



**UNIVERSIDAD DE TALCA**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Herramienta para la optimización de redes de  
distribución de agua potable**

**GABRIEL GONZALO ALEXANDER SANHUEZA FUENTES**

Profesor Guía: JIMMY GUTIERREZ BAHAMONDES

Memoria para optar al título de  
Ingeniero Civil en Computación

Curicó – Chile  
mes, año



**UNIVERSIDAD DE TALCA**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

## **Herramienta para la optimización de redes de distribución de agua potable**

**GABRIEL GONZALO ALEXANDER SANHUEZA FUENTES**

Profesor Guía: JIMMY GUTIERREZ BAHAMONDES

Profesor Informante: PROFESOR INFORMANTE 1

Profesor Informante: PROFESOR INFORMANTE 2

Memoria para optar al título de  
Ingeniero Civil en Computación

El presente documento fue calificado con nota: \_\_\_\_\_

Curicó – Chile

mes, año

*Dedicado a ...*

## AGRADECIMIENTOS

Agradecimientos a ...

## TABLA DE CONTENIDOS

	página
<b>Dedicatoria</b>	<b>I</b>
<b>Agradecimientos</b>	<b>II</b>
<b>Tabla de Contenidos</b>	<b>III</b>
<b>Índice de Figuras</b>	<b>V</b>
<b>Índice de Tablas</b>	<b>VI</b>
<b>Resumen</b>	<b>VII</b>
 <b>1. Introducción</b>	 <b>8</b>
1.1. Presentación del problema . . . . .	8
1.2. Hipótesis . . . . .	8
1.3. Objetivos . . . . .	8
1.4. Propuesta de solución . . . . .	9
1.5. Alcances . . . . .	10
1.6. Trabajo relacionado . . . . .	11
 <b>2. Marco Teórico</b>	 <b>12</b>
2.1. Metodología de desarrollo . . . . .	12
2.1.1. Metodología iterativo e incremental . . . . .	12
2.2. Metodología de evaluación . . . . .	13
2.2.1. Casos de estudio . . . . .	13
2.3. Herramientas para la implementación del software . . . . .	13
2.4. Red de distribución de agua potable . . . . .	14
2.4.1. Componentes físicos de una red . . . . .	14
2.4.2. Epanet . . . . .	15
2.5. Optimización . . . . .	16
2.6. Heurística . . . . .	16
2.7. Metaheurística . . . . .	17

2.7.1. Algoritmos Evolutivos . . . . .	17
2.7.2. Algoritmo Genético . . . . .	19
2.7.3. Conceptos para la optimización multiobjetivo . . . . .	19
2.7.4. Algoritmo NSGA-II (Nondominated Genetic Algorithm) . . . .	21
<b>3. Propuesta De Solución</b>	<b>25</b>
<b>4. Evaluación De La Solución</b>	<b>26</b>
<b>5. Conclusiones Y Trabajos Futuros</b>	<b>27</b>
<b>Glosario</b>	<b>28</b>
<b>Bibliografía</b>	<b>29</b>
<b>Anexos</b>	
<b>A: Documento de especificación de requisitos</b>	<b>32</b>
<b>B: Documento de diseño</b>	<b>33</b>
<b>C: Documento de casos de prueba</b>	<b>34</b>
<b>D: Cuestionario para la evaluación de la aplicación</b>	<b>35</b>

## ÍNDICE DE FIGURAS

	página
2.1. Metodología iterativo e incremental [13] . . . . .	13
2.2. Componentes físicos de un sistema de distribución de agua [15] . . . .	15
2.3. Pseudocódigo algoritmo evolutivo . . . . .	17
2.4. Ejemplo de dominancia y óptimo de Pareto . . . . .	20
2.5. Ejemplo frente de Pareto . . . . .	21
2.6. Pseudocódigo de la función de remplazo utilizada en el algoritmo NSGA-II [6] . . . . .	22
2.7. Procedimiento NSGA-II [6] . . . . .	23
2.8. Pseudocódigo de la función de remplazo utilizada en el algoritmo NSGA-II [6] . . . . .	24
3.1. La primera figura de la memoria . . . . .	25

## ÍNDICE DE TABLAS

	página
3.1. La primera tabla de la memoria . . . . .	25



## RESUMEN

La escasez de agua potable es un problema que esta presente a nivel mundial y que es de difícil solución. Por esta razón, existe la necesidad de utilizar conscientemente el recurso hidráulico y buscar maneras de optimizar su distribución.

Dentro de las redes de distribución de agua potable, encontramos diferentes tipos de problema. Estos problemas pueden agruparse principalmente en dos clases, las cuales son diseño y operación. La primera busca la optimización en la configuración y elección de los elementos que componen la red previo a su construcción. En cuanto a la segunda, esta busca la optimización de las características operacionales de una red ya construida.

Los algoritmos metaheurísticos han demostrado ser un mecanismo eficiente y con un buen desempeño en la resolución de problemas asociados a la redes de distribución de agua potable. Esto debido a que permiten recorrer el espacio de búsqueda del problema en busca de soluciones factibles.

Con el fin de afrontar este problema se llevara a cabo el desarrollo de un sistema el cual permita buscar y evaluar la factibilidad de las soluciones a problemas presentes en las redes de distribución de agua potable. Este sistema abordara dos problemas, uno de diseño y uno de operación. El primero de estos consiste en la optimización de los costos de inversión, para el cual se utilizara un enfoque monoobjetivo. En cuanto al segundo, este sera la optimización de los costos energéticos y el régimen de bombeo utilizando un enfoque multiobjetivo.

