# Resumen

# Overview

# Introducción

Es un poco hablar de lo que estás haciendo y de que otras herramientas existen actualmente para hacer lo mismo. También deberías hablar aquí de las decisiones básicas a la hora de hacer la aplicación (ej: ¿Por qué has usado Java?) y los objetivos de la misma.

# Algoritmos genéticos

Los algoritmos genéticos se apoyan en los principios de la evolución mediante selección natural, enunciados por Charles Darwin en su libro El Origen de las especies. Estos principios pueden resumirse así:

* Cada individuo tiende a transmitir sus rasgos a su progenie.
* Sin embargo, la naturaleza produce individuos con rasgos diferentes.
* Los individuos más adaptados, tienden a producir mayor progenie.
* Durante largos periodos de tiempo se puede acumular la variación produciendo nuevas especies completamente adaptadas a nichos particulares.

Los componentes básicos de un algoritmo genético son los siguientes:

* Una representación para los individuos.
* Una medida del grado de adaptación de un individuo al medio, la función de calidad.
* Un operador de selección, con probabilidad de selección de cada individuo proporcional a su calidad.
* Un operador emparejamiento o reproducción, que dará lugar a nuevos individuos en la siguiente generación.
* Un operador mutación, capaz de alterar las características de los nuevos individuos, incrementando la riqueza genética de la población.

En cada generación se crea un nuevo conjunto de individuos utilizando el material genético de los mejores individuos de la generación anterior.

## Representación de individuos

Los individuos pueden representarse de diversas formas. Dependiendo de la naturaleza de esta representación, la implementación de los operadores básico será diferente.

Habitualmente, la teoría clásica de algoritmos genéticos utiliza par a la representación la idea de cromosoma. Un cromosoma es el conjunto de los parámetros que determinan la estructura de un individuo. Cada uno de estos parámetros recibe el nombre de gen.

Existen diversos modos de representar los genes y por tanto los cromosomas. Una posibilidad habitual es utilizar valores binarios. Se asigna un número de bits a cada gen.

Sin embargo, también pueden existir representaciones que asignen a cada cromosoma un valor entero, real o cualquier otro tipo de datos específicos al campo de aplicación del algoritmo.

## Algoritmo general

El algoritmo trabaja sobre una población de individuos, representados por sus cromosomas. Cada uno de estos individuos tiene asociado un valor de su bondad determinado por la función de calidad o de coste.

El funcionamiento genérico de un algoritmo genético podría ser el siguiente:

Generar una población aleatoria de Ni individuos

Mientras que no se cumpla el criterio de terminación

Crear una nueva población

Seleccionar padres

Cruzar los padres con probabilidad pc

Mutar los individuos resultantes con probabilidad pm

Finalmente devolver como solución el individuo con mayor calidad de la población

Los criterios de terminación pueden ser variados:

* Los mejores individuos de la población generada representan soluciones suficientemente buenas.
* Se han alcanzado el número de generaciones especificado.
* La población ha convergido. Cuando la mayoría de los cromosomas de la población tienen la mayoría de sus genes iguales se dice que la población ha convergido. Cuando se da esta situación, el hecho de repetir sucesivas iteraciones no va a producir mejores resultados.

## Operador de selección

Este operador sirve para extraer individuos de una población, de modo que la probabilidad de extracción de un individuo sea proporcional al valor tomado por la función de calidad. De este modo Existen diversos métodos.

### Método de la ruleta

Se trata de un método de selección entre N elementos no equiprobables. A cada uno de los individuos se le asigna una probabilidad acumulada en función de su calidad. Se genera un número aleatorio dentro del rango 0,1 y se selecciona el individuo correspondiente.

### Selección por torneo

En la selección por torneo, se elige un número de individuos de forma aleatoria de entre los individuos de la población. Los individuos seleccionados se comparan entre sí y se elige el que tiene mayor aptitud.

En función del número de individuos que participen en el torneo se modifica la presión de selección. Cuando el número de individuos es elevado, la presión de selección es alta, los peores individuos apenas tienen posibilidades de ser seleccionados y la búsqueda de soluciones se centra en el espacio próximo a las mejores soluciones actuales. Por el contrario cuando el número de individuos es reducido, los peores tienen más posibilidades de ser seleccionados y la búsqueda de soluciones puede explorar nuevas regiones del espacio de búsqueda.

## Operador de cruce

El operador de cruce intercambia información genética entre dos individuos de la población.

Existen diversos algoritmos de cruce.

