

BiCIAM

Generado por Doxygen 1.9.8

1 Índice de espacios de nombres	1
1.1 Lista de paquetes	1
2 Índice jerárquico	3
2.1 Jerarquía de clases	3
3 Índice de clases	7
3.1 Lista de clases	7
4 Índice de archivos	11
4.1 Lista de archivos	11
5 Documentación de espacios de nombres	15
5.1 Paquete <code>es.ull.esit.app.config.tspdynamic</code>	15
5.2 Paquete <code>es.ull.esit.app.evolutionary_algorithms.complement</code>	15
5.3 Paquete <code>es.ull.esit.app.factory_interface</code>	16
5.4 Paquete <code>es.ull.esit.app.factory_method</code>	16
5.5 Paquete <code>es.ull.esit.app.local_search.acceptation_type</code>	17
5.5.1 Descripción detallada	17
5.6 Paquete <code>es.ull.esit.app.local_search.candidate_type</code>	17
5.6.1 Descripción detallada	18
5.7 Paquete <code>es.ull.esit.app.local_search.complement</code>	18
5.8 Paquete <code>es.ull.esit.app.metaheuristics.strategy</code>	18
5.9 Paquete <code>es.ull.esit.app.metaheuristics.generators</code>	18
5.10 Paquete <code>es.ull.esit.app.problem.definition</code>	19
5.11 Paquete <code>es.ull.esit.app.problem.extension</code>	19
5.12 Paquete <code>es.ull.esit.app.problem_operators</code>	20
5.13 Paquete <code>es.ull.esit.app.strategy</code>	20
6 Documentación de clases	21
6.1 Referencia de la clase <code>es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate</code>	21
6.1.1 Descripción detallada	22
6.1.2 Documentación de funciones miembro	22
6.1.2.1 <code>acceptCandidate()</code>	22
6.2 Referencia de la clase <code>es.ull.esit.app.local_search.acceptation_type.AcceptAnyone</code>	23
6.2.1 Descripción detallada	23
6.2.2 Documentación de funciones miembro	24
6.2.2.1 <code>acceptCandidate()</code>	24
6.3 Referencia de la clase <code>es.ull.esit.app.local_search.acceptation_type.AcceptBest</code>	24
6.3.1 Descripción detallada	25
6.3.2 Documentación de funciones miembro	25
6.3.2.1 <code>acceptCandidate()</code>	25
6.4 Referencia de la clase <code>es.ull.esit.app.local_search.acceptation_type.AcceptMulticase</code>	26
6.4.1 Descripción detallada	27

6.4.2 Documentación de funciones miembro	27
6.4.2.1 acceptCandidate()	27
6.5 Referencia de la clase es.ull.esit.app.local_search.acceptation_type.AcceptNotBad	28
6.5.1 Descripción detallada	28
6.5.2 Documentación de funciones miembro	29
6.5.2.1 acceptCandidate()	29
6.6 Referencia de la clase es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT	29
6.6.1 Descripción detallada	30
6.6.2 Documentación de funciones miembro	30
6.6.2.1 acceptCandidate()	30
6.7 Referencia de la clase es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU	31
6.7.1 Descripción detallada	32
6.7.2 Documentación de funciones miembro	32
6.7.2.1 acceptCandidate()	32
6.8 Referencia de la clase es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated	32
6.8.1 Descripción detallada	33
6.8.2 Documentación de funciones miembro	33
6.8.2.1 acceptCandidate()	33
6.9 Referencia de la clase es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu	34
6.9.1 Descripción detallada	34
6.9.2 Documentación de funciones miembro	35
6.9.2.1 acceptCandidate()	35
6.10 Referencia de la enumeración es.ull.esit.app.local_search.acceptation_type.AcceptType	35
6.10.1 Descripción detallada	35
6.10.2 Documentación de datos miembro	35
6.10.2.1 AcceptAnyone	35
6.10.2.2 AcceptBest	36
6.10.2.3 AcceptMulticase	36
6.10.2.4 AcceptNotBad	36
6.10.2.5 AcceptNotBadT	36
6.10.2.6 AcceptNotBadU	36
6.10.2.7 AcceptNotDominated	36
6.10.2.8 AcceptNotDominatedTabu	36
6.11 Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation	37
6.11.1 Descripción detallada	38
6.11.2 Documentación de funciones miembro	38
6.11.2.1 fillPath()	38
6.11.2.2 mutation()	38
6.11.2.3 sortedPathValue()	38
6.11.3 Documentación de datos miembro	39
6.11.3.1 path	39
6.12 Referencia de la enumeración es.ull.esit.app.local_search.candidate_type.CandidateType	39

6.12.1 Descripción detallada	39
6.12.2 Documentación de datos miembro	39
6.12.2.1 GreaterCandidate	39
6.12.2.2 NotDominatedCandidate	39
6.12.2.3 RandomCandidate	39
6.12.2.4 SmallerCandidate	40
6.13 Referencia de la clase <code>es.ull.esit.app.local_search.candidate_type.CandidateValue</code>	40
6.13.1 Descripción detallada	40
6.13.2 Documentación de constructores y destructores	40
6.13.2.1 <code>CandidateValue()</code> [1/2]	40
6.13.2.2 <code>CandidateValue()</code> [2/2]	40
6.13.3 Documentación de funciones miembro	41
6.13.3.1 <code>getTabusolution()</code>	41
6.13.3.2 <code>newSearchCandidate()</code>	41
6.13.3.3 <code>setTabusolution()</code>	42
6.13.3.4 <code>stateCandidate()</code>	42
6.14 Referencia de la clase <code>es.ull.esit.app.problem.definition.Codification</code>	43
6.14.1 Descripción detallada	43
6.14.2 Documentación de funciones miembro	43
6.14.2.1 <code>getAleatoryKey()</code>	43
6.14.2.2 <code>getVariableAleatoryValue()</code>	43
6.14.2.3 <code>getVariableCount()</code>	44
6.14.2.4 <code>validState()</code>	44
6.15 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.Crossover</code>	45
6.15.1 Descripción detallada	45
6.15.2 Documentación de funciones miembro	45
6.15.2.1 <code>crossover()</code>	45
6.16 Referencia de la enumeración <code>es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType</code>	46
6.16.1 Descripción detallada	46
6.16.2 Documentación de datos miembro	46
6.16.2.1 <code>ONE_POINT_CROSSOVER</code>	46
6.16.2.2 <code>UNIFORM_CROSSOVER</code>	46
6.17 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.Distribution</code>	46
6.17.1 Descripción detallada	47
6.17.2 Documentación de funciones miembro	47
6.17.2.1 <code>distribution()</code>	47
6.18 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm</code>	47
6.18.1 Descripción detallada	49
6.18.2 Documentación de constructores y destructores	49
6.18.2.1 <code>DistributionEstimationAlgorithm()</code>	49
6.18.3 Documentación de funciones miembro	49
6.18.3.1 <code>awardUpdateREF()</code>	49

6.18.3.2 generate()	50
6.18.3.3 getCountRef()	50
6.18.3.4 getDistributionType()	50
6.18.3.5 getfathersList()	51
6.18.3.6 getGeneratorType()	51
6.18.3.7 getListCountBetterGender()	51
6.18.3.8 getListCountGender()	52
6.18.3.9 getListReference()	52
6.18.3.10 getListStateRef()	52
6.18.3.11 getReference()	52
6.18.3.12 getReferenceList()	53
6.18.3.13 getReplaceType()	53
6.18.3.14 getSelectionType()	53
6.18.3.15 getSonList()	53
6.18.3.16 getTrace()	54
6.18.3.17 getType()	54
6.18.3.18 getWeight()	54
6.18.3.19 maxValue()	54
6.18.3.20 setCountRef()	55
6.18.3.21 setDistributionType()	55
6.18.3.22 setGeneratorType()	55
6.18.3.23 setInitialReference()	55
6.18.3.24 setListReference()	56
6.18.3.25 setReplaceType()	56
6.18.3.26 setSelectionType()	56
6.18.3.27 setWeight()	56
6.18.3.28 updateReference()	57
6.19 Referencia de la enumeración <code>es.ull.esit.app.evolutionary_algorithms.complement.DistributionType</code>	57
6.19.1 Descripción detallada	57
6.19.2 Documentación de datos miembro	57
6.19.2.1 UNIVARIATE	57
6.20 Referencia de la clase <code>es.ull.esit.app.local_search.acceptation_type.Dominance</code>	58
6.20.1 Descripción detallada	58
6.20.2 Documentación de funciones miembro	58
6.20.2.1 dominance()	58
6.20.2.2 listDominance()	58
6.21 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.EvolutionStrategies</code>	59
6.21.1 Descripción detallada	60
6.21.2 Documentación de constructores y destructores	60
6.21.2.1 EvolutionStrategies()	60
6.21.3 Documentación de funciones miembro	61
6.21.3.1 awardUpdateREF()	61

6.21.3.2 generate()	62
6.21.3.3 getCountRef()	62
6.21.3.4 getListCountBetterGender()	63
6.21.3.5 getListCountGender()	63
6.21.3.6 getListStateRef()	63
6.21.3.7 getListStateReference()	63
6.21.3.8 getReference()	63
6.21.3.9 getReferenceList()	64
6.21.3.10 getSonList()	64
6.21.3.11 getTrace()	64
6.21.3.12 getType()	65
6.21.3.13 getTypeGenerator()	65
6.21.3.14 getWeight()	65
6.21.3.15 setCountRef()	65
6.21.3.16 setInitialReference()	66
6.21.3.17 setListStateReference()	66
6.21.3.18 setStateRef()	66
6.21.3.19 setTypeGenerator()	66
6.21.3.20 setWeight()	67
6.21.3.21 updateReference()	67
6.22 Referencia de la clase es.ull.esit.app.problem.extension.FactoresPonderados	68
6.22.1 Descripción detallada	68
6.22.2 Documentación de funciones miembro	68
6.22.2.1 evaluationState()	68
6.23 Referencia de la clase es.ull.esit.app.factory_method.FactoryAcceptCandidate	69
6.23.1 Descripción detallada	70
6.23.2 Documentación de funciones miembro	70
6.23.2.1 createAcceptCandidate()	70
6.24 Referencia de la clase es.ull.esit.app.factory_method.FactoryCandidate	70
6.24.1 Descripción detallada	71
6.24.2 Documentación de funciones miembro	71
6.24.2.1 createSearchCandidate()	71
6.25 Referencia de la clase es.ull.esit.app.factory_method.FactoryCrossover	72
6.25.1 Descripción detallada	73
6.25.2 Documentación de funciones miembro	73
6.25.2.1 createCrossover()	73
6.26 Referencia de la clase es.ull.esit.app.factory_method.FactoryDistribution	74
6.26.1 Descripción detallada	74
6.26.2 Documentación de funciones miembro	75
6.26.2.1 createDistribution()	75
6.27 Referencia de la clase es.ull.esit.app.factory_method.FactoryFatherSelection	76
6.27.1 Descripción detallada	76

