarios lo encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby (Ruby et al., 2009).

7.3.5 Angular

Marco de desarrollo de código abierto para el desarrollo de aplicaciones web y móviles basadas en el lenguaje de programación JavaScript. Fue creado por Google y es utilizado por muchas empresas y desarrolladores de todo el mundo para crear aplicaciones web modernas y escalables. Esta herramienta se caracteriza por presentar un enfoque en la arquitectura basada en componentes que permiten ser reutilizados, adicionalmente, ofrece una serie de herramientas como son: Motor de plantillas basado en HTML, Biblioteca de componentes visuales y estilos de diseño, Enrutador para crear aplicaciones con navegación, y Herramienta de depuración y pruebas (Farhi, 2017).

7.3.6 Django

Marco de desarrollo de código abierto para el desarrollo de aplicaciones web basadas en el lenguaje de programación Python. Fue creado con el objetivo de ayudar a los desarrolladores a crear aplicaciones web rápidamente y de manera sencilla. Django incluye distintas características

para el desarrollo de aplicaciones web, como son: Sistema de enrutamiento y navegación, Sistema de plantillas para crear la interfaz de usuario de la aplicación, Gestor de base de datos y un lenguaje de consultas para el almacenamiento y recuperación de datos, y Gestor de usuarios y permisos para proteger y restringir el acceso a ciertas funcionalidades de la aplicación (Hourieh, 2008).

7.3.7 Python

Es un lenguaje de programación de alto nivel y propósito general que se utiliza ampliamente en aplicaciones de desarrollo web, análisis de datos, inteligencia artificial y otras áreas. Es un lenguaje fácil de aprender y de leer, con una sintaxis clara y concisa que hace que sea popular entre principiantes y profesionales.

Python también cuenta con una gran cantidad de bibliotecas y marcos de trabajo que facilitan el desarrollo de aplicaciones complejas. Además, es multiplataforma, lo que significa que puede ser utilizado en diferentes sistemas operativos, como Windows, MacOS y Linux.

7.4 Herramienta para la generación de soluciones al problema de distribución de espacios

El presente proyecto consta de los siguientes componentes 1. Herramienta en entorno web para el diseño de planta, 2. Modelo de optimización multiobjetivo *quadratic assignment problem*, 3. Heurística para el cómputo de ponderaciones de la función multiobjetivo, 4. Resolución del problema mediante el algoritmo de recocido simulado, y 5. Gestión de usuarios de la herramienta.

7.4.1 Prototipo de herramienta

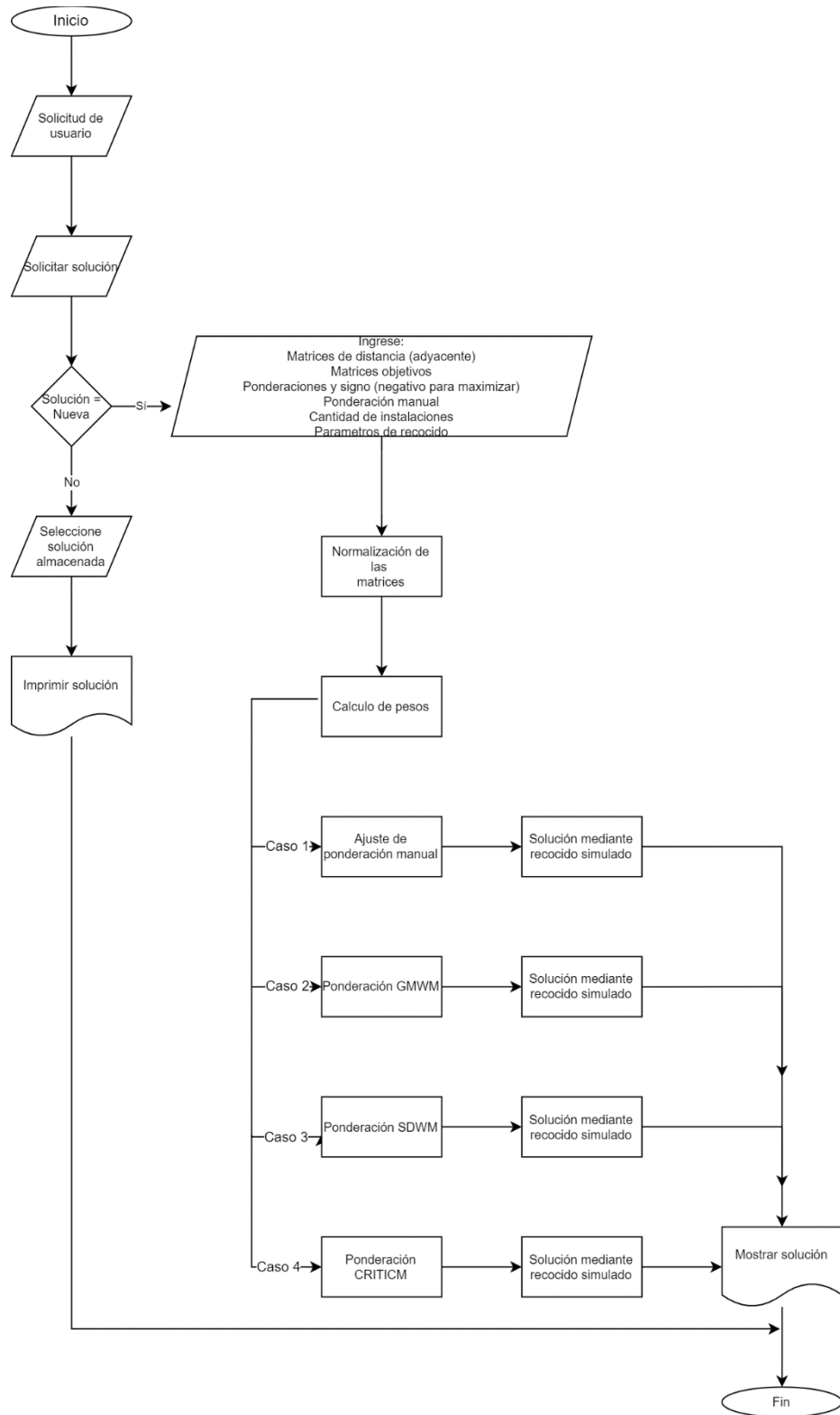
La herramienta en entorno web desarrollada utiliza Python y Django como tecnologías de base. A continuación, se resumen las generalidades de la herramienta la cual se encuentra en el enlace: [FLPWeb](#). Al ingresar a la herramienta, el usuario se encuentra en la pantalla de bienvenida y una barra de navegación en la esquina superior derecha, donde puede iniciar sesión o crear un

nuevo usuario (registrar). Una vez iniciada la sesión, el usuario puede elegir entre o crear una nueva (Nueva Simulación) o cargar una simulación existente (Historial de simulaciones). Si opta por cargar una simulación, se desplaza a la pantalla donde visualiza los datos de la simulación guardada, tiene la opción de modificar los datos de la simulación u obtener nuevamente los pesos objetivos, para luego volver a calcular los datos de la simulación. Caso contrario, si el usuario decide crear una nueva simulación, se desplaza a una nueva pantalla, que es el inicio del proceso de simulación. Esta pantalla incluye la carga de la información y el preprocesamiento de los datos. Se deben cargar las matrices relacionadas con las características del problema de diseño de instalaciones, definir los parámetros del algoritmo de recocido simulado y especificar el tipo y peso de las matrices de las funciones objetivo.

Una vez finalizado el preprocesamiento, se visualizan los resultados y se cargan las matrices desde archivos en formato csv. Luego, en una nueva ventana se realiza una validación de los datos previo a la implementación del algoritmo de recocido simulado para resolver el Modelo de optimización multiobjetivo QAP (*QAP: Quadratic Assignment Problem, Problema de Asignación Cuadrática*) y se muestran los resultados finales. En resumen, la herramienta permite cargar y procesar datos, aplicar el algoritmo de recocido simulado y visualizar los resultados obtenidos, este proceso de flujo de tareas relacionadas con el modelo de optimización se registran en la Figura 22. Finalmente, las interfaces de usuario de la herramienta se visualizan en la subsección 7.5.3 *Desarrollo del prototipo 3: Frontend, conexión con servidor.*

Figura 22

Diagrama de flujo sobre las generalidades de la herramienta



Para el diseño de la herramienta se tuvieron presente los requerimientos funcionales y no funcionales descritos en la subsección 7.2 Documento de requisitos, adicionalmente, en la Tabla 21 se describen las herramientas utilizadas tanto para el desarrollo del FLPWeb como para el análisis de resultados. Adicionalmente, durante la etapa de diseño se determinan los diferentes Diagrama de casos de uso, Diagrama de estado, Diagramas de actividades, y Diagramas de clases.

Tabla 21

Herramientas de desarrollo utilizadas

Herramienta	Artefacto
Python 3.8.10	Programación.
Visual studio code	Editor de código fuente.
Django 4.1.5	Framework web
PostgreSQL	Gestor de Base de Datos
Draw.io	Diagramas de casos de uso.
Microsoft Word	Documentación del proyecto
Microsoft Excel	Reportes y validación manual de matrices y operaciones
Minitab 19	Análisis y modelamiento estadístico
Scribeshow	Crear guías de usuario

Para el desarrollo del prototipo de la herramienta FLPWeb, se utilizó el lenguaje de programación Python en su versión 3.8.10, esta decisión está basada en la facilidad y practicidad que el lenguaje ofrece al manipular matrices y vectores a través de su librería Numpy, la cual se trabajó en su versión 1.24.1. Esta librería se utilizó en toda la lógica de la aplicación aprovechando la cantidad de funciones, métodos y recursos, ya desarrollados en la misma para hacer más sencilla la programación de la algoritmia que se incluye en el prototipo. Además, se utiliza el framework de alto nivel Django en su versión 4.1.5, el cual es de código abierto, manejable y reutilizable entre otros.

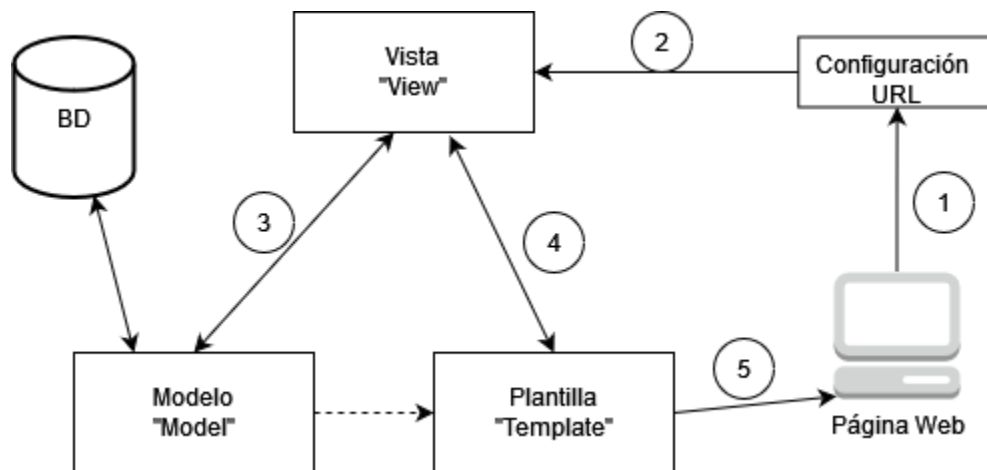
7.4.2 Arquitectura Django

El Framework que se utilizó es Django el cual trabaja con el Modelo Vista Plantilla (Model View Template por sus siglas en inglés), que permite separar tres lógicas importantes en una aplicación web; la primera es la lógica del almacenamiento de información en una base de datos la cual es manejada por la parte del Modelo, la segunda es la lógica de negocio/aplicación, en el caso de FLPWeb, la lógica que lleva para calcular y generar soluciones al problema de distribución de planta que es manejada por la Vista.

Finalmente, la Plantilla maneja todo lo que corresponde a la presentación al usuario, esto es, todo lo que el usuario podría visualizar e interactuar en su navegador web. En la Figura 23 se puede visualizar el esquema del MVT, se puede ver que el flujo inicia con el envío de una solicitud de la página web, segundo el Conf URL interpreta la solicitud y ubica la vista apropiada, tercero la vista interactúa con el modelo para obtener datos, cuarto la vista llama la plantilla y por último da respuesta a la página de la solicitud enviada inicialmente

Figura 23

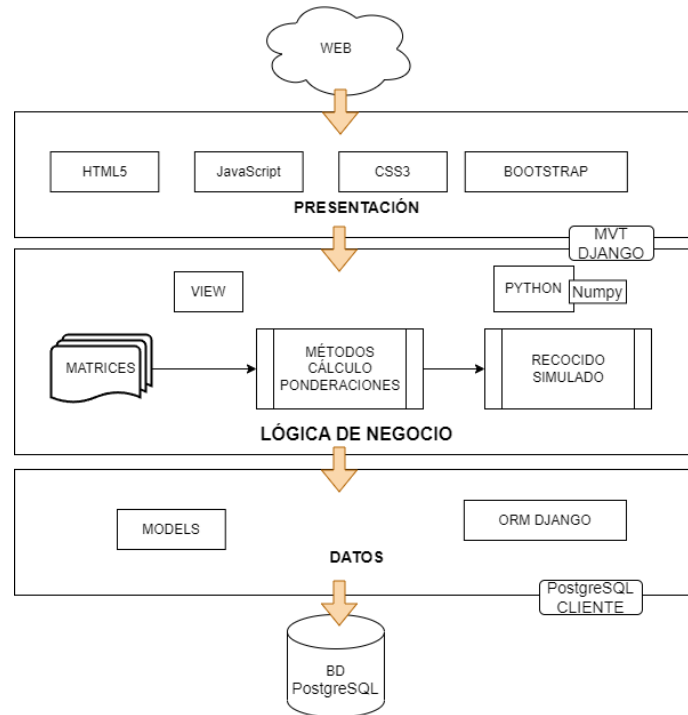
Diagrama MVT Django



7.4.3 Arquitectura de la herramienta

Figura 24

Arquitectura herramienta FLPWeb



En la Figura 24 se puede visualizar la arquitectura de la herramienta, la cual es una adaptación de la arquitectura del Framework Django. Inicialmente el usuario accede a través de un navegador web a la herramienta, en la capa de presentación se tiene las tecnologías que permiten la interacción de la herramienta con el usuario, a través del MVT de Django la herramienta internamente recibe los datos de las matrices y la simulación para ejecutar la lógica de negocio de esta aplicación, esta consiste en normalizar las matrices, calcular las ponderaciones de los objetivos, correr la metaheurística recocido simulado y poder mostrar los resultados. Finalmente, la capa de datos que permite interactuar con la Base de Datos a través del Object Relational Mapper, que permite mapear los objetos de la herramienta directamente en la base de datos.

7.4.4 Diseño de datos

Las tablas principales de la herramienta hacen referencia a la Simulación donde se almacenan todos los datos del problema que se va a resolver; Solución en la que se encuentran los datos de la mejor solución objetivo junto al vector solución, Matriz aquí se almacenan los datos propios de una matriz, tanto si es objetivo o distancia y Método de Pesos Objetivos en la que se tienen los datos para los cuatro métodos de cálculos de pesos objetivos que se utilizan en la heurística. Finalmente, como se ve en la Figura 25 también se añadió una tabla de Usuario para poder guardar la información de este junto a las simulaciones que pueda ejecutar.

7.4.5 Diagrama de casos de uso

En la Figura 26 se relacionan los casos de uso que se han especificado para la herramienta, estos son basados en la interacción que el usuario tendría al utilizar y ejecutar la herramienta desarrollada. Se considera un único usuario altamente especializado con conocimiento avanzado en programación matemática discreta y no lineal con énfasis en diseño de plantas, este usuario podrá resolver el problema de asignación cuadrática multiobjetivo aprovechando el potencial de la herramienta, de lo contrario, un perfil no especializado requeriría de una capacitación para darle sentido y buen uso a la herramienta.