### Cruce en un punto

En este algoritmo se forma un vector compuesto por los genes del individuo. A continuación se selecciona un punto de corte en el vector de forma aleatoria generando dos segmentos diferenciados en cada vector. Las colas resultantes de cada vector se intercambian entre sí resultando dos vectores descendientes que heredan información genética de cada uno de los padres.

### Cruce en dos puntos

Se trata de una evolución del algoritmo de cruce en un punto. En este caso se seleccionan dos puntos en el vector de genes. Se intercambian entre ambos vectores los intervalos delimitados por los dos puntos.

## Operador de mutación

Se trata de un operador que provoca que se modifique alguno de los genes del individuo con un valor aleatorio. Esto suele hacerse con una probabilidad preestablecida generalmente muy baja.

El efecto de este operador es el de mantener la diversidad genética de una población, escapar de los óptimos locales y desplazar a los individuos hacia zonas del espacio de búsqueda que no pueden alcanzarse mediante el resto de operadores.

## Limitaciones de los algoritmos genéticos

En numerosas ocasiones los algoritmos genéticos tienden a converger hacia valores óptimos locales en lugar de soluciones óptimas globales del problema. Tienden a buscar mejoras a corto plazo desdeñando mejoras a largo plazo. Este problema puede solventarse modificando la función de evaluación, aumentando la probabilidad de mutación o mediante el uso de operadores de selección que mantengan la diversidad genética en la población.

# Diseño

Con los requisitos, análisis de casos de uso, etc. También debes de especificar los componentes que tiene, poner un diagrama de clases e indicar que hace cada una de ellas. Indicar cómo se evalúan las expresiones, cómo se organiza la interfaz gráfica... No incluyas código (esto irá en el CD) a no ser que fuese imprescindible. Se trata de documentar las clases indicando que hacen y cómo está organizada la aplicación no de describir exhaustivamente cada línea del programa...

## Requisitos

El objetivo de este proyecto es el de desarrollar una herramienta software para resolver problemas de optimización mediante el uso de algoritmos genéticos. El problema de optimización estará expresado en forma de una función de coste que puede tomar únicamente valores positivos.

En el análisis del problema se han detectado los siguientes requisitos funcionales.

### Introducción de la función de coste

Se trata de la función que se va a tratar de optimizar mediante el uso de la herramienta. Esta función tiene como parámetros una serie de números reales y tiene como resultado un valor positivo en el conjunto de los números reales:

J: Rn → Rn

Los posibles elementos que podrán incluirse en la función son

* Números reales.
* Parámetros, expresados por su nombre.
* Los siguientes operadores: +, -, /, \*, √, sin, cos, tan, asin, atan, abs, log, log10, log2
* Los símbolos de paréntesis “(“ y “)”.

### Codificación del vector de parámetros o cromosoma

El cromosoma o vector de parámetros es el conjunto de parámetros que se aplican a la función de coste. Está formado por una serie de elementos o genes, que consisten en los parámetros de la función de evaluación.

= [x1, …, xn]

Cada uno de los genes es un número real acotado entre un valor mínimo y un valor máximo.

xi [ximin, ximax]

### Introducción del tamaño de la población

El tamaño de la población es el número de elementos (cromosomas) con los que trabajará el algoritmo. Se trata de un valor numérico entero. Inicialmente tendrá un valor por defecto de 100 elementos.

### Establecimiento del número de generaciones

El número de generaciones determinará el número de iteraciones que se realizarán durante la ejecución del algoritmo. Se trata de un valor numérico entero. Este valor deberá poderse introducir por pantalla. Inicialmente tendrá un valor por defecto de 100 generaciones.

### Establecimiento del número de eras

El número de eras determinará el número de ejecuciones del algoritmo que se realizarán. Se realizan varias ejecuciones distintas del algoritmo para evitar problemas en la ejecución como la aparición de superelementos que dominen por completo la evolución. Se trata de un valor numérico entero que debe poderse introducir por pantalla. Inicialmente tendrá un valor por defecto de 10 eras.

### Selección de elitismo

Se debe poder seleccionar si el algoritmo operará con elitismo o no. El selector estará activado por defecto.

### Introducción de la probabilidad de cruce

La probabilidad de cruce determinará la probabilidad de que los cromosomas sean seleccionados para aplicar el operador de cruce. Se trata de un valor numérico real entre 0 y 1 que debe poderse introducir por pantalla. Inicialmente tendrá un valor por defecto de 0.2.