6.27.2 Documentación de funciones miembro	77
6.27.2.1 createSelectFather()	77
6.28 Referencia de la clase <code>es.ull.esit.app.factory_method.FactoryGenerator</code>	78
6.28.1 Descripción detallada	78
6.28.2 Documentación de funciones miembro	79
6.28.2.1 createGenerator()	79
6.29 Referencia de la clase <code>es.ull.esit.app.factory_method.FactoryLoader</code>	79
6.29.1 Descripción detallada	79
6.29.2 Documentación de funciones miembro	80
6.29.2.1 getInstance()	80
6.30 Referencia de la clase <code>es.ull.esit.app.factory_method.FactoryMutation</code>	80
6.30.1 Descripción detallada	81
6.30.2 Documentación de funciones miembro	81
6.30.2.1 createMutation()	81
6.31 Referencia de la clase <code>es.ull.esit.app.factory_method.FactoryReplace</code>	82
6.31.1 Descripción detallada	83
6.31.2 Documentación de funciones miembro	83
6.31.2.1 createReplace()	83
6.32 Referencia de la clase <code>es.ull.esit.app.factory_method.FactorySampling</code>	84
6.32.1 Descripción detallada	84
6.32.2 Documentación de funciones miembro	85
6.32.2.1 createSampling()	85
6.33 Referencia de la clase <code>es.ull.esit.app.factory_method.FactorySolutionMethod</code>	86
6.33.1 Descripción detallada	86
6.33.2 Documentación de funciones miembro	87
6.33.2.1 createdSolutionMethod()	87
6.34 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection</code>	87
6.34.1 Descripción detallada	88
6.34.2 Documentación de funciones miembro	88
6.34.2.1 selection()	88
6.35 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace</code>	89
6.35.1 Descripción detallada	89
6.35.2 Documentación de funciones miembro	90
6.35.2.1 replace()	90
6.36 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.Generator</code>	91
6.36.1 Descripción detallada	92
6.36.2 Documentación de funciones miembro	92
6.36.2.1 awardUpdateREF()	92
6.36.2.2 generate()	93
6.36.2.3 getListCountBetterGender()	93
6.36.2.4 getListCountGender()	94
6.36.2.5 getReference()	94

6.36.2.6	getReferenceList()	94
6.36.2.7	getSonList()	95
6.36.2.8	getTrace()	95
6.36.2.9	getType()	95
6.36.2.10	getWeight()	96
6.36.2.11	setInitialReference()	96
6.36.2.12	setWeight()	96
6.36.2.13	updateReference()	97
6.36.3	Documentación de datos miembro	98
6.36.3.1	countBetterGender	98
6.36.3.2	countGender	98
6.36.3.3	listCountBetterGender	98
6.37	Referencia de la enumeración <code>es.ull.esit.app.metaheuristics.generators.GeneratorType</code>	98
6.37.1	Descripción detallada	99
6.37.2	Documentación de datos miembro	99
6.37.2.1	DistributionEstimationAlgorithm	99
6.37.2.2	EvolutionStrategies	99
6.37.2.3	GeneticAlgorithm	99
6.37.2.4	HillClimbing	99
6.37.2.5	HillClimbingDistance	99
6.37.2.6	HillClimbingRestart	99
6.37.2.7	LimitThreshold	100
6.37.2.8	MultiCaseSimulatedAnnealing	100
6.37.2.9	MultiGenerator	100
6.37.2.10	MultiobjectiveHillClimbingDistance	100
6.37.2.11	MultiobjectiveHillClimbingRestart	100
6.37.2.12	MultiobjectiveStochasticHillClimbing	100
6.37.2.13	MultiobjectiveTabuSearch	100
6.37.2.14	ParticleSwarmOptimization	100
6.37.2.15	RandomSearch	101
6.37.2.16	SimulatedAnnealing	101
6.37.2.17	StochasticHillClimbing	101
6.37.2.18	TabuSearch	101
6.38	Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm</code>	101
6.38.1	Descripción detallada	103
6.38.2	Documentación de constructores y destructores	103
6.38.2.1	GeneticAlgorithm()	103
6.38.3	Documentación de funciones miembro	103
6.38.3.1	awardUpdateREF()	103
6.38.3.2	generate()	103
6.38.3.3	getCountRef()	104
6.38.3.4	getGeneratorType()	104

6.38.3.5	getListCountBetterGender()	104
6.38.3.6	getListCountGender()	105
6.38.3.7	getListState()	105
6.38.3.8	getListStateRef()	105
6.38.3.9	getReference()	105
6.38.3.10	getReferenceList()	106
6.38.3.11	getSonList()	106
6.38.3.12	getTrace()	106
6.38.3.13	getType()	106
6.38.3.14	getWeight()	107
6.38.3.15	setCountRef()	107
6.38.3.16	setGeneratorType()	108
6.38.3.17	setInitialReference()	108
6.38.3.18	setListState()	108
6.38.3.19	setStateRef()	108
6.38.3.20	setWeight()	109
6.38.3.21	updateReference()	109
6.39	Referencia de la clase es.ull.esit.app.local_search.candidate_type.GreaterCandidate	110
6.39.1	Descripción detallada	110
6.39.2	Documentación de funciones miembro	111
6.39.2.1	stateSearch()	111
6.40	Referencia de la clase es.ull.esit.app.metaheuristics.generators.HillClimbing	112
6.40.1	Descripción detallada	113
6.40.2	Documentación de constructores y destructores	113
6.40.2.1	HillClimbing()	113
6.40.3	Documentación de funciones miembro	113
6.40.3.1	awardUpdateREF()	113
6.40.3.2	generate()	114
6.40.3.3	getGeneratorType()	114
6.40.3.4	getListCountBetterGender()	114
6.40.3.5	getListCountGender()	114
6.40.3.6	getReference()	115
6.40.3.7	getReferenceList()	115
6.40.3.8	getSonList()	115
6.40.3.9	getTrace()	115
6.40.3.10	getType()	116
6.40.3.11	getWeight()	116
6.40.3.12	setGeneratorType()	116
6.40.3.13	setInitialReference()	116
6.40.3.14	setStateRef()	116
6.40.3.15	setTypeCandidate()	117
6.40.3.16	setWeight()	117

6.40.3.17 updateReference()	117
6.41 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.HillClimbingRestart</code>	118
6.41.1 Descripción detallada	119
6.41.2 Documentación de constructores y destructores	119
6.41.2.1 HillClimbingRestart()	119
6.41.3 Documentación de funciones miembro	119
6.41.3.1 awardUpdateREF()	119
6.41.3.2 generate()	120
6.41.3.3 getCount()	120
6.41.3.4 getCountBetterGender()	120
6.41.3.5 getCountCurrent()	121
6.41.3.6 getCountGender()	121
6.41.3.7 getGeneratorType()	121
6.41.3.8 getListCountBetterGender()	121
6.41.3.9 getListCountGender()	122
6.41.3.10 getReference()	122
6.41.3.11 getReferenceList()	122
6.41.3.12 getSonList()	122
6.41.3.13 getTrace()	123
6.41.3.14 getType()	123
6.41.3.15 getWeight()	123
6.41.3.16 setCount()	123
6.41.3.17 setCountBetterGender()	124
6.41.3.18 setCountCurrent()	124
6.41.3.19 setCountGender()	124
6.41.3.20 setGeneratorType()	124
6.41.3.21 setInitialReference()	125
6.41.3.22 setStateRef()	125
6.41.3.23 setTypeCandidate()	125
6.41.3.24 setWeight()	125
6.41.3.25 updateReference()	126
6.42 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryAcceptCandidate</code>	126
6.42.1 Descripción detallada	126
6.42.2 Documentación de funciones miembro	126
6.42.2.1 createAcceptCandidate()	126
6.43 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryCandidate</code>	127
6.43.1 Descripción detallada	127
6.43.2 Documentación de funciones miembro	127
6.43.2.1 createSearchCandidate()	127
6.44 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryCrossover</code>	128
6.44.1 Descripción detallada	128
6.44.2 Documentación de funciones miembro	128

6.44.2.1 createCrossover()	128
6.45 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryDistribution</code>	129
6.45.1 Descripción detallada	129
6.45.2 Documentación de funciones miembro	129
6.45.2.1 createDistribution()	129
6.46 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryFatherSelection</code>	130
6.46.1 Descripción detallada	130
6.46.2 Documentación de funciones miembro	130
6.46.2.1 createSelectFather()	130
6.47 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryGenerator</code>	131
6.47.1 Descripción detallada	131
6.47.2 Documentación de funciones miembro	131
6.47.2.1 createGenerator()	131
6.48 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryMutation</code>	132
6.48.1 Descripción detallada	132
6.48.2 Documentación de funciones miembro	132
6.48.2.1 createMutation()	132
6.49 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactoryReplace</code>	133
6.49.1 Descripción detallada	133
6.49.2 Documentación de funciones miembro	133
6.49.2.1 createReplace()	133
6.50 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFactorySolutionMethod</code>	134
6.50.1 Descripción detallada	134
6.50.2 Documentación de funciones miembro	134
6.50.2.1 createdSolutionMethod()	134
6.51 Referencia de la interface <code>es.ull.esit.app.factory_interface.IFFSampling</code>	135
6.51.1 Descripción detallada	135
6.51.2 Documentación de funciones miembro	135
6.51.2.1 createSampling()	135
6.52 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.InstanceDE</code>	136
6.52.1 Descripción detallada	137
6.52.2 Documentación de funciones miembro	137
6.52.2.1 isTerminate()	137
6.52.2.2 run()	137
6.52.2.3 setTerminate()	137
6.53 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.InstanceEE</code>	138
6.53.1 Descripción detallada	139
6.53.2 Documentación de funciones miembro	139
6.53.2.1 isTerminate()	139
6.53.2.2 run()	139
6.53.2.3 setTerminate()	139
6.54 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.InstanceGA</code>	140