Adicionalmente se pretende que el sistema a desarrollar pueda ser ampliado en futuros trabajos, lo cual permitirá agregar nuevos algoritmos metaheurísticos y operadores asociados a estos, así como la implementación de nuevos problemas a resolver dentro de las redes de distribución de agua potable.

Para el desarrollo de este proyecto se opto por utilizar la metodología de desarrollo Iterativo e Incremental.

Para la evaluación del sistema a desarrollar se usara la metodología de caso de estudio cuyo resultado permitirá determinar la utilidad del software desarrollado.

# 1. Introducción

---

En la actualidad la escasez de agua potable es

## 1.1. Presentación del problema

Los encargados de implementar sistemas de distribución de agua potable, no cuentan con suficientes herramientas y tiempo para su correcta gestión. Por lo tanto, no es posible utilizar los recursos asociados de forma eficiente. Además, las herramientas existentes no satisfacen las necesidades de usabilidad y costo, debido a que son poco intuitivas y de pago.

El escoger las especificaciones de una red de agua potable ya es de por sí difícil debido a que hay que evaluar el rendimiento general del sistema alternando entre distintas configuraciones en busca de una solución que sea eficaz. Debido a esto, el uso de herramientas automatizadas que evalúen el rendimiento de las diversas combinaciones posibles viene a ser necesario.

A lo anterior se suma el hecho de que los interesados en esta área no manejan herramientas informáticas.

## 1.2. Hipótesis

## 1.3. Objetivos

Para abarcar la problemática se fijan los siguientes objetivos

### Objetivo general

- Diseñar y desarrollar una aplicación de apoyo a la toma de decisiones, integrando algoritmos de optimización aplicados al problema de diseño y operación en redes de distribución de agua potable.

### Objetivos específicos

1. Generar soluciones frente al problema monoobjetivo de diseño de redes de distribución de agua potable a través de la implementación de algoritmos genéticos.
2. Generar soluciones frente al problema multiobjetivo de operación de redes de distribución de agua potable a través de la implementación de NSGA-II.
3. Diseñar e implementar la interfaz gráfica del sistema de optimización de redes de agua potable desarrollado durante este proyecto.
4. Evaluar el desempeño de los algoritmos, contrastando los resultados obtenidos en redes de benchmarking con óptimos conocidos.

## 1.4. Propuesta de solución

La solución que se propone para abordar el problema consiste en el desarrollo de una aplicación que permita buscar soluciones a dos de los problemas existentes en las redes de agua potable. Además, de hacer uso de una arquitectura que permita fácilmente extender el programa para abarcar nuevos problemas. Para lograr esto último la aplicación debe quedar bien documentada. Por lo tanto, lo que se tendrá al término del desarrollo del proyecto será un software escalable que tratara con dos problemas relacionados a la distribución de agua potable y que podrá ser ampliado en futuros trabajos.

Los problemas que se abordaran en el contexto de optimización de redes de agua potable serán:

- **Problema de diseño:** Se trata este problema desde un enfoque monoobjetivo implementando un algoritmo genético que buscará la optimización de los costo de inversión variando el diámetro de las tuberías.

- **Problema de operación:** Este problema se abordará desde el enfoque multiobjetivo. Para esto se implementará el algoritmo NSGA-II y se buscará la optimización de los objetivos de costos energéticos y el número de encendidos y apagados de las bombas.

Tanto el algoritmo genético como el NSGA-II, permiten la utilización de distintos operadores de cruzamiento y mutación que también serán implementados para ser utilizados.

La solución planteada supone además el diseño e implementación de una interfaz gráfica para ayudar al usuario en el uso de la herramienta.

## 1.5. Alcances

Los alcances propuestos para este proyecto serán los siguientes:

- El sistema permitirá la carga y la visualización de la red gráficamente.
- El sistema permitirá visualizar la configuración básica almacenada en el archivo `inp` de los elementos de la red.
- El sistema solo resolverá dos clases de problemas de optimización, uno monoobjetivo y el otro multiobjetivo. El problema monoobjetivo será el de los costos de inversión. En cuanto al problema multiobjetivo, este será el de los costos energéticos y el número de encendidos y apagado de las bombas.
- El sistema únicamente contará con dos algoritmos implementados los cuales serán el algoritmo genético y NSGA-II. El algoritmo genético será el usado para tratar el problema monoobjetivo, mientras que NSGA-II será aplicado al multiobjetivo.
- El sistema permitirá visualizar el o los resultados obtenidos al finalizar la ejecución del algoritmo.
- El sistema permitirá generar archivos `.inp` con la configuración de la solución obtenida a través de los algoritmos. Debido a que el archivo `.inp` establece una gran cantidad de configuraciones, las únicas que se permitirán modificar serán las configuraciones asociadas a las conexiones (junctions) y tuberías (pipes).