### Introducción de la probabilidad de mutación

La probabilidad de mutación determina la probabilidad de que a un cromosoma se le aplique el operador de mutación. Se trata de un valor numérico real entre 0 y 1 que debe poderse introducir por pantalla. Inicialmente tendrá un valor por defecto de 0.015.

### Selección de un operador de selección

El usuario debe poder elegir entre distintos operadores de selección. En concreto debe poder elegir entre los siguientes operadores:

* Método de la ruleta
* Selección por torneo

### Validación de la configuración

Antes de comenzar la ejecución del algoritmo debe validarse que la configuración seleccionada es válida. En particular deben validarse los siguientes aspectos:

* Que la función de coste introducida es sintácticamente correcta.
* Que la configuración del cromosoma es correcta y congruente con la función de coste. Esto es que los genes introducidos coinciden en número y nombre con los parámetros de la función introducida.

### Inicialización de la población

Al inicio de la ejecución del algoritmo genético se debe generar de forma aleatoria una población de individuos que cumplan con las restricciones especificadas para el cromosoma y cada uno de sus genes.

### Evaluación de la función de coste

Dado un cromosoma con valores debe poderse evaluar la función de coste para los genes que contiene.

### Operador de selección

Se ejecutará el operador de selección que se haya configurado en la herramienta. Ver punto 5.1.9

### Operador de cruce

El operador de cruce que se utilizará es el operador de cruce en un punto con la probabilidad de cruce introducida anteriormente.

### Operador de mutación

El operador de mutación que se aplicará es el de mutación uniforme con la probabilidad configurada inicialmente.

### Ejecución del algoritmo

Debe ejecutarse un algoritmo genético según la configuración introducida. El número de eras determinará el número de ejecuciones distintas del algoritmo que se realizarán. El número de generaciones determinará el número de iteraciones que realizará el algoritmo. El tamaño de la población determinará el número de elementos con los que trabajará y las probabilidades de mutación y de cruce determinarán la probabilidad con que se aplicarán los operadores de mutación y de cruce.

### Cancelación de la ejecución

La ejecución del algoritmo debe poder cancelarse en cualquier momento. Al cancelar la ejecución se mostrarán los resultados finales con los datos calculados hasta el momento de la cancelación.

### Resultados parciales

Durante la ejecución del algoritmo deben mostrarse los datos que se vayan calculando, así como información del progreso de la ejecución y del tiempo transcurrido. En concreto deben mostrarse los siguientes datos:

* Tiempo de ejecución del programa.
* Número de Era actual.
* Número de generación actual.
* Mejor cromosoma obtenido hasta el momento.
* Valor medio de la función de coste durante la generación anterior.
* Desviación estándar de la función de coste en la generación anterior.
* Mejora porcentual obtenido en el mejor valor de la función de coste en la generación anterior.
* Valor medio de los mejores valores de la función de coste obtenidos a lo largo de las distintas eras.

### Resultados finales

Al finalizar la ejecución del algoritmo deben mostrarse los datos de los cálculos realizados. Los datos que se deben mostrar son:

* El mejor cromosoma obtenido en cada era.
* El mejor valor de la función de coste obtenido en cada era.

Además deben mostrarse presentaciones gráficas de los datos obtenidos. En concreto

* El mejor valor de la función de coste obtenido a lo largo de cada era.

### Guardado de la configuración

La herramienta debe permitir almacenar en un fichero los datos configurados para la ejecución del algoritmo. Los datos que deben almacenarse son:

* La función de coste.
* La codificación del cromosoma.
* El tamaño de la población.
* El número de eras a ejecutar.
* El número de generaciones.
* La probabilidad de mutación.
* La probabilidad de cruce.
* La existencia de elitismo.
* El operador de selección a emplear.

### Carga de la configuración guardada

La herramienta debe poder cargar una configuración a partir del fichero guardado. Los datos que debe poder cargar son los especificados en el punto 5.1.20.