6.54.1 Descripción detallada	140
6.54.2 Documentación de funciones miembro	141
6.54.2.1 isTerminate()	141
6.54.2.2 run()	141
6.54.2.3 setTerminate()	141
6.55 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.LimitRoulette</code>	141
6.55.1 Descripción detallada	142
6.55.2 Documentación de funciones miembro	142
6.55.2.1 getGenerator()	142
6.55.2.2 getLimitHigh()	142
6.55.2.3 getLimitLow()	142
6.55.2.4 setGenerator()	142
6.55.2.5 setLimitHigh()	143
6.55.2.6 setLimitLow()	143
6.56 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.LimitThreshold</code>	143
6.56.1 Descripción detallada	145
6.56.2 Documentación de constructores y destructores	145
6.56.2.1 LimitThreshold()	145
6.56.3 Documentación de funciones miembro	145
6.56.3.1 awardUpdateREF()	145
6.56.3.2 generate()	145
6.56.3.3 getGeneratorType()	146
6.56.3.4 getListCountBetterGender()	146
6.56.3.5 getListCountGender()	146
6.56.3.6 getReference()	147
6.56.3.7 getReferenceList()	147
6.56.3.8 getSonList()	147
6.56.3.9 getTrace()	147
6.56.3.10 getType()	148
6.56.3.11 getWeight()	148
6.56.3.12 setGeneratorType()	148
6.56.3.13 setInitialReference()	148
6.56.3.14 setStateRef()	149
6.56.3.15 setTypeCandidate()	149
6.56.3.16 setWeight()	149
6.56.3.17 updateReference()	149
6.57 Referencia de la clase <code>es.ull.esit.app.problem.extension.MetricasMultiobjetivo</code>	150
6.57.1 Descripción detallada	150
6.57.2 Documentación de funciones miembro	150
6.57.2.1 calcularDispersion()	150
6.57.2.2 calcularDistanciaGeneracional()	151
6.57.2.3 calcularMax()	151

6.57.2.4	calcularMedia()	151
6.57.2.5	calcularMin()	152
6.57.2.6	calcularTasaError()	152
6.58	Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing</code>	153
6.58.1	Descripción detallada	154
6.58.2	Documentación de constructores y destructores	154
6.58.2.1	<code>MultiCaseSimulatedAnnealing()</code>	154
6.58.3	Documentación de funciones miembro	154
6.58.3.1	<code>awardUpdateREF()</code>	154
6.58.3.2	<code>generate()</code>	155
6.58.3.3	<code>getListCountBetterGender()</code>	155
6.58.3.4	<code>getListCountGender()</code>	156
6.58.3.5	<code>getReference()</code>	156
6.58.3.6	<code>getReferenceList()</code>	156
6.58.3.7	<code>getSonList()</code>	157
6.58.3.8	<code>getTinitial()</code>	157
6.58.3.9	<code>getTrace()</code>	157
6.58.3.10	<code>getType()</code>	157
6.58.3.11	<code>getTypeGenerator()</code>	158
6.58.3.12	<code>getWeight()</code>	158
6.58.3.13	<code>setInitialReference()</code>	158
6.58.3.14	<code>setStateRef()</code>	158
6.58.3.15	<code>setTfinal()</code>	159
6.58.3.16	<code>setTinitial()</code>	159
6.58.3.17	<code>setTypeGenerator()</code>	159
6.58.3.18	<code>setWeight()</code>	159
6.58.3.19	<code>tfinal()</code>	160
6.58.3.20	<code>updateReference()</code>	160
6.59	Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.MultiGenerator</code>	161
6.59.1	Documentación de constructores y destructores	163
6.59.1.1	<code>MultiGenerator()</code> [1/2]	163
6.59.1.2	<code>MultiGenerator()</code> [2/2]	163
6.59.2	Documentación de funciones miembro	163
6.59.2.1	<code>awardUpdateREF()</code>	163
6.59.2.2	<code>copy()</code>	163
6.59.2.3	<code>createInstanceGeneratorsBPP()</code>	163
6.59.2.4	<code>destroyMultiGenerator()</code>	163
6.59.2.5	<code>generate()</code>	164
6.59.2.6	<code>getActiveGenerator()</code>	164
6.59.2.7	<code>getListCountBetterGender()</code>	164
6.59.2.8	<code>getListCountGender()</code>	164
6.59.2.9	<code>getListGenerators()</code>	165

6.59.2.10	getReference()	165
6.59.2.11	getReferenceList()	165
6.59.2.12	getSonList()	165
6.59.2.13	getTrace()	165
6.59.2.14	getType()	166
6.59.2.15	getWeight()	166
6.59.2.16	initializeGenerators()	166
6.59.2.17	initializeListGenerator()	166
6.59.2.18	roulette()	166
6.59.2.19	searchState()	166
6.59.2.20	setActiveGenerator()	167
6.59.2.21	setGeneratortype()	167
6.59.2.22	setInitialReference()	167
6.59.2.23	setListGeneratedPP()	167
6.59.2.24	setListGenerators()	167
6.59.2.25	setWeight()	167
6.59.2.26	tournament()	168
6.59.2.27	updateAwardImp()	168
6.59.2.28	updateAwardSC()	168
6.59.2.29	updateReference()	168
6.59.2.30	updateWeight()	169
6.59.3	Documentación de datos miembro	169
6.59.3.1	activeGenerator	169
6.59.3.2	listGeneratedPP	169
6.59.3.3	listStateReference	169
6.60	Referencia de la clase es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance	169
6.60.1	Documentación de constructores y destructores	171
6.60.1.1	MultiobjectiveHillClimbingDistance()	171
6.60.2	Documentación de funciones miembro	171
6.60.2.1	awardUpdateREF()	171
6.60.2.2	distanceCalculateAdd()	171
6.60.2.3	generate()	171
6.60.2.4	getGeneratorType()	172
6.60.2.5	getListCountBetterGender()	172
6.60.2.6	getListCountGender()	172
6.60.2.7	getReference()	172
6.60.2.8	getReferenceList()	173
6.60.2.9	getSonList()	173
6.60.2.10	getTrace()	173
6.60.2.11	getType()	173
6.60.2.12	getWeight()	174
6.60.2.13	setGeneratorType()	174

6.60.2.14 setInitialReference()	174
6.60.2.15 setStateRef()	174
6.60.2.16 setWeight()	174
6.60.2.17 updateReference()	175
6.60.3 Documentación de datos miembro	175
6.60.3.1 candidatevalue	175
6.60.3.2 distanceSolution	175
6.60.3.3 generatorType	176
6.60.3.4 ifacceptCandidate	176
6.60.3.5 listStateReference	176
6.60.3.6 listTrace	176
6.60.3.7 sizeNeighbors	176
6.60.3.8 stateReferenceHC	176
6.60.3.9 strategy	176
6.60.3.10 typeAcceptation	176
6.60.3.11 typeCandidate	176
6.60.3.12 weight	177
6.61 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart</code>	177
6.61.1 Descripción detallada	178
6.61.2 Documentación de constructores y destructores	178
6.61.2.1 <code>MultiobjectiveHillClimbingRestart()</code>	178
6.61.3 Documentación de funciones miembro	179
6.61.3.1 <code>awardUpdateREF()</code>	179
6.61.3.2 <code>generate()</code>	179
6.61.3.3 <code>getGeneratorType()</code>	179
6.61.3.4 <code>getListCountBetterGender()</code>	180
6.61.3.5 <code>getListCountGender()</code>	180
6.61.3.6 <code>getReference()</code>	180
6.61.3.7 <code>getReferenceList()</code>	180
6.61.3.8 <code>getSizeNeighbors()</code>	181
6.61.3.9 <code>getSonList()</code>	181
6.61.3.10 <code>getTrace()</code>	181
6.61.3.11 <code>getType()</code>	181
6.61.3.12 <code>getWeight()</code>	182
6.61.3.13 <code>setGeneratorType()</code>	182
6.61.3.14 <code>setInitialReference()</code>	182
6.61.3.15 <code>setSizeNeighbors()</code>	182
6.61.3.16 <code>setStateRef()</code>	183
6.61.3.17 <code>setWeight()</code>	183
6.61.3.18 <code>updateReference()</code>	183
6.62 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing</code>	184
6.62.1 Descripción detallada	185

6.62.2 Documentación de constructores y destructores	185
6.62.2.1 MultiobjectiveStochasticHillClimbing()	185
6.62.3 Documentación de funciones miembro	185
6.62.3.1 awardUpdateREF()	185
6.62.3.2 generate()	186
6.62.3.3 getGeneratorType()	186
6.62.3.4 getListCountBetterGender()	186
6.62.3.5 getListCountGender()	187
6.62.3.6 getReference()	187
6.62.3.7 getReferenceList()	187
6.62.3.8 getSonList()	187
6.62.3.9 getTrace()	188
6.62.3.10 getType()	188
6.62.3.11 getWeight()	188
6.62.3.12 setGeneratorType()	188
6.62.3.13 setInitialReference()	189
6.62.3.14 setStateRef()	189
6.62.3.15 setWeight()	189
6.62.3.16 updateReference()	189
6.63 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch</code>	190
6.63.1 Descripción detallada	192
6.63.2 Documentación de constructores y destructores	192
6.63.2.1 MultiobjectiveTabuSearch()	192
6.63.3 Documentación de funciones miembro	192
6.63.3.1 awardUpdateREF()	192
6.63.3.2 generate()	192
6.63.3.3 getListCountBetterGender()	193
6.63.3.4 getListCountGender()	193
6.63.3.5 getReference()	193
6.63.3.6 getReferenceList()	194
6.63.3.7 getSonList()	194
6.63.3.8 getStateReferenceTS()	194
6.63.3.9 getTrace()	194
6.63.3.10 getType()	195
6.63.3.11 getTypeGenerator()	195
6.63.3.12 getWeight()	195
6.63.3.13 setInitialReference()	195
6.63.3.14 setStateRef()	196
6.63.3.15 setStateReferenceTS()	196
6.63.3.16 setTypeCandidate()	196
6.63.3.17 setTypeGenerator()	196
6.63.3.18 setWeight()	197