- El sistema permitirá generar archivos con las soluciones obtenidas para cada problema, es decir, el valor de los objetivos y las variables de decisión involucradas.
- El sistema permitirá la visualización de una gráfica la cual permitirá visualizar la variación de la solución durante la ejecución de los algoritmos.
- La grafica unicamente estará disponible en problemas con uno o dos objetivos.
- La comparación con las redes de benchmarking consistirá unicamente, en presentar una tabla comparativa entre los resultados presentes en la literatura y los obtenidos a través de nuestro sistema, para cada uno de los problemas de redes de distribución de agua potable de nuestro sistema.

Este proyecto no contempla la creación de la red por lo que estas deberán ser ingresadas como entradas al programa, es decir, estas deberán ser creadas usando EPANET o manualmente, pero siguiendo el formato establecido por EPANET. Además, esta herramienta únicamente podrá ser ocupada en equipos cuyo sistema operativo sea Windows debido a que se realizan llamadas a librerías nativas.

## 1.6. Trabajo relacionado

## 2. Marco Teórico

---

### 2.1. Metodología de desarrollo

Para el desarrollo de este proyecto se optó por la metodología iterativa e incremental. La elección de esta metodología se debe al hecho de que este genera suficiente documentación para ser utilizada en el desarrollo de futuros proyectos derivados de este trabajo.

#### 2.1.1. Metodología iterativo e incremental

El desarrollo iterativo e incremental es una metodología la cual lleva a cabo el desarrollo de un proyecto de software dividiéndolo en iteraciones que generan un incremento, el cual contribuye en el desarrollo del producto final.

Cada iteración se compone de las fases de análisis, diseño, implementación y testing como se muestra en la figura 2.1. La fase de análisis se encarga de llevar a cabo la obtención y definición de los requerimientos del software. Durante la etapa de diseño se realiza la conceptualización del software basado en los requerimientos definidos anteriormente. Durante la implementación se codifican las funcionalidades siguiendo las directivas establecidas durante el diseño, con el fin de satisfacer los requerimientos. Y finalmente, durante la fase de testing, se valida y verifica la correctitud de las funcionalidades implementadas, así como el cumplimiento de los requisitos.

El hecho de llevar a cabo un desarrollo iterativo permite la obtención de retroalimentación del producto que se está desarrollando tempranamente y de esta manera poder refinar el trabajo en etapas posteriores del desarrollo. [19, 11, 10, 1].

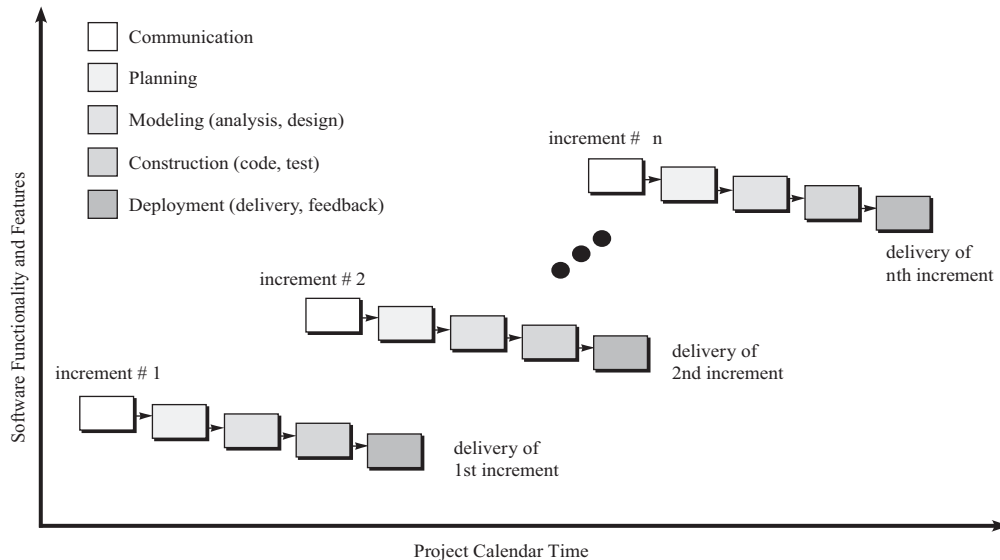


Figura 2.1: Metodología iterativo e incremental [13]

## 2.2. Metodología de evaluación

### 2.2.1. Casos de estudio

## 2.3. Herramientas para la implementación del software

### Java

Java es un lenguaje de programación de alto nivel orientado a objetos y de propósito general. Un programa java se ejecuta sobre la maquina virtual llamada la Java Virtual Machine, la cual le da a este lenguaje la característica de ser multi-plataforma. Adicionalmente, java incorpora el soporte para multi-hilos, una poderosa herramienta que permite la ejecución de distintas instrucciones de código al mismo tiempo [7]. Además, este lenguaje también incorpora una característica conocida como el recolector de basura, el cual se encarga de limpiar la memoria de objetos que ya no están siendo utilizados. Fue anunciado por Sun Microsystems en Mayo de 1995 [17].

### Java Reflection

Característica de java que permite que un programa se auto examine. Esta característica está disponible a través de la Java Reflection API, la cual cuenta con

métodos para obtener los meta-object de las clases, métodos, constructores, campos o parámetros. Esta API también permite crear nuevos objetos cuyo tipo era desconocido al momento de compilar el programa [5].