## Casos de uso

El único actor que se ha identificado es el usuario final de la herramienta. Del análisis de los requisitos se han identificado los siguientes casos de uso:

|  |  |
| --- | --- |
| **CU1** | **Introducción de la función de coste** |
| **Descripción:**  Permite al usuario introducir la función de coste que se pretende optimizar mediante el uso de la herramienta | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna | |
| **Flujo Normal:**  El usuario introduce la función que desea optimizar. | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  No aplica | |

|  |  |
| --- | --- |
| **CU2** | **Introducción del número de eras** |
| **Descripción:**  Permite al usuario introducir un valor numérico entre uno y cien representando el número de eras que se desea que se ejecuten | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna | |
| **Flujo Normal:**  El usuario introduce en el campo que contiene las generaciones un número entero entre uno y cien. | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  No aplica | |

|  |  |
| --- | --- |
| **CU3** | **Introducción del número de generaciones** |
| **Descripción:**  Permite al usuario introducir un valor numérico entre uno y mil representando el número de generaciones que se desea que se ejecuten | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna | |
| **Flujo Normal:**  El usuario introduce un número entero entre uno y mil. | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  No aplica | |

|  |  |
| --- | --- |
| **CU4** | **Introducción de la codificación de los cromosomas** |
| **Descripción:**  Permite al usuario introducir la codificación de los cromosomas que va a emplear el algoritmo. | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna | |
| **Flujo Normal:**   1. Por cada gen que componga el cromosoma el usuario introduce el nombre, valor mínimo, valor máximo y la precisión numérica que se va a emplear. 2. El usuario pulsa el botón de añadir. 3. La herramienta valida que el gen tenga un nombre, que no exista ningún gen con el mismo nombre y que el valor máximo no sea inferior al valor mínimo. 4. La herramienta añade el gen a la lista de genes del cromosoma | |
| **Flujo Alternativo:**   1. El sistema valida que el gen tenga nombre, que el nombre no esté repetido y que el valor máximo no sea inferior al valor mínimo. Si los datos del gen no son válidos, l herramienta muestra un mensaje indicando el error. | |
| **Post condiciones:** | |

|  |  |
| --- | --- |
| **Nombre:** | **Introducción del tamaño de la población** |
| **Descripción:**  Permite al usuario introducir el número de cromosomas que contiene la población durante la ejecución del algoritmo genético | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna | |
| **Flujo Normal:**  El usuario introduce en el campo tamaño de la población un valor numérico entre uno y mil | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  No aplica | |

|  |  |
| --- | --- |
| **CU5** | **Introducción de la probabilidad de cruce** |
| **Descripción:**  Permite al usuario introducir el valor que se utilizará como probabilidad para ejecutar el operador de cruce en el algoritmo genético. | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna | |
| **Flujo Normal:**  El usuario introduce en el campo probabilidad de cruce un valor numérico entre cero y uno. | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  No aplica | |

|  |  |
| --- | --- |
| **CU6** | **Introducción de la probabilidad de mutación** |
| **Descripción:**  Permite al usuario introducir el valor que se utilizará como probabilidad para ejecutar el operador de cruce en el algoritmo genético. | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna | |
| **Flujo Normal:**  El usuario introduce en el campo probabilidad de cruce un valor numérico un valor numérico entre cero y uno. | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  No aplica | |

|  |  |
| --- | --- |
| **CU7** | **Guardar configuración** |
| **Descripción:**  Permite al usuario guardar en un fichero los datos de configuración del algoritmo. | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Se han introducido valores de configuración en la herramienta. | |
| **Flujo Normal:**   1. El usuario pulsa en el botón de guardar configuración. 2. El usuario selecciona un fichero en el que guardar su configuración. 3. La herramienta almacenará en el fichero los datos de configuración existentes:    1. Número de eras.    2. Número de generaciones.    3. Tamaño de la población    4. Probabilidad de mutación.    5. Probabilidad de cruce.    6. Función de coste.    7. Codificación de cromosomas | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  Se creará un fichero con los datos de configuración. | |

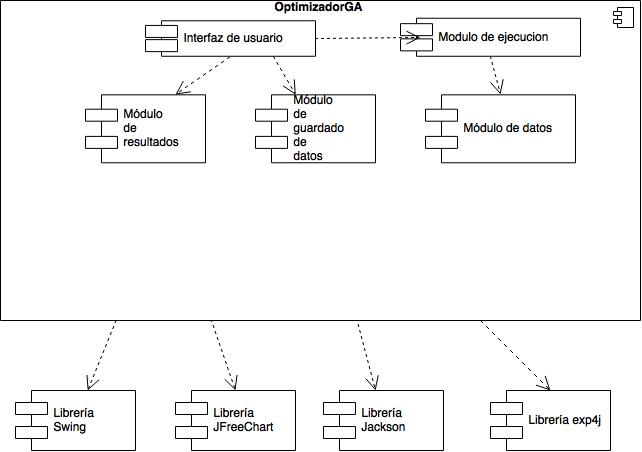
|  |  |
| --- | --- |
| **CU8** | **Cargar configuración** |
| **Descripción:**  Permite al usuario cargar los datos de configuración almacenados en un fichero. | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  Ninguna. | |
| **Flujo Normal:**   1. El usuario pulsa en el botón de cargar configuración. 2. El usuario selecciona el fichero que contiene la configuración. 3. En la pantalla principal de la herramienta aparecerán introducidos los parámetros de configuración que contiene el fichero. Estos parámetros podrán ser:    1. Número de eras.    2. Número de generaciones.    3. Tamaño de la población    4. Probabilidad de mutación.    5. Probabilidad de cruce.    6. Función de coste.    7. Codificación de cromosomas | |
| **Flujo Alternativo:**  No aplica | |
| **Post condiciones:**  La herramienta marcará los parámetros de configuración que contiene el fichero. | |