6.63.3.19 updateReference()	197
6.64 Referencia de la clase es.ull.esit.app.problem.extension.MultiObjetivoPuro	198
6.64.1 Descripción detallada	198
6.64.2 Documentación de funciones miembro	198
6.64.2.1 evaluationState()	198
6.65 Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.Mutation	199
6.65.1 Descripción detallada	199
6.65.2 Documentación de funciones miembro	199
6.65.2.1 mutation()	199
6.66 Referencia de la clase es.ull.esit.app.problem_operators.MutationOperator	200
6.66.1 Descripción detallada	201
6.66.2 Documentación de funciones miembro	201
6.66.2.1 generatedNewState()	201
6.66.2.2 generateRandomState()	201
6.67 Referencia de la enumeración es.ull.esit.app.evolutionary_algorithms.complement.MutationType	202
6.67.1 Descripción detallada	202
6.67.2 Documentación de datos miembro	202
6.67.2.1 AIO_MUTATION	202
6.67.2.2 ONE_POINT_MUTATION	202
6.67.2.3 TOW_POINTS_MUTATION	202
6.68 Referencia de la clase es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate	203
6.68.1 Descripción detallada	203
6.68.2 Documentación de funciones miembro	204
6.68.2.1 stateSearch()	204
6.69 Referencia de la clase es.ull.esit.app.problem.definition.ObjectiveFunction	204
6.69.1 Descripción detallada	205
6.69.2 Documentación de funciones miembro	205
6.69.2.1 evaluation()	205
6.69.2.2 getTypeProblem()	205
6.69.2.3 getWeight()	205
6.69.2.4 setTypeProblem()	206
6.69.2.5 setWeight()	206
6.70 Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover	206
6.70.1 Descripción detallada	207
6.70.2 Documentación de funciones miembro	207
6.70.2.1 crossover()	207
6.71 Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation	208
6.71.1 Descripción detallada	208
6.71.2 Documentación de funciones miembro	209
6.71.2.1 mutation()	209
6.72 Referencia de la clase es.ull.esit.app.problem.definition.Operator	210
6.72.1 Descripción detallada	210

6.72.2 Documentación de funciones miembro	210
6.72.2.1 generatedNewState()	210
6.72.2.2 generateRandomState()	211
6.73 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.Particle</code>	211
6.73.1 Descripción detallada	213
6.73.2 Documentación de constructores y destructores	213
6.73.2.1 Particle() [1/2]	213
6.73.2.2 Particle() [2/2]	213
6.73.3 Documentación de funciones miembro	213
6.73.3.1 awardUpdateREF()	213
6.73.3.2 generate()	214
6.73.3.3 getListCountBetterGender()	214
6.73.3.4 getListCountGender()	214
6.73.3.5 getReference()	215
6.73.3.6 getReferenceList()	215
6.73.3.7 getSonList()	215
6.73.3.8 getStateActual()	215
6.73.3.9 getStatePBest()	216
6.73.3.10 getTrace()	216
6.73.3.11 getType()	216
6.73.3.12 getVelocity()	216
6.73.3.13 getWeight()	216
6.73.3.14 setInitialReference()	216
6.73.3.15 setStateActual()	217
6.73.3.16 setStatePBest()	217
6.73.3.17 setVelocity()	217
6.73.3.18 setWeight()	217
6.73.3.19 updateReference()	218
6.74 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization</code>	219
6.74.1 Descripción detallada	221
6.74.2 Documentación de constructores y destructores	221
6.74.2.1 ParticleSwarmOptimization()	221
6.74.3 Documentación de funciones miembro	221
6.74.3.1 awardUpdateREF()	221
6.74.3.2 gBestInicial()	222
6.74.3.3 generate()	222
6.74.3.4 getConstriction()	222
6.74.3.5 getCountCurrentIterPSO()	222
6.74.3.6 getCountParticle()	223
6.74.3.7 getCountParticleBySwarm()	223
6.74.3.8 getCountRef()	223
6.74.3.9 getGBest()	223

6.74.3.10	getGeneratorType()	224
6.74.3.11	getLBest()	224
6.74.3.12	getLearning1()	224
6.74.3.13	getLearning2()	224
6.74.3.14	getListCountBetterGender()	225
6.74.3.15	getListCountGender()	225
6.74.3.16	getListParticle()	225
6.74.3.17	getListStateReference()	225
6.74.3.18	getReference()	226
6.74.3.19	getReferenceList()	226
6.74.3.20	getSonList()	226
6.74.3.21	getStateReferencePSO()	226
6.74.3.22	getTrace()	227
6.74.3.23	getType()	227
6.74.3.24	getWeight()	227
6.74.3.25	getWmax()	227
6.74.3.26	getWmin()	228
6.74.3.27	inicialiceLBest()	228
6.74.3.28	isBinary()	228
6.74.3.29	setBinary()	228
6.74.3.30	setConstriction()	228
6.74.3.31	setCountCurrentIterPSO()	229
6.74.3.32	setCountRef()	229
6.74.3.33	setGBest()	229
6.74.3.34	setGeneratorType()	229
6.74.3.35	setInitialReference()	230
6.74.3.36	setLearning1()	230
6.74.3.37	setLearning2()	230
6.74.3.38	setListParticle()	230
6.74.3.39	setListStateReference()	231
6.74.3.40	setStateReferencePSO()	231
6.74.3.41	setWeight()	231
6.74.3.42	setWmax()	232
6.74.3.43	setWmin()	232
6.74.3.44	updateReference()	232
6.75	Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling</code>	233
6.75.1	Descripción detallada	234
6.75.2	Documentación de funciones miembro	234
6.75.2.1	listState()	234
6.75.2.2	sampling()	234
6.76	Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.Probability</code>	235
6.76.1	Descripción detallada	235

6.76.2 Documentación de funciones miembro	235
6.76.2.1 getKey()	235
6.76.2.2 getProbability()	235
6.76.2.3 getValue()	235
6.76.2.4 setKey()	235
6.76.2.5 setProbability()	236
6.76.2.6 setValue()	236
6.77 Referencia de la clase <code>es.ull.esit.app.problem.definition.Problem</code>	236
6.77.1 Descripción detallada	237
6.77.2 Documentación de constructores y destructores	237
6.77.2.1 Problem()	237
6.77.3 Documentación de funciones miembro	237
6.77.3.1 evaluate()	237
6.77.3.2 getCodification()	238
6.77.3.3 getFactorySolutionMethod()	238
6.77.3.4 getFunction()	238
6.77.3.5 getOperator()	238
6.77.3.6 getPossibleValue()	239
6.77.3.7 getState()	239
6.77.3.8 getTypeProblem()	239
6.77.3.9 getTypeSolutionMethod()	239
6.77.3.10 newSolutionMethod()	239
6.77.3.11 setCodification()	240
6.77.3.12 setFactorySolutionMethod()	240
6.77.3.13 setFunction()	240
6.77.3.14 setOperator()	241
6.77.3.15 setPossibleValue()	241
6.77.3.16 setState()	241
6.77.3.17 setTypeProblem()	241
6.77.3.18 setTypeSolutionMethod()	242
6.78 Referencia de la enumeración <code>es.ull.esit.app.problem.definition.Problem.ProblemType</code>	242
6.78.1 Documentación de datos miembro	242
6.78.1.1 MAXIMIZAR	242
6.78.1.2 MINIMIZAR	242
6.79 Referencia de la clase <code>es.ull.esit.app.local_search.candidate_type.RandomCandidate</code>	243
6.79.1 Descripción detallada	243
6.79.2 Documentación de funciones miembro	244
6.79.2.1 stateSearch()	244
6.80 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.RandomSearch</code>	244
6.80.1 Descripción detallada	246
6.80.2 Documentación de constructores y destructores	246
6.80.2.1 RandomSearch()	246

6.80.3 Documentación de funciones miembro	246
6.80.3.1 awardUpdateREF()	246
6.80.3.2 generate()	246
6.80.3.3 getListCountBetterGender()	247
6.80.3.4 getListCountGender()	247
6.80.3.5 getListStateReference()	247
6.80.3.6 getReference()	247
6.80.3.7 getReferenceList()	248
6.80.3.8 getSonList()	248
6.80.3.9 getTrace()	248
6.80.3.10 getType()	248
6.80.3.11 getTypeGenerator()	249
6.80.3.12 getWeight()	249
6.80.3.13 setInitialReference()	249
6.80.3.14 setListStateReference()	249
6.80.3.15 setTypeGenerator()	250
6.80.3.16 setWeight()	250
6.80.3.17 updateReference()	250
6.81 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.Range</code>	251
6.81.1 Descripción detallada	251
6.81.2 Documentación de funciones miembro	251
6.81.2.1 getData()	251
6.81.2.2 getMax()	251
6.81.2.3 getMin()	252
6.81.2.4 setData()	252
6.81.2.5 setMax()	252
6.81.2.6 setMin()	252
6.82 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.Replace</code>	253
6.82.1 Descripción detallada	253
6.82.2 Documentación de funciones miembro	253
6.82.2.1 replace()	253
6.83 Referencia de la enumeración <code>es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType</code>	254
6.83.1 Descripción detallada	254
6.83.2 Documentación de datos miembro	254
6.83.2.1 GENERATIONAL_REPLACE	254
6.83.2.2 STEADY_STATE_REPLACE	254
6.84 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection</code>	255
6.84.1 Descripción detallada	255
6.84.2 Documentación de funciones miembro	256
6.84.2.1 selection()	256
6.85 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.Sampling</code>	257
6.85.1 Descripción detallada	257