### **Java Annotation**

Característica de java para agregar metadatos a elementos de java (clases, métodos, parámetros, etc.) [7]. Las anotaciones no tienen efecto directo sobre el código, pero cuando son usadas junto con otras herramientas pueden llegar a ser muy útiles. Estas herramientas pueden analizar estas anotaciones y realizar acciones en base a estas, por ejemplo, generar archivos adicionales como clases de java, archivos xml, entre otras; ser analizadas durante la ejecución del programa vía Java Reflection, para crear objetos cuyo tipo no conocemos en tiempo de compilación; etc.

## **2.4. Red de distribución de agua potable**

Conjunto de elementos enlazados de tal manera que permite suministrar cierta cantidad de agua a una presión establecida [12].

### **Caudal**

Cantidad de agua que se mueve a través de un segmento de la red.

#### **2.4.1. Componentes físicos de una red**

A continuación se define los componentes que conforman una red de agua potable [15], los cuales se aprecian en la figura 2.2:

### **Nudos de caudal**

Son los puntos o extremos de una tubería, los cuales también permiten que estas se unan. Estos nudos pueden actuar como nudos de demanda a través de los cuales el flujo abandona la red.

### **Embalse**

Es una fuente de alimentación externa.



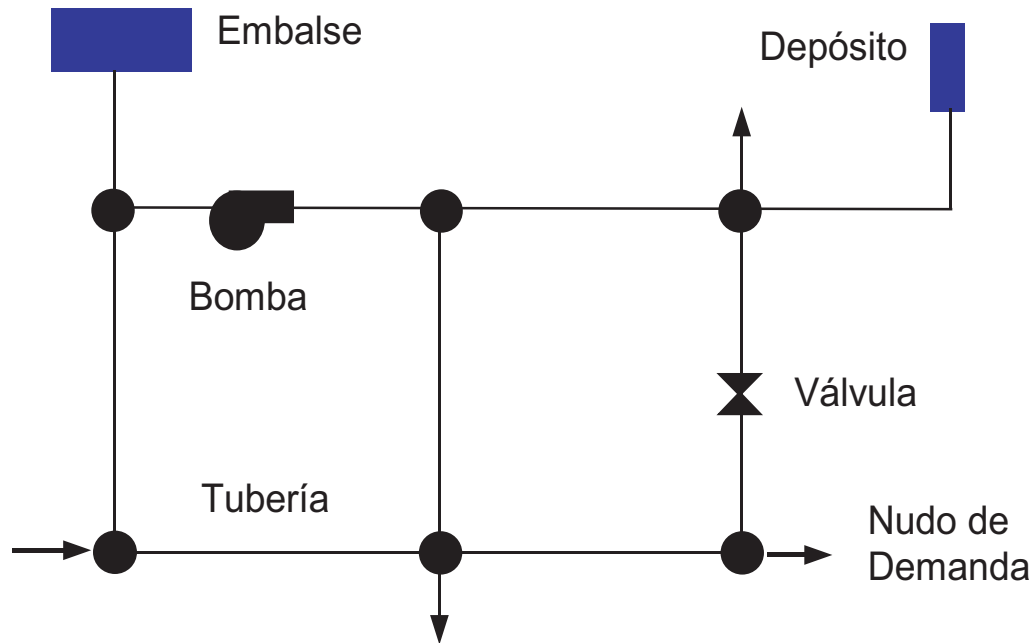


Figura 2.2: Componentes físicos de un sistema de distribución de agua [15]

### Deposito

Son elementos con la capacidad de almacenar agua.

### Tubería

Son los elementos a través de los cuales transita el agua de un nudo a otro.

### Bomba

Elementos que permiten impulsar el líquido con el fin de elevarlo a una posición superior.

### Válvula

Elementos que limitan la presión o el caudal que transita en un punto de la red.

#### 2.4.2. Epanet

Software que permite simular el comportamiento hidráulico y la calidad del agua en redes de distribución de aguas compuesta por tuberías, nodos, bombas, válvulas

y tanques de almacenamiento [15]. Este software cuenta también con una librería dinámica conocida bajo el nombre de Epanet Programming Toolkit, la cual cuenta con un conjunto de funciones para realizar simulaciones desde diferentes entornos de desarrollo como C, C++, VB, Java, etc [16].

## 2.5. Optimización

La optimización consiste en maximizar o minimizar un conjunto de funciones que matemáticamente pueden ser expresadas de la siguiente forma:

$$f_1(x), f_2(x), \dots, f_N(x), \quad x = (x_1, \dots, x_d) | x \in X$$

sujeto a una serie de condiciones

$$h_j(x) = 0, j = 1, 2, \dots, J$$

$$g_k(x) \leq 0, k = 1, 2, \dots, K$$

siendo  $f_1, \dots, f_N$  funciones objetivos;  $x_1, \dots, x_d$  variables de decisión, pertenecientes al espacio de búsqueda  $X$ ; y  $h_j$  junto con  $g_k$ , una serie de restricciones [20]. De acuerdo a la cantidad de funciones objetivos que se tenga, se establece que si  $N = 1$  la optimización es **monoobjetivo**, mientras que para  $N \geq 2$  se conoce como **multi-objetivo** [20]. En este punto se debe tener en cuenta que los objetivos planteados deben encontrarse en contradicción.

Debido a la definición de las restricciones es posible dividir el espacio de búsqueda en dos regiones [4]:

- Soluciones factibles: Compuesto por los elementos pertenecientes al espacio de búsqueda que satisfacen todas las restricciones.
- Soluciones no factibles: Integrado por aquellos elementos que no complen todas las restricciones.