|  |  |
| --- | --- |
| **CU9** | **Resolución de problemas de optimización** |
| **Descripción:**  Hace que la herramienta resuelva el problema de optimización configurado. | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  La herramienta debe estar correctamente configurada | |
| **Flujo Normal:**   1. El usuario pulsa en el botón de ejecutar. 2. El sistema comprueba que existe una función de coste correcta. 3. El sistema valida que se han configurado los cromosomas de forma correcta y congruente con la función de coste 4. El sistema comienza la ejecución de un algoritmo genético con las propiedades configuradas. 5. El sistema muestra resultados parciales durante su ejecución. 6. Una vez finalizada la ejecución, el sistema muestra los resultados finales del cálculo. | |
| **Flujo Alternativo:**   1. Si la función de coste no es correcta, el sistema muestra un mensaje de error 2. Si la configuración de los cromosomas no es coherente con la función de coste o no es correcta, el sistema muestra un mensaje de error. | |
| **Post condiciones:**  Al finalizar la ejecución, el sistema mostrará los resultados finales del cálculo. | |

|  |  |
| --- | --- |
| **CU10** | **Cancelación de la ejecución** |
| **Descripción:**  El usuario puede cancelar en todo momento la ejecución. | |
| **Actores:**  El usuario final | |
| **Precondiciones:**  El sistema debe estar calculando la solución a un problema de optimización. | |
| **Flujo Normal:**   1. El usuario pulsa en el botón de cancelar. 2. La ejecución se detiene antes de finalizar. | |
| **Flujo Alternativo:**  No aplica. | |
| **Post condiciones:**  La ejecución se detendrá sin haber finalizado. | |

## Componentes del sistema

Los componentes que conforman el sistema, así como sus dependencias con componentes de terceros, se presentan en el siguiente diagrama.



Como puede observarse, el sistema está formado por un único componente. Este componente está compuesto por varios módulos:

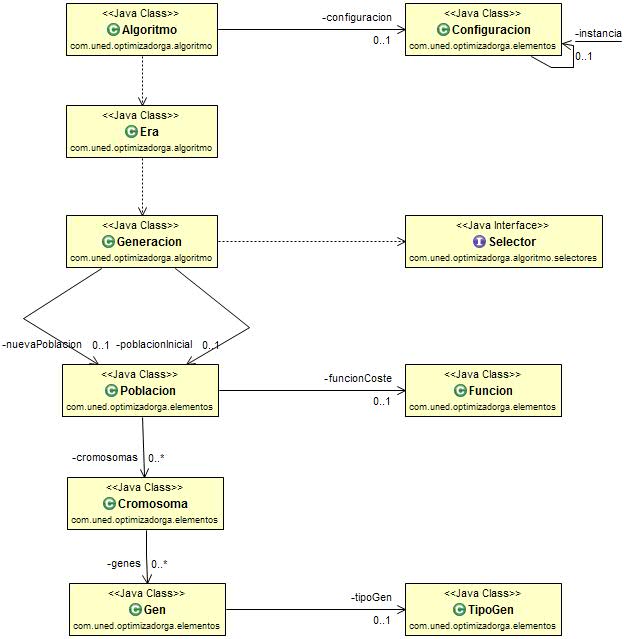
* **Interfaz de usuario:** es el módulo encargado de la interacción con el usuario.
* **Módulo de ejecución:** este módulo se encarga de gestionar la ejecución del algoritmo.
* **Módulo de datos:** este módulo se encarga de gestionar el modelo de datos con los que trabaja la aplicación.
* **Módulo de resultados:** este módulo está encargado de mostrar los resultados al usuario.
* **Módulo de guardado de datos:** este módulo se encarga de almacenar los resultados y la configuración de la ejecución.

