

Software para la optimización de redes de distribución de agua potable

“JHawanet Framework”

Gabriel Sanhueza

Defensa de Título
Universidad de Talca

Profesores Guías
Jimmy Gutiérrez Bahamondes
Daniel Mora Melia

Agosto, 2020



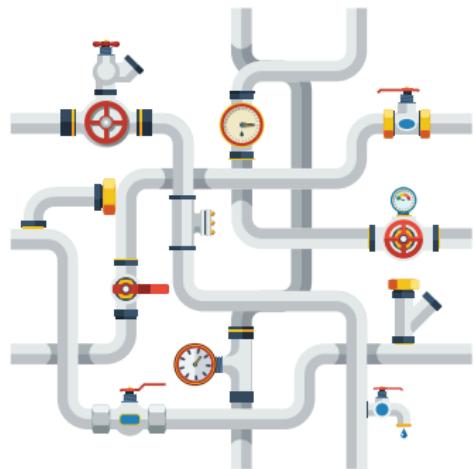
Contexto

- Escasez de agua.
- Sistemas que deben estar las 24 horas del día activo.
- Múltiples criterios para optimizar los sistemas de distribución.
- Las redes de agua (RDA) involucran dos tipos de problemas:
 - Problema de diseño.
 - Problema de operación.



Problema

- No se cuenta con las suficientes herramientas y tiempo para la correcta gestión de las redes de distribución de agua potable.
- El escoger las especificaciones es una tarea difícil debido a que hay que evaluar el rendimiento general del sistema.

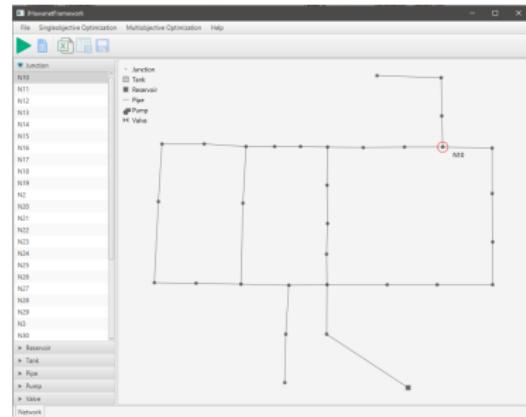


Propuesta de solución

Aplicación de escritorio extensible que permite optimizar procesos de diseño y operación en RDA.

Problemas implementados:

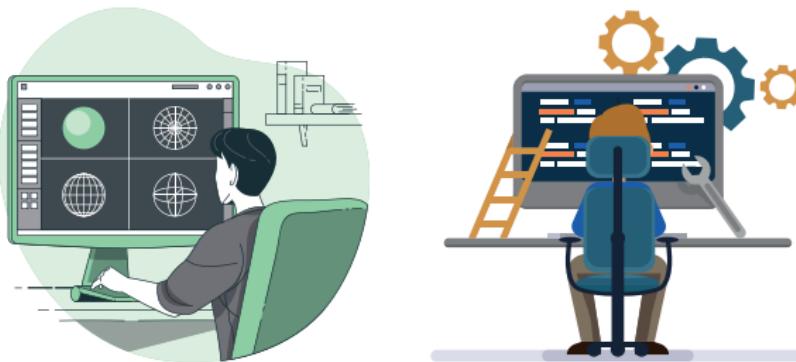
- Problema de diseño de RDA basado en el costo de tuberías.
- Problema de operación basado en el Régimen de bombeo.



Tipos de Usuarios

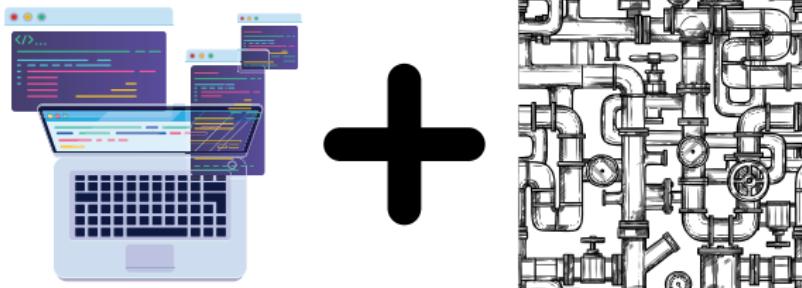
Se contemplan dos tipos de usuarios:

- Usuarios que trabajan directamente con la aplicación gráfica.
- Usuarios que utilizan la estructura para acoplar nuevos problemas desde el punto de vista computacional.



Objetivo general

- Diseñar y desarrollar una aplicación extensible de escritorio para optimizar el diseño y operación de una red de distribución de agua.