Figura 25

Diagrama de Base de Datos de FLPWeb

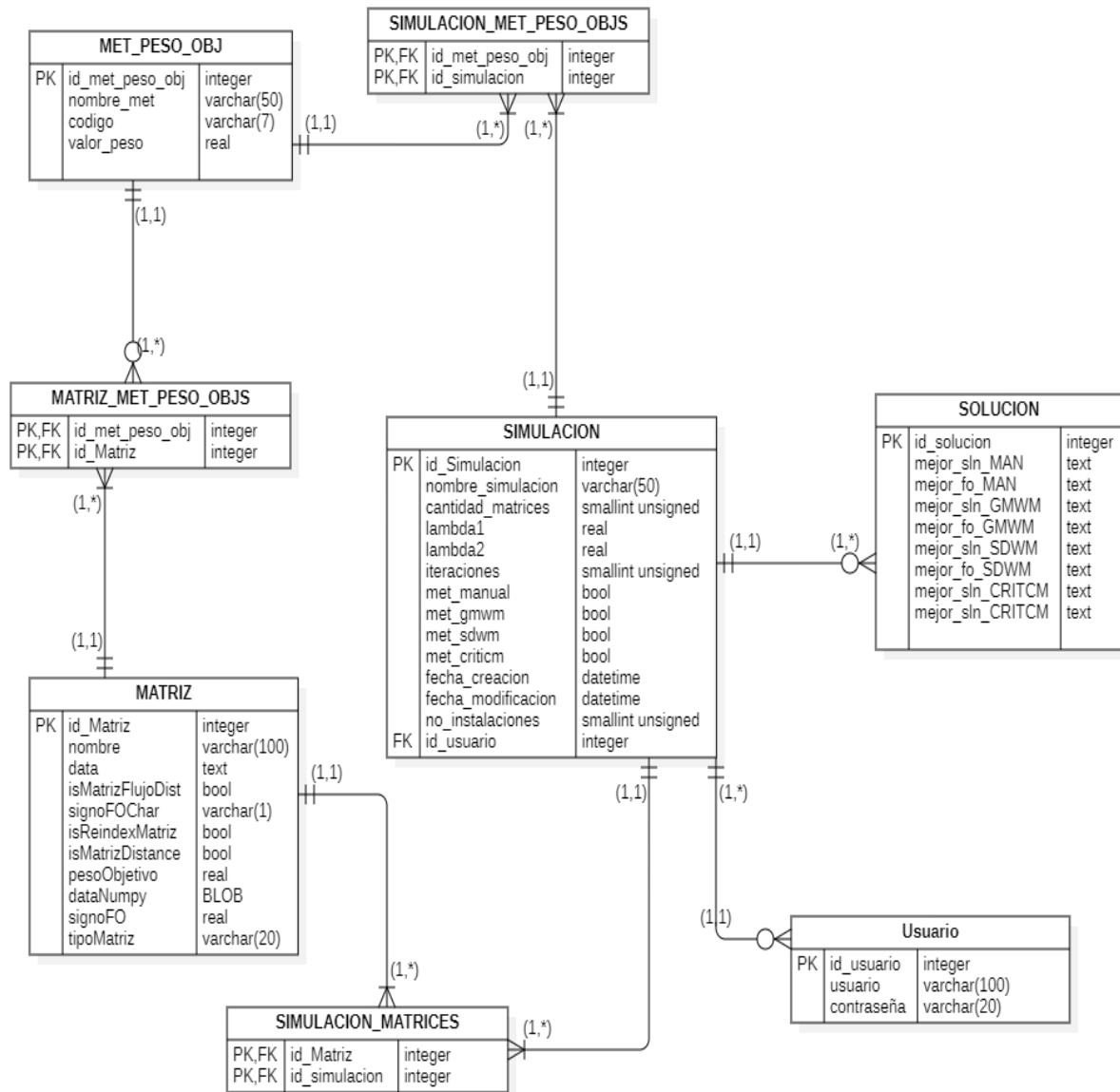


Figura 26

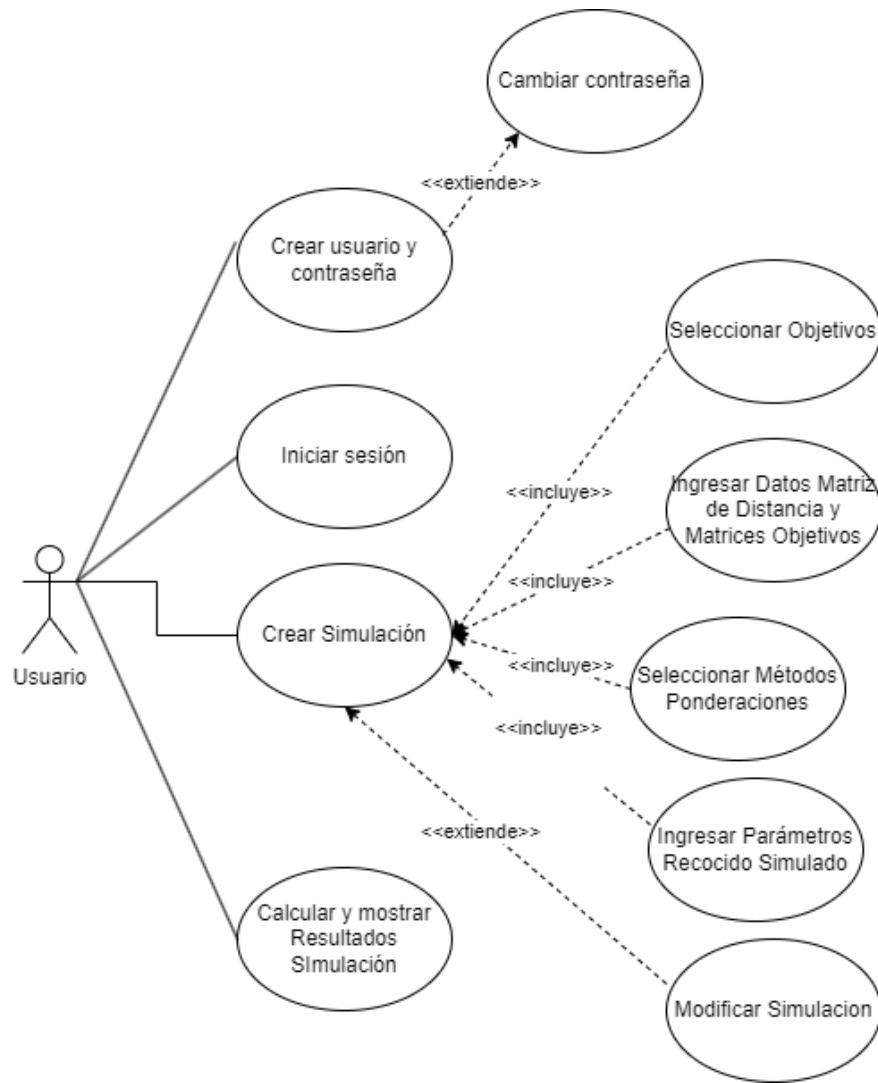


Diagrama de casos de uso

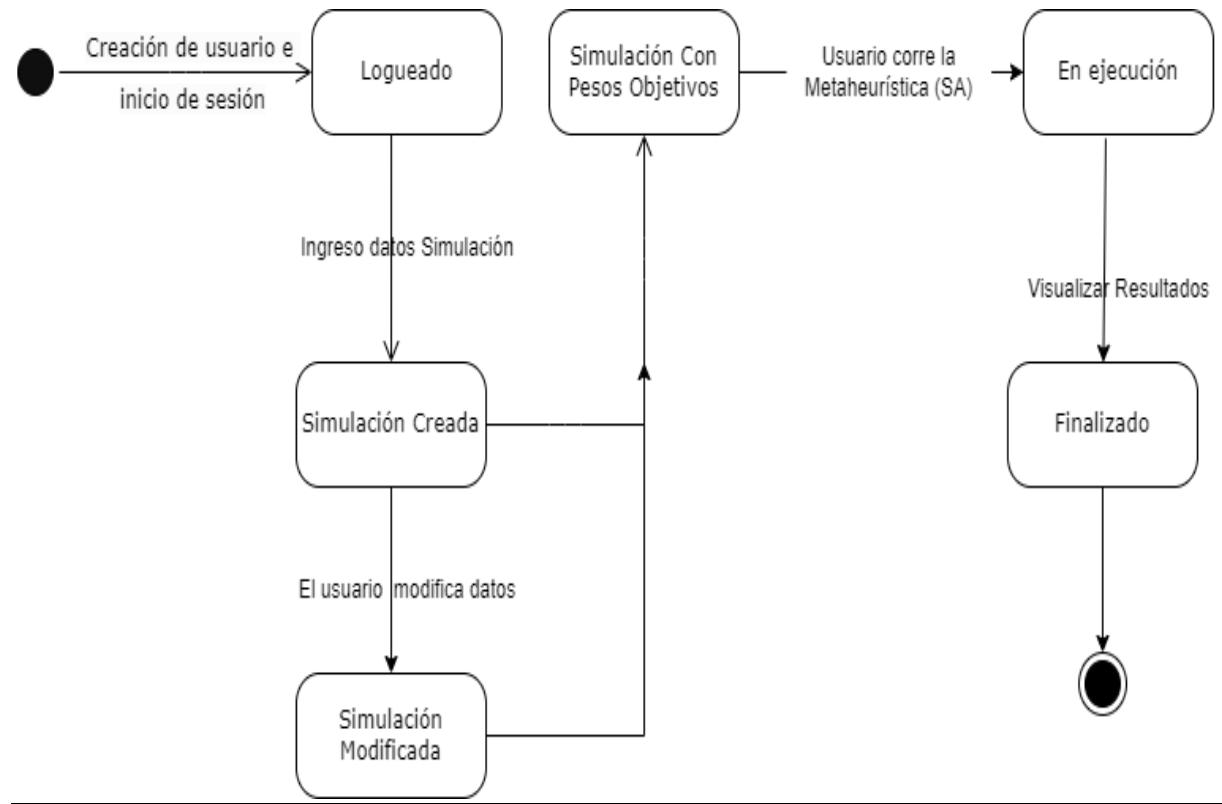
7.4.6 Diagrama de estado

En la Figura 27 es posible visualizar los estados que puede tener la herramienta, nótese que el usuario es el único que interactúa con la herramienta, permitiéndole tener el control de todos los parámetros tanto del problema de optimización discreta no lineal, como de la configuración del

algoritmo de recocido simulado. En este sentido y debido al perfil especializado del mismo, el usuario puede tener control total para realizar las modificaciones que quiera sobre los datos ingresados en el FLPWeb. Debido a la codificación en inglés el algoritmo metaheurístico aparece con las siglas SA.

Figura 27

Diagrama de estados



7.4.7 Diagramas de actividades

En este apartado se presentan los diagramas de actividades los casos de uso descritos anteriormente para el prototipo de la herramienta FLPWeb. De manera general, hay tres grandes

grupos de actividades. El primer grupo de se relaciona con las actividades de ingreso o registro de los usuarios a la herramienta (Figura 28), actividades enfocadas en el modelamiento del problema de asignación de instalaciones a ubicaciones mediante el enfoque de asignación cuadrática, así como la configuración del algoritmo de solución enfocado en un recocido simulado modificado (Figura 29), y el ingreso del usuario para la consulta de soluciones previamente creadas (Figura 30).

Figura 28

Diagrama de Actividades Crear Usuario - Iniciar Sesión

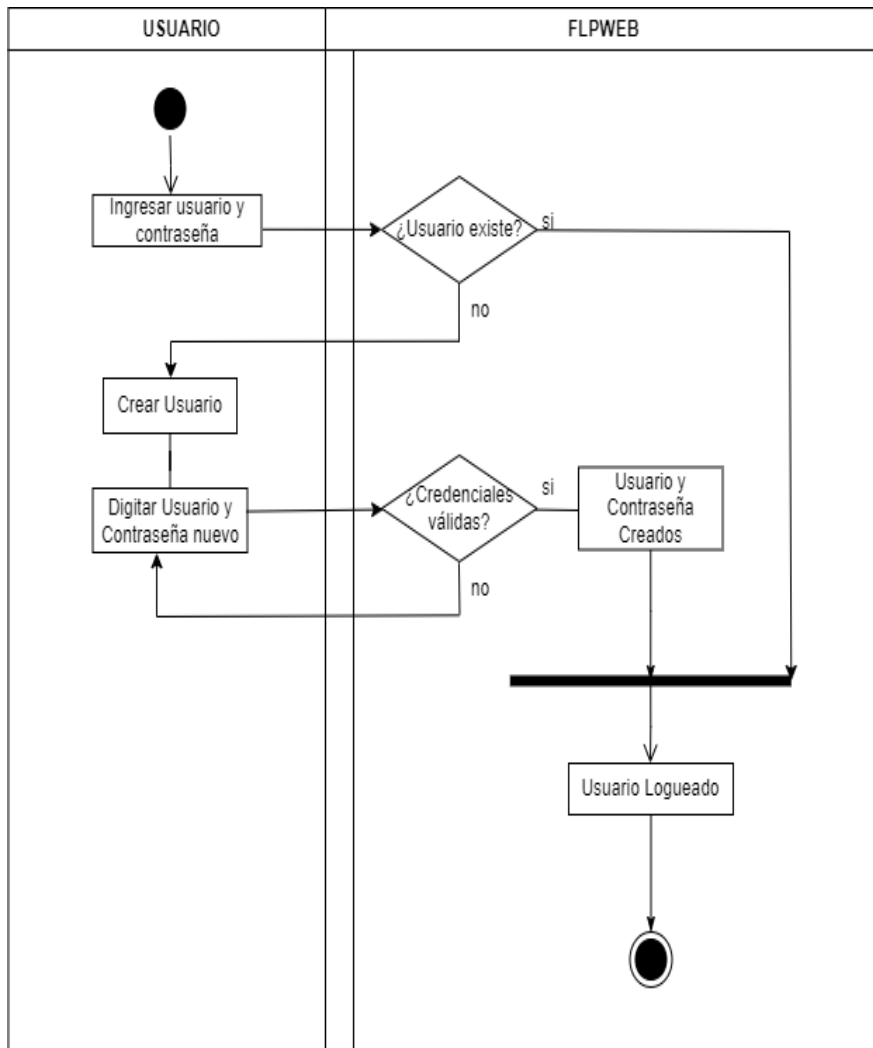


Figura 29

Diagrama de Actividades Simulación

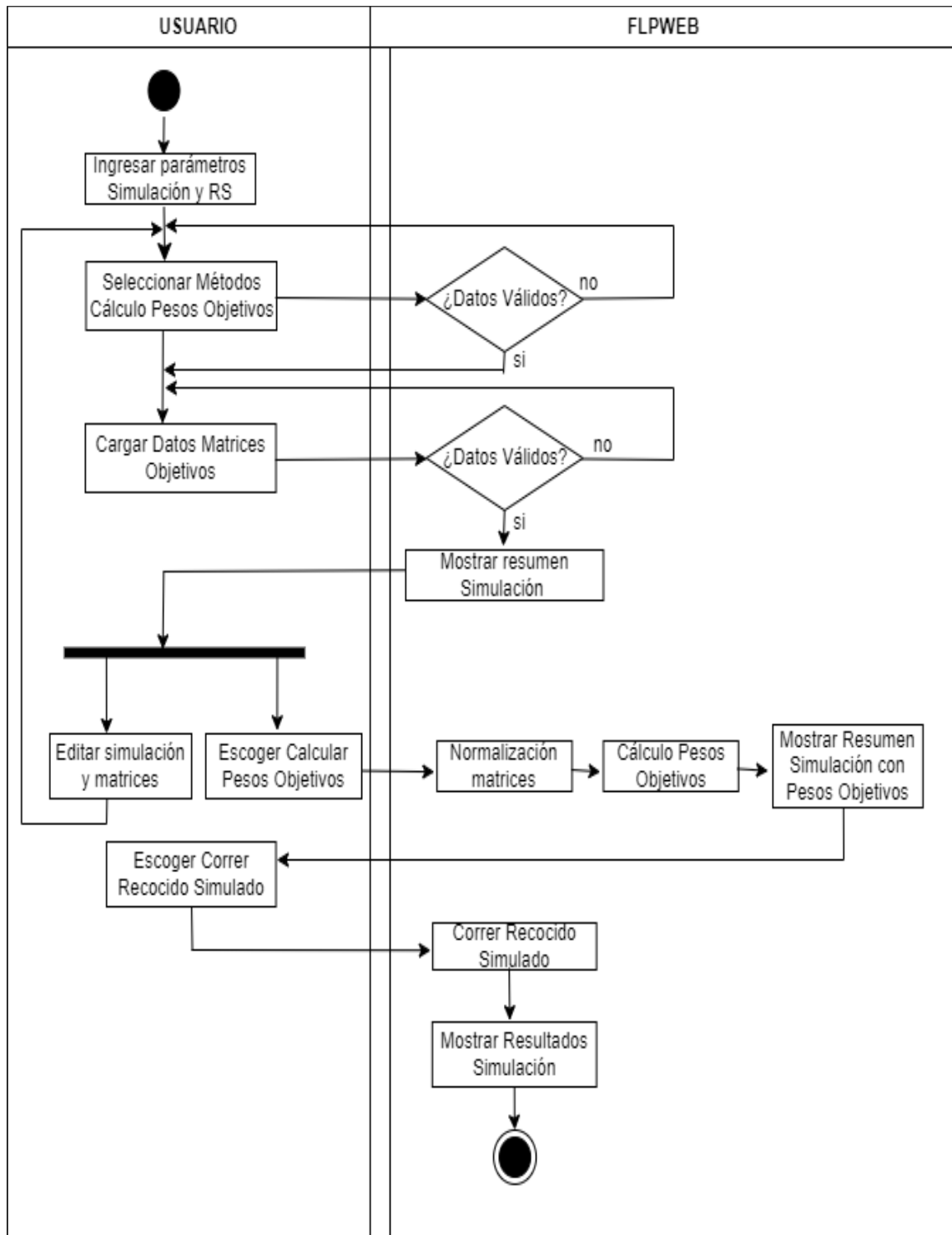
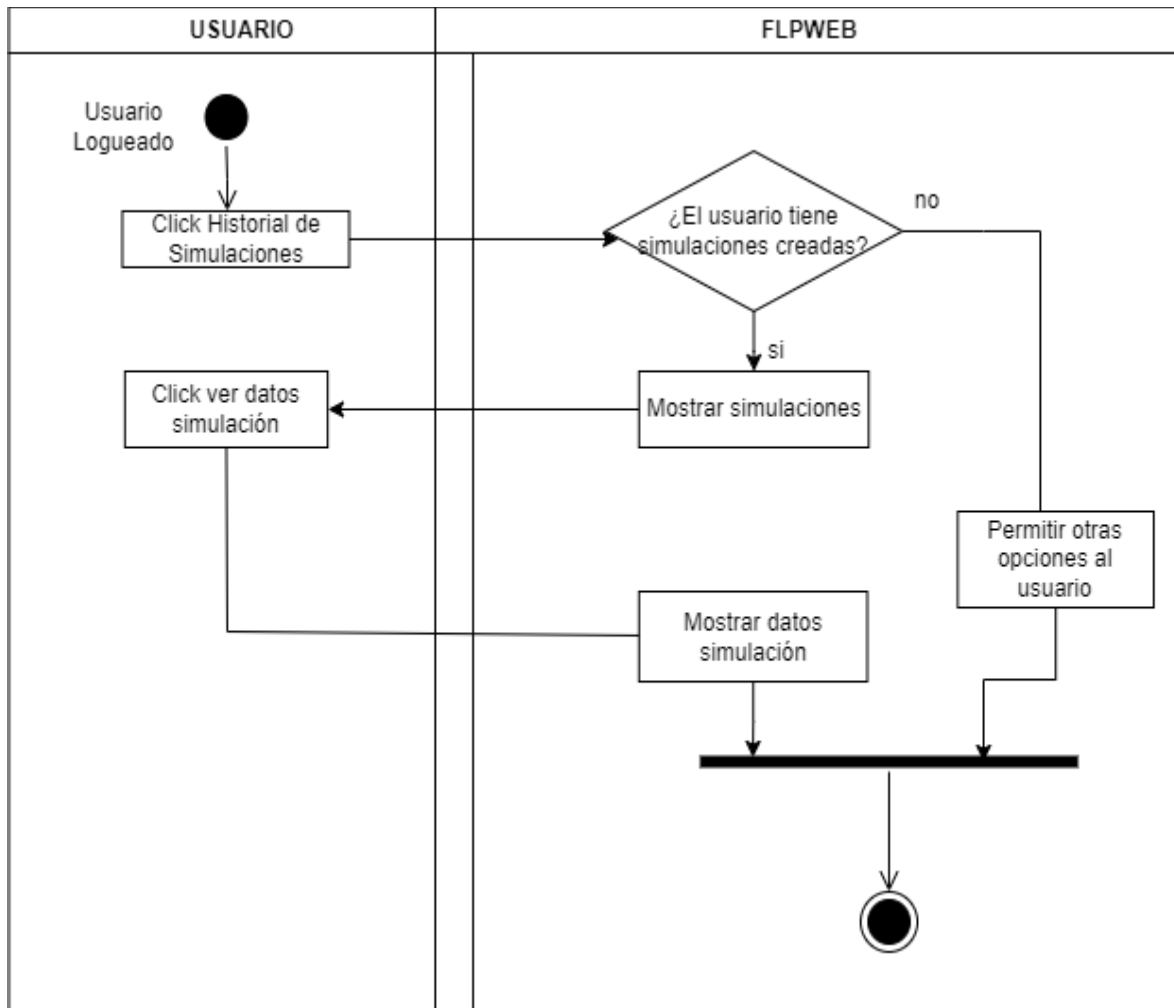


Figura 30

Diagrama de Actividades Consulta de soluciones

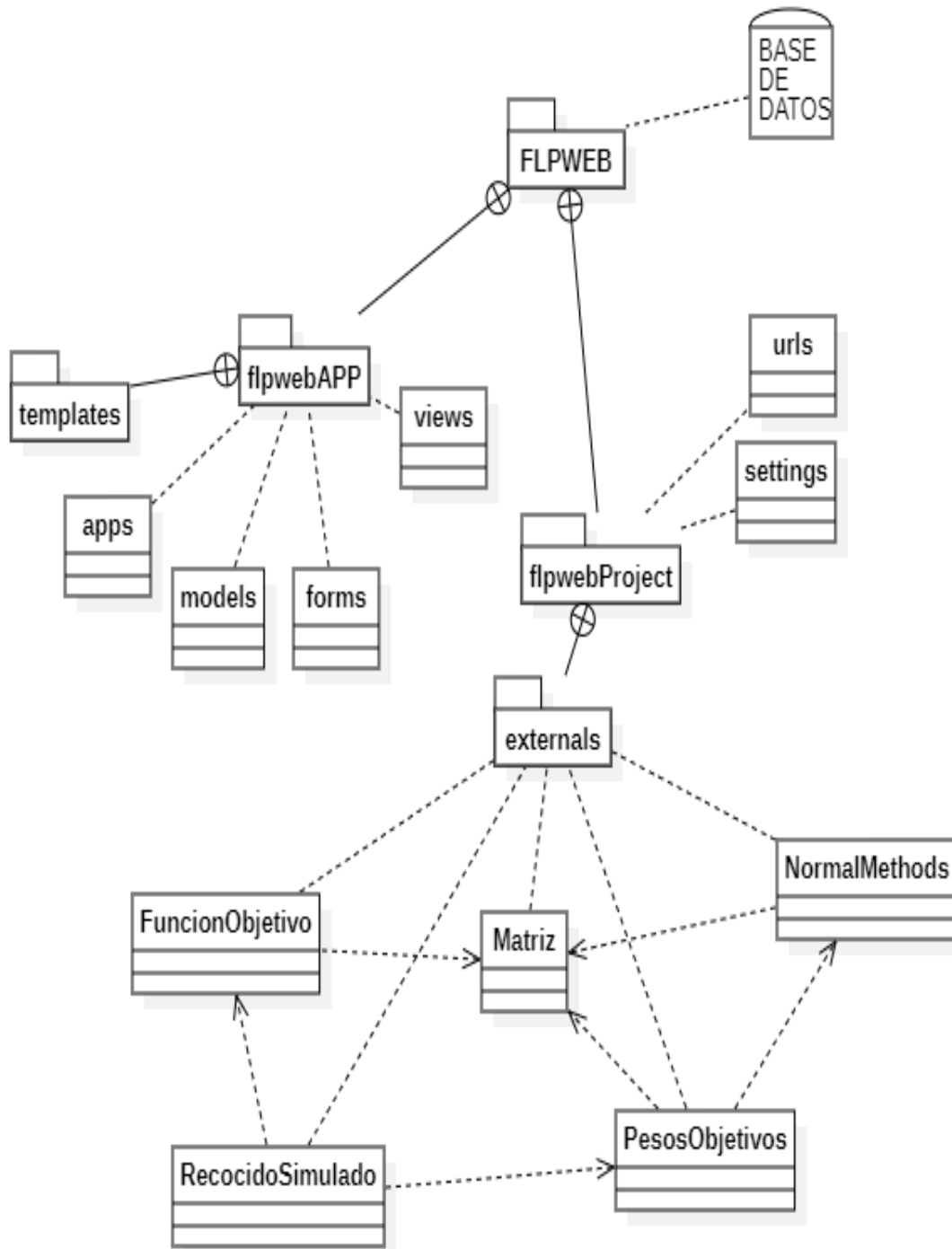


7.4.8 Diagramas de clases

Para representar el diseño de la herramienta web mediante diagramas de clases, se trabajó desde lo general hasta lo más específico, por esta razón, se realizaron tres diagramas: Diagrama de clases general donde se muestran los paquetes y algunas dependencias entra las clases que se tienen a nivel macro, luego están los Diagramas de clases del paquete external y el Diagrama de clases del paquete *flpwebAPP*.