Además, la herramienta realiza uso de determinadas librerías externas:

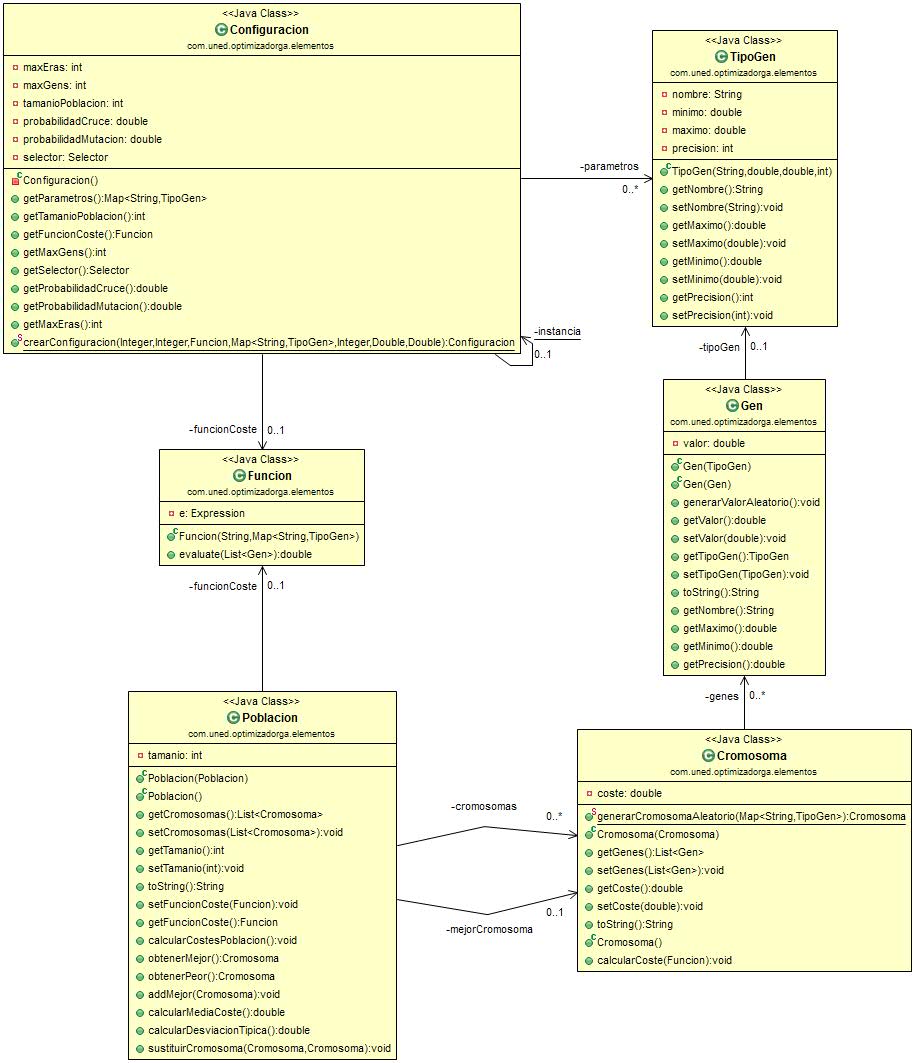
* **Swing:** librería estándar de java empleada para la construcción de interfaces gráficos de usuario.
* **JFreeChart:** librería destinada a la visualización de gráficas. Se utiliza para mostrar los resultados de la ejecución de forma gráfica. Dispone de licencia LGPL, que permite su uso de forma libre. http://www.jfree.org/jfreechart/
* **Jackson:** librería destinada al manejo de datos en formato JSON. Se utiliza para el almacenamiento de resultados y configuración en ficheros. Dispone de licencia Apache 2.0, que permite su uso de forma libre. http://wiki.fasterxml.com/JacksonHome
* **Exp4j:** librería destinada a la evaluación de expresiones matemáticas. Emplea el algoritmo shunting yard desarrollado por Edsger Dijkstra. Se utiliza para la evaluación de la función de coste. Dispone de licencia Apache 2.0, que permite su uso de forma libre. http://www.objecthunter.net/exp4j

## Diagrama de clases

A continuación se presenta el diagrama de las clases diseñado:



### Modelo de datos



El paquete elementos contiene las clases que almacenan los datos de la aplicación. A continuación se describen las clases implicadas en esta área.