Objetivos específicos

- ① Diseñar software orientado a la optimización de RDA basado en la arquitectura lógica del framework multiobjetivo Jmetal.
- ② Implementar un algoritmo metaheurístico de optimización monoobjetivo para aplicar al problema de diseño de RDA.
- ③ Implementar un algoritmo metaheurístico de optimización multiobjetivo para aplicar al problema de Régimen de bombeo en RDA.
- ④ Diseñar e implementar la interfaz gráfica del sistema de optimización de redes de agua potable desarrollado durante este proyecto.

Iterativa e incremental

Fases:

- Análisis
- Diseño
- Implementación
- Pruebas

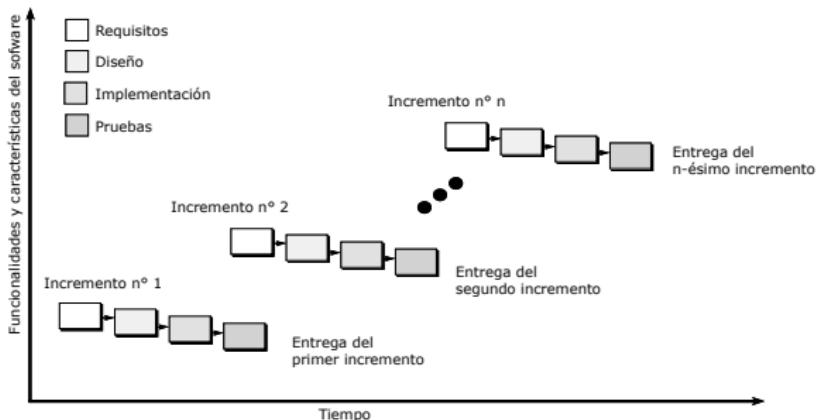


Figura: Metodología iterativa e incremental Pressman (2009)

Planificación

- **Iteración 1:** Requisitos y Arquitectura.
- **Iteración 2:** Problema monoobjetivo y Algoritmo Genético (GA).
- **Iteración 3:** Interfaz gráfica (GUI).
- **Iteración 4:** Problema multiobjetivo y Algoritmo Non-Dominated Sorting Genetic Algorithm II (NSGAII).
- **Iteración 5:** Experimentos y simulación hidráulica.
- **Iteración 6:** Afinación de detalles.

Tecnología utilizada



Lenguaje de
Programación



Interfaz Gráfica



Simulaciones Hidráulicas

Requisitos

- Captura, priorización y especificación formal de los requisitos.
- 32 Requisitos de usuario.
- Documento de especificación formal de requisitos.

| | |
|---|--|
| RU005 – Exportar los resultados de los algoritmos aplicados en dos archivos, uno para las variables y otro para los objetivos. | |
| Descripción: | Se deben poder exportar las soluciones generadas por la ejecución de los algoritmos sobre un problema a un conjunto de archivos. Específicamente, serían 2 archivos. El primer archivo debe guardar las variables de las soluciones mientras que el segundo archivo los objetivos de las soluciones. |
| Fuente: | Jimmy Gutiérrez |
| Prioridad: | Moderada |
| Estabilidad: | Intransable |
| Fecha de actualización: | 09/09/2019 |
| Estado: | Cumple |
| Incremento: | 3 |
| Tipo: | Funcional |

Diseño

- Diseñar los módulos de la aplicación y la interacción entre ellos.
- Documento de diseño.