Figura 31

Diagrama de clases General de FLPWeb

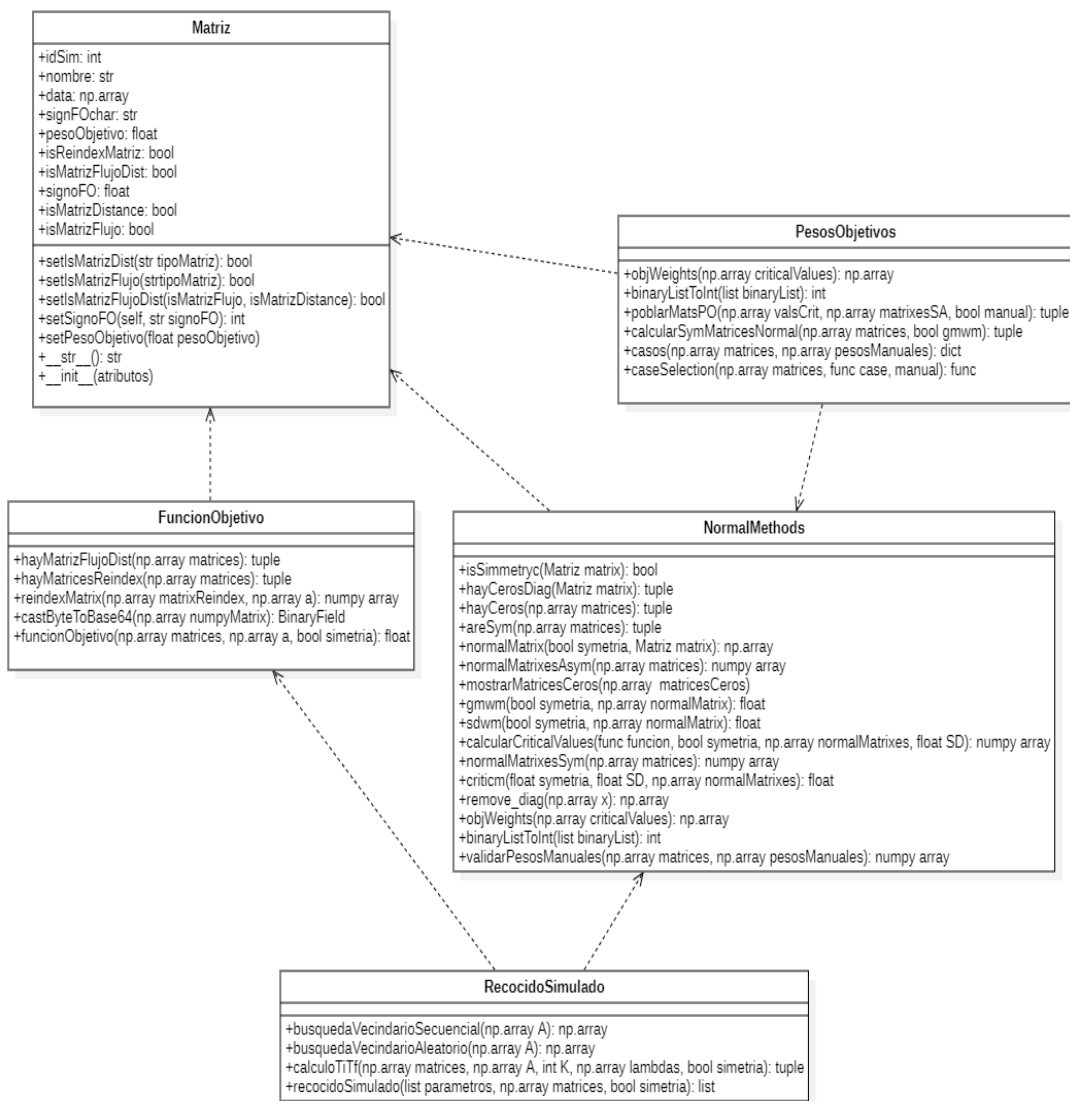


En la Figura 31 se puede observar el paquete principal donde está contenida toda la herramienta web, el cual es FLPWEB, allí están los paquetes flpwebAPP y flpwebProject, ambos

paquetes creados por el framework Django, donde se puede separar la dinámica del backend (algoritmia incluida en el paquete externals), de la lógica web que está en el paquete flpwebAPP. Además, en el paquete principal se encuentra la Base de Datos de la herramienta la cual como ya se mencionó se utilizó PostgreSQL.

Figura 32

Diagrama de clases Paquete externals

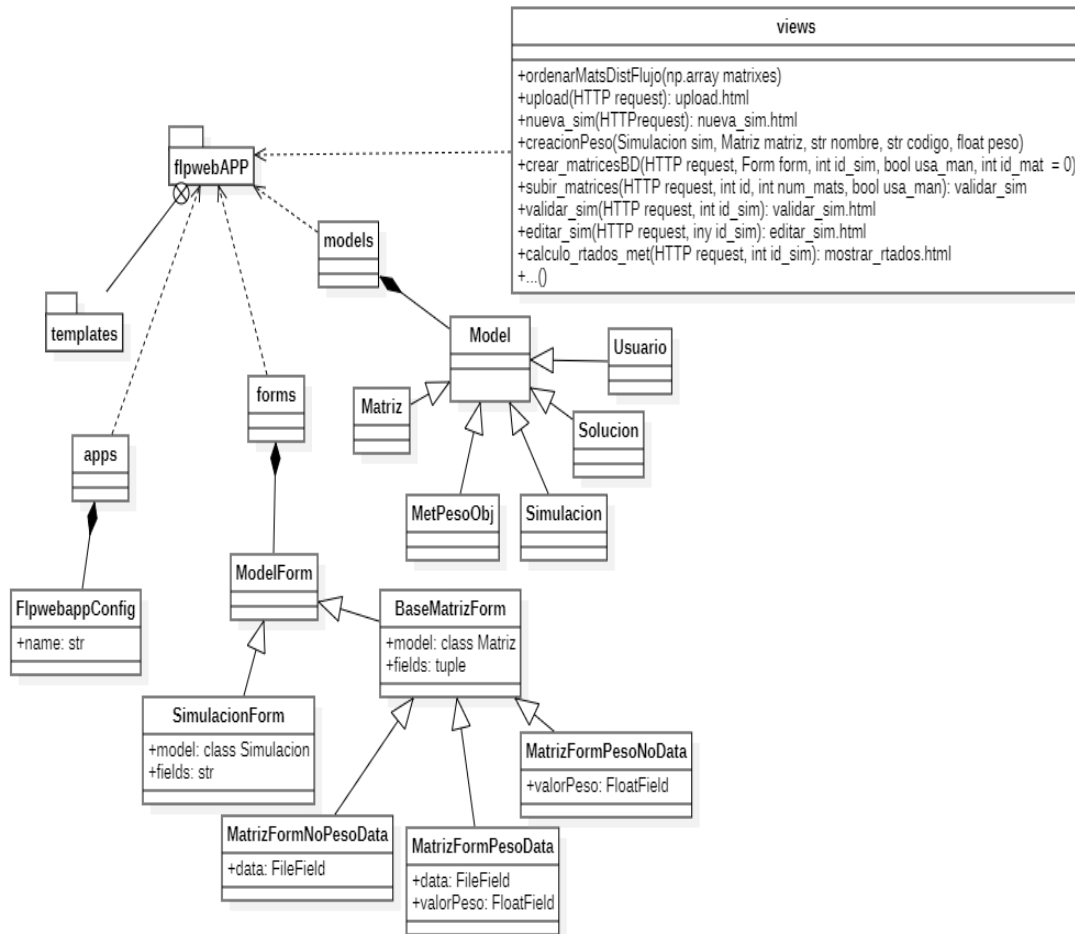


Toda la algoritmia en la cual está sustentado el desarrollo de este prototipo web, se encuentra en el paquete externals (Ver Figura 32), en este se encuentra la clase Matriz que contiene todos los métodos y atributos para poder modelar el problema, además, están las clases PesosObjetivos, NormalMethods, FuncionObjetivo y RecocidoSimulado, que son las que se encargan de generar todos los cálculos que se tomaron de la investigación. Esta información se puede encontrar en la subsección 7.4.10.

Para finalizar el apartado de los diagramas de clases, se presenta el *Diagrama de clases del paquete flpwebAPP* (Ver Figura 33), Dentro de este paquete se encuentran todos los archivos de las páginas web que se utilizan en la herramienta (carpeta templates), también se encuentra la clase views que es la que permite la comunicación entre el backend y las solicitudes que realice el usuario a través del navegador web, esto lo realiza utilizando funciones y conectando con las urls que están definidos en el paquete flpwebProject, además, en esta clase se utilizan todos los métodos o funciones necesarios para poder obtener y mostrar la información necesaria para el usuario. También se encuentra la clase models que contiene a la superclase Model para poder heredar el comportamiento de un modelo, en Django un modelo es la representación de una tabla de la Base de Datos, por esta razón, no se especificaron los atributos de estas tablas, ya que se encuentran en el Diagrama de Base de Datos de la Figura 25. En la clase form están contenidos los formularios que se utilizaron, los cuales heredan de la clase ModelForm de Django.

Figura 33

Diagrama de clases del paquete flpwebAPP



7.4.9 Modelo de optimización multiobjetivo para resolver un problema de distribución de planta

El problema de asignación cuadrática (*quadratic assignment problem, QAP*) es un problema de optimización combinatoria que implica encontrar la asignación óptima de instalaciones a ubicaciones de manera que se minimice el costo total de la asignación (Forghani et al., 2012; Ramkumar et al., 2009a; Samanta et al., 2018a). En el QAP, hay un conjunto de

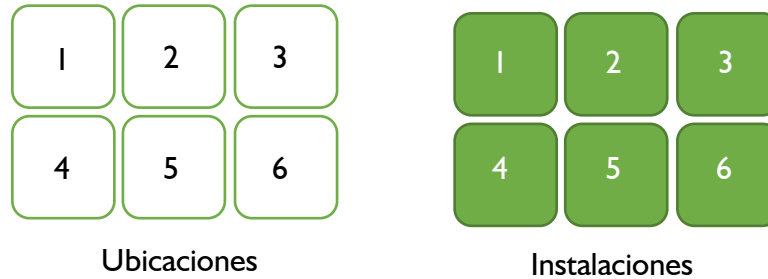
instalaciones (*facilities*) y un conjunto de ubicaciones (*locations*), y el objetivo es asignar cada instalación a una de las ubicaciones de manera que se minimice el costo total.

El costo de una asignación está determinado por una función de costo cuadrática que tiene en cuenta las distancias entre las ubicaciones, así como la interacción entre las instalaciones. Según diversos autores en la literatura, se estima que la solución del QAP es de tipo NP-hard (Fon Y Wong, 2010), lo que significa que es computacionalmente difícil encontrar una solución óptima para problemas a gran escala y, de manera general, requiere de la implementación de algoritmos aproximados para dar una solución. Si bien el presente proyecto de investigación propone la resolución de un problema de optimización multiobjetivo para una cantidad indeterminada de objetivos, su formulación se basa en el trabajo propuesto por Singh y Singh (2010a). A continuación, se describen las generalidades del modelo de optimización.

7.4.9.1 Supuestos. Un diseñador de planta o tomador de decisiones debe determinar la forma en la cual diferentes instalaciones serán organizadas o ubicadas dentro de una planta. Se supone que este diseño no va a cambiar en el tiempo, es decir, no se contemplan diseño de planta dinámicos derivados de cambios en la capacidad o demanda. Adicionalmente, la cantidad de máquinas en cada una de las instalaciones es la indicada y no se contemplan fracciones de áreas, es decir, se puede suponer que el número de máquinas por instalación es unitario o que atiende a las características que defina el diseñador. En este sentido, no se discriminan las áreas ni la orientación de cada instalación y éstas pueden ser vistas como del mismo tamaño. Además, se parte de la base en que hay tantas instalaciones como ubicaciones disponibles existen, los tamaños de ambas son regulares (cuadrados) y su distribución es rectangular como se ve en la Figura 34.

Figura 34

Representación gráfica de la distribución a trabajar



7.4.9.2 Conjuntos.

T Número de objetivos $\{1,2,3, \dots, T\}$

n Número de instalaciones y número de ubicaciones $\{1,2,3, \dots, n\}$

7.4.9.3 Parámetros.

f_{ik} Flujo de trabajo de la instalación i a la instalación k

d_{jl} Distancia de la ubicación j a la ubicación l

cr_{ik} Tasa de cercanía de la instalación i a la instalación k

mht_{jl} Tiempo de distribución de materiales de la ubicación j a la ubicación l

hm_{ik} Movimientos peligrosos de la instalación i a la instalación k

w_t Ponderación de cada objetivo t

A_{ijkl} Representa la combinación cuantitativa y cualitativa de factores en el problema de optimización, también conocido como *multiobjective Facility layout Problem* (mFLP).

7.4.9.4 Variables.

X_{ij} Variable binaria, vale 1 si la instalación i es asignada a la ubicación j

7.4.9.5 Objetivos. Siguiendo por Singh y Singh (2010a), el presente trabajo calibra el modelo con cuatro funciones objetivo las cuales son descritas a continuación.

$F_1(X)$ minimización de flujo de trabajo.

$$F_1(X) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n f_{ik} d_{jl} X_{ij} X_{kl} \quad (1)$$

$F_2(X)$ maximización de tasa de cercanía.

$$F_2(X) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n cr_{ik} X_{ij} X_{kl} \quad (2)$$

$F_3(X)$ minimización de tiempo de distribución de materiales.

$$F_3(X) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n mht_{ik} X_{ij} X_{kl} \quad (3)$$

$F_4(X)$ maximización de movimientos peligrosos.

$$F_4(X) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n hm_{ik}d_{jl}X_{ij}X_{kl} \quad (4)$$

Como es un problema de optimización multiobjetivo resuelto mediante una combinación convexa, la función resultante es:

$$Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n A_{ijkl}X_{ij}X_{kl} \quad (5)$$

7.4.9.6 Restricciones.

Cada instalación i sólo puede ser asignada a una ubicación j

$$\sum_{j=1}^n X_{ij} = 1 \quad \forall i = 1, 2, 3, \dots, n \quad (6)$$

Cada ubicación j sólo puede relacionarse con una instalación i

$$\sum_{i=1}^n X_{ij} = 1 \quad \forall j = 1, 2, 3, \dots, n \quad (7)$$

La variable de decisión es binaria

$$X_{ij} \in \{0, 1\} \quad (8)$$

La combinación de funciones objetivos debe ser una combinación convexa.

$$\sum_{t=1}^T w_t = 1 \quad (9)$$

Como es un problema de minimización, para el ejemplo la combinación de funciones objetivos para las dos primeras funciones de optimización debe ser:

$$A_{ijkl} = w_1 * f_{ik} * d_{jl} - w_2 * cr_{ik} \text{ si } T = 2 \quad (10)$$

De manera análoga, para el ejemplo la combinación de funciones objetivos para las cuatro funciones de optimización es:

$$A_{ijkl} = w_1 * f_{ik} * d_{jl} + w_3 * mht_{jl} + w_4 * hm_{ik} - w_2 * cr_{ik} \text{ si } T = 4 \quad (11)$$

De manera general se obtiene:

$$A_{ijkl} = \sum_{t=1}^T w_t F_t(X), \text{ donde } F_t \text{ es una función de minimización} \quad (12)$$

7.4.10 Heurística

El presente proyecto implementa una heurística para la estimación de ponderaciones para la resolución de un problema de optimización multiobjetivo como si fuese un problema mono-objetivo. Particularmente, hace uso de la estrategia propuesta por Singh y Singh (2010). Esta heurística se divide en dos grandes etapas previas a la solución por metaheurística (en este caso, recocido simulado), la primera es la normalización de matrices de relación entre instalaciones, por ejemplo, matrices de flujo de material, de contaminación cruzada, tiempos de espera, etc. Lo

anterior se busca ya que dependiendo de las magnitudes de los parámetros relacionados con las funciones objetivos la ponderación puede sesgarse hacia un objetivo. Por otra parte, la segunda fase se enfoca en el cumplimiento de requisitos descritos en la Tabla 12

7.4.10.1 Normalización La normalización de las matrices se basa en la división de todos los valores de relación (costo, distancia, tiempos, etc.) en el valor total de la matriz. Se considera una matriz de costo E_{ijt} , donde i indica instalación, j ubicación, y t el objetivo, se reemplaza la matriz de costo normalizada E'_{ijt} de acuerdo con la ecuación (13).

$$E'_{ijt} = \frac{E_{ijt}}{\sum_{i=1}^n \sum_{j=1, j \neq i}^n |E_{ijt}|} \quad \forall t = 1, 2, 3, \dots, T \quad (13)$$

7.4.10.2 Cálculo de pesos Una vez indicados los pesos por parte del tomador de decisiones, la herramienta realiza cuatro operaciones para estimar cuatro posibles ponderaciones diferentes y, por consiguiente, cuatro posibles soluciones al problema de optimización multiobjetivo, esas cuatro transformaciones son respectivamente

7.4.10.2.1 Ponderación manual. En esta ponderación el tomador de decisiones ajusta un valor subjetivo de importancia a cada objetivo y el algoritmo lo ajusta para generar una combinación convexa, es decir, que la suma de las ponderaciones sea igual a 1.0. Durante esta etapa el usuario ingresa un vector de pesos de objetivos w_t donde la ubicación indica la ponderación de cada objetivo t . Este vector es sobre escrito según lo indicado en la ecuación (14).

$$w_t = \frac{w_t}{\sum_{t=1}^T |w_t|} \quad \forall t = 1, 2, 3, \dots, T \quad (14)$$

7.4.10.2.2 Ponderación GMWM. También conocido como método de ponderación por media geométrica, esta ponderación sólo se puede usar cuando todos los valores dentro de una matriz son diferentes de cero (salvo por la diagonal), ya que al ser una productoria dicho valor en cero genera un peso de cero y, por tanto, no se puede determinar. De hecho, en la subsección 7.5.6 Análisis de instancias de referencia probadas en la literatura, se evidencia que diferentes modelos en la literatura no pueden ser evaluados haciendo uso de esta estrategia de ponderación.

$$w_t = \frac{G_t}{\sum_{t=1}^T G_t} \quad \forall t = 1, 2, 3, \dots, T; \quad \forall E'_{ijt} > 0 \quad i \neq j \quad (15)$$

Donde

$$G_t = \sqrt[n(n-1)]{\prod_{\substack{i=1 \\ i \neq j}}^n \prod_{j=1}^n |E'_{ijt}|} \quad \forall t = 1, 2, 3, \dots, T \quad (16)$$

7.4.10.2.3 Ponderación SDWM. También conocido como método de ponderación por desviación estándar. En esta aproximación se le da más peso a aquellas matrices objetivo que mayor dispersión tiene, para ello se calcula la desviación estándar de cada matriz objetivo σ_t el peso asignado corresponde con la porción total de desviación:

$$w_t = \frac{\sigma_t}{\sum_{t=1}^T \sigma_t} \quad \forall t = 1, 2, 3, \dots, T \quad (17)$$

Donde

$$\sigma_t = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1, j \neq i}^n (|E'_{ijt}| - |\bar{E}'_{ijt}|)^2}{n(n-1)}} \quad \forall t = 1, 2, 3, \dots, T \quad (18)$$

7.4.10.2.4 Ponderación CRITICM. También conocido como método de ponderación por criterio de correlación, método originalmente propuesto por Diakoulaki (1995) el cual se basa en darle mayor peso a aquellos objetivos que se encuentren mayormente correlacionados según un criterio de agregación multiplicativa C_t que depende del cálculo de la desviación estándar señalado en la ecuación (18).

$$C_t = \sigma_t \sum_{\substack{k=1 \\ t \neq k}}^T (1 - r_{tk}) \quad \forall t = 1, 2, 3, \dots, T \quad (19)$$

Donde

$$r_{tk} = \frac{\sum_{i=1}^n \sum_{j=1, j \neq i}^n [(|E'_{ijt}| - |\bar{E}'_{ijt}|)_t (|E'_{ikt}| - |\bar{E}'_{ikt}|)_k]}{n(n-1)\sigma_t\sigma_k} \quad \forall t = 1, 2, 3, \dots, T \quad (20)$$

De forma tal que el peso ponderado corresponde con la porción de correlación

$$w_t = \frac{C_t}{\sum_{t=1}^T C_t} \quad \forall t = 1, 2, 3, \dots, T \quad (21)$$

7.4.11 Recocido simulado

Es un algoritmo de optimización inspirado en el proceso físico de recocido de metales. Se utiliza para encontrar la mejor solución a un problema dado, sujeto a restricciones y objetivos. Como es un algoritmo metaheurístico, éste genera soluciones aleatorias y evalúa su calidad

utilizando una función de evaluación (*fitness*). Si la solución es mejor que la anterior, se acepta. Si es peor, se acepta con una probabilidad basada en la diferencia de calidad entre la solución actual y la anterior y en una temperatura que va disminuyendo a medida que avanza el proceso. El aceptar soluciones las cuales no necesariamente mejoran la solución actual favorece la exploración del espacio de búsqueda con el fin de evitar óptimos locales.

El algoritmo es sensible a los parámetros de temperatura y velocidad de enfriamiento. Una temperatura más alta permite una mayor exploración, mientras que una temperatura más baja se enfoca más en la explotación. La tasa de enfriamiento determina qué tan rápido disminuye la temperatura, lo que afecta la duración de la fase de exploración. A continuación, se presenta el pseudocódigo del recocido simulado.