6.85.2 Documentación de funciones miembro	257
6.85.2.1 <code>sampling()</code>	257
6.86 Referencia de la enumeración <code>es.ull.esit.app.evolutionary_algorithms.complement.SamplingType</code>	258
6.86.1 Descripción detallada	258
6.86.2 Documentación de datos miembro	258
6.86.2.1 <code>PROBABILISTIC_SAMPLING</code>	258
6.87 Referencia de la clase <code>es.ull.esit.app.local_search.candidate_type.SearchCandidate</code>	259
6.87.1 Descripción detallada	259
6.87.2 Documentación de funciones miembro	259
6.87.2.1 <code>stateSearch()</code>	259
6.88 Referencia de la enumeración <code>es.ull.esit.app.evolutionary_algorithms.complement.SelectionType</code>	260
6.88.1 Descripción detallada	260
6.88.2 Documentación de datos miembro	260
6.88.2.1 <code>ROULETTE_SELECTION</code>	260
6.88.2.2 <code>TRUNCATION_SELECTION</code>	260
6.89 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing</code>	261
6.89.1 Descripción detallada	263
6.89.2 Documentación de constructores y destructores	263
6.89.2.1 <code>SimulatedAnnealing()</code> [1/2]	263
6.89.2.2 <code>SimulatedAnnealing()</code> [2/2]	263
6.89.3 Documentación de funciones miembro	263
6.89.3.1 <code>awardUpdateREF()</code> [1/2]	263
6.89.3.2 <code>awardUpdateREF()</code> [2/2]	264
6.89.3.3 <code>generate()</code> [1/2]	264
6.89.3.4 <code>generate()</code> [2/2]	264
6.89.3.5 <code>getAlpha()</code> [1/2]	265
6.89.3.6 <code>getAlpha()</code> [2/2]	265
6.89.3.7 <code>getCountIterationsT()</code> [1/2]	265
6.89.3.8 <code>getCountIterationsT()</code> [2/2]	265
6.89.3.9 <code>getListCountBetterGender()</code> [1/2]	266
6.89.3.10 <code>getListCountBetterGender()</code> [2/2]	266
6.89.3.11 <code>getListCountGender()</code> [1/2]	266
6.89.3.12 <code>getListCountGender()</code> [2/2]	266
6.89.3.13 <code>getReference()</code> [1/2]	267
6.89.3.14 <code>getReference()</code> [2/2]	267
6.89.3.15 <code>getReferenceList()</code> [1/2]	267
6.89.3.16 <code>getReferenceList()</code> [2/2]	267
6.89.3.17 <code>getSonList()</code> [1/2]	268
6.89.3.18 <code>getSonList()</code> [2/2]	268
6.89.3.19 <code>getTfinal()</code> [1/2]	268
6.89.3.20 <code>getTfinal()</code> [2/2]	268
6.89.3.21 <code>getTinitial()</code> [1/2]	269

6.89.3.22 getTinitial() [2/2]	269
6.89.3.23 getTrace() [1/2]	269
6.89.3.24 getTrace() [2/2]	269
6.89.3.25 getType() [1/2]	270
6.89.3.26 getType() [2/2]	270
6.89.3.27 getTypeGenerator() [1/2]	270
6.89.3.28 getTypeGenerator() [2/2]	270
6.89.3.29 getWeight() [1/2]	271
6.89.3.30 getWeight() [2/2]	271
6.89.3.31 setAlpha() [1/2]	271
6.89.3.32 setAlpha() [2/2]	271
6.89.3.33 setCountIterationsT() [1/2]	272
6.89.3.34 setCountIterationsT() [2/2]	272
6.89.3.35 setInitialReference() [1/2]	272
6.89.3.36 setInitialReference() [2/2]	272
6.89.3.37 setStateRef() [1/2]	273
6.89.3.38 setStateRef() [2/2]	273
6.89.3.39 setTfinal() [1/2]	273
6.89.3.40 setTfinal() [2/2]	273
6.89.3.41 setTinitial() [1/2]	274
6.89.3.42 setTinitial() [2/2]	274
6.89.3.43 setTypeGenerator() [1/2]	274
6.89.3.44 setTypeGenerator() [2/2]	274
6.89.3.45 setWeight() [1/2]	275
6.89.3.46 setWeight() [2/2]	275
6.89.3.47 updateReference() [1/2]	275
6.89.3.48 updateReference() [2/2]	275
6.90 Referencia de la clase <code>es.ull.esit.app.local_search.candidate_type.SmallerCandidate</code>	276
6.90.1 Descripción detallada	277
6.90.2 Documentación de funciones miembro	277
6.90.2.1 stateSearch()	277
6.91 Referencia de la clase <code>es.ull.esit.app.problem.extension.SolutionMethod</code>	278
6.91.1 Descripción detallada	278
6.91.2 Documentación de funciones miembro	278
6.91.2.1 evaluationState()	278
6.92 Referencia de la clase <code>es.ull.esit.app.problem.definition.State</code>	279
6.92.1 Descripción detallada	279
6.92.2 Documentación de constructores y destructores	279
6.92.2.1 State() [1/3]	279
6.92.2.2 State() [2/3]	280
6.92.2.3 State() [3/3]	280
6.92.3 Documentación de funciones miembro	280

6.92.3.1 comparator()	280
6.92.3.2 copy()	280
6.92.3.3 distance()	281
6.92.3.4 getCode()	281
6.92.3.5 getCopy()	281
6.92.3.6 getEvaluation()	281
6.92.3.7 getNumber()	282
6.92.3.8 getTypeGenerator()	282
6.92.3.9 setCode()	282
6.92.3.10 setEvaluation()	282
6.92.3.11 setNumber()	282
6.92.3.12 setTypeGenerator()	283
6.92.4 Documentación de datos miembro	283
6.92.4.1 code	283
6.92.4.2 evaluation	283
6.92.4.3 number	283
6.92.4.4 typeGenerator	283
6.93 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace</code>	284
6.93.1 Descripción detallada	284
6.93.2 Documentación de funciones miembro	285
6.93.2.1 max_value()	285
6.93.2.2 min_value()	285
6.93.2.3 replace()	285
6.94 Referencia de la clase <code>es.ull.esit.app.local_search.complement.StopExecute</code>	286
6.94.1 Descripción detallada	286
6.94.2 Documentación de funciones miembro	286
6.94.2.1 stopIterations()	286
6.95 Referencia de la clase <code>es.ull.esit.app.metaheuristics.strategy.Strategy</code>	287
6.95.1 Descripción detallada	288
6.95.2 Documentación de funciones miembro	288
6.95.2.1 calculateOffLinePerformance()	288
6.95.2.2 destroyExecute()	288
6.95.2.3 executeStrategy()	288
6.95.2.4 getBestState()	289
6.95.2.5 getCountCurrent()	289
6.95.2.6 getCountMax()	289
6.95.2.7 getGenerator()	289
6.95.2.8 getListKey()	290
6.95.2.9 getListRefPoblacFinal()	290
6.95.2.10 getListStates()	290
6.95.2.11 getMapGenerators()	290
6.95.2.12 getProblem()	290

6.95.2.13	getStopexecute()	291
6.95.2.14	getStrategy()	291
6.95.2.15	getThreshold()	291
6.95.2.16	getUpdateparameter()	291
6.95.2.17	initialize()	291
6.95.2.18	initializeGenerators()	292
6.95.2.19	newGenerator()	292
6.95.2.20	setBestState()	293
6.95.2.21	setCountCurrent()	293
6.95.2.22	setCountMax()	293
6.95.2.23	setGenerator()	294
6.95.2.24	setListRefPoblacFinal()	294
6.95.2.25	setListStates()	294
6.95.2.26	setMapGenerators()	294
6.95.2.27	setProblem()	295
6.95.2.28	setStopexecute()	295
6.95.2.29	setThreshold()	295
6.95.2.30	setUpdateparameter()	295
6.95.2.31	update()	296
6.95.2.32	updateCountGender()	296
6.95.2.33	updateRef()	296
6.95.2.34	updateRefGenerator()	296
6.95.2.35	updateRefMultiG()	297
6.95.2.36	updateWeight()	297
6.96	Referencia de la clase es.ull.esit.app.strategy.Strategy	297
6.96.1	Descripción detallada	298
6.96.2	Documentación de funciones miembro	298
6.96.2.1	calculateOffLinePerformance()	298
6.96.2.2	destroyExecute()	298
6.96.2.3	executeStrategy()	298
6.96.2.4	getBestState()	299
6.96.2.5	getCountCurrent()	299
6.96.2.6	getCountMax()	299
6.96.2.7	getListKey()	300
6.96.2.8	getProblem()	300
6.96.2.9	getStopexecute()	300
6.96.2.10	getStrategy()	300
6.96.2.11	getThreshold()	300
6.96.2.12	getUpdateparameter()	301
6.96.2.13	initialize()	301
6.96.2.14	initializeGenerators()	301
6.96.2.15	newGenerator()	302

6.96.2.16 setBestState()	302
6.96.2.17 setCountCurrent()	302
6.96.2.18 setCountMax()	303
6.96.2.19 setProblem()	303
6.96.2.20 setStopexecute()	303
6.96.2.21 setThreshold()	303
6.96.2.22 setUpdateparameter()	304
6.96.2.23 update()	304
6.96.2.24 updateCountGender()	304
6.96.2.25 updateRef()	304
6.96.2.26 updateRefGenerator()	305
6.96.2.27 updateRefMultiG()	305
6.96.2.28 updateWeight()	305
6.96.3 Documentación de datos miembro	305
6.96.3.1 generator	305
6.96.3.2 notDominated	305
6.96.3.3 threshold	306
6.97 Referencia de la enumeración <code>es.ull.esit.app.local_search.complement.StrategyType</code>	306
6.97.1 Descripción detallada	306
6.97.2 Documentación de datos miembro	306
6.97.2.1 NORMAL	306
6.97.2.2 TABU	306
6.98 Referencia de la clase <code>es.ull.esit.app.metaheuristics.generators.TabuSearch</code>	307
6.98.1 Descripción detallada	308
6.98.2 Documentación de constructores y destructores	308
6.98.2.1 TabuSearch()	308
6.98.3 Documentación de funciones miembro	308
6.98.3.1 awardUpdateREF()	308
6.98.3.2 generate()	309
6.98.3.3 getListCountBetterGender()	309
6.98.3.4 getListCountGender()	309
6.98.3.5 getReference()	310
6.98.3.6 getReferenceList()	310
6.98.3.7 getSonList()	310
6.98.3.8 getTrace()	310
6.98.3.9 getType()	311
6.98.3.10 getTypeGenerator()	311
6.98.3.11 getWeight()	311
6.98.3.12 setInitialReference()	311
6.98.3.13 setStateRef()	312
6.98.3.14 setTypeCandidate()	312
6.98.3.15 setTypeGenerator()	312