## 2.6. Heurística

Técnicas estocásticas que permitan acotar la búsqueda de soluciones en problemas cuyo espacio de búsqueda es de gran tamaño y que hacen inviable la aplicación de

técnicas deterministas por el costo de tiempo que implican. Con la utilización de estas técnicas se espera encontrar soluciones buenas en un tiempo razonable, pero esto no está garantizado [20, 14].

## 2.7. Metaheurística

Algoritmos que permiten resolver un amplio rango de problemas de optimización empleando técnicas con algún grado de aleatoriedad para encontrar soluciones a un problema. Estos algoritmos no garantizan que la solución encontrada sea la óptima, pero permiten obtener generalmente aproximaciones a esta. La diferencia entre heurísticas y metaheurísticas, es que esta última puede ser aplicada a un amplio conjunto de problemas sin necesidad de realizar grandes cambios en el algoritmo, mientras que las heurísticas generalmente son aplicadas a un dominio específico [20, 3, 9].

### 2.7.1. Algoritmos Evolutivos

Conjunto de algoritmos inspirado en la teoría de la evolución de Darwin acerca de la capacidad de la naturaleza para evolucionar seres vivos bien adaptados a su entorno. Estos algoritmos hacen uso de diversos mecanismos entre los que se encuentran la selección, mutación y cruzamiento sobre los individuos de una población con el fin de generar una nueva generación de individuos [3]. En la Figura 2.3 muestra un pseudocódigo de los pasos generales de un algoritmo evolutivo.

---

#### Algoritmo 1: Algoritmo Evolutivo

---

```

1 población ← crearPoblaciónInicial()
2 evaluarPoblación(población)
3 mientras la condición de termino no ha sido alcanzada hacer
4   | poblaciónSeleccionada ← selección(población)
5   | poblaciónDecendiente ← cruzamiento(poblaciónSeleccionada)
6   | poblaciónDecendiente ← mutación (poblaciónDecendiente)
7   | poblaciónDecendiente ← evaluarPoblación (poblaciónDecendiente)
8   | población ← reemplazar (población, poblaciónDecendiente)
9 fin
```

---

Figura 2.3: Pseudocódigo algoritmo evolutivo

Primero, se crea la población inicial. Luego, se evalúan los objetivos de dicha población y se itera hasta que la condición de termino haya sido alcanzada. La condición de término puede ser, por ejemplo un máximo numero de evaluaciones o evaluaciones sin mejoras en los resultados. Dentro del ciclo se realiza la selección sobre la población con el fin de determinar las soluciones que serán usadas en los operadores de cruzamiento y mutación. Finalmente, se remplaza la población inicial con la descendiente.

### **Población**

Conjunto de soluciones candidatas sobre las cual opera el algoritmo. Durante cada iteración del algoritmo se generan nuevas soluciones que son agregadas a la población a la vez que se remueven otras. Una solución en la población se conoce como individuo [8].

Los individuos pueden ser representados de diversas maneras, entre ellas se encuentra la representación binaria (1 y 0), la real (Los números reales), etc. Para la representación binaria cada variable se codifica como un conjunto de bits, lo cual forma una cadena binaria. Por ejemplo, la representación binaria de la solución (2, 4, 6, 8) formada por enteros de 4 bits correspondería a

0010 0100 0110 1000

En cambio, para la representación real esta se presenta como un vector, en donde cada valor que forma este vector pertenece a los números reales, es decir,

$$v = (x_1, x_2, \dots, x_n), \text{ en donde } v \in \mathbb{R}^n$$

### **Selección**

La selección es un mecanismo utilizado por los algoritmos evolutivos para escoger a los individuos mas aptos los cuales serán usados para la reproducción [8]. Existen numerosos algoritmos de selección que pueden ser utilizados en los algoritmos evolutivos.

### Cruzamiento

El cruzamiento es un mecanismo usado para generar nuevas soluciones a partir de dos o mas individuos seleccionados [2, 3].

### Mutación

La mutación es un operador el cual permite mantener la diversidad en la descendencia [3] realizando modificaciones en ciertas partes de la solución.

#### 2.7.2. Algoritmo Genético

El algoritmo genético es una estrategia de búsqueda de soluciones. Para realizar esto, el algoritmo parte desde un conjunto de soluciones denominada población he iterativamente, lleva a cabo un proceso de reproducción, generando nuevas soluciones [8]. Este algoritmo pertenece a la categoría de algoritmos evolutivos y por lo tanto puede usar el mismo esquema presentado en la figura 2.3.

Los individuos en el contexto del algoritmo genético son llamados cromosomas, los cuales como se menciona en la sección de los algoritmos evolutivos pueden ser representados de diversas maneras.

#### 2.7.3. Conceptos para la optimización multiobjetivo

Como resultado de un proceso de optimización multiobjetivo no existe una única solución a un problema, sino que se tiene un conjunto de soluciones. Es por ello que a continuación se presentaran una serie de criterios que permitirán analizar y determinar el conjunto de soluciones optimas. Los criterios son los siguientes [18]:

#### Dominancia de Pareto

Sean  $u$  y  $v$  vectores pertenecientes a  $\mathbb{R}^n$ , se dice que  $u$  domina a  $v$  (se denota como  $u \preceq v$ ) si, y sólo si (en el caso de minimización):  $\forall i \in \{1, 2, \dots, n\} | f_i(u) \leq f_i(v) \wedge \exists j \in \{1, 2, \dots, n\} | f_j(u) < f_j(v)$

Es decir, para que una solución domine a otra, cada uno de sus objetivos debe ser mejores o iguales y al menos en uno de ellos este debe ser mejor.