Figura 35

Pseudocódigo del recocido simulado

Paso 0: Defina la temperatura inicial, la longitud de periodo, el periodo de enfriamiento y los criterios de parada

Paso 1: Selecciona aleatoriamente la solución inicial A como solución inicial para el recocido simulado.

Paso 2: Elija una temperatura inicial $T_{inicial} > 0$

Paso 3: Elija la función de actualización de la temperatura, es decir, el programa de recocido (o enfriamiento).

Paso 4: Elija la duración de longitud L.

Paso 5: Generar la solución A' en la vecindad de A intercambiando dos instalaciones (se seleccionan ambas instalaciones al azar).

Paso 6: Calcular $\Delta Z = z(A') - z(A)$

Paso 7:

Si $\Delta Z < 0$ entonces reemplaza A por A', 0 de lo contrario, vaya al paso 9.

Paso 8: Si el valor aleatorio $U\{0,1\} < \exp\left(-\frac{\Delta Z}{t}\right)$ entonces también reemplace A por A', es decir $A' = A$

Paso 9: Repita los pasos 5 a 8 hasta que $L = QK$. Donde, $Q > 1$ y $K = n(n-1)/2$.

Paso 10: Calcule la siguiente temperatura según la función de cambio de temperatura tomada en el paso 3 y repita los pasos 5 a 9 para la siguiente temperatura.

Paso 12: Repita estos pasos hasta que se cumplan los criterios de parada.

Considerando el Paso 0 descrito en la Figura 35 es necesario determinar las temperaturas iniciales, periodos de enfriamiento y criterios de parada. Particularmente, el artículo seminal sugiere utilizar valores altos de temperatura ya que este facilita aceptar las soluciones en la primera etapa del algoritmo con una probabilidad del 80%. En cuanto al periodo de enfriamiento las funciones más populares son:

Función aritmética: $t_{k+1} = t - \text{Constante}$, $k = 0, 1, \dots, K$ (el paso es constante)

Función geométrica: $t_{k+1} = \alpha t_k$, $k = 0, 1, \dots, K$, $\alpha < 1$ y $t_0 = \text{Temperatura inicial}$

Función logarítmica: $t_k = \frac{\text{constante}}{\log(k+2)}$, $k = 0, 1, \dots, K$

Función inversa: $t_{k+1} = \frac{t_k}{1+\beta t_k}$, $k = 0, 1, \dots, K$, $\beta \ll t_0$, $t_0 = \text{Temperatura inicial}$, y $t_k =$

$$\frac{\text{Constante}}{1+k}$$

Finalmente, los criterios de parada son:

Se ha alcanzado un número total determinado de iteraciones;

Si no se ha obtenido el número de aceptación previamente definido para un ensayo dado

Si no se alcanza la temperatura final dada

Si no hay mejora luego de un número determinado de iteraciones

7.4.11.1 Algoritmo Recocido Simulado Modificado. Una vez calculadas las relaciones entre los diferentes objetivos se resuelve el problema como un problema mono objetivo. Se genera una solución factible inicial A . Para ello las instalaciones se asignan a ubicaciones de manera secuencial, de modo que la instalación 1 está en la ubicación 1, la instalación 2 está en la ubicación 2 y así sucesivamente.

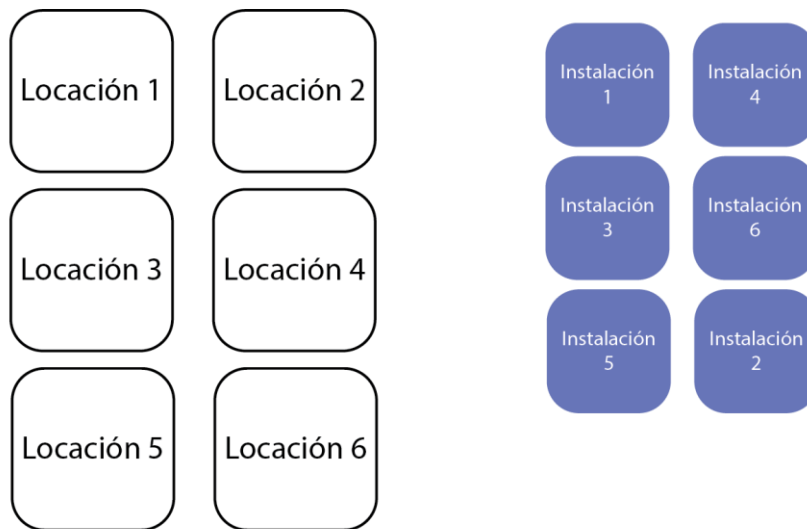
Considere el ejemplo de la Figura 36 para el cual existen seis instalaciones que serán asignadas a sus respectivos espacios de manera ordenada como se observa en la Figura 37. Después el algoritmo calcula el valor de la función objetivo $Z(A)$, donde A es la asignación inicial secuencial. Después, se adopta un proceso de intercambio bidireccional para mejorar la calidad de la solución. Por lo tanto, para una única iteración se examinará un número $n(n - 1)/2$ de intercambios para aceptar o rechazar la solución, donde las soluciones vecinas de A se buscan en un orden fijo, no al azar. El orden de la búsqueda se define por la secuencia $(1, 2), (1, 3), \dots, (1, n), (2, 3), \dots, (2, n), \dots, (n - 1, 2n)$. Es decir, Se necesitan $K = n(n - 1)/2$ ensayos para explorar toda la vecindad de A . A modo de ejemplo, para un problema con $n = 6$ como el descrito en la Figura 37 se necesitan un total de $K = 15$ intercambios. Decimos que se realiza una iteración cuando se realizan K ensayos (evaluaciones de la función objetivo).

Por ejemplo, se selecciona la instalación en la posición 1 que en la solución base corresponde con la instalación 1 y se hace cinco intercambios, donde la posición pivote se colorea en verde oscuro y las posiciones a intercambiar en verde claro (Figura 38). Una vez finalizada la evaluación de todas las cinco funciones objetivo el contador de posiciones aumenta a la segunda y se realizan cuatro nuevos intercambios (Figura 39) para un total acumulado de 9 soluciones vecinas. Nuevamente, se continúa el mismo proceso aumentando la posición de la instalación

pivote y reduciendo la cantidad de intercambios como se ve en la Figura 40 para tres intercambios, Figura 41 para dos intercambios, y Figura 42 para un acumulado de 15 intercambios.

Figura 36

Ejemplo para seis instalaciones en seis locaciones



Si siguiendo esta estrategia se generan para el ejemplo 15 soluciones diferentes a la inicial y se selecciona aquella con valor de función objetivo más alto, en caso de empate, se selecciona al azar. Al igual que el en el algoritmo de recocido simulado general, la temperatura inicial y las temperaturas finales no son fijas, sino que las temperaturas se consideran en función del delta o cambio mínimo ΔZ_{min} ; delta máximo (ΔZ_{max}) y delta promedio (ΔZ_{avg}). Donde ΔZ_{min} y ΔZ_{max} son las diferencias mínimas y máximas obtenidas luego de R intercambios aleatorios donde el número de intercambios aleatorios corresponde con la mitad de los intercambios ordenados $R = K/2$, y ΔZ_{avg} es el cambio promedio de Z es ese mismo número de intercambios. Por otra parte,

las expresiones de T_I (temperatura inicial) y T_F (temperatura final) incluyen los términos ΔZ_{min} , ΔZ_{max} , y ΔZ_{avg} según las siguientes expresiones.

$$T_I = (1 - \lambda_1 - \lambda'_1)\Delta Z_{min} + \lambda_1\Delta Z_{avg} + \lambda'_1\Delta Z_{max} \quad (22)$$

$$T_F = (1 - \lambda_2 - \lambda'_2)\Delta Z_{min} + \lambda_2\Delta Z_{avg} + \lambda'_2\Delta Z_{max} \quad (23)$$

Donde $\lambda_1, \lambda'_1, \lambda_2, \lambda'_2$ son números reales entre cero y uno denominado los pesos o factores de temperatura, y las ecuaciones (22) (23) representan combinaciones convexas de los diferentes deltas calculados. Adicionalmente, si se fijan valores de λ'_1, λ'_2 a $\{0,0\}$ se obtienen las siguientes ecuaciones sugeridas en la literatura:

$$T_I = (1 - \lambda_1)\Delta Z_{min} + \lambda_1\Delta Z_{avg} \quad (24)$$

$$T_F = (1 - \lambda_2)\Delta Z_{min} + \lambda_2\Delta Z_{avg} \quad (25)$$

Figura 37

Asignación inicial



Figura 38

Cinco interambios para la instalación 1 y su vecindad



Figura 39

Cuatro intercambios para la instalación 2 y su vecindad



Figura 40

Tres intercambios para la instalación 3 y su vecindad



Figura 41

Dos intercambios para la instalación 4 y su vecindad

**Figura 42**

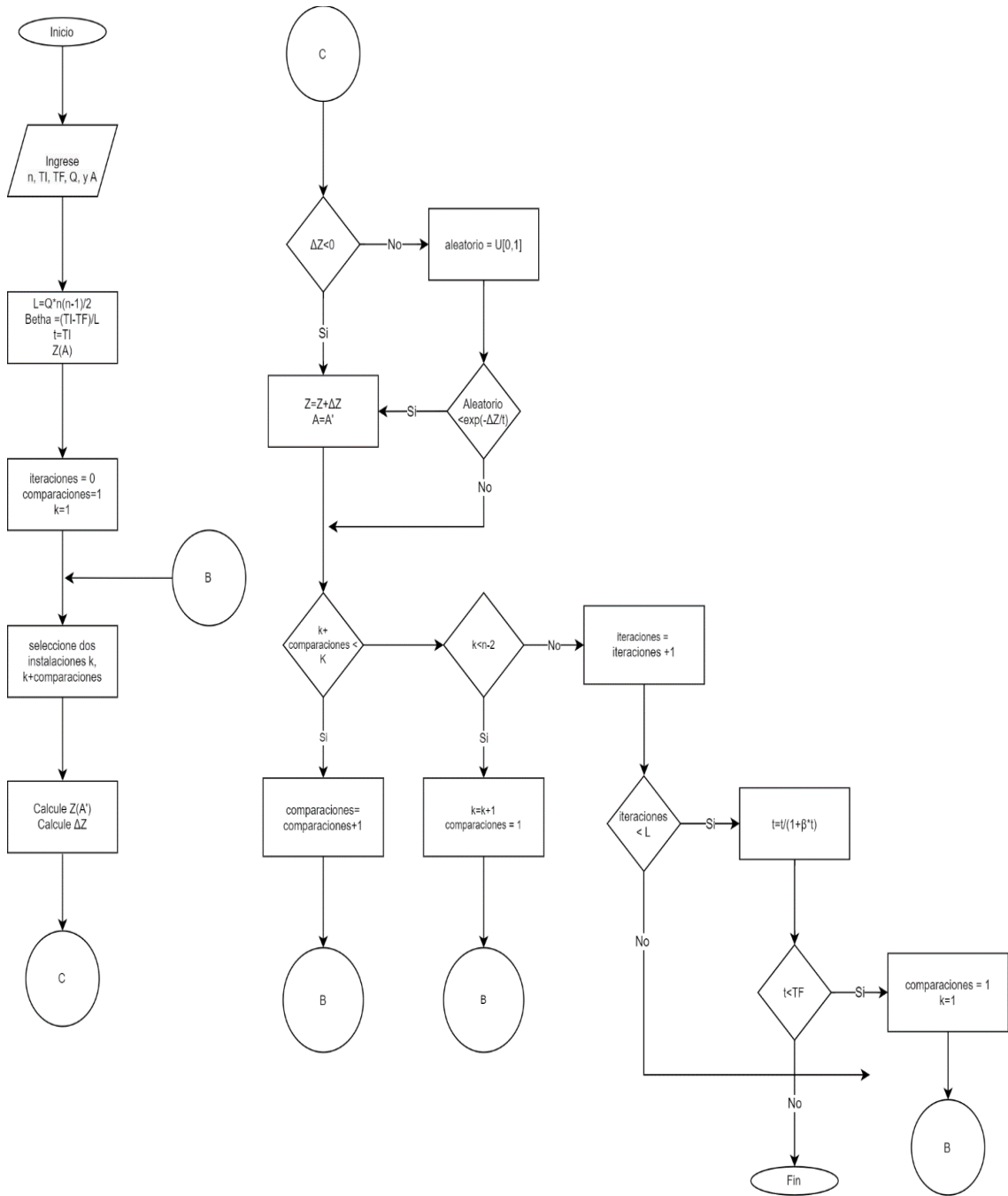
Un intercambio para la instalación 5 y su vecindad



Finalmente, en la Figura 43 se presenta el diagrama de flujo del algoritmo de recocido simulado modificado que se implementó en el FLPWeb.

Figura 43

Diagrama de flujo del algoritmo de recocido simulado modificado para resolver el QAP



7.5 Desarrollo y evaluación del prototipo

7.5.1 Desarrollo del prototipo 1: Algoritmo heurístico y recocido simulado

El primer prototipo de FLPWeb se enfocaba en el desarrollo en Python de una serie de instrucciones que incluían un algoritmo heurístico para estimar las ponderaciones de los objetivos (Manual, GMWM, SDWM, CRITICM), donde la fuente de información (matrices) se cargaba de manera manual en archivos csv. Adicionalmente, el primer prototipo incluía el algoritmo de recocido simulado para solucionar un problema de asignación cuadrática. Este enfoque se tomó debido a la complejidad del QAP, ya que es un problema NP-hard, y se requiere de algoritmos metaheurísticos para poder encontrar soluciones factibles en un tiempo razonable. En esta fase se cumplió con la entrada de datos mediante un archivo de configuración, entrada de datos de la matriz de distancias, entrada de datos de la matriz de cada objetivo y la normalización de estos. Utilizando el algoritmo de recocido simulado, junto con el algoritmo heurístico para las ponderaciones, se logró desarrollar una herramienta capaz de resolver el QAP de manera eficiente y precisa. De hecho, durante este primer prototipo se realizó la validación estadística del algoritmo de recocido simulado determinando así los parámetros sugeridos para generar las soluciones en tiempos razonables (subsección 7.5.4).

Es importante mencionar que se desarrolló el paquete `externals` que se puede visualizar a detalle en la Figura 44, como un diagrama de clases que muestra las relaciones que existen entre ellas. Además, en la Figura 45 se da una pequeña vista del código que se creó en el editor Visual Studio Code.

Figura 44

Paquete externals

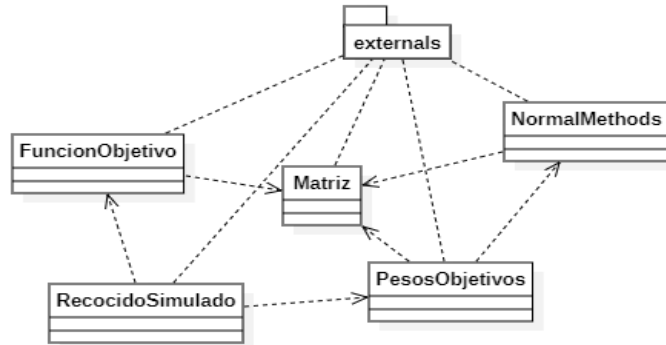


Figura 45

```

1 import numpy as np
2 from matplotlib import pyplot as plt
3 from random import shuffle
4 import random as random
5 import math
6 import time
7 import copy
8 from itertools import combinations, product
9
10
11
12
13 def hayMatrizFlujoDist(matrices):
14     """
15     Averigua si hay alguna matriz de flujo o de distancia en las matrices
16     Return:
17     hayMatrFlujoDist :True si existe, False si no la hay
18     matrizFlujoDist: la matriz de Flujo y la de distancia de todo el vector
19     index: retorna la posicion donde está la matriz de flujo
20     """
21     isMatrizFlujoDistArray = [x.isMatrizFlujoDist for x in matrices] #Almacena los true o false
22     isMatrizFlujoDistArray = np.asarray(isMatrizFlujoDistArray)
23     matrizFlujoDist = [x for x in matrices if x.isMatrizFlujoDist == True] #Almacena la matriz de flujo)
24     hayMatrFlujoDist = isMatrizFlujoDistArray.any()
25     index = [ i for i,x in enumerate(matrices) if x.isMatrizFlujoDist == True] #Almacena el indice
26     if(hayMatrFlujoDist):
27         return hayMatrFlujoDist, matrizFlujoDist, index
28     else:
29         return hayMatrFlujoDist, 0, 0
30
31
32 def hayMatricesPorIndex(matrices):

```

Muestra código primer prototipo

7.5.2 Desarrollo del prototipo 2: Backend

Una vez solucionado el problema FLP considerando dos y cuatro objetivos de optimización validados con resultados de la literatura (ver Singh y Singh (2010)), se modificó el algoritmo para que fuera posible resolver una cantidad indeterminada de objetivos. Es de resaltar que dicha característica aumentó la complejidad del algoritmo de solución ya que la heurística debía ser ajustada a una mayor cantidad de matrices, es decir, la generalización de casos para la segunda versión del prototipo implicó ajustar la “lógica del negocio”. Durante esta fase se realiza la validación del modelo con problemas distintos al propuesto por los autores seminales (ver subsecciones 7.5.5 y 7.5.6). A partir de este ejercicio se logró permitir cargar sin problemas los archivos de configuración de cada caso de estudio, no se evidenciaron errores de desbordamiento de memoria, bucles infinitos, ni errores de sintaxis o dimensiones de los objetos matriciales. Se cumplió con el parámetro de extensibilidad y persistencia. Adicionalmente, durante esta versión de la herramienta se hizo la integración con ORM Django y la base de datos PostgreSQL. En la Figura 46 se muestra una parte del código y el árbol de los archivos que se utilizaron en este prototipo. También, se empezó a trabajar con un servidor local en el puerto 8000, esto para poder ir mostrando en el navegador las vistas que se iban desarrollando en la herramienta, en la Figura 47 y Figura 48 se puede observar la interfaz de este prototipo.

Figura 46

Muestra código segundo prototipo

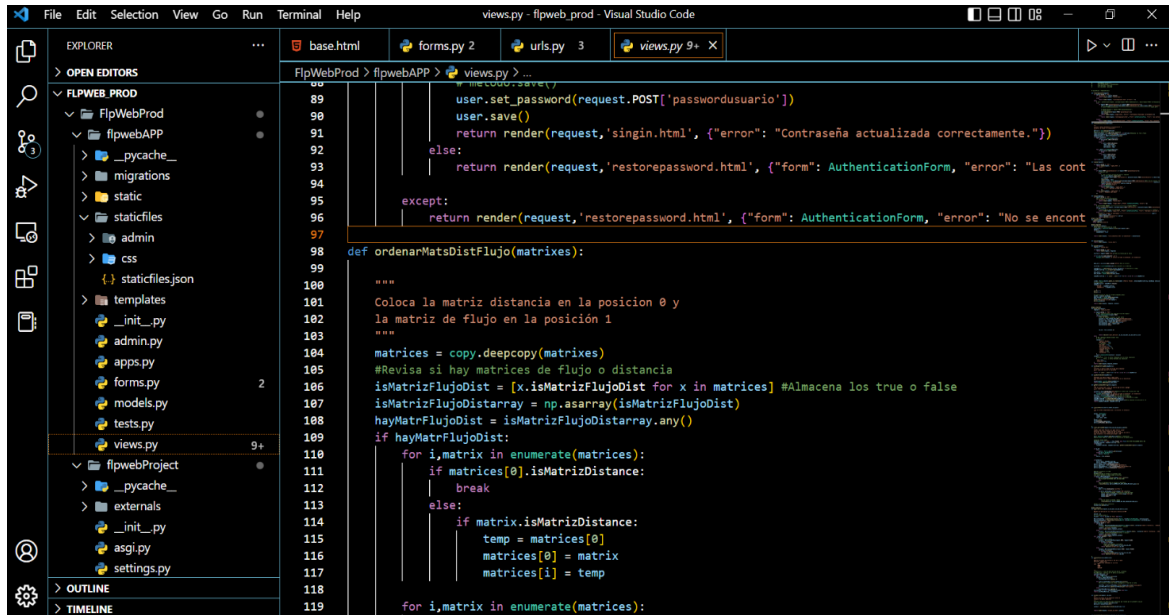


Figura 47

Interfaz página inicio segundo prototipo

The screenshot shows a web browser window with a form for configuring a simulation. The form includes several input fields and checkboxes. The browser's address bar shows 'localhost:8000/nueva_sim/'.