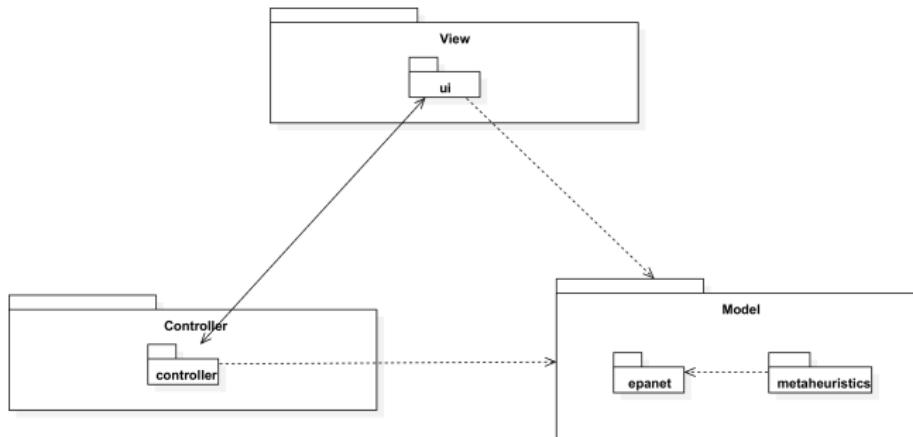


Figura: Arquitectura Lógica Modelo-Vista-Controlador

Implementación

Java Reflection y Java Annotation

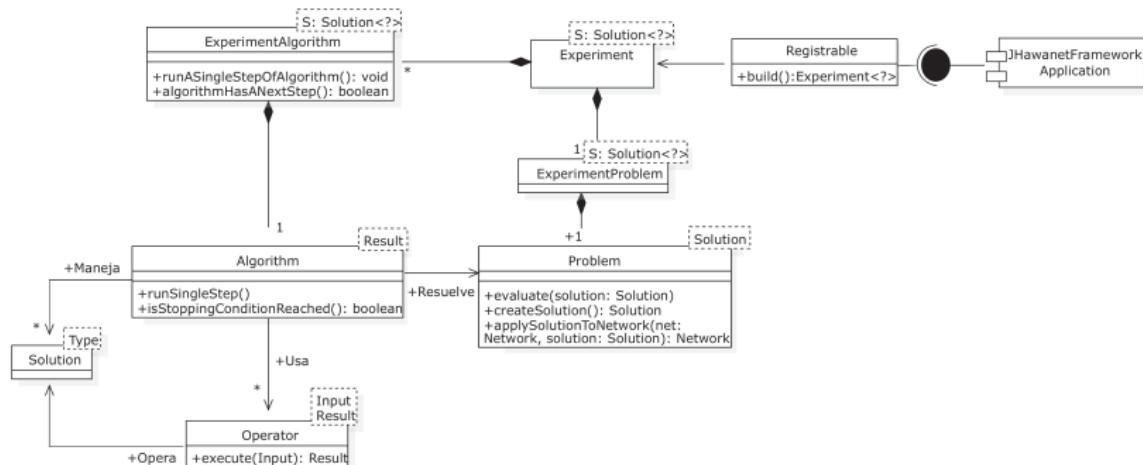


Figura: Diagrama de relaciones entre la aplicación y el módulo de metaheurísticas

Implementación

Problema de diseño de RDA basado en el costo de tuberías Pereyra et al. (2017).

Objetivo

$$\text{Costo de inversión} = \sum_{i=1}^N (C_i \times D_i \times L_i)$$

Implementación

Problema de operación basado en el Régimen de bombeo Gutiérrez-Bahamondes et al. (2019).

Objetivo 1

$$C_E(S) = \sum_{n=1}^{NP} \sum_{t=0}^{NT-1} (P_c(t) \times E_c(n, t) \times S(n, t))$$

Consumo energético

$$E_c(n, t) = \frac{10^{-3} \times \gamma \times Q(n, t) \times h(n, t)}{e(n, t)}$$

Objetivo 2

$$C_N(S) = \sum_{n=1}^{NP} \sum_{t=0}^{NT-1} r_t$$

Implementación

Algoritmo Genético

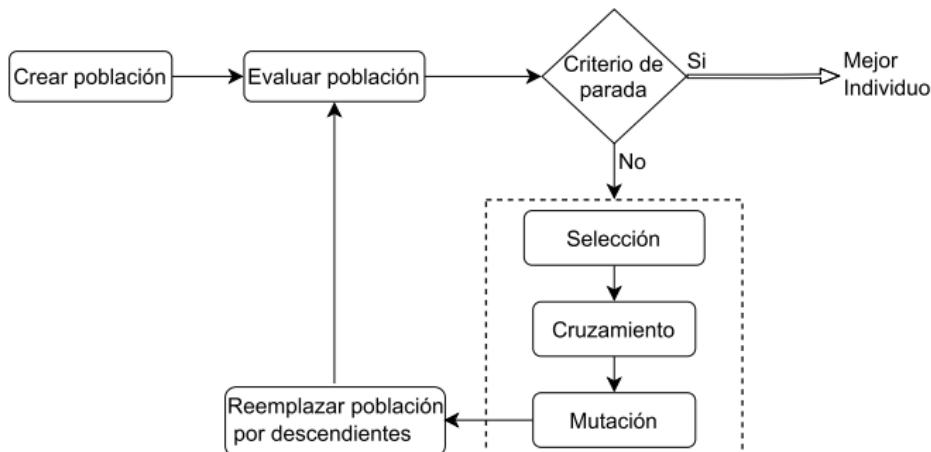


Figura: Procedimiento Algoritmo Genético Bonelli and Begliardo (2016)