En el ejemplo de la figura 2.4 se muestra los vectores A,B,C,D de los cuales C y B dominan a D, y A domina a C B y D. Nótese que C y B no son dominantes entre

sí.

### Óptimo de Pareto

Una solución  $u$  es un óptimo de Pareto si no hay otra solución  $v$  en el espacio  $\Omega$ , tal que  $v$  domine a  $u$ , es decir, para  $u, v \in \mathbb{R}^n$   $\nexists v \preceq u$ . Los óptimos de Pareto también se conocen bajo el nombre de solución no-dominada. En la figura 2.4 el vector A sería un óptimo de Pareto.

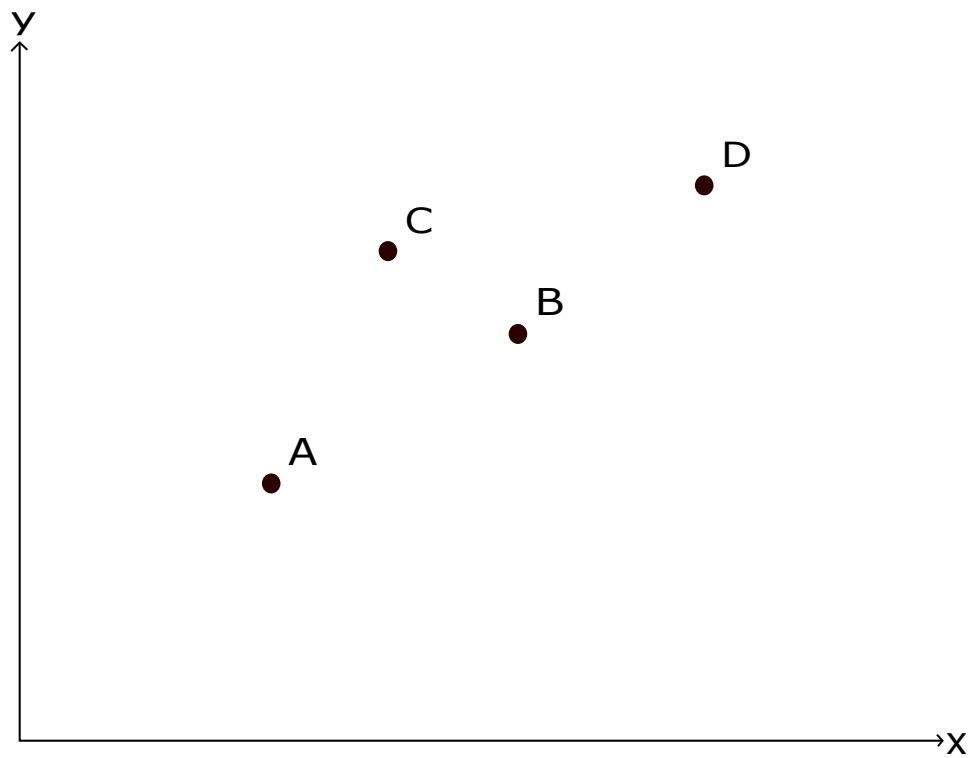


Figura 2.4: Ejemplo de dominancia y óptimo de Pareto

### Frontera de Pareto

La frontera de Pareto es el conjunto de todas las soluciones no dominadas las cuales componen las soluciones óptimas al problema multiobjetivo. En la figura 2.5 los puntos rojos componen la frontera de Pareto.