Nombre de la Simulación:	<input type="text" value="Prueba"/>
Método Manual:	<input checked="" type="checkbox"/>
Método Media Geométrica:	<input checked="" type="checkbox"/>
Método Desviación Estándar:	<input checked="" type="checkbox"/>
Método Matriz de Correlaciones:	<input checked="" type="checkbox"/>
Cantidad de instalaciones:	<input type="text" value="6"/>
Cantidad de Matrices:	<input type="text" value="3"/>
Parámetro Lambda 1:	<input type="text" value="0,5"/>
Parámetro Lambda 2:	<input type="text" value="0,05"/>
Parámetro Cantidad Iteraciones:	<input type="text" value="1"/>

Figura 48

Validación simulación segundo prototipo

La Simulación se hará con los siguientes datos

3
 Nombre de la simulación Prueba
 En la que va utilizar los métodos de Cálculo de Objetivos:
 Los parámetros de la metaheurística Recocido Simulado son:

- Lambda 1: 0.5
- Lambda 2: 0.05
- Cantidad de iteraciones: 1

[Editar Simulación](#)

Los datos de las matrices que va a utilizar son:

Nombre	Matriz	Signo en la Función Objetivo	Tipo de Matriz	¿Se debe reindexar la Matriz?	Pesos Método Manual
d	0.0 1.0 2.0 1.0 2.0 3.0 1.0 0.0 1.0 2.0 1.0 2.0 2.0 1.0 0.0 3.0 2.0 1.0 1.0 2.0 3.0 0.0 1.0 2.0 2.0 1.0 2.0 1.0 0.0 1.0 3.0 2.0 1.0 2.0 1.0 0.0	+	Distancia	Si	1.0000
fluj	0.0 4.0 6.0 2.0 4.0 4.0 4.0 0.0 4.0 2.0 2.0 8.0 6.0 4.0 0.0 2.0 2.0 6.0 2.0 2.0 2.0 0.0 6.0 2.0 4.0 2.0 2.0 6.0 0.0 10.0 4.0 8.0 6.0 2.0 10.0 0.0	+	Flujo de Materiales	Si	0.7000
tasa	0.0 5.0 3.0 2.0 6.0 4.0 5.0 0.0 5.0 2.0 6.0 2.0 3.0 5.0 0.0 1.0 2.0 1.0 2.0 2.0 1.0 0.0 2.0 2.0 6.0 6.0 2.0 2.0 0.0 6.0 4.0 2.0 1.0 2.0 6.0 0.0	+	Otra	No	0.3000

[Editar Todas las Matrices](#)

[Obtener Pesos Objetivos](#)

7.5.3 Desarrollo del prototipo 3: Frontend, conexión con servidor y guía de usuario

Durante la tercera versión del prototipo se implementaron las características del frontend del FLPWeb, es decir, se realizó un primer diseño de la interfaz gráfica de usuario (GUI) de la herramienta, cubriendo componentes como el diseño de pantallas, menús, botones, formularios y otros elementos visuales como la impresión de los resultados. Igualmente, se desarrolló el código HTML, CSS y JavaScript que permita mostrar la GUI de manera adecuada en los distintos navegadores web, haciendo unas primeras pruebas en Chrome, Edge, Firefox y Brave. Luego fueron implementadas las funcionalidades de la herramienta utilizando JavaScript. Esto incluye la

validación de formularios, el procesamiento de datos, la comunicación con el backend y la presentación de resultados.

En este prototipo también se incluyó la noción de usuario con contraseña para poder tener las simulaciones almacenadas y poder visualizarlas cuando se requiera. Finalmente, la herramienta se desplegó en un servidor en la nube llamado Railway (Railway Corp, 2023), el cual permite cargar todos los archivos de la herramienta. La url en la que se encuentra alojada la herramienta está en [FLPWeb](#).

En las Figuras a continuación se describen las pantallas principales del último prototipo de la herramienta. Desde la Figura 49 a la Figura 53 se puede visualizar la página de inicio y las vistas concernientes a la creación de usuario e inicio de sesión, además, la vista donde se pueden consultar las simulaciones guardadas. La Figura 54 y Figura 55 muestran las dos páginas para ingresar los datos tanto de la simulación, como de las matrices de esta; para editar estas dos, la vista que se utiliza es la misma, aprovechando el principio de Django DRY (Don't Repeat Yourself: No te repitas a ti mismo), el cual dice que se debe reutilizar todo el código lo máximo que sea posible.

Las vistas que se muestran en la Figura 56 y Figura 57 son las encargadas de validar los datos ingresados por el usuario para luego poder calcular los resultados finales que son mostrados en la vista de la Figura 58.

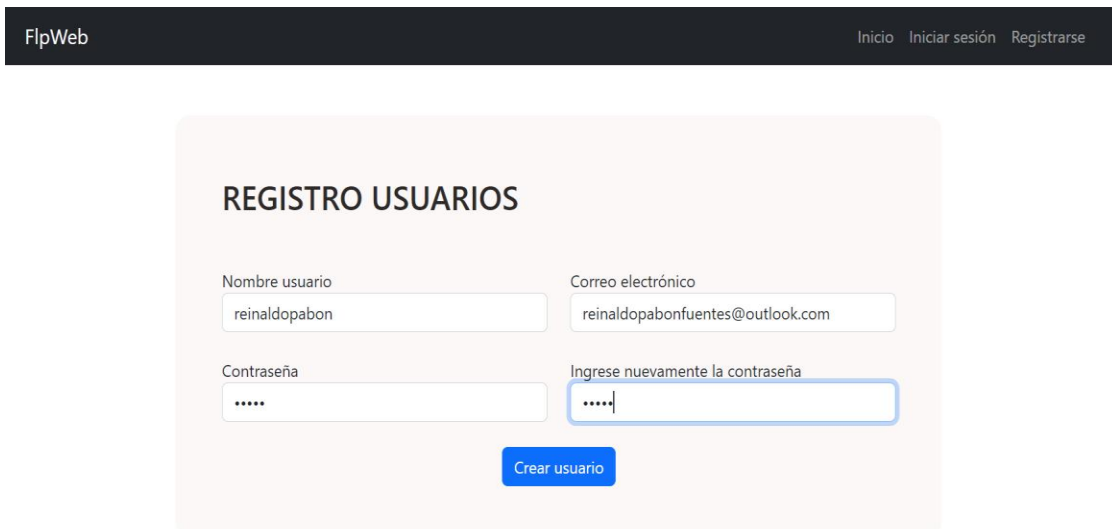
Figura 49*Página inicio prototipo***Figura 50***Vista crear usuarios*

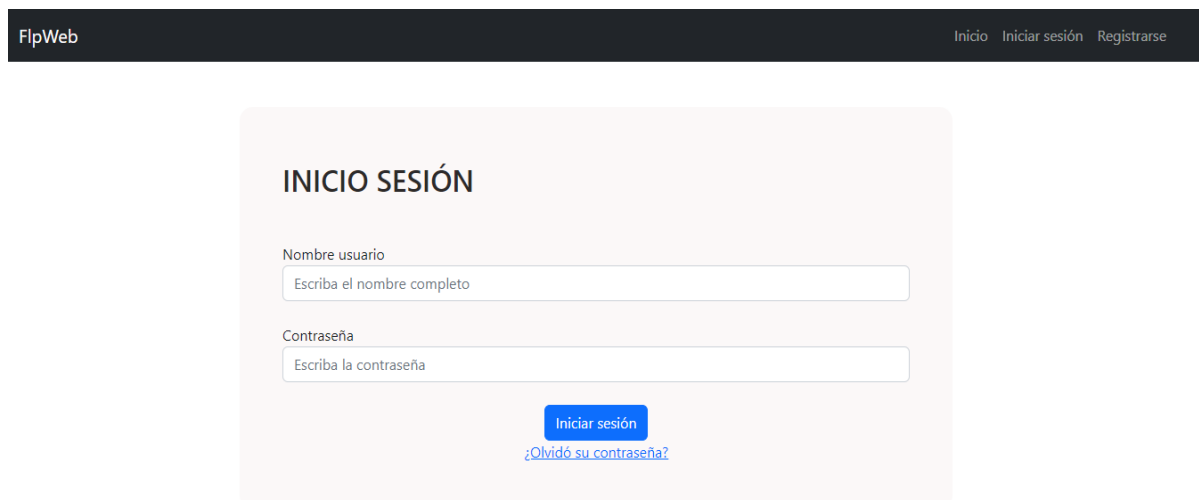
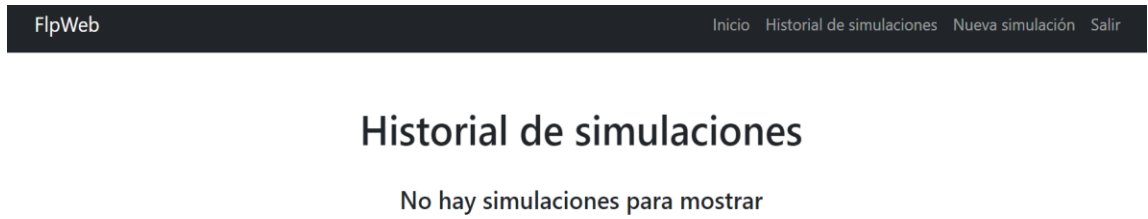
Figura 51*Vista de inicio con usuario logueado***Figura 52***Vista iniciar sesión*

Figura 53*Vista historial de simulaciones***Figura 54***Vista registro de simulación*

The screenshot shows the 'Registro de simulación' form. At the top, there is a dark navigation bar with 'FlpWeb' on the left and 'Inicio', 'Historial de simulaciones', 'Nueva simulación', and 'Salir' on the right. The main content area has a heading 'Registro de simulación' and a form with the following fields and options:

- Nombre de la Simulación:
- Seleccione los métodos deseados:
 - Método Manual:
 - Método Media Geométrica:
 - Método Desviación Estándar:
 - Método Matriz de Correlaciones:
- Cantidad de instalaciones:
- Cantidad de Matrices:
- Ingrese parametros recocido simulado:
 - Parámetro Lambda 1:
 - Parámetro Lambda 2:
 - Parámetro Cantidad Iteraciones:
-

Figura 55

Vista ingresar datos de las matrices

FlpWeb Inicio Historial de simulaciones Nueva simulación Salir

Ingresa los datos para cada matriz

Nombre de la matriz/objetivo:

Signo en la Función Objetivo:

¿Qué tipo de Matriz es?

¿La Matriz debe ser reindexada?

Archivo CSV de la Matriz: Ninguno archivo selec.

Valor peso de la Matriz:

Nombre de la matriz/objetivo:

Signo en la Función Objetivo:

¿Qué tipo de Matriz es?

¿La Matriz debe ser reindexada?

Archivo CSV de la Matriz: Ninguno archivo selec.

Valor peso de la Matriz:

Nombre de la matriz/objetivo:

Signo en la Función Objetivo:

¿Qué tipo de Matriz es?

¿La Matriz debe ser reindexada?

Archivo CSV de la Matriz: Ninguno archivo selec.

Valor peso de la Matriz:

Figura 56

Vista validar datos simulación

FlpWeb Inicio Historial de simulaciones Nueva simulación Salir

La Simulación se hará con los siguientes datos

Simulación Prueba 6 Matrices - ID : 17

Métodos de Cálculo de Objetivos Aplicados	Parámetros de la metaheurística Recursivo Simulado
Método Manual	Lambda 1: 0.5
Método de la Media Geométrica	Lambda 2: 0.05
Método de la Desviación Estándar	Cantidad de iteraciones: 1
Método de Matriz de Correlaciones	

Datos de las Matrices

Nombre	Matriz	Signo	Tipo de Matriz	Reindexar	Pesos Metodo Manual																																				
Distancia	<table border="1" style="font-size: 0.8em; border-collapse: collapse;"> <tr><td>0,0</td><td>1,0</td><td>2,0</td><td>1,0</td><td>2,0</td><td>3,0</td></tr> <tr><td>1,0</td><td>0,0</td><td>1,0</td><td>2,0</td><td>1,0</td><td>2,0</td></tr> <tr><td>2,0</td><td>1,0</td><td>0,0</td><td>3,0</td><td>2,0</td><td>1,0</td></tr> <tr><td>1,0</td><td>2,0</td><td>3,0</td><td>0,0</td><td>1,0</td><td>2,0</td></tr> <tr><td>2,0</td><td>1,0</td><td>2,0</td><td>1,0</td><td>0,0</td><td>1,0</td></tr> <tr><td>3,0</td><td>2,0</td><td>1,0</td><td>2,0</td><td>1,0</td><td>0,0</td></tr> </table>	0,0	1,0	2,0	1,0	2,0	3,0	1,0	0,0	1,0	2,0	1,0	2,0	2,0	1,0	0,0	3,0	2,0	1,0	1,0	2,0	3,0	0,0	1,0	2,0	2,0	1,0	2,0	1,0	0,0	1,0	3,0	2,0	1,0	2,0	1,0	0,0	+	Distancia	Sí	1.0000
0,0	1,0	2,0	1,0	2,0	3,0																																				
1,0	0,0	1,0	2,0	1,0	2,0																																				
2,0	1,0	0,0	3,0	2,0	1,0																																				
1,0	2,0	3,0	0,0	1,0	2,0																																				
2,0	1,0	2,0	1,0	0,0	1,0																																				
3,0	2,0	1,0	2,0	1,0	0,0																																				

Figura 57

Vista simulación con pesos objetivos

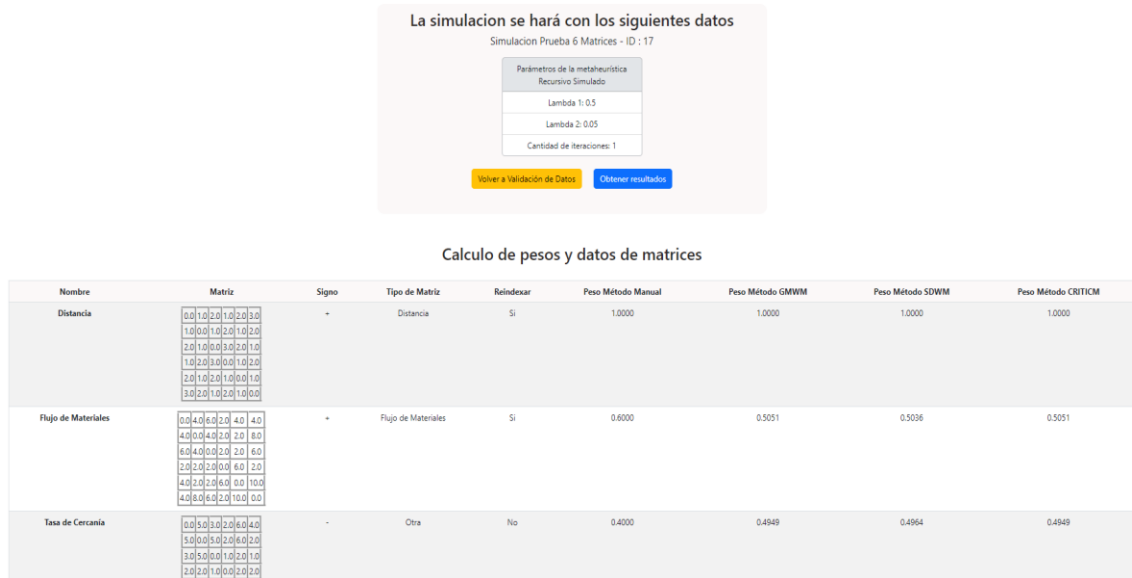
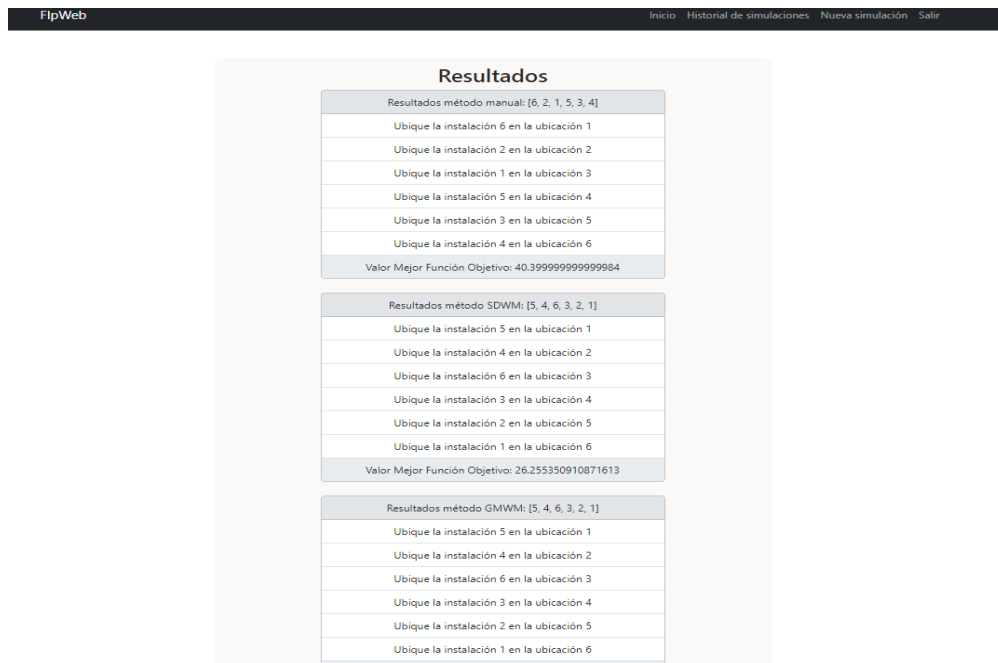


Figura 58

Vista mostrar resultados



La guía de usuario, se creó mediante una herramienta web llamada ScribeHow (Labs, 2022), que permitió crear dos guías de usuario, en la [Guía de Usuario No. 1](#) se explican las actividades relacionadas con el Usuario y crear una simulación. En la [Guía de Usuario No. 2](#) se cubren las actividades de cálculo de resultados y de consultar simulaciones guardadas. Ambas guías se encuentran consolidadas en un archivo en formato PDF **Apéndice B. Guía de Usuario**

Es importante mencionar que como delimitantes de esta versión, la herramienta no pudo ser probada con 40 usuarios en simultáneo ya que este requerimiento originalmente propuesto desconocía de lo especializado que es el perfil de usuario y las pruebas fueron realizadas enteramente por el autor del trabajo y un asesor externo.

Por otro lado, en el **Apéndice C. Matrices de Prueba.zip** se encuentran los archivos en formato CSV de tres problemas de la literatura con tres tamaños diferentes esto es: 6 instalaciones con 2 objetivos de Rosenblatt (1979), 8 instalaciones con 4 objetivos de Chen y Sha (2005) y 15 instalaciones con 2 objetivos de C.W. Chen (1999); para poder utilizar la guía en caso de querer practicar con la herramienta.

Finalmente, todo el código desarrollado del tercer y final prototipo de la herramienta web se puede encontrar en el [Repositorio de GitHub](#).

7.5.4 Evaluación estadística del algoritmo de recocido simulado

Con el fin de probar qué valores eran mejores para los parámetros de ingreso del algoritmo recocido simulado, se realiza un experimento factorial 3^3 donde los tres factores son los tres parámetros del algoritmo: Número de iteraciones (Q), λ_1 y λ_2 que según la literatura consultada recomiendan utilizar 50, 0.5 y 0.05 (Matai, 2015). En la Tabla 22 se muestran los detalles utilizados para el experimento realizado. En este experimento se trabajaron con tres niveles para cada factor,

cabe resaltar que en la literatura se encuentra que como los valores de lambda deben ser inferiores a 1.0, el presente trabajo utiliza los valores correspondientes a los cuartiles teóricos de una distribución uniforme continua entre 0 y 1 ya que esta es la distribución estadística que sigue el generador de números pseudoaleatorio.

Así mismo, estos umbrales son configurados como alto, medio y bajo, es importante mencionar que para el experimento se toma como nivel medio el valor recomendado por (los autores) y para el valor alto y bajo, se aumentó y disminuyó en un 25% el valor utilizado como valor medio respectivamente; además, se codificó cada nivel como -1, 0 y 1 para los niveles bajo, medio y alto respectivamente. Si no se hacen réplicas, el diseño propuesto presentaría un total de 27 corridas y estaría sobre saturado lo cual no permitiría evaluar interacciones, así que se proponen tres réplicas para un total de 81 corridas, donde la métrica de calidad corresponde con la evaluación de la función objetivo. Finalmente, se resalta que el diseño fue aplicado a un problema de optimización el cual tiene una instancia de 20x20 propuesta por los autores de referencia (C.W. Chen, 1999).

Tabla 22

Niveles del diseño factorial

Nivel	Código	Q	Factores	
			lambda ₁	lambda ₂
Bajo	-1	25	0,25	0,025
Medio	0	50	0,5	0,05
Alto	1	75	0,75	0,075

Inicialmente se realiza una estadística descriptiva de la respuesta, de la cual se obtiene que los datos no se comportan de una forma normal, además, se nota que se tiene solo un dato atípico (Figura 59). Adicionalmente, según los resultados para Q (Figura 60), al aumentar el número de iteraciones en promedio no se mejora la respuesta, pero es menos variable, es decir, aparentemente la búsqueda se concentra más y se obtienen resultados menos dispersos; a pesar de esto, se verifica en la gráfica de resumen que para el nivel alto donde hay menos variabilidad, este nivel no presenta un comportamiento normal y presenta dos datos atípicos, esto se puede explicar en que al estar los datos más agrupados (tener menor varianza), cualquier dato con un valor diferente se convierte en un dato atípico. Por otro lado, en los otros dos niveles se observa que los datos están un poco más dispersos y sí se comportan de forma normal.