#### TipoGen

Contiene la codificación de los genes. Esta codificación se representa mediante un nombre, un valor máximo, un valor mínimo y la precisión deseada para los valores del gen.

##### Atributos

* **Nombre**. De tipo String, representa el nombre que tendrá el gen.
* **Mínimo**. De tipo doble, representa el valor mínimo que pueden tomar los genes.
* **Máximo**. De tipo doble, representa el valor mínimo que pueden tomar los genes.
* **Precisión**. De tipo entero, representa el número de valores decimales con los que trabajará la herramienta para el gen.

##### Métodos

Dado que esta clase es un contenedor de propiedades, únicamente dispone de métodos de obtención y establecimiento de sus propiedades.

##### Relaciones

La clase forma parte de relaciones con las siguientes clases

* Con la clase Gen en una relación de agregación.
* Con la clase Configuración en una relación de agregación.

#### Gen

Esta clase contiene los genes que formarán parte de un cromosoma o individuo.

##### Atributos

* **Valor**. De tipo doble, contiene el valor que tiene un gen para la función de coste.
* **TipoGen**. De tipo TipoGen, representa la relación de agregación que se establece entre un gen y el elemento TipoGen que contiene su codificación.

##### Métodos

* **generarValorAleatorio**: Método que inicializa un gen con un valor aleatorio que cumpla con las restricciones especificadas por su tipología. Este método se emplea tanto en la generación de la población inicial como en el operador de mutación.

##### Constructores

* Constructor que recibe un tipo gen como parámetro. Este constructor se emplea para crear un gen nuevo sin un valor inicial.
* Constructor que recibe otro gen como parámetro. Este constructor se emplea para crear un gen nuevo copia de un gen ya existente.

##### Relaciones

* Con la clase TipoGen en una relación de agregación.
* Con la clase Cromosoma en una relación de agregación.

#### Cromosoma

Esta clase contiene los datos de un cromosoma.

##### Atributos

* **Coste**: el valor calculado mediante la función de coste para el cromosoma concreto.
* **Genes**: la lista de genes que forman un cromosoma. Representa la relación de agregación que se establece entre un cromosoma y sus genes.

##### Métodos

* **CalcularCoste**: Método que recibe como parámetro la función de coste y calcula el coste del cromosoma.

##### Constructores

* Constructor que recibe un cromosoma como parámetro. Este constructor se utiliza para crear un cromosoma copia de otro.
* Método **generarCromosomaAleatorio**. Se trata de un método estático que crea cromosomas con un valores aleatorios para sus genes

##### Relaciones

* De agregación con la clase Gen.
* De agregación con la clase Población para representar los cromosomas que conforman una población.
* De asociación con la clase Población para representar el mejor de los cromosomas de una población.

#### Población

Esta clase contiene todos los cromosomas que son los individuos que la forman.

##### Atributos

* **Cromosomas**: Un listado de los cromosomas que componen la población.
* **Tamaño**: El tamaño de la población con la que trabaja el algoritmo.
* **Función de coste**: La función de coste para la que se aplicará el algoritmo.
* **Mejor Cromosoma**: apunta al mejor cromosoma de la población.

##### Métodos

* **calcularCostesPoblacion**: calcula el coste para la función de coste de cada individuo de la población.
* **obtenerMejor**: obtiene el mejor individuo de la población.
* **obtenerPeor**: obtiene el peor individuo de la población.
* **calcularMediaCoste**: obtiene la media del valor de la función de coste para todos los cromosomas de la población.
* **calcularDesviaciónTipica**: calcula la desviación típica de la función de coste para todos los cromosomas de la población.
* **sustituirCromosoma**: Sustituye en una población un cromosoma por otro.

##### Constructores

* **Poblacion**: Constructor público que inicializa los cromosomas.
* **copiarPoblacionVacia**: Método de factoría estático que crea una nueva instancia de una población copiando sus datos de configuración (tamaño y función de coste).
* **generarPoblacionInicializada**: Método de factoría estático que crea una nueva instancia de una población tomando el tamaño y la función de coste de un objeto Configuración y generando los cromosomas con valores aleatorios.

##### Relaciones

* Con la clase Cromosoma, una relación de agregación, a través del atributo cromosomas.
* Con la clase Cromosoma, una relación de asociación, a través del atributo mejorCromosoma.
* Con la clase Funcion, una relación de asociación, a través del atributo función.