Implementación

Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

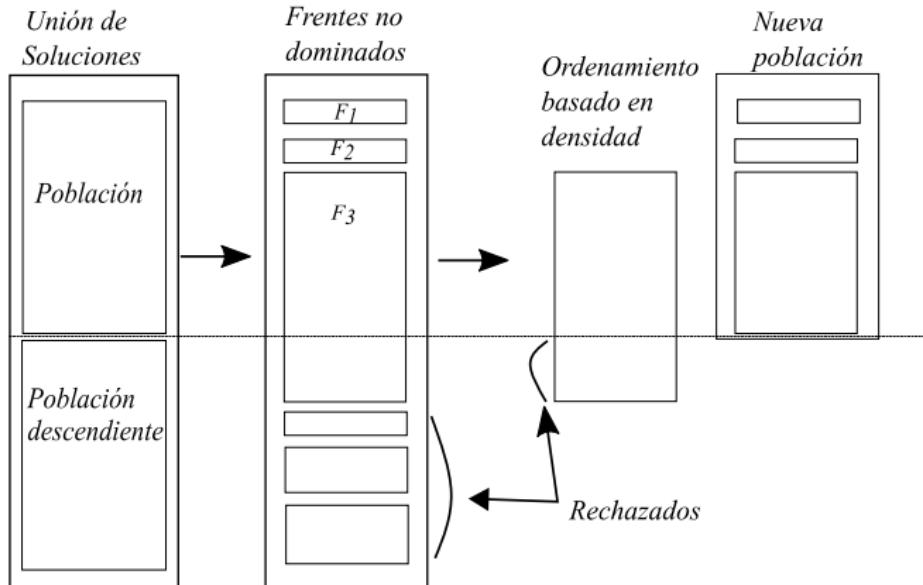


Figura: Procedimiento NSGAII Deb et al. (2002)

Introducción

Metodología de desarrollo

Desarrollo

Evaluación

Conclusiones y Trabajos Futuros

Requisitos

Diseño

Implementación

Pruebas

Implementación

Pruebas

- Realización de pruebas y su especificación formal.
- 39 Pruebas automatizadas.
- 16 Pruebas manuales.
- Documento de especificación formal de pruebas.

| | |
|-----------------------|---|
| Test ID: | AT009 |
| Título: | Número máximo de evaluaciones no válido. |
| Característica: | Validar parámetro <i>maxEvaluations</i> . |
| Objetivo: | Validar que el parámetro <i>maxEvaluations</i> no sea negativo. Si el parámetro es negativo debe lanzarse una excepción. |
| Configuración: | Instancia de la clase <i>GeneticAlgorithm</i> inicializada. |
| Datos de prueba: | Cualquier entero menor que 0. |
| Acciones de prueba: | Llamar al método <i>setMaxEvaluations</i> con un argumento negativo. |
| Resultados esperados: | Una excepción. |

Figura: Ejemplo de especificación de un caso de prueba

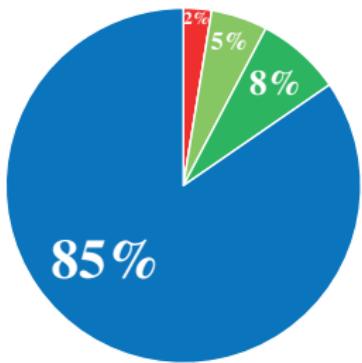
Diseño del caso

- Elección del caso
 - *La aplicación desarrollada*
- Objetivos de la investigación
 - *¿Cómo el sistema desarrollado trabaja en la práctica?*
- Características a evaluar
 - Funcionalidad
 - Usabilidad
 - Utilidad



Análisis de datos

Funcionalidad



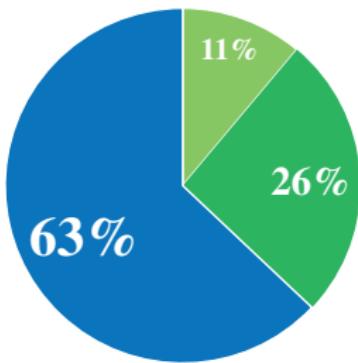
Usabilidad

$$\bar{x}_{SUS} = 79,17$$

$$s_{SUS} = 9,46$$

| SUS Score | Calificación |
|-----------|--------------|
| > 80,3 | Excelente |
| 68 - 80,3 | Buena |
| 68 | Regular |
| 51 - 68 | Mala |
| < 51 | Terrible |

Utilidad



- Totalmente de acuerdo
- Totalmente en desacuerdo

- De acuerdo
- En desacuerdo

- Ni de acuerdo ni en desacuerdo

Figura: Resultados de la recolección de datos

Conclusiones

- Software para la optimización de los procesos de diseño y operación de RDA.
- Es posible extender el sistema para mejorar su uso con diferentes algoritmos y diferentes problemas.
- Todos los objetivos establecidos fueron cumplidos satisfactoriamente.
- Los criterios de funcionalidad, usabilidad y utilidad fueron cumplidos, concluyendo que el sistema trabaja bien en la práctica.

Trabajos Futuros

- Mejorar el formato de salida.
 - Recortar el número de decimales utilizados para mostrar los resultados.
 - Permitir exportar la imagen de la red.
 - Guardar las imágenes exportadas en un formato vectorial.
 - Permitir agregar fórmulas LaTeX en la descripción del algoritmo.
- Agregar nuevos algoritmos, operadores y problemas.
- Permitir comparar algoritmos utilizados para resolver un problema.