En cuanto a λ_1 , se encuentra que la variable respuesta es normal para todos los niveles (Figura 61); sin embargo, al aumentar λ_2 se disminuye la variabilidad. Aparentemente no hay diferencia, por esto se realiza una prueba matemática para corroborar que no hay diferencia entre los niveles (Figura 62). Los resultados de las corridas, la instancia de calibración y el resumen de la tabulación se encuentran en el Apéndice D. Resultados y Tabulación Calibración Parámetros RS.zip

Figura 59

Gráfica resumen de las corridas

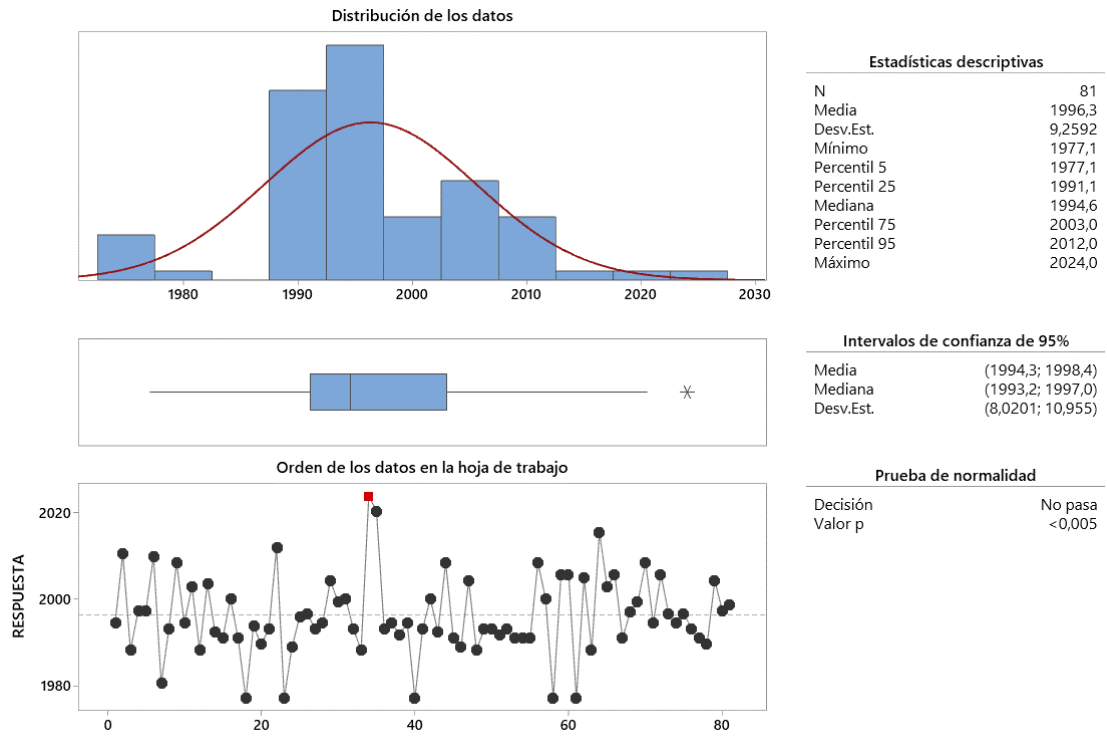


Figura 60

Gráfica resumen de las corridas dividida por niveles del factor Q

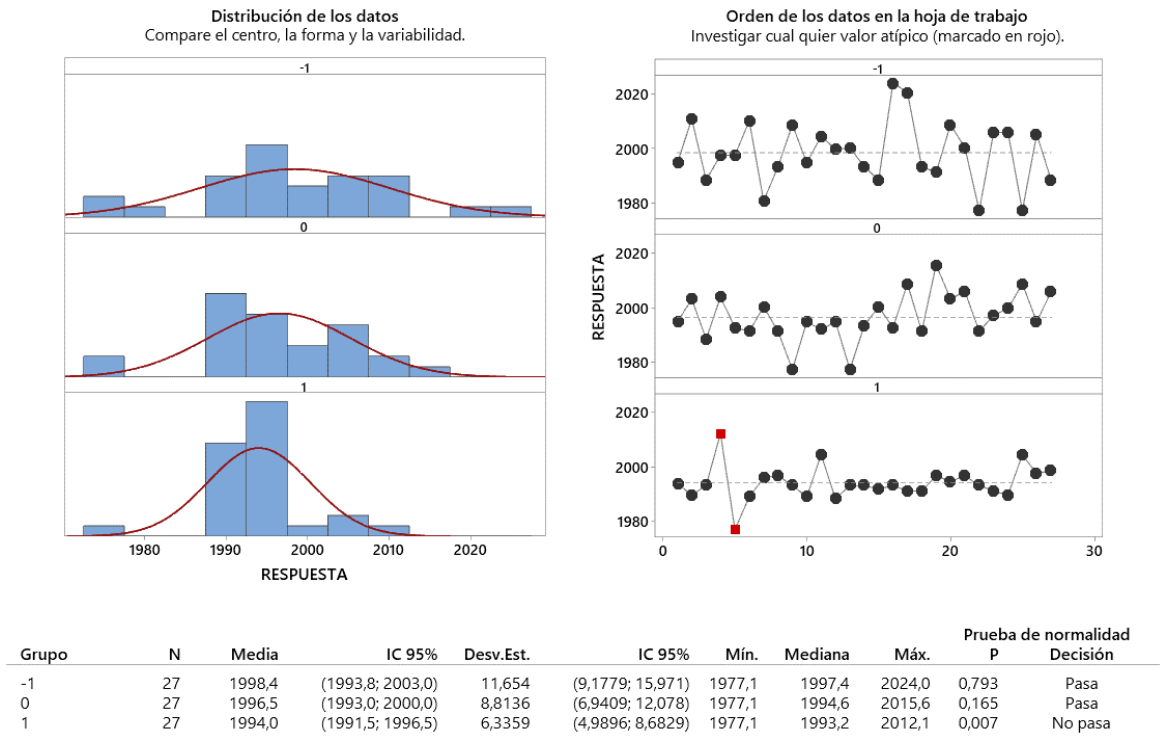


Figura 61

Gráfica resumen de las corridas dividida por niveles del factor Lambda1

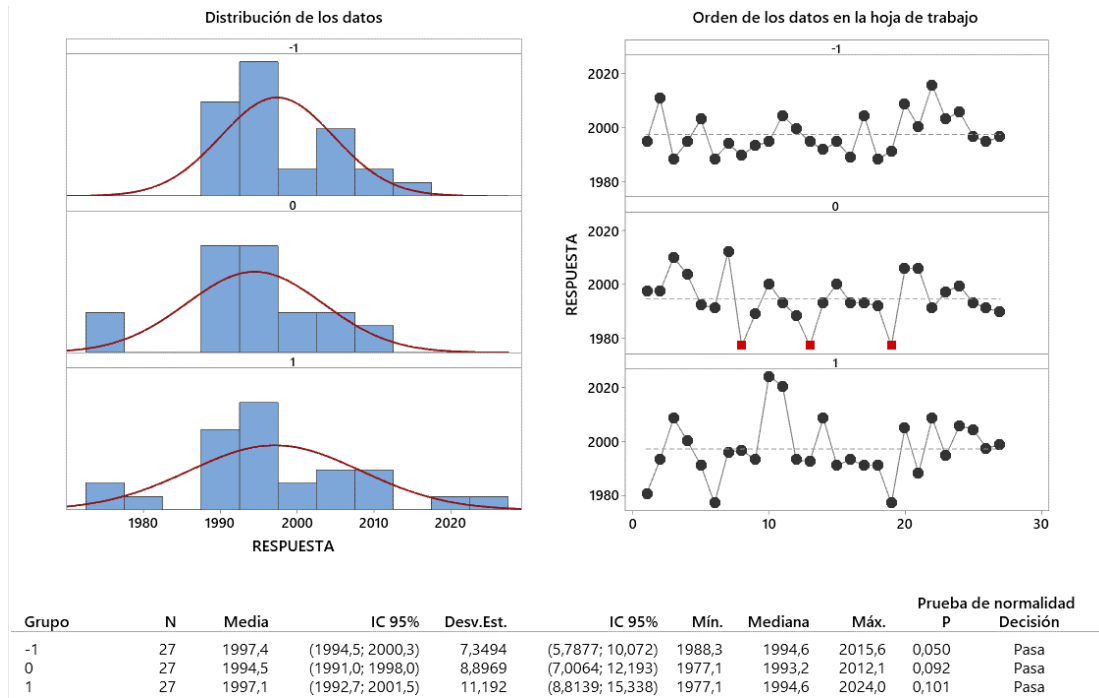
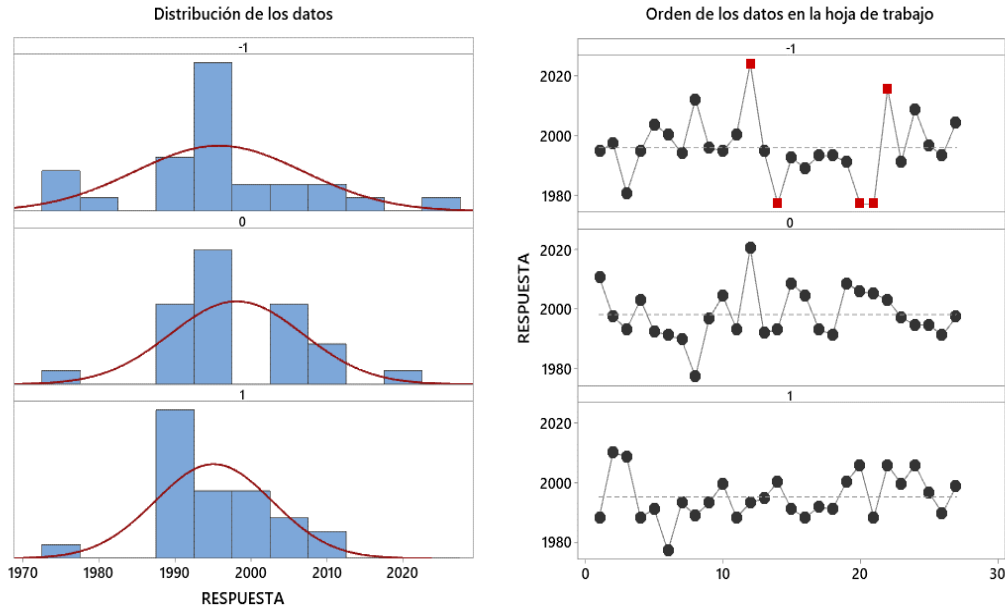


Figura 62

Gráfica resumen de las corridas dividida por niveles del factor Lambda2



Grupo	N	Media	IC 95%	Desv.Est.	IC 95%	Mín.	Mediana	Máx.	Prueba de normalidad	
									P	Decisión
-1	27	1995,8	(1991,4; 2000,2)	11,105	(8,7454; 15,219)	1977,1	1994,6	2024,0	0,035	No pasa
0	27	1998,1	(1994,7; 2001,6)	8,7063	(6,8564; 11,931)	1977,1	1996,7	2020,5	0,050	Pasa
1	27	1995,1	(1992,0; 1998,1)	7,7118	(6,0732; 10,568)	1977,1	1993,2	2010,0	0,063	Pasa

Considerando el diseño experimental propuesto anteriormente, luego de la inspección descriptiva se procede a realizar el ANOVA. Esta prueba implica el cálculo de varias estadísticas y compararlas con un valor crítico para determinar si hay diferencias significativas entre los grupos. En la Tabla 23 se indican los resultados de éstos, si el valor p es menor que 0,05 se puede decir que hay efecto del factor o sus interacciones, según lo registrado en la misma se concluye con un nivel de significancia de 5% que el cambio en niveles no incide sobre el valor de la función objetivo y, por tanto, no deben ser restrictivos dentro del desarrollo de software a implementar. Estos resultados se alinean con la teoría, ya que algunos autores proponen de manera empírica utilizar el valor medio de los parámetros 1 y 2, es decir, $Q=50$, $\lambda_1=0.5$, acompañado. No obstante, estos resultados no se pueden comparar en su totalidad ya que en la literatura se sugiere utilizar un valor de $\lambda_2=0.05$. A pesar de esto, se decide utilizar $Q=25$, debido a que en la práctica el algoritmo

al aumentar las iteraciones aumenta proporcionalmente el tiempo de cómputo y la herramienta necesita dar respuestas rápidas.

Tabla 23

Tabla ANOVA

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
A	2	263,86	131,93	1,39	0,257
B	2	139,43	69,72	0,74	0,483
C	2	135,15	67,58	0,71	0,494
A*B	4	51,65	12,91	0,14	0,968
A*C	4	595,48	148,87	1,57	0,195
B*C	4	243,31	60,83	0,64	0,634
A*B*C	8	319,78	39,97	0,42	0,903
Error	54	5109,93	94,63		
Total	80	6858,59			

7.5.5 Análisis del algoritmo de recocido simulado para la solución del QAP

Para determinar la capacidad del algoritmo para hallar soluciones óptimas a problemas mono-objetivos QAP, se hace uso de una base de datos de problemas de asignación cuadrática (Burkard, Çela, Rendl, y Karisch, 1997).

Las instancias de NVR enumeradas en esta página web son un conjunto específico de QAP propuestos por Neville y Von Neumann como problemas de prueba para algoritmos de optimización (Nugent et al., 1968). Cabe resaltar que la matriz de distancias utilizadas en estos problemas hace uso de una distancia de Manhattan asociada a las cuadrículas rectangulares que representan las locaciones.

Según los autores, las instancias de tamaño $n = \{14, 16, 17, 18, 21, 22, 24, 25\}$ se construyeron a partir de las instancias más grandes y fueron eliminadas filas y columnas (Clausen Y Perregaard, 1997), igualmente, a partir de este trabajo se obtienen las soluciones óptimas para ser comparadas con el algoritmo de recocido simulado implementado. Para el caso de las instancias $n = \{21, 22\}$, la solución óptima fue demostrada por Brungger (1997) para la instancia de $n = 24$. Las instancias de tamaño $n = \{27, 28\}$ se construyeron a partir de la instancia de tamaño $n = 30$ eliminando las tres o dos últimas instalaciones, respectivamente, y estas tres instancias ($n = \{27, 28, 30\}$) fueron (Anstreicher Y Brixius, 2001a, 2001b) para las de 27 y 28 y Anstreicher y Brixius para las de 30 (2001a, 2001b). De este ejercicio se resalta que el algoritmo implementado logra hallar las soluciones óptimas en tiempos razonables, como se puede ver en la Tabla 24. El detalle de los resultados instancia por instancia y la tabulación resumen se encuentra en el Apéndice E. Resultados y Tabulación QAPlib.zip

Tabla 24*Tabla de resultados para las 15 instancias QAP*

Instancia	n	Método	Solución	Z
NUG12	12	Óptima	[12, 7, 9, 3, 4, 8, 11, 1, 5, 6, 10, 2]	578
		FLPWeb	[12, 7, 9, 3, 4, 8, 11, 1, 5, 6, 10, 2]	578
NUG14	14	Óptima	[9, 8, 13, 2, 1, 11, 7, 14, 3, 4, 12, 5, 6, 10]	1014
		FLPWeb	[1, 2, 13, 8, 9, 4, 3, 14, 7, 11, 10, 6, 5, 12]	1016
NUG15	15	Óptima	[1, 2, 13, 8, 9, 4, 3, 14, 7, 11, 10, 15, 6, 5, 12]	1150
		FLPWeb	[12, 5, 6, 15, 10, 11, 7, 14, 3, 4, 9, 8, 13, 2, 1]	1150
NUG16a	16	Óptima	[9, 14, 2, 15, 16, 3, 10, 12, 8, 11, 6, 5, 7, 1, 4, 13]	1610
		FLPWeb	[9, 14, 2, 15, 16, 3, 10, 12, 8, 11, 6, 5, 7, 1, 4, 13]	1610
NUG16b	16	Óptima	[16, 12, 13, 8, 4, 2, 9, 11, 15, 10, 7, 3, 14, 6, 1, 5]	1240
		FLPWeb	[5, 3, 11, 8, 1, 7, 9, 13, 6, 10, 2, 12, 14, 15, 4, 16]	1240
NUG17	17	Óptima	[16, 15, 2, 14, 9, 11, 8, 12, 10, 3, 4, 1, 7, 6, 13, 17, 5]	1732
		FLPWeb	[16, 15, 2, 14, 9, 11, 8, 12, 10, 3, 4, 1, 7, 6, 13, 17, 5]	1732
NUG18	18	Óptima	[10, 3, 14, 2, 18, 6, 7, 12, 15, 4, 5, 1, 11, 8, 17, 13, 9, 16]	1930
		FLPWeb	[9, 3, 10, 6, 13, 14, 2, 12, 7, 5, 18, 15, 8, 1, 17, 16, 4, 11]	1930
NUG20	20	Óptima	[18, 14, 10, 3, 9, 4, 2, 12, 11, 16, 19, 15, 20, 8, 13, 17, 5, 7, 1, 6]	2570
		FLPWeb	[6, 1, 7, 5, 17, 13, 8, 20, 15, 19, 16, 11, 12, 2, 4, 9, 3, 10, 14, 18]	2570
NUG21	21	Óptima	[4, 21, 3, 9, 13, 2, 5, 14, 18, 11, 16, 10, 6, 15, 20, 19, 8, 7, 1, 12, 17]	2438
		FLPWeb	[20, 19, 8, 7, 1, 12, 17, 14, 18, 11, 16, 10, 6, 15, 4, 21, 3, 9, 13, 2, 5]	2438
NUG22	22	Óptima	[2, 21, 9, 10, 7, 3, 1, 19, 8, 20, 17, 5, 13, 6, 12, 16, 11, 22, 18, 14, 15]	3596
		FLPWeb	[15, 14, 4, 18, 22, 11, 16, 12, 6, 13, 5, 17, 20, 8, 19, 1, 3, 7, 10, 9, 21, 2]	3596
NUG24	24	Óptima	[17, 8, 11, 23, 4, 20, 15, 19, 22, 18, 3, 14, 1, 10, 7, 9, 16, 21, 24, 12, 6, 13, 5, 2]	3488
		FLPWeb	[17, 24, 1, 19, 8, 20, 15, 12, 22, 18, 23, 14, 5, 6, 10, 7, 3, 11, 2, 13, 21, 9, 16, 4]	3488
NUG25	25	Óptima	[5, 11, 20, 15, 22, 2, 25, 8, 9, 1, 18, 16, 3, 6, 19, 24, 21, 14, 7, 10, 17, 12, 4, 23, 13]	3744
		FLPWeb	[22, 1, 19, 10, 13, 15, 9, 6, 7, 23, 20, 8, 3, 14, 4, 11, 25, 16, 21, 12, 5, 2, 18, 24, 17]	3744
NUG27	27	Óptima	[23, 18, 3, 1, 27, 17, 5, 12, 7, 15, 4, 26, 8, 19, 20, 2, 24, 21, 14, 10, 9, 13, 22, 25, 6, 16, 11]	5234
		FLPWeb	[21, 13, 9, 25, 7, 11, 3, 16, 4, 2, 6, 26, 10, 19, 22, 8, 27, 20, 5, 12, 24, 17, 1, 23, 18, 15, 14]	5234
NUG28	28	Óptima	[18, 21, 9, 1, 28, 20, 11, 3, 13, 12, 10, 19, 14, 22, 15, 2, 25, 16, 4, 23, 7, 17, 24, 26, 5, 27, 8, 6]	5166
		FLPWeb	[18, 21, 9, 1, 28, 20, 11, 3, 13, 12, 10, 19, 14, 22, 15, 2, 25, 16, 4, 23, 7, 17, 24, 26, 5, 27, 8, 6]	5166
NUG30	30	Óptima	[5, 12, 6, 13, 2, 21, 26, 24, 10, 9, 29, 28, 17, 1, 8, 7, 19, 25, 23, 22, 11, 16, 30, 4, 15, 18, 27, 3, 14, 20]	6124
		FLPWeb	[5, 12, 6, 13, 2, 21, 26, 24, 10, 9, 29, 28, 17, 1, 8, 7, 19, 25, 23, 22, 11, 16, 30, 4, 15, 18, 27, 3, 14, 20]	6124

7.5.6 Análisis de instancias de referencia probadas en la literatura

Como última etapa de validación de la herramienta, se ejecuta el algoritmo de recocido simulado para la solución de problemas mQAP y las instancias descritas en la Tabla 25. Se encuentra que el algoritmo implementado da soluciones factibles e incluso mejores que algunas heurísticas usadas por autores en la literatura científica; no obstante, se destaca que se desconoce cuán diferente son las soluciones óptimas halladas por los autores en la literatura, por lo que no es posible determinar un error o gap respecto al óptimo. Para obtener los resultados mostrados en la Tabla 25, es necesario utilizar las matrices que se encuentran en el Apéndice C. Matrices de Prueba.zip, además, seleccionar que las matrices de Flujo sean reindexadas, con signo positivo en la función objetivo; las matrices de tasa de cercanía con signo negativo en la función objetivo, no se deben reindexar y signo positivo en la función objetivo.