6.98.3.16 setWeight()	312
6.98.3.17 updateReference()	313
6.99 Referencia de la clase <code>es.ull.esit.app.local_search.complement.TabuSolutions</code>	313
6.99.1 Descripción detallada	313
6.99.2 Documentación de funciones miembro	313
6.99.2.1 filterNeighborhood()	313
6.99.3 Documentación de datos miembro	314
6.99.3.1 listTabu	314
6.99.3.2 maxelements	314
6.100 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation</code>	314
6.100.1 Descripción detallada	315
6.100.2 Documentación de funciones miembro	315
6.100.2.1 mutation()	315
6.101 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection</code>	316
6.101.1 Descripción detallada	316
6.101.2 Documentación de funciones miembro	317
6.101.2.1 ascOrderBetter()	317
6.101.2.2 orderBetter()	317
6.101.2.3 selection()	317
6.102 Referencia de la clase <code>es.ull.esit.app.config.tspdynamic.TSPState</code>	318
6.102.1 Descripción detallada	318
6.102.2 Documentación de funciones miembro	318
6.102.2.1 getIdCity()	318
6.102.2.2 getValue()	318
6.102.2.3 setIdCity()	318
6.102.2.4 setValue()	319
6.103 Referencia de la enumeración <code>es.ull.esit.app.problem.extension.TypeSolutionMethod</code>	319
6.103.1 Documentación de datos miembro	319
6.103.1.1 FactoresPonderados	319
6.103.1.2 MultiObjetivoPuro	319
6.104 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover</code>	320
6.104.1 Descripción detallada	320
6.104.2 Documentación de funciones miembro	321
6.104.2.1 crossover()	321
6.104.2.2 mascara()	321
6.105 Referencia de la clase <code>es.ull.esit.app.evolutionary_algorithms.complement.Univariate</code>	322
6.105.1 Descripción detallada	322
6.105.2 Documentación de funciones miembro	323
6.105.2.1 distribution()	323
6.105.2.2 getListKey()	323
6.106 Referencia de la clase <code>es.ull.esit.app.local_search.complement.UpdateParameter</code>	323
6.106.1 Descripción detallada	324

6.106.2 Documentación de funciones miembro	324
6.106.2.1 updateParameter()	324
7 Documentación de archivos	325
7.1 Referencia del archivo src/main/java/es/ull/esit/app/config/tspdynamic/TSPState.java	325
7.2 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ AIOMutation.java	325
7.3 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Crossover.java	326
7.4 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Crossover↵ Type.java	326
7.5 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ Distribution.java	327
7.6 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ DistributionType.java	327
7.7 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Father↵ Selection.java	328
7.8 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ GenerationalReplace.java	328
7.9 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation.java	329
7.10 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation↵ Type.java	329
7.11 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoint↵ Crossover.java	330
7.12 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoint↵ Mutation.java	330
7.13 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ ProbabilisticSampling.java	331
7.14 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ Probability.java	332
7.15 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Range.java	332
7.16 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replace.java	332
7.17 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replace↵ Type.java	333
7.18 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Roulette↵ Selection.java	333
7.19 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ Sampling.java	334
7.20 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Sampling↵ Type.java	334
7.21 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Selection↵ Type.java	335
7.22 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Steady↵ StateReplace.java	335
7.23 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Tow↵ PointsMutation.java	336
7.24 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ TruncationSelection.java	336

7.25 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Truncation← Selection.java	337
7.26 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Uniform← Crossover.java	338
7.27 Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/← Univariate.java	338
7.28 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryAcceptCandidate.java	339
7.29 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryCandidate.java . . .	340
7.30 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryCrossover.java	340
7.31 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryDistribution.java . . .	341
7.32 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryFatherSelection.java .	342
7.33 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryGenerator.java	342
7.34 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryMutation.java	343
7.35 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactoryReplace.java	344
7.36 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFactorySolutionMethod.java .	344
7.37 Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFSampling.java	345
7.38 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryAcceptCandidate.java .	346
7.39 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryCandidate.java	346
7.40 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryCrossover.java	347
7.41 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryDistribution.java	348
7.42 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryFatherSelection.java . .	348
7.43 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryGenerator.java	349
7.44 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryLoader.java	350
7.45 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryMutation.java	350
7.46 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryReplace.java	351
7.47 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactorySampling.java	352
7.48 Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactorySolutionMethod.java . .	352
7.49 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/Acceptable← Candidate.java	353
7.50 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptAnyone.java	354
7.51 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptBest.java .	354
7.52 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/Accept← Multicase.java	355
7.53 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBad.java	356
7.54 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadT.java	356
7.55 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNot← BadU.java	357
7.56 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNot← Dominated.java	358
7.57 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNot← DominatedTabu.java	358
7.58 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptType.java	359
7.59 Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/Dominance.java .	359
7.60 Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/CandidateType.java	360

7.61 Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/CandidateValue.java	360
7.62 Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/GreaterCandidate.java	361
7.63 Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/NotDominatedCandidate.java	361
7.64 Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/RandomCandidate.java	362
7.65 Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/SearchCandidate.java	363
7.66 Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/SmallerCandidate.java	363
7.67 Referencia del archivo src/main/java/es/ull/esit/app/local_search/complement/StopExecute.java	364
7.68 Referencia del archivo src/main/java/es/ull/esit/app/local_search/complement/StrategyType.java	364
7.69 Referencia del archivo src/main/java/es/ull/esit/app/local_search/complement/TabuSolutions.java	364
7.70 Referencia del archivo src/main/java/es/ull/esit/app/local_search/complement/UpdateParameter.java	365
7.71 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/DistributionEstimationAlgorithm.java	366
7.72 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/EvolutionStrategies.java	366
7.73 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/Generator.java	367
7.74 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/GeneratorType.java	368
7.75 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/GeneticAlgorithm.java	368
7.76 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbing.java	369
7.77 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbingRestart.java	369
7.78 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceDE.java	370
7.79 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceEE.java	371
7.80 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceGA.java	372
7.81 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/LimitRoulette.java	372
7.82 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/LimitThreshold.java	373
7.83 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/MultiCaseSimulatedAnnealing.java	373
7.84 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/MultiGenerator.java	374
7.85 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingDistance.java	375
7.86 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingRestart.java	376
7.87 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveStochasticHillClimbing.java	376
7.88 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveTabuSearch.java	377
7.89 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/Particle.java	378
7.90 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/ParticleSwarmOptimization.java	379
7.91 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/RandomSearch.java	379
7.92 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java	380
7.93 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java	381

7.94 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/TabuSearch.java . . .	382
7.95 Referencia del archivo src/main/java/es/ull/esit/app/problem/definition/Codification.java	382
7.96 Referencia del archivo src/main/java/es/ull/esit/app/problem/definition/ObjectiveFunction.java	383
7.97 Referencia del archivo src/main/java/es/ull/esit/app/problem/definition/Operator.java	383
7.98 Referencia del archivo src/main/java/es/ull/esit/app/problem/definition/Problem.java	384
7.99 Referencia del archivo src/main/java/es/ull/esit/app/problem/definition/State.java	385
7.100 Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/FactoresPonderados.java .	385
7.101 Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/MetricasMultiobjetivo.java .	386
7.102 Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/MultiObjetivoPuro.java . . .	387
7.103 Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/SolutionMethod.java	387
7.104 Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/TypeSolutionMethod.java . .	388
7.105 Referencia del archivo src/test/java/es/ull/esit/app/problem/extension/TypeSolutionMethod.java . .	388
7.106 Referencia del archivo src/main/java/es/ull/esit/app/problem_operators/MutationOperator.java . . .	389
7.107 Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/strategy/Strategy.java	389
7.108 Referencia del archivo src/main/java/es/ull/esit/app/strategy/Strategy.java	390
7.109 Referencia del archivo src/test/java/es/ull/esit/app/config/tspdynamic/TSPStateTest.java	391
7.110 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ AIOMutationTest.java	391
7.111 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ GenerationalReplaceTest.java	392
7.112 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoint↵ CrossoverTest.java	393
7.113 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoint↵ MutationTest.java	394
7.114 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ ProbabilisticSamplingTest.java	394
7.115 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ ProbabilityTest.java	395
7.116 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Range↵ Test.java	396
7.117 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Roulette↵ SelectionTest.java	396
7.118 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Steady↵ StateReplaceTest.java	397
7.119 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Tow↵ PointsMutationTest.java	398
7.120 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Uniform↵ CrossverTest.java	398
7.121 Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/↵ UnivarariteTest.java	399
7.122 Referencia del archivo src/test/java/es/ull/esit/app/factory_interface/FactoryInterfacesTest.java . . .	400
7.123 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryAcceptCandidateTest.java	400
7.124 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryCandidateTest.java . . .	401
7.125 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryCrossoverTest.java . . .	402
7.126 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryDamplingTest.java . . .	402