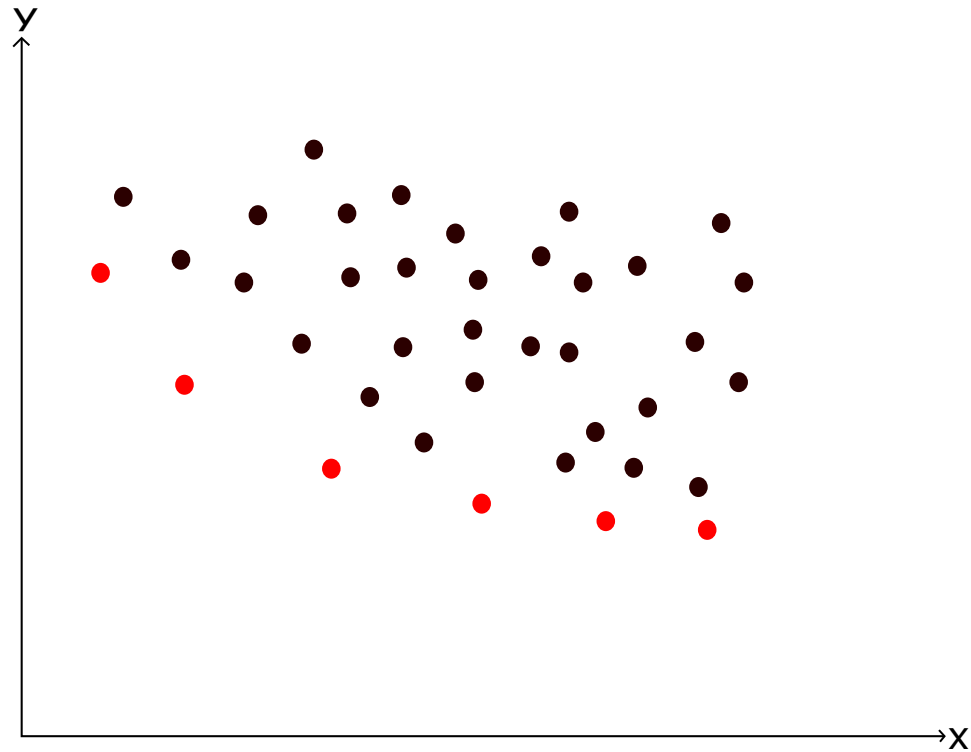


Figura 2.5: Ejemplo frente de Pareto

#### 2.7.4. Algoritmo NSGA-II (Nondominated Genetic Algorithm)

El algoritmo NSGA-II [6] pertenece a la categoría de algoritmo evolutivo multi-objetivo (MOEA). Este algoritmo al igual que el algoritmo genético hace uso de los operadores de cruzamiento, mutación y reproducción para encontrar un conjunto de soluciones optimas a problemas que cuentan con más de un objetivo. Adicionalmente, NSGA-II añade conceptos y operadores adicionales los cuales permiten mejorar su rendimiento y la calidad de las soluciones obtenidas. NSGA-II puede ser implementado siguiendo los mismos pasos de el algoritmo evolutivo mostrados en la figura 2.3, utilizando la función de remplazo mostrada en la figura 2.8.

En la figura 2.8 se puede ver que el proceso de remplazo dentro del algoritmo genético consiste en unir la población actual y la población descendiente (formada por los elementos resultantes del operador de selección y sobre la que se han aplicado el cruzamiento y la mutación) en un solo elemento llamada *unionPoblación*. Luego, esta es enviada a un procedimiento el cual ordena y categoriza la población en diversos frentes de acuerdo al concepto de dominancia de Pareto. Después, de que la población

a sido categorizada se procede a iterar sobre los frentes y añadir sus elementos a una nueva población con el cuidado de no sobrepasar el tamaño de la población deseada ( $N$ ). En caso de que uno de los frentes no pueda ser añadido en su totalidad por sobrepasar dicho tamaño, se llevara a cabo un proceso por el cual se ordenaran las soluciones en dicho frente basadas en un criterio conocido como densidad de las soluciones. Una vez las soluciones se encuentren ordenadas se rellenara la nueva población hasta alcanzar el tamaño de deseado  $N$ . Se puede ver un ejemplo de este procedimiento gráficamente en la figura 2.7.

---

**Algoritmo 2:** Función de remplazo para el algoritmo NSGA-II

---

```

1 Function remplazar(población, poblaciónDescendiente)
2   unionPoblacion  $\leftarrow$  población  $\cup$  poblaciónDescendiente
3    $F \leftarrow$  ordenarPorFrentesNoDominados (unionPoblacion)
4   /*  $F = (F_1, F_2, \dots)$  */
5   nuevaPoblacion  $\leftarrow \emptyset$ 
6    $i = i + 1$ 
7   /* Hasta que nuevaPoblacion este lleno */
8   mientras ( $|nuevaPoblacion| + |F_i| \leq N$ ) hacer
9     /* Calcular y asignar la densidad a cada solución del
10      frente  $F_i$  */
11     asignarDensidad ( $F_i$ )
12     /* Añadir a nuevaPoblacion las soluciones del frente  $F_i$  */
13     nuevaPoblacion  $\leftarrow$  nuevaPoblacion  $\cup F_i$ 
14      $i = i + 1$ 
15   fin
16   /* Ordenar el frente  $F_i$  usando el comparador de densidad */
17   ordenar ( $F_i, \prec_n$ )
18   /* Elegir los primeros  $N - |nuevaPoblacion|$  */
19   nuevaPoblacion  $\leftarrow$  nuevaPoblacion  $\cup F_i[1 : N - |nuevaPoblacion|]$ 
20   retornar nuevaPoblacion
21 fin

```

---

Figura 2.6: Pseudocódigo de la función de remplazo utilizada en el algoritmo NSGA-II [6]



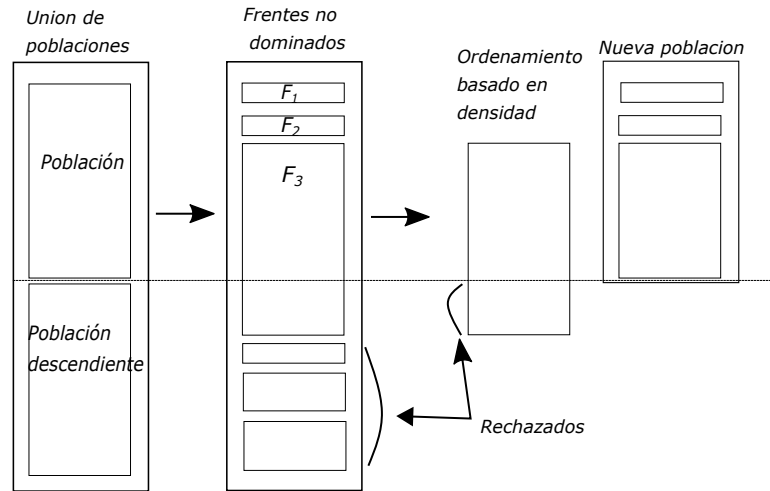


Figura 2.7: Procedimiento NSGA-II [6]

### Ordenamiento de soluciones en frentes no dominados

Uno de los procedimientos presentados en la figura 2.8 consistía en ordenar las soluciones en lo denominado como frentes no dominados. El proceso para llevar a cabo esto se presenta en [6] y consiste en:

### Densidad de estimación (Crowing Distance)

Otra funcion presente en la figura 2.8 es la asignación de densidad cuyo procedimiento es mostrado en la figura ...