Los resultados que se comparan en la columna Enfoque, son los encontrados por Singh y Singh (2010) y Matai (2015), respectivamente. La tabulación y el resumen de los resultados se encuentran en el Apéndice F. Resultados y Tabulación Instancias Literatura.zip

Tabla 25

Tabla de resultados resumen para instancias halladas en la literatura para el mQAP

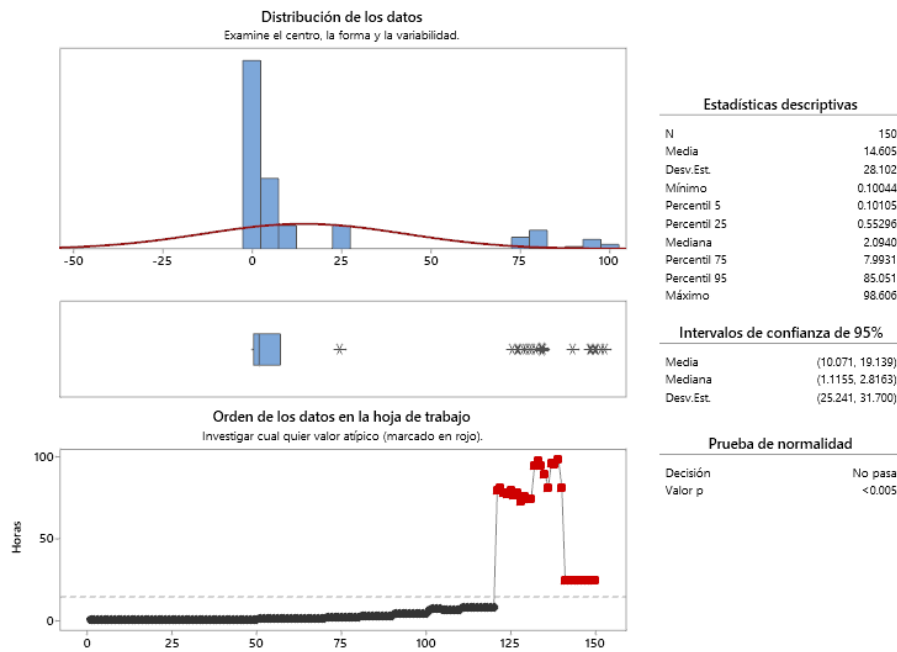
Problema	Enfoque	Método GMWM	Método SDWM	Método CRITIC M
n = 6 (Rosenblatt, 1979)	Singh y Singh	22.0076	22.2191	22.2191
	R. Matai	22.0076	22.2191	22.2191
	<i>FLPWeb</i>	22.011	22.2148	22.2148
n = 8 (Chen Y Sha, 2005)	Singh y Singh	N/A	50.0504	42.6571
	R. Matai	N/A	50.0504	42.6571
	<i>FLPWeb</i>	N/A	50.0623	42.6616
n = 8 (Dutta Y Sahu, 1982)	Singh y Singh	105.78	127.0816	127.0816
	R. Matai	52.8928	127.0816	127.0816
	<i>FLPWeb</i>	52.9156	127.0623	127.0623
n = 12 (Fortenberry Y Cox, 1985)		1050.407	1162.764	
	Singh y Singh	5	5	1162.7645
		1050.407	1162.764	
	R. Matai	5	5	1162.7645
		1050.430	1162.868	
	<i>FLPWeb</i>	7	6	1162.8686
n = 15 (C.-W. Chen, 1999)			1411.110	
	Singh y Singh	N/A	8	1411.1108
			1411.110	
	R. Matai	N/A	8	1411.1108
	<i>FLPWeb</i>	N/A	1392.245	1392.2452

Ahora bien, es importante resaltar que durante la fase de calibración se menciona que el algoritmo logra generar soluciones eficaces en tiempo de cómputo razonables; no obstante, no se ha detallado el desempeño del modelo en función del número de horas que requiere para dar una solución. En la Figura 63 se presenta un resumen estadístico descriptivo, se resalta que existe una tendencia hacia el valor de 14 horas; no obstante, en la figura se encuentran agregados los valores sin distinguir por instancias y, debido a que la solución óptima requiere de algoritmos con complejidades NP-hard según la literatura consultada, es relevante realizar el ejercicio considerando las distintas instancias. Ahora bien, la anterior afirmación toma mayor relevancia

cunado se destaca que los tiempos mínimos de cómputo son de 0.1 horas y los máximos de 98.6 horas. Sobre todo, considerando que el horizonte de las decisiones es estratégico y se espera que los diseños propuestos por la herramienta sean implementados durante meses o años.

Figura 63

Gráfica resumen de los tiempos de cómputo en horas para la solución con recocido simulado

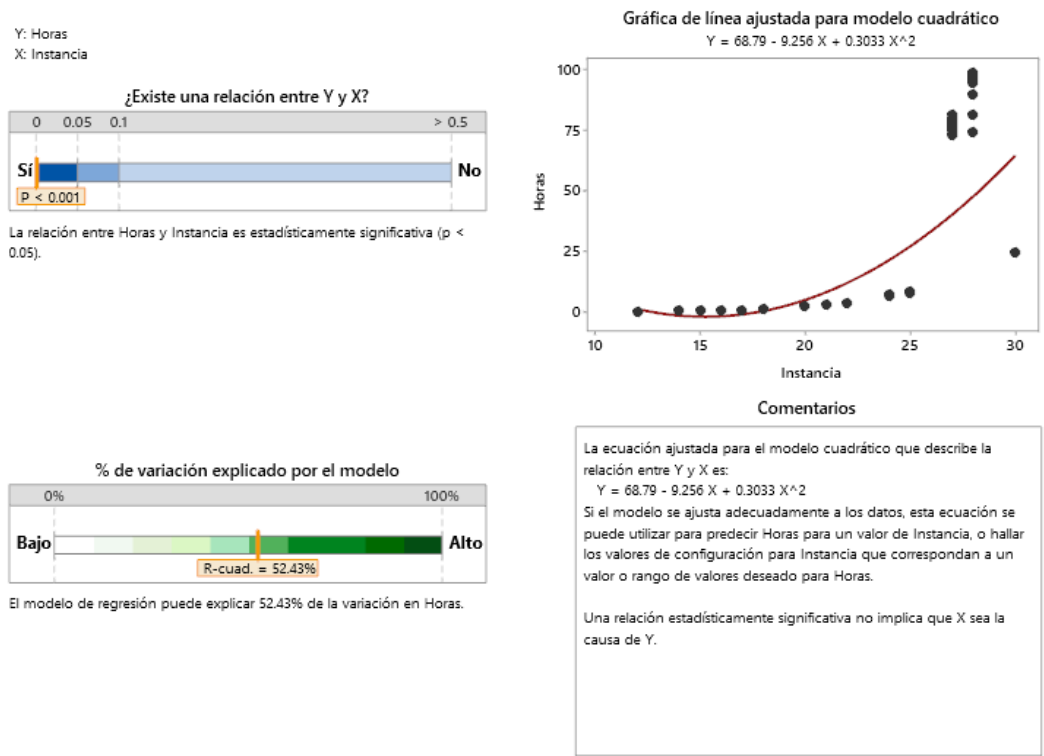


Por otra parte, se calcula el tiempo de ejecución del recocido simulado considerando el número de instalaciones como predictor. Se encuentra que el tiempo en horas puede ser modelado como $Y = 68.79 - 9.256X + 0.3033X^2$ (donde X es el número de instalaciones y Y el tiempo en horas), el modelo explica el fenómeno con un R cuadrado de 52.43%, lo que indica que existe un margen de error en ciertos escenarios. De hecho, la mayor diferencia o error se percibe en instancias de mayor tamaño, esto puede deberse a la cantidad de objetivos que deben ser normalizados y procesados de manera simultánea, es decir, dos problemas con la misma cantidad

de instancias pueden variar drásticamente según la cantidad de objetivos a ponderar. Adicionalmente, si existen ceros en las matrices objetivos se descarta el método de ponderación por gradiente lo cual acota el problema, ya que sólo se deben calcular tres métodos de ponderación. La Figura 64 contiene el resumen de resultados de la aplicación de un modelo de regresión.

Figura 64

Modelo de regresión para la estimación de tiempo de cómputo del algoritmo recocido simulado



8. Conclusiones

El problema de la distribución de planta, también conocido como problema de la localización de instalaciones FLP, ha sido ampliamente estudiado en la literatura y es un tema de gran importancia en la ingeniería industrial para la toma de decisiones y la ingeniería de sistemas para el desarrollo de algoritmos eficientes y eficaces. Una variante importante de este problema es el problema de asignación de cuadrados QAP, que se utiliza para representar la asignación de instalaciones a locaciones.

Cabe resaltar que según diversos autores el QAP es un problema NP-hard, es necesario utilizar algoritmos metaheurísticos para encontrar soluciones aceptables. Una opción ampliamente implementada es el algoritmo de recocido simulado, que se basa en el proceso de enfriamiento de metales y ha demostrado ser muy efectivo para resolver este tipo de problemas. No obstante, este tipo de soluciones es encontrada principalmente en trabajos de investigación y no en herramientas de acceso libre en el mercado. Es decir, la implementación de herramientas para solucionar el QAP es escasa.

Por otra parte, debido a la complejidad y potencia que tiene el algoritmo de recocido simulado, la simplificación del problema de toma de decisiones obtenido mediante la implementación del mQAP con ponderación de objetivos y a las características de los potenciales

usuarios (usuarios altamente especializados que puedan tomar decisiones a largo plazo o investigadores en el área). El prototipo de herramienta desarrollado FLPWeb es sencillo, ya que su poder se centra en la algoritmia y no en elementos como la integración de APIs, manejo de bases de datos relacionales o no relacionales, integración de distintos frameworks para desarrollar herramientas multiplataformas, ni en características distintas a los requerimientos identificados durante la ejecución del proyecto. Estas características le permiten al usuario altamente especializado optimizar la solución del problema, sin tener que preocuparse por aspectos técnicos.

En cuanto a los resultados de las pruebas realizadas se encuentra lo siguiente: el algoritmo implementado da soluciones factibles e incluso mejores a algunas heurísticas usadas por autores tales como Singh, Singh (2010) y Matai (2015). El algoritmo implementado logra hallar las soluciones óptimas en tiempos razonables cuando se compara con la librería QAPlib.

En el sentido de la implementación de la herramienta en un público más general, cabe mencionar como potencial usuario de una herramienta que resuelva el QAP un estudiante de distintas áreas como son: ingeniería, matemática aplicada o física. No obstante, para tal fin es importante contar con un banco de ejercicios de ejemplo o materiales alojados en plataformas de acceso masivo como cursos MOOC. Lograr llegar a este tipo de usuarios puede potenciar mejoras en funcionalidades de futuras versiones de la herramienta. Como son la portabilidad, la gestión de usuario, la implementación de distintos algoritmos de solución, la inclusión de nuevos problemas de optimización, la experiencia de usuario, la accesibilidad, el banco de información, entre otros.

Por último, es importante mencionar que esta herramienta es de gran utilidad para aquellas personas interesadas en el diseño de planta y en la optimización multiobjetivo, ya que permite aplicar conocimientos teóricos y mejorar el desempeño profesional. Además, la implementación

del algoritmo de recocido simulado en la solución de problemas mQAP demuestra su eficacia y efectividad en la obtención de soluciones de alta calidad en un tiempo razonable.

9. Recomendaciones

Considerando las limitaciones establecidas durante la creación de la herramienta FLP en entorno WEB, tanto desde el punto de vista investigativo incluyendo el campo de acción, horizonte de búsqueda y enfoque de solución, así como el enfoque de desarrollo de software en el cual los requerimientos fueron obtenidos mediante orientaciones por parte del cuerpo director, se proponen las siguientes formas de mejorar la disponibilidad y efectividad de FLPWeb.

Una posible solución para mejorar la disponibilidad de herramientas para solucionar el QAP es permitir el uso de nuevos algoritmos metaheurísticos, como el algoritmo genético, búsqueda tabú, o el método de colonias de hormigas. Además, es importante evaluar aspectos como la usabilidad, accesibilidad y calidad de la herramienta, así como la experiencia de usuario en general, para garantizar su efectividad.

Otra opción es generar proyectos de práctica en docencia en áreas como ingeniería, matemática aplicada o física, con el fin de crear material de capacitación y promover el uso de estas herramientas. Lo anterior puede ocurrir en la generación de material disponible en línea como cursos MOOC, tutoriales guiados solucionando los ejemplos de clase, o talleres guiados. El impacto generado no es sólo en los estudiantes quienes podrán conocer un poco más del enfoque de aplicación de metaheurísticas para solucionar problemas de optimización no lineal, sino en los

mismos instructores o tutores quienes pueden usar la herramienta como un laboratorio de investigación en pedagogía.

Asimismo, se podría considerar la implementación de distintos modelos para solucionar el FLP, diferentes al mQAP, a fin de brindar opciones adicionales a los usuarios. No obstante, debido a la naturaleza del problema, el algoritmo de solución y la información requerida para describir el proceso de toma de decisiones en un modelo de programación matemática, esta implementación puede requerir del desarrollo de más proyectos de grado similares que puedan acceder al Código fuente, o la implementación de retos de programación en cursos de Diseño de Sistemas Productivos o cursos de Programación competitiva o desarrollo de soluciones creativas a problemas complejos.

Es importante también fortalecer en las escuelas de Estudios Industriales y Empresariales junto a la escuela de Ingeniería de Sistemas e Informática la unión para posibles proyectos de grado conjuntos, debido a que ambas comparten el enfoque de resolver problemas mediante el uso de la algoritmia, siendo este una potencial fuente de conocimiento para la Universidad Industrial de Santander.

Finalmente, se sugiere la utilización de otros supuestos para poder generar nuevos proyectos e investigaciones; esto es, utilizar distribuciones irregulares (no cuadradas como se trabajó en este proyecto), utilizar la metodología de la planificación sistemática de la distribución (SLP) propuesta por Muther (1973) para generar objetivos, utilizar una representación continua donde se las instalaciones se puedan ubicar en cualquier lugar de la planta sin sobreponerse entre ellas.

Referencias Bibliográficas

Alizon, F., Dallery, Y., Feillet, D., Y Michelon, P. (2007). Solution of a facility layout problem in a final assembly workshop using constraint programming. *INFOR*, 45(2), 65–73.

<https://doi.org/10.3138/infor.45.2.65>

Allahyari, M. Z., Y Azab, A. (2018). Mathematical modeling and multi-start search simulated annealing for unequal-area facility layout problem. *Expert Systems with Applications*, 91,

46–62. <https://doi.org/10.1016/j.eswa.2017.07.049>

Anstreicher, K. M., Y Brixius, N. W. (2001a). Solving quadratic assignment problems using convex quadratic programming relaxations. *Optimization Methods and Software*, 16(1–4),

49–68. <https://doi.org/10.1080/10556780108805828>

Anstreicher, K. M., Y Brixius, N. W. (2001b). A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming*, 89(3), 341–

357. <https://doi.org/10.1007/PL00011402>

Armour, G. C., Y Buffa, E. S. (1963). A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities. *Management Science*, 9(2), 294–309.

<https://doi.org/10.1287/mnsc.9.2.294>

- Atencio, F. N., Y Rodado, D. N. (2016). A Sule's Method initiated genetic algorithm for solving QAP formulation in facility layout design: A real world application. *Journal of Theoretical and Applied Information Technology*, 84(2), 157–196.
- Azimi, P., Y Soofi, P. (2017). An ANN-based optimization model for facility layout problem using simulation technique. *Scientia Iranica*, 24(1), 364–377.
<https://doi.org/10.24200/SCI.2017.4040>
- Banduka, N., Mladineo, M., Y Eric, M. (2017). Designing a layout using Schmigalla method combined with software tool vistable. *International Journal of Simulation Modelling*, 16(3), 375–385. [https://doi.org/10.2507/IJSIMM16\(3\)1.379](https://doi.org/10.2507/IJSIMM16(3)1.379)
- Bass, L., Clements, P., Y Kazman, R. (2012). Software Architecture in Practice. *Vasa*, 2nd, 1–426. <https://doi.org/10.1024/0301-1526.32.1.54>
- Beck, K. (1999). Extreme Programming Explained: Embrace Change. In *XP Series*.
- Beckmann, M., Y Koopmans, T. C. (1957). Assignment Problems and the Location of Economic Activities. *Econometrica*, 25(1), 53–76.
- Bibik, I. (2018). How to Kill the Scrum Monster: Quick Start to Agile Scrum Methodology and the Scrum Master Role. In *How to Kill the Scrum Monster: Quick Start to Agile Scrum Methodology and the Scrum Master Role*. <https://doi.org/10.1007/978-1-4842-3691-8>
- Bozer, Y. A., Y Wang, C.-T. (2012). A graph-pair representation and MIP-model-based heuristic for the unequal-area facility layout problem. *European Journal of Operational Research*, 218(2), 382–391. <https://doi.org/10.1016/j.ejor.2011.10.052>

- Bozorgi, N., Abedzadeh, M., Y Zeinali, M. (2015). Tabu search heuristic for efficiency of dynamic facility layout problem. *International Journal of Advanced Manufacturing Technology*, 77(1–4). <https://doi.org/10.1007/s00170-014-6460-9>
- Brungger, A., Marzetta, A., Clausen, J., Y Perregaard, M. (1997). Joining forces in solving large-scale quadratic assignment problems in parallel. *Proceedings 11th International Parallel Processing Symposium*, 418–427. <https://doi.org/10.1109/IPPS.1997.580936>
- BURKARD, R. E., ÇELA, E. ., RENDL, F., Y KARISCH, S. E. (1997). QAPlib. Revisado 10, 2022, from <https://www.opt.math.tugraz.at/qaplib/inst.html#NVR>
- Castillo, I., Y Westerlund, T. (2005). An ϵ -accurate model for optimal unequal-area block layout design. *Computers and Operations Research*, 32(3). [https://doi.org/10.1016/S0305-0548\(03\)00246-6](https://doi.org/10.1016/S0305-0548(03)00246-6)
- Chan, W. K., Y Malmborg, C. J. (2013). On the effectiveness of Monte Carlo simulation and heuristic search for solving large-scale block layout problems. *International Journal of Production Research*, 51(14), 4258–4272. <https://doi.org/10.1080/00207543.2012.762129>
- Chang, M.-S., Y Ku, T.-C. (2013). A slicing tree representation and QCP-model-based heuristic algorithm for the unequal-area block facility layout problem. *Mathematical Problems in Engineering*, 2013. <https://doi.org/10.1155/2013/853586>
- Chen *, C.-W., Y Sha, D. Y. (2005). Heuristic approach for solving the multi-objective facility layout problem. *International Journal of Production Research*, 43(21), 4493–4507. <https://doi.org/10.1080/00207540500056383>

- Chen, C.-W. (1999). A design approach to the multi-objective facility layout problem. *International Journal of Production Research*, 37(5), 1175–1196.
<https://doi.org/10.1080/002075499191463>
- Chen, Y., Jiang, Y., Wahab, M. I. M., Y Long, X. (2015). The facility layout problem in non-rectangular logistics parks with split lines. *Expert Systems with Applications*, 42(21), 7768–7780. <https://doi.org/10.1016/j.eswa.2015.06.009>
- Chwif, L., Pereira Barretto, M. R., Y Moscato, L. A. (1998). A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36(1–2), 125–132.
[https://doi.org/10.1016/S0166-3615\(97\)00106-1](https://doi.org/10.1016/S0166-3615(97)00106-1)
- Clausen, J., Y Perregaard, M. (1997). Solving Large Quadratic Assignment Problems in Parallel. *Computational Optimization and Applications*, 8(2), 111–127.
<https://doi.org/10.1023/A:1008696503659>
- Cockburn, A. (2004). Crystal clear a human-powered methodology for small teams. In *Addison-Wesley*.
- Cooper, L. (1963). Location-Allocation Problems. *Operations Research*, 11(3), 331–343.
<https://doi.org/10.1287/opre.11.3.331>
- Derakhshan Asl, A., Wong, K. Y., Y Tiwari, M. K. (2015). Unequal-area stochastic facility layout problems: Solutions using improved covariance matrix adaptation evolution strategy, particle swarm optimisation, and genetic algorithm. *International Journal of Production Research*, 54(3), 799–823. <https://doi.org/10.1080/00207543.2015.1070217>