#### Función

Clase que contiene la función de coste y utilidades para evaluarla

##### Atributos

* **Expresión**: la expresión matemática que realizará la evaluación. Se trata de una referencia a una clase externa net.objecthunter.exp4j.Expression que es la que se ha utilizado para realizar la evaluación de expresiones matemáticas.

##### Métodos:

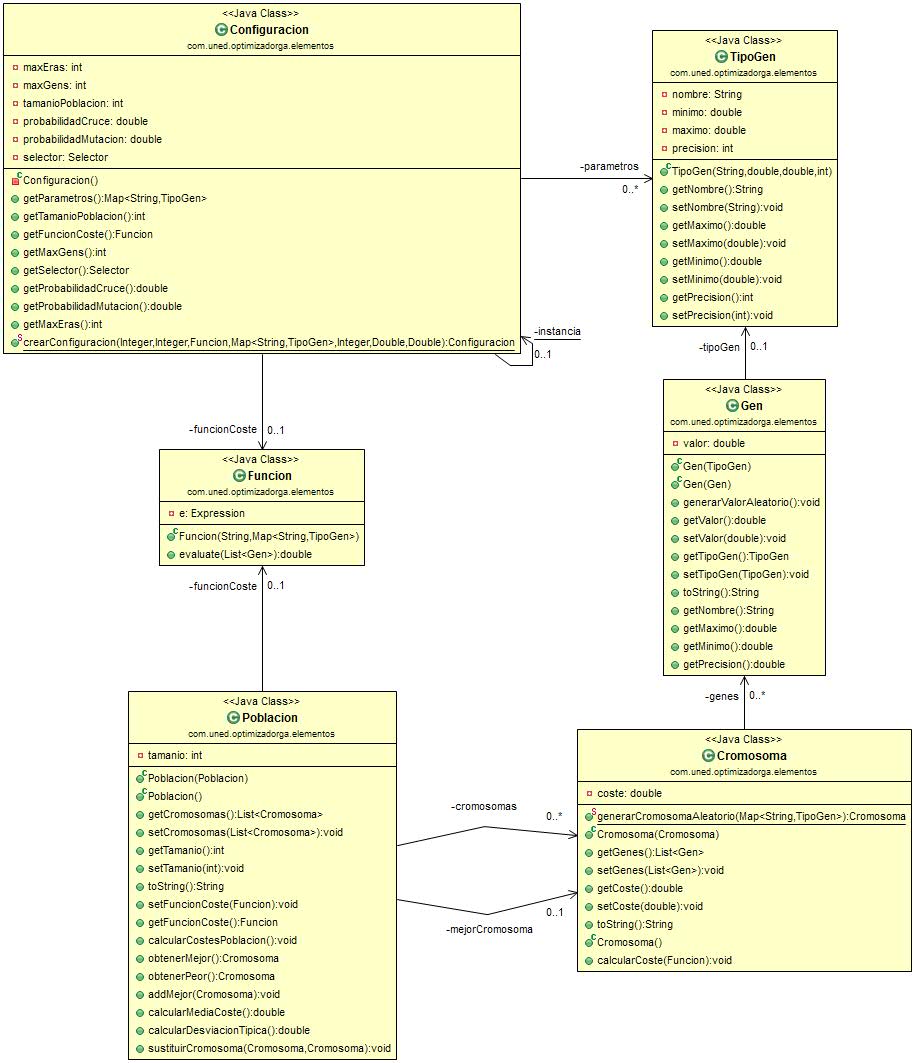
* **evaluate**: recibe una lista de genes que utiliza para evaluar un cromosoma. Devuelve el valor de la función de coste.

##### Constructores

* Dispone de un constructor público que recibe la expresión matemática y el listado de los tipos de gen que aplican.

#### Configuración

### Módulo de ejecución del algoritmo



### Módulo de interfaz gráfica y presentación de resultados

## Flujo de ejecución

Aquí irían diagramas de secuencia y de actividad de la ejecución de un programa. Por

## Interfaz gráfica

# Funcionamiento de la aplicación

Esta es la parte descriptiva de cómo se instala (si es necesario hacerlo) y usa la herramienta y las partes que tiene la GUI.

# Ejemplos de ejecución

Deberías probar la herramienta con todos las funciones propuestas en las directrices y comparar el óptimo teórico con el obtenido por el AG.

# Planificación y presupuesto

Describe cuanto has tardado, cómo has dividido el trabajo y cuanto costaría tu aplicación (un diagrama de gant con las partes del proyecto ayuda mucho en esta fase)

# Conclusiones

# Bibliografía