### Comparador de densidad (Crowing Distance comparator)

Este operador compara las soluciones basados en dos conceptos los cuales son el ranking de dominación y la densidad de soluciones.

---

**Algoritmo 3:** Función de remplazo para el algoritmo NSGA-II
 

---

```

1 Function ordenarPorFrentesNoDominados (población)
2    $F \leftarrow \emptyset$ 
3   para cada  $p \in$  población hacer
4      $p.S_p = \emptyset$ 
5      $p.n_p = 0$ 
6     para cada  $q \in$  población hacer
7       si  $p \prec q$  entonces
8          $p.S_p = p.S_p \cup \{q\}$ 
9       si no, si  $q \prec p$  entonces
10         $p.n_p = p.n_p + 1$ 
11      fin
12      si  $p.n_p = 0$  entonces
13         $p.rank = 1$ 
14         $F_1 = F_1 \cup \{p\}$ 
15      fin
16    fin
17  fin
18 fin

```

---

Figura 2.8: Pseudocódigo de la función de remplazo utilizada en el algoritmo NSGA-II [6]

### 3. Propuesta De Solución

---

Sólo para incluir figuras y tablas.



Figura 3.1: La primera figura de la memoria

(aquí debiera ir la tabla)

Cuadro 3.1: La primera tabla de la memoria

## 4. Evaluación De La Solución

---

## 5. Conclusiones Y Trabajos Futuros

---

# Glosario

**El primer término:** Este es el significado del primer término, realmente no se bien lo que significa pero podría haberlo averiguado si hubiese tenido un poco mas de tiempo.

**El segundo término:** Este si se lo que significa pero me da lata escribirlo...

# Bibliografía

- [1] Adel Alshamrani and Abdullah Bahattab. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, 12(1):106–111, 2015.
- [2] Christian Blum and Andrea Roli. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.
- [3] Ilhem Boussaïd, Julien Lepagnot, and Patrick Siarry. A survey on optimization metaheuristics. *Information Sciences*, 237:82–117, 2013.
- [4] Omid Bozorg-Haddad, Mohammad Solgi, and Hugo A Loáiciga. *Meta-Heuristic and Evolutionary Algorithms for Engineering Optimization*. John Wiley & Sons, Incorporated, Newark, UNITED STATES, 2017.
- [5] Mathias Braux and Jacques Noyé. Towards partially evaluating reflection in Java. *ACM SIGPLAN Notices*, 34(11):2–11, 1999.
- [6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002.
- [7] James Gosling, Bill Joy, Guy Steele, Gilad Bracha, and Alex Buckley. The Java® Language Specification Java SE 8 Edition, 2015.
- [8] Dorothea Heiss-Czedik. An Introduction to Genetic Algorithms. *Artificial Life*, 3(1):63–65, 1997.

- [9] Sean Luke. *Essentials of Metaheuristics*. Lulu, second edition, 2013.
- [10] Robert C Martin. Iterative and Incremental Development ( IID ). *Design*, (Iid), 1999.
- [11] Susan M. Mitchell and Carolyn B. Seaman. A comparison of software cost, duration, and quality for waterfall vs. iterative and incremental development: A systematic review. *2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009*, (February 2008):511–515, 2009.
- [12] Daniel Mora. DISEÑO DE REDES DE DISTRIBUCIÓN DE AGUA MEDIANTE ALGORITMOS EVOLUTIVOS. ANÁLISIS DE EFICIENCIA. 2012.
- [13] Roger S Pressman. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 2009.
- [14] Marc H.J. Romanycia and Francis Jeffrey Pelletier. What is a heuristic? *Computational Intelligence*, 1(1):47–58, 1985.
- [15] Lewis Rossman. *EPANET 2.0 en Español. Analisis Hidraulico y de Calidad del Agua en Redes de Distribución de Agua. Manual del Usuario*. 2017.
- [16] Lewis A. Rossman. The EPANET Programmer's Toolkit for Analysis of Water Distribution Systems. In *WRPMD'99*, pages 1–10, Reston, VA, jun 1999. American Society of Civil Engineers.
- [17] C L Sabharwal. Java, Java, Java. *IEEE Potentials*, 17(3):33–37, 1998.
- [18] David a. Van Veldhuizen and Gary B Lamont. Evolutionary Computation and Convergence to a Pareto Front. *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, 1998.
- [19] R Victor. Iterative and Incremental Development: A Brief History. 2003.
- [20] Xin She Yang. Metaheuristic Optimization. *Scholarpedia*, 6(2011):11472, 2015.



**ANEXOS**

## A. Documento de especificación de requisitos

---

## B. Documento de diseño

---

## C. Documento de casos de prueba

---

## D. Cuestionario para la evaluación de la aplicación

---