- Diakoulaki, D., Mavrotas, G., Y Papayannakis, L. (1995). Determining objective weights in multiple criteria problems: The critic method. *Computers Y Operations Research*, 22(7), 763–770. [https://doi.org/10.1016/0305-0548\(94\)00059-H](https://doi.org/10.1016/0305-0548(94)00059-H)
- Dooley, J. F. (2017). Software Development, Design and Coding. In *Software Development, Design and Coding*. <https://doi.org/10.1007/978-1-4842-3153-1>
- Drira, A., Pierreval, H., Y Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2), 255–267. <https://doi.org/10.1016/j.arcontrol.2007.04.001>
- Dutta, K. N., Y Sahu, S. (1982). A multigoal heuristic for facilities design problems: MUGHAL. *International Journal of Production Research*, 20(2), 147–154. <https://doi.org/10.1080/00207548208947756>
- Farhi, O. (2017). Reactive Programming with Angular and ngrx. In *Reactive Programming with Angular and ngrx*. <https://doi.org/10.1007/978-1-4842-2620-9>
- Ficko, M., Y Palcic, I. (2013). Designing a layout using the modified triangle method, and genetic algorithms. *International Journal of Simulation Modelling*, 12(4), 237–251. [https://doi.org/10.2507/IJSIMM12\(4\)3.244](https://doi.org/10.2507/IJSIMM12(4)3.244)
- Fon, C. W., Y Wong, K. Y. (2010). Investigating the performance of bees algorithm in solving quadratic assignment problems. *International Journal of Operational Research*, 9(3), 241–257. <https://doi.org/10.1504/IJOR.2010.035519>
- Forghani, K., Khamseh, A. A., Y Mohammadi, M. (2012). Integrated quadratic assignment and continuous facility layout problem. *International Journal of Industrial Engineering Computations*, 3(5), 787–806. <https://doi.org/10.5267/j.ijiec.2012.07.001>

- Fortenberry, J. C., Y Cox, J. F. (1985). Multiple criteria approach to the facilities layout problem. *International Journal of Production Research*, 23(4), 773–782.
<https://doi.org/10.1080/00207548508904746>
- Friedrich, C., Klausnitzer, A., Y Lasch, R. (2018). Integrated slicing tree approach for solving the facility layout problem with input and output locations based on contour distance. *European Journal of Operational Research*, 270(3), 837–851.
<https://doi.org/10.1016/j.ejor.2018.01.001>
- García-Hernández, L., Pérez-Ortiz, M., Araúzo-Azofra, A., Salas-Morera, L., Y Hervás-Martínez, C. (2014). An evolutionary neural system for incorporating expert knowledge into the UA-FLP. *Neurocomputing*, 135, 69–78. <https://doi.org/10.1016/j.neucom.2013.01.068>
- Garcia-Hernandez, L., Pierreval, H., Salas-Morera, L., Y Arauzo-Azofra, A. (2013). Handling qualitative aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm. *Applied Soft Computing Journal*, 13(4), 1718–1727.
<https://doi.org/10.1016/j.asoc.2013.01.003>
- Garrison, E. (2018). A History of Engineering and Technology Artful Methods. In *A History of Engineering and Technology Artful Methods*. <https://doi.org/10.1201/9780203751749>
- Hales, L., Y Muther, R. (2015). The Systematic Layout Planning Pattern. In *Systematic Layout Planning*.
- Harmonosky, C. M., Y Tothero, G. K. (1992). A multi-factor plant layout methodology. *International Journal of Production Research*, 30(8), 1773–1789.
<https://doi.org/10.1080/00207549208948121>

- Hernández Gress, E. S., Mora-Vargas, J., Herrera Del Canto, L. E., Y Díaz-Santillán, E. (2011). A genetic algorithm for optimal unequal-area block layout design. *International Journal of Production Research*, 49(8), 2183–2195. <https://doi.org/10.1080/00207540903130868>
- Hosseini-Nasab, H., Y Emami, L. (2013). A hybrid particle swarm optimisation for dynamic facility layout problem. *International Journal of Production Research*, 51(14). <https://doi.org/10.1080/00207543.2013.774486>
- Hourieh, A. (2008). Learning Website Development with Django. In *Birmingham: Packt Publishing Ltd.*
- IEEE. (1990). IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos. CA: *IEEE Computer Society*, 610.12-199. <https://doi.org/10.1109/IEEESTD.1990.101064>
- Janes, A., Y Succi, G. (2014). Lean software development in action. In *Lean Software Development in Action* (Vol. 9783642005039). <https://doi.org/10.1007/978-3-642-00503-9>
- Jiang, S., Y Nee, A. Y. C. (2013). A novel facility layout planning and optimization methodology. *CIRP Annals - Manufacturing Technology*, 62(1), 483–486. <https://doi.org/10.1016/j.cirp.2013.03.133>
- Jokar, M. R. A., Y Sangchooli, A. S. (2011). Constructing a block layout by face area. *International Journal of Advanced Manufacturing Technology*, 54(5–8), 801–809. <https://doi.org/10.1007/s00170-010-2960-4>
- Kang, S., Y Chae, J. (2017). Harmony search for the layout design of an unequal area facility. *Expert Systems with Applications*, 79, 269–281. <https://doi.org/10.1016/j.eswa.2017.02.047>

- Kapoor, V., Y Tak, S. S. (2015). A multicriteria clustering approach to facility layout generation. *International Journal of Multicriteria Decision Making*, 5(1/2), 59–86.
<https://doi.org/10.1504/IJMCDM.2015.067945>
- Kaveh, A., Y Sharafi, P. (2011). Charged system search algorithm for minimax and Minisum facility layout problems. *Asian Journal of Civil Engineering (Building and Housing)*, 12(6), 703–718.
- Khilwani, N., Shankar, R., Y Tiwari, M. K. (2008). Facility layout problem: An approach based on a group decision-making system and psychoclonal algorithm. *International Journal of Production Research*, 46(4), 895–927. <https://doi.org/10.1080/00207540600943993>
- Komarudin, Y Wong, K. Y. (2010). Applying Ant System for solving Unequal Area Facility Layout Problems. *European Journal of Operational Research*, 202(3), 730–746.
<https://doi.org/10.1016/j.ejor.2009.06.016>
- Kulkarni, M. H., Bhatwadekar, S. G., Y Thakur, H. M. (2015). A literature review of facility planning and plant layouts. *International Journal of Engineering Sciences Y Research Technology*, 4(3), 35–42.
- Kulturel-Konak, S., Y Konak, A. (2013). Linear programming based genetic algorithm for the unequal area facility layout problem. *International Journal of Production Research*, 51(14), 4302–4324. <https://doi.org/10.1080/00207543.2013.774481>
- Kusiak, A., Y Heragu, S. S. (1987). The facility layout problem. In *European Journal of Operational Research* (Vol. 29, Issue 3, pp. 229–251). [https://doi.org/10.1016/0377-2217\(87\)90238-4](https://doi.org/10.1016/0377-2217(87)90238-4)

- Labs, C. (2022). Scribe. Recuperado de <https://scribeshow.com/home>
- Latifi, S. E. S. E. E., Mohammadi, E., Y Khakzad, N. (2017). Process plant layout optimization with uncertainty and considering risk. *Computers and Chemical Engineering*, 106, 224–242. <https://doi.org/10.1016/j.compchemeng.2017.05.022>
- Laudon, K. C., Y Laudon, J. P. (2012). *Sistemas de información gerencial* (L. M. Cruz Castillo, Ed.; 12 ed). Pearson Educación.
- Leno, I. J., Sankar, S. S., Y Ponnambalam, S. G. (2018). MIP model and elitist strategy hybrid GA–SA algorithm for layout design. *Journal of Intelligent Manufacturing*, 29(2), 369–387. <https://doi.org/10.1007/s10845-015-1113-x>
- Li, D., Chen, Z., Y Zhao, R. (2013). An adaptive genetic algorithm for digital facility layout optimization under the multi-species and variable-batch production mode. *Open Automation and Control Systems Journal*, 5(1), 7–14. <https://doi.org/10.2174/18744444301305010007>
- Li, J., Y Smith, A. E. (2018). Block layout for attraction-based enterprises. *European Journal of Operational Research*, 266(3), 1100–1112. <https://doi.org/10.1016/j.ejor.2017.10.028>
- Liu, J., Wang, D., Y Yan, X. (2021). Tabu search algorithm for dynamic facility layout problem. *Huazhong Keji Daxue Xuebao (Ziran Kexue Ban)/Journal of Huazhong University of Science and Technology (Natural Science Edition)*, 49(2). <https://doi.org/10.13245/j.hust.210206>
- Liu, J., Zhang, H., He, K., Y Jiang, S. (2018). Multi-objective particle swarm optimization algorithm based on objective space division for the unequal-area facility layout problem.

Expert Systems with Applications, 102, 179–192.

<https://doi.org/10.1016/j.eswa.2018.02.035>

Liu, Q., Y Meller, R. D. (2007). A sequence-pair representation and MIP-model-based heuristic for the facility layout problem with rectangular departments. *IIE Transactions (Institute of Industrial Engineers)*, 39(4), 377–394. <https://doi.org/10.1080/07408170600844108>

Martinez-Gomez, J., Nápoles-Rivera, F., Ponce-Ortega, J. M., Serna-González, M., Y El-Halwagi, M. M. (2014a). Siting optimization of facility and unit relocation with the simultaneous consideration of economic and safety issues. *Industrial and Engineering Chemistry Research*, 53(10), 3950–3958. <https://doi.org/10.1021/ie402242u>

Martinez-Gomez, J., Nápoles-Rivera, F., Ponce-Ortega, J. M., Serna-González, M., Y El-Halwagi, M. M. (2014b). Siting optimization of facility and unit relocation with the simultaneous consideration of economic and safety issues. *Industrial and Engineering Chemistry Research*, 53(10), 3950–3958. <https://doi.org/10.1021/ie402242u>

Masrom, S., Abidin, S. Z. Z., Omar, N., Rahman, A. S. A., Y Rizman, Z. I. (2017). Dynamic parameterizations of particle swarm optimization and genetic algorithm for facility layout problem. *ARNP Journal of Engineering and Applied Sciences*, 12(10).

Matai, R. (2015). Solving multi objective facility layout problem by modified simulated annealing. *Applied Mathematics and Computation*, 261, 302–311.

<https://doi.org/10.1016/j.amc.2015.03.107>

Matai, R., Singh, S. P., Y Mittal, M. L. (2012). Non-greedy systematic neighbourhood search heuristic for multi-objective facility layout problem. *International Journal of Services and Operations Management*, 12(1), 118–138. <https://doi.org/10.1504/IJSOM.2012.046677>

- Matai, R., Singh, S. P., Y Mittal, M. L. (2013a). A non-greedy systematic neighbourhood search heuristic for solving facility layout problem. *International Journal of Advanced Manufacturing Technology*, 68(5–8), 1665–1675. <https://doi.org/10.1007/s00170-013-4965-2>
- Matai, R., Singh, S. P., Y Mittal, M. L. (2013b). Modified simulated annealing based approach for multi objective facility layout problem. *International Journal of Production Research*, 51(14), 4273–4288. <https://doi.org/10.1080/00207543.2013.765078>
- McKendall, A. R., Y Liu, W. H. (2012). New Tabu search heuristics for the dynamic facility layout problem. *International Journal of Production Research*, 50(3). <https://doi.org/10.1080/00207543.2010.545446>
- mdn web docs. (2022, December 12). *JavaScript | MDN*. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Mell, P., Y Grance, T. (2011). The NIST definition of cloud computing. *NIST Special Publication*, 145, 7. <https://doi.org/10.1136/emj.2010.096966>
- Meller, R. D., Y Bozer, Y. A. (1997). Alternative approaches to solve the multi-floor facility layout problem. *Journal of Manufacturing Systems*, 16(3), 192–203. [https://doi.org/10.1016/S0278-6125\(97\)88887-5](https://doi.org/10.1016/S0278-6125(97)88887-5)
- Mohamadi, A., Ebrahimnejad, S., Y Tavakkoli-Moghaddam, R. (2018). A novel two-stage approach for solving a bi-objective facility layout problem. *International Journal of Operational Research*, 31(1), 49–87. <https://doi.org/10.1504/IJOR.2018.088557>

- Moore, J. M. (1974). Computer aided facilities design: An international survey. *International Journal of Production Research*, 12(1), 21–44. <https://doi.org/10.1080/00207547408919538>
- Muther, R. (1973). *Systematic Layout Planning* (2 ed.). Cahners Books.
- Navidi, H., Bashiri, M., Y Bidgoli, M. M. (2012). A heuristic approach on the facility layout problem based on game theory. *International Journal of Production Research*, 50(6), 1512–1527. <https://doi.org/10.1080/00207543.2010.550638>
- Niederst, J. (2012). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web ... - Jennifer Niederst Robbins - Google Libros* (Simon St. Laurent, Ed.; 4th ed., Vol. 1). https://books.google.com.co/books?hl=esYlr=Yid=A-tltyafYmECYoi=fndYpg=PR11Ydq=html+cssYots=J6KHdN0CFRYSig=f5FyHpFgcaTWo8lMplt32rmps5EYredir_esc=y#v=onepageYq=html%20cssYf=false
- Nourelfath, M., Nahas, N., Y Montreuil, B. (2007). Coupling ant colony optimization and the extended great. *Engineering Optimization*, 39(8), 953–968. <https://doi.org/10.1080/03052150701551461>
- Nugent, C. E., Vollmann, T. E., Y Ruml, J. (1968). An Experimental Comparison of Techniques for the Assignment of Facilities to Locations. *Operations Research*, 16(1), 150–173. <https://doi.org/10.1287/opre.16.1.150>
- Palomo-Romero, J. M., Salas-Morera, L., Y García-Hernández, L. (2017). An island model genetic algorithm for unequal area facility layout problems. *Expert Systems with Applications*, 68, 151–162. <https://doi.org/10.1016/j.eswa.2016.10.004>

- Paper, C. (2014). *A COMPARISON OF THREE NATURE-INSPIRED METAHEURISTICS FOR THE FACILITY LAYOUT PROBLEM* | Sinem Buyuksaatci - Academia.edu. June.
- Peng, Y., Zeng, T., Fan, L., Han, Y., Xia, B., Y Wang, X. (2018). An Improved Genetic Algorithm Based Robust Approach for Stochastic Dynamic Facility Layout Problem. *Discrete Dynamics in Nature and Society*, 2018. <https://doi.org/10.1155/2018/1529058>
- php.net. (2023, January 4). *PHP: ¿Qué es PHP?* . <https://www.php.net/manual/es/intro-what-is.php>
- Pourvaziri, H., Y Naderi, B. (2014). A hybrid multi-population genetic algorithm for the dynamic facility layout problem. *Applied Soft Computing Journal*, 24. <https://doi.org/10.1016/j.asoc.2014.06.051>
- Pressman, R., Y Troya, J. (2007). Ingeniería del software. In *CITEG Revista Arbitrada* (Vol. 1, Issue 2).
- Railway Corp. (2023). Railway. Retrieved from <https://railway.app/>
- Ramkumar, A. S., Ponnambalam, S. G., Y Jawahar, N. (2009a). A new iterated fast local search heuristic for solving QAP formulation in facility layout design. *Robotics and Computer-Integrated Manufacturing*, 25(3), 620–629. <https://doi.org/10.1016/j.rcim.2008.03.022>
- Ramkumar, A. S., Ponnambalam, S. G., Y Jawahar, N. (2009b). A population-based hybrid ant system for quadratic assignment formulations in facility layout design. *International Journal of Advanced Manufacturing Technology*, 44(5–6), 548–558. <https://doi.org/10.1007/s00170-008-1849-y>

- Rosenblatt, M. J. (1979). The facilities layout problem: a multi-goal approach. *International Journal of Production Research*, 17(4), 323–332.
<https://doi.org/10.1080/00207547908919617>
- Ruby, P., Thomas, D., Y Hunt, A. (2009). Programming Ruby 1.9: The Pragmatic Programmers' Guide. In *Networks*.
- Sahni, S., Y Gonzalez, T. (1976). P-Complete Approximation Problems. *Journal of the ACM*, 23(3), 555–565. <https://doi.org/10.1145/321958.321975>
- Samanta, S., Philip, D., Y Chakraborty, S. (2018a). Bi-objective dependent location quadratic assignment problem: Formulation and solution using a modified artificial bee colony algorithm. *Computers and Industrial Engineering*, 121, 8–26.
<https://doi.org/10.1016/j.cie.2018.05.018>
- Samanta, S., Philip, D., Y Chakraborty, S. (2018b). Bi-objective dependent location quadratic assignment problem: Formulation and solution using a modified artificial bee colony algorithm. *Computers and Industrial Engineering*, 121, 8–26.
<https://doi.org/10.1016/j.cie.2018.05.018>
- Sangwan, K. S. (2010). Fuzziness in materials flow and plant layout. In K. C. Y Y. M. (Eds.), *Studies in Fuzziness and Soft Computing* (Vol. 252). https://doi.org/10.1007/978-3-642-12052-7_15
- Scholz, D., Petrick, A., Y Domschke, W. (2009). STaTS: A Slicing Tree and Tabu Search based heuristic for the unequal area facility layout problem. *European Journal of Operational Research*, 197(1), 166–178. <https://doi.org/10.1016/j.ejor.2008.06.028>

- Search Technology, Inc. (2018). *The Vantage Point* (No. 11).
- Shariatzadeh, N., Sivard, G., Y Chen, D. (2012). Software evaluation criteria for rapid factory layout planning, design and simulation. *Procedia CIRP*, 3(1), 299–304.
<https://doi.org/10.1016/j.procir.2012.07.052>
- Sherrell, L. (2013). Waterfall Model. In *Encyclopedia of Sciences and Religions*.
https://doi.org/10.1007/978-1-4020-8265-8_200285
- Sikaroudi, A. M. E., Y Shahanaghi, K. (2016). Facility layout by collision detection and force exertion heuristics. *Journal of Manufacturing Systems*, 41, 21–30.
<https://doi.org/10.1016/j.jmsy.2016.07.001>
- Singh, S. P., Y Singh, V. K. (2010). An improved heuristic approach for multi-objective facility layout problem. *International Journal of Production Research*, 48(4), 1171–1194.
<https://doi.org/10.1080/00207540802534731>
- Singh, S. P., Y Singh, V. K. (2011). Three-level AHP-based heuristic approach for a multi-objective facility layout problem. *International Journal of Production Research*, 49(4), 1105–1125. <https://doi.org/10.1080/00207540903536148>
- Solimanpur, M., Y Jafari, A. (2008). Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm. *Computers and Industrial Engineering*, 55(3), 606–619. <https://doi.org/10.1016/j.cie.2008.01.018>
- Taghavi, A., Y Murat, A. (2011). A heuristic procedure for the integrated facility layout design and flow assignment problem. *Computers and Industrial Engineering*, 61(1), 55–63.
<https://doi.org/10.1016/j.cie.2011.02.011>

- Talbi, E. G. (2009). Metaheuristics: From Design to Implementation. In *Metaheuristics: From Design to Implementation*. <https://doi.org/10.1002/9780470496916>
- Tasadduq, I. A. A., Imam, M. H. H., Ahmad, A., Y Tasadduq IA, Imam MH, A. A. (2015). A hybrid algorithm for optimising facility layout. *South African Journal of Industrial Engineering*, 26(1), 120–134. <https://doi.org/10.7166/26-1-1013>
- Tompkins, J. A., White, J. A., Bozer, Y. A., Y Tanchoco, J. M. A. (2011). *Planeación de Instalaciones* (S. R. Cervantes González, Ed.; 4th ed.). Cengage Learning Editores.
- Tong, J., van Luttervelt, C. A., Wedman, F. W., Y Bikker, H. (2003). Computer-guided system for design of factories. *International Journal of Computer Integrated Manufacturing*, 16(7–8). <https://doi.org/10.1080/0951192031000115705>
- Turgay, S. (2018). Multi objective simulated annealing approach for facility layout design. *International Journal of Mathematical, Engineering and Management Sciences*, 3(4), 365–380.
- Tuzkaya, G., Gülsün, B., Tuzkaya, U. R., Onut, S., Y Bildik, E. (2013). A comparative analysis of meta-heuristic approaches for facility layout design problem: A case study for an elevator manufacturer. *Journal of Intelligent Manufacturing*, 24(2), 357–372. <https://doi.org/10.1007/s10845-011-0599-0>
- Uddin, M. S. (2015). Hybrid Genetic Algorithm and Variable Neighborhood Search for Dynamic Facility Layout Problem. *Open Journal of Optimization*, 04(04). <https://doi.org/10.4236/ojop.2015.44015>

- Xiao, Y., Seo, Y., Y Seo, M. (2013). A two-step heuristic algorithm for layout design of unequal-sized facilities with input/output points. *International Journal of Production Research*, 51(14), 4200–4222. <https://doi.org/10.1080/00207543.2012.752589>
- Yang, T., Y Kuo, C. (2003). A hierarchical AHP/DEA methodology for the facilities layout design problem. *European Journal of Operational Research*, 147(1), 128–136. [https://doi.org/10.1016/S0377-2217\(02\)00251-5](https://doi.org/10.1016/S0377-2217(02)00251-5)
- Zha, S., Guo, Y., Huang, S., Wu, Q., Y Tang, P. (2020). A hybrid optimization approach for unequal-sized dynamic facility layout problems under fuzzy random demands. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 234(3). <https://doi.org/10.1177/0954405419883046>
- Zhao, Y., Y Wallace, S. W. S. W. S. W. (2014). Integrated facility layout design and flow assignment problem under uncertainty. *INFORMS Journal on Computing*, 26(4), 1–11. <https://doi.org/10.1287/ijoc.2014.0599>