7.127 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryDistributionTest.java . . .	403
7.128 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryFatherSelectionTest.java	404
7.129 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryGeneratorTest.java . . .	404
7.130 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryLoaderTest.java	405
7.131 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryMutationTest.java	406
7.132 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryReplaceTest.java	406
7.133 Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactorySotutionMethodTest.java	407
7.134 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/ AcceptNotBad← Test.java	408
7.135 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/Acceptable← CandidateTest.java	408
7.136 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptAnyone← Test.java	409
7.137 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptBestTest.java	410
7.138 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/Accept← MulticaseTest.java	411
7.139 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBad← TTest.java	411
7.140 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBad← UTest.java	412
7.141 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNot← DominatedTabuTest.java	413
7.142 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNot← DominatedTest.java	413
7.143 Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/DominanceTest.java	414
7.144 Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/CandidateValue← Test.java	415
7.145 Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/Greater← CandidateTest.java	415
7.146 Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/NotDominated← CandidatedTest.java	416
7.147 Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/Random← CandidateTest.java	417
7.148 Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/Smaller← CandidateTest.java	417
7.149 Referencia del archivo src/test/java/es/ull/esit/app/local_search/complement/StopExecuteTest.java	418
7.150 Referencia del archivo src/test/java/es/ull/esit/app/local_search/complement/TabuSolutionTest.java	419
7.151 Referencia del archivo src/test/java/es/ull/esit/app/local_search/complement/UpdateParameter← Test.java	419
7.152 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/Distribution← EstimationAlgorithmTest.java	420
7.153 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/EvolutionStrategies← Test.java	421
7.154 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/GeneratorTest.java	421
7.155 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/GenericAlgorithms← Test.java	422

7.156 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/HillClimbingRestart↵ Test.java	423
7.157 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/HillClimbingTest.java	423
7.158 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/InstanceDETest.java	424
7.159 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/InstanceEETest.java	425
7.160 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/InstanceGATest.java	425
7.161 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/LimitRouletteTest.java	426
7.162 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/LimitThresholdTest.java	427
7.163 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/MultiCaseSimulated↵ AnnealingTest.java	427
7.164 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/Multiobjective↵ ClimbingRestartTet.java	428
7.165 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/Multiobjective↵ StochasticHillClimbingTest.java	429
7.166 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveTabu↵ SearchTest.java	429
7.167 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/ParticleSwarm↵ OptimizationTest.java	430
7.168 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/ParticleTest.java	431
7.169 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/RandomSearchTest.java	432
7.170 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/generators/TabuSearchTest.java	432
7.171 Referencia del archivo src/test/java/es/ull/esit/app/problem/definition/CodificationTest.java	433
7.172 Referencia del archivo src/test/java/es/ull/esit/app/problem/definition/ObjectiveFunctionTest.java	434
7.173 Referencia del archivo src/test/java/es/ull/esit/app/problem/definition/OperatorTest.java	434
7.174 Referencia del archivo src/test/java/es/ull/esit/app/problem/definition/ProblemTest.java	435
7.175 Referencia del archivo src/test/java/es/ull/esit/app/problem/definition/StateTest.java	436
7.176 Referencia del archivo src/test/java/es/ull/esit/app/problem/extension/FactoresPonderadosTest.java	436
7.177 Referencia del archivo src/test/java/es/ull/esit/app/problem/extension/MetricasMultiobjetivoTest.java	437
7.178 Referencia del archivo src/test/java/es/ull/esit/app/problem/extension/SolutionMethodTest.java	438
7.179 Referencia del archivo src/test/java/es/ull/esit/app/problem_operators/MutationOperatorTest.java	438
7.180 Referencia del archivo src/test/java/es/ull/esit/app/metaheuristics/strategy/StrategyTest.java	439
7.181 Referencia del archivo src/test/java/es/ull/esit/app/strategy/StrategyTest.java	440

Índice alfabético

441

Capítulo 1

Índice de espacios de nombres

1.1. Lista de paquetes

Estos son los paquetes con breves descripciones (si están disponibles):

es.ull.esit.app.config.tspdynamic	15
es.ull.esit.app.evolutionary_algorithms.complement	15
es.ull.esit.app.factory_interface	16
es.ull.esit.app.factory_method	16
es.ull.esit.app.local_search.acceptation_type	17
es.ull.esit.app.local_search.candidate_type	17
es.ull.esit.app.local_search.complement	18
es.ull.esit.app.metaheuristics.strategy	18
es.ull.esit.app.metaheuristics.generators	18
es.ull.esit.app.problem.definition	19
es.ull.esit.app.problem.extension	19
es.ull.esit.app.problem_operators	20
es.ull.esit.app.strategy	20

Capítulo 2

Índice jerárquico

2.1. Jerarquía de clases

Este listado de herencia está ordenado de forma general pero no está en orden alfabético estricto:

es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate	21
es.ull.esit.app.local_search.acceptation_type.AcceptAnyone	23
es.ull.esit.app.local_search.acceptation_type.AcceptBest	24
es.ull.esit.app.local_search.acceptation_type.AcceptMulticase	26
es.ull.esit.app.local_search.acceptation_type.AcceptNotBad	28
es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT	29
es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU	31
es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated	32
es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu	34
es.ull.esit.app.local_search.acceptation_type.AcceptType	35
es.ull.esit.app.local_search.candidate_type.CandidateType	39
es.ull.esit.app.local_search.candidate_type.CandidateValue	40
es.ull.esit.app.problem.definition.Codification	43
es.ull.esit.app.evolutionary_algorithms.complement.Crossover	45
es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover	206
es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover	320
es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType	46
es.ull.esit.app.evolutionary_algorithms.complement.Distribution	46
es.ull.esit.app.evolutionary_algorithms.complement.Univariate	322
es.ull.esit.app.evolutionary_algorithms.complement.DistributionType	57
es.ull.esit.app.local_search.acceptation_type.Dominance	58
es.ull.esit.app.factory_method.FactoryLoader	79
es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection	87
es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection	255
es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection	316
es.ull.esit.app.metaheuristics.generators.Generator	91
es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm	47
es.ull.esit.app.metaheuristics.generators.EvolutionStrategies	59
es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm	101
es.ull.esit.app.metaheuristics.generators.HillClimbing	112
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart	118
es.ull.esit.app.metaheuristics.generators.LimitThreshold	143
es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing	153
es.ull.esit.app.metaheuristics.generators.MultiGenerator	161

es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance	169
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart	177
es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing	184
es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch	190
es.ull.esit.app.metaheuristics.generators.Particle	211
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization	219
es.ull.esit.app.metaheuristics.generators.RandomSearch	244
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing	261
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing	261
es.ull.esit.app.metaheuristics.generators.TabuSearch	307
es.ull.esit.app.metaheuristics.generators.GeneratorType	98
es.ull.esit.app.factory_interface.IFFactoryAcceptCandidate	126
IFFactoryAcceptCandidate	
es.ull.esit.app.factory_method.FactoryAcceptCandidate	69
es.ull.esit.app.factory_interface.IFFactoryCandidate	127
IFFactoryCandidate	
es.ull.esit.app.factory_method.FactoryCandidate	70
es.ull.esit.app.factory_interface.IFFactoryCrossover	128
IFFactoryCrossover	
es.ull.esit.app.factory_method.FactoryCrossover	72
es.ull.esit.app.factory_interface.IFFactoryDistribution	129
IFFactoryDistribution	
es.ull.esit.app.factory_method.FactoryDistribution	74
es.ull.esit.app.factory_interface.IFFactoryFatherSelection	130
IFFactoryFatherSelection	
es.ull.esit.app.factory_method.FactoryFatherSelection	76
es.ull.esit.app.factory_interface.IFFactoryGenerator	131
IFFactoryGenerator	
es.ull.esit.app.factory_method.FactoryGenerator	78
es.ull.esit.app.factory_interface.IFFactoryMutation	132
IFFactoryMutation	
es.ull.esit.app.factory_method.FactoryMutation	80
es.ull.esit.app.factory_interface.IFFactoryReplace	133
IFFactoryReplace	
es.ull.esit.app.factory_method.FactoryReplace	82
es.ull.esit.app.factory_interface.IFFactorySolutionMethod	134
IFFactorySolutionMethod	
es.ull.esit.app.factory_method.FactorySolutionMethod	86
es.ull.esit.app.factory_interface.IFFSampling	135
IFFSampling	
es.ull.esit.app.factory_method.FactorySampling	84
es.ull.esit.app.metaheuristics.generators.LimitRoulette	141
es.ull.esit.app.problem.extension.MetricasMultiobjetivo	150
es.ull.esit.app.evolutionary_algorithms.complement.Mutation	199
es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation	37
es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation	208
es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation	314
es.ull.esit.app.evolutionary_algorithms.complement.MutationType	202
es.ull.esit.app.problem.definition.ObjetivoFunction	204
es.ull.esit.app.problem.definition.Operator	210
Operator	
es.ull.esit.app.problem_operators.MutationOperator	200
es.ull.esit.app.evolutionary_algorithms.complement.Probability	235
es.ull.esit.app.problem.definition.Problem	236
es.ull.esit.app.problem.definition.Problem.ProblemType	242
es.ull.esit.app.evolutionary_algorithms.complement.Range	251
es.ull.esit.app.evolutionary_algorithms.complement.Replace	253
es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace	89

es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace	284
es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType	254
Runnable	
es.ull.esit.app.metaheuristics.generators.InstanceDE	136
es.ull.esit.app.metaheuristics.generators.InstanceEE	138
es.ull.esit.app.metaheuristics.generators.InstanceGA	140
es.ull.esit.app.evolutionary_algorithms.complement.Sampling	257
es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling	233
es.ull.esit.app.evolutionary_algorithms.complement.SamplingType	258
es.ull.esit.app.local_search.candidate_type.SearchCandidate	259
es.ull.esit.app.local_search.candidate_type.GreaterCandidate	110
es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate	203
es.ull.esit.app.local_search.candidate_type.RandomCandidate	243
es.ull.esit.app.local_search.candidate_type.SmallerCandidate	276
es.ull.esit.app.evolutionary_algorithms.complement.SelectionType	260
es.ull.esit.app.problem.extension.SolutionMethod	278
es.ull.esit.app.problem.extension.FactoresPonderados	68
es.ull.esit.app.problem.extension.MultiObjetivoPuro	198
es.ull.esit.app.problem.definition.State	279
es.ull.esit.app.local_search.complement.StopExecute	286
es.ull.esit.app.metaheuristics.strategy.Strategy	287
es.ull.esit.app.strategy.Strategy	297
es.ull.esit.app.local_search.complement.StrategyType	306
es.ull.esit.app.local_search.complement.TabuSolutions	313
es.ull.esit.app.config.tspdynamic.TSPState	318
es.ull.esit.app.problem.extension.TypeSolutionMethod	319
es.ull.esit.app.local_search.complement.UpdateParameter	323

Capítulo 3

Índice de clases

3.1. Lista de clases

Lista de clases, estructuras, uniones e interfaces con breves descripciones:

es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate	21
es.ull.esit.app.local_search.acceptation_type.AcceptAnyone	23
es.ull.esit.app.local_search.acceptation_type.AcceptBest	24
es.ull.esit.app.local_search.acceptation_type.AcceptMulticase	26
es.ull.esit.app.local_search.acceptation_type.AcceptNotBad	28
es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT	29
es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU	31
es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated	32
es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu	34
es.ull.esit.app.local_search.acceptation_type.AcceptType	35
es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation	37
es.ull.esit.app.local_search.candidate_type.CandidateType	39
es.ull.esit.app.local_search.candidate_type.CandidateValue	40
es.ull.esit.app.problem.definition.Codification	43
es.ull.esit.app.evolutionary_algorithms.complement.Crossover	45
es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType	46
es.ull.esit.app.evolutionary_algorithms.complement.Distribution	46
es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm	47
es.ull.esit.app.evolutionary_algorithms.complement.DistributionType	57
es.ull.esit.app.local_search.acceptation_type.Dominance	58
es.ull.esit.app.metaheuristics.generators.EvolutionStrategies	59
es.ull.esit.app.problem.extension.FactoresPonderados	68
es.ull.esit.app.factory_method.FactoryAcceptCandidate	69
es.ull.esit.app.factory_method.FactoryCandidate	70
es.ull.esit.app.factory_method.FactoryCrossover	72
es.ull.esit.app.factory_method.FactoryDistribution	74
es.ull.esit.app.factory_method.FactoryFatherSelection	76
es.ull.esit.app.factory_method.FactoryGenerator	78
es.ull.esit.app.factory_method.FactoryLoader	79
es.ull.esit.app.factory_method.FactoryMutation	80
es.ull.esit.app.factory_method.FactoryReplace	82
es.ull.esit.app.factory_method.FactorySampling	84
es.ull.esit.app.factory_method.FactorySolutionMethod	86
es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection	87
es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace	89

es.ull.esit.app.metaheuristics.generators.Generator	91
es.ull.esit.app.metaheuristics.generators.GeneratorType	98
es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm	101
es.ull.esit.app.local_search.candidate_type.GreaterCandidate	110
es.ull.esit.app.metaheuristics.generators.HillClimbing	112
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart	118
es.ull.esit.app.factory_interface.IFFactoryAcceptCandidate	126
es.ull.esit.app.factory_interface.IFFactoryCandidate	127
es.ull.esit.app.factory_interface.IFFactoryCrossover	128
es.ull.esit.app.factory_interface.IFFactoryDistribution	129
es.ull.esit.app.factory_interface.IFFactoryFatherSelection	130
es.ull.esit.app.factory_interface.IFFactoryGenerator	131
es.ull.esit.app.factory_interface.IFFactoryMutation	132
es.ull.esit.app.factory_interface.IFFactoryReplace	133
es.ull.esit.app.factory_interface.IFFactorySolutionMethod	134
es.ull.esit.app.factory_interface.IFFSampling	135
es.ull.esit.app.metaheuristics.generators.InstanceDE	136
es.ull.esit.app.metaheuristics.generators.InstanceEE	138
es.ull.esit.app.metaheuristics.generators.InstanceGA	140
es.ull.esit.app.metaheuristics.generators.LimitRoulette	141
es.ull.esit.app.metaheuristics.generators.LimitThreshold	143
es.ull.esit.app.problem.extension.MetricasMultiobjetivo	150
es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing	153
es.ull.esit.app.metaheuristics.generators.MultiGenerator	161
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance	169
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart	177
es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing	184
es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch	190
es.ull.esit.app.problem.extension.MultiObjetivoPuro	198
es.ull.esit.app.evolutionary_algorithms.complement.Mutation	199
es.ull.esit.app.problem_operators.MutationOperator	200
es.ull.esit.app.evolutionary_algorithms.complement.MutationType	202
es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate	203
es.ull.esit.app.problem.definition.ObjectiveFunction	204
es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover	206
es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation	208
es.ull.esit.app.problem.definition.Operator	210
es.ull.esit.app.metaheuristics.generators.Particle	211
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization	219
es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling	233
es.ull.esit.app.evolutionary_algorithms.complement.Probability	235
es.ull.esit.app.problem.definition.Problem	236
es.ull.esit.app.problem.definition.Problem.ProblemType	242
es.ull.esit.app.local_search.candidate_type.RandomCandidate	243
es.ull.esit.app.metaheuristics.generators.RandomSearch	244
es.ull.esit.app.evolutionary_algorithms.complement.Range	251
es.ull.esit.app.evolutionary_algorithms.complement.Replace	253
es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType	254
es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection	255
es.ull.esit.app.evolutionary_algorithms.complement.Sampling	257
es.ull.esit.app.evolutionary_algorithms.complement.SamplingType	258
es.ull.esit.app.local_search.candidate_type.SearchCandidate	259
es.ull.esit.app.evolutionary_algorithms.complement.SelectionType	260
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing	261
es.ull.esit.app.local_search.candidate_type.SmallerCandidate	276
es.ull.esit.app.problem.extension.SolutionMethod	278
es.ull.esit.app.problem.definition.State	279
es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace	284

es.ull.esit.app.local_search.complement.StopExecute	286
es.ull.esit.app.metaheuristics.strategy.Strategy	287
es.ull.esit.app.strategy.Strategy	297
es.ull.esit.app.local_search.complement.StrategyType	306
es.ull.esit.app.metaheuristics.generators.TabuSearch	307
es.ull.esit.app.local_search.complement.TabuSolutions	313
es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation	314
es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection	316
es.ull.esit.app.config.tspdynamic.TSPState	318
es.ull.esit.app.problem.extension.TypeSolutionMethod	319
es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover	320
es.ull.esit.app.evolutionary_algorithms.complement.Univariate	322
es.ull.esit.app.local_search.complement.UpdateParameter	323

Capítulo 4

Índice de archivos

4.1. Lista de archivos

Lista de todos los archivos con breves descripciones:

src/main/java/es/ull/esit/app/config/tspdynamic/TSPState.java	325
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/AIOMutation.java	325
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Crossover.java	326
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/CrossoverType.java	326
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Distribution.java	327
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/DistributionType.java	327
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/FatherSelection.java	328
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/GenerationalReplace.java	328
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation.java	329
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/MutationType.java	329
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointCrossover.java	330
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointMutation.java	330
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/ProbabilisticSampling.java	331
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Probability.java	332
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Range.java	332
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replace.java	332
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/ReplaceType.java	333
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/RouletteSelection.java	333
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Sampling.java	334
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SamplingType.java	334
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SelectionType.java	335
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SteadyStateReplace.java	335
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/TowPointsMutation.java	336
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/TruncationSelection.java	336
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/UniformCrossover.java	338
src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Univariate.java	338
src/main/java/es/ull/esit/app/factory_interface/IFFactoryAcceptCandidate.java	339
src/main/java/es/ull/esit/app/factory_interface/IFFactoryCandidate.java	340
src/main/java/es/ull/esit/app/factory_interface/IFFactoryCrossover.java	340
src/main/java/es/ull/esit/app/factory_interface/IFFactoryDistribution.java	341
src/main/java/es/ull/esit/app/factory_interface/IFFactoryFatherSelection.java	342
src/main/java/es/ull/esit/app/factory_interface/IFFactoryGenerator.java	342
src/main/java/es/ull/esit/app/factory_interface/IFFactoryMutation.java	343
src/main/java/es/ull/esit/app/factory_interface/IFFactoryReplace.java	344
src/main/java/es/ull/esit/app/factory_interface/IFFactorySolutionMethod.java	344

src/main/java/es/ull/esit/app/factory_interface/IFFSampling.java	345
src/main/java/es/ull/esit/app/factory_method/FactoryAcceptCandidate.java	346
src/main/java/es/ull/esit/app/factory_method/FactoryCandidate.java	346
src/main/java/es/ull/esit/app/factory_method/FactoryCrossover.java	347
src/main/java/es/ull/esit/app/factory_method/FactoryDistribution.java	348
src/main/java/es/ull/esit/app/factory_method/FactoryFatherSelection.java	348
src/main/java/es/ull/esit/app/factory_method/FactoryGenerator.java	349
src/main/java/es/ull/esit/app/factory_method/FactoryLoader.java	350
src/main/java/es/ull/esit/app/factory_method/FactoryMutation.java	350
src/main/java/es/ull/esit/app/factory_method/FactoryReplace.java	351
src/main/java/es/ull/esit/app/factory_method/FactorySampling.java	352
src/main/java/es/ull/esit/app/factory_method/FactorySolutionMethod.java	352
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptableCandidate.java	353
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptAnyone.java	354
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptBest.java	354
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptMulticase.java	355
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBad.java	356
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadT.java	356
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadU.java	357
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominated.java	358
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominatedTabu.java	358
src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptType.java	359
src/main/java/es/ull/esit/app/local_search/acceptation_type/Dominance.java	359
src/main/java/es/ull/esit/app/local_search/candidate_type/CandidateType.java	360
src/main/java/es/ull/esit/app/local_search/candidate_type/CandidateValue.java	360
src/main/java/es/ull/esit/app/local_search/candidate_type/GreaterCandidate.java	361
src/main/java/es/ull/esit/app/local_search/candidate_type/NotDominatedCandidate.java	361
src/main/java/es/ull/esit/app/local_search/candidate_type/RandomCandidate.java	362
src/main/java/es/ull/esit/app/local_search/candidate_type/SearchCandidate.java	363
src/main/java/es/ull/esit/app/local_search/candidate_type/SmallerCandidate.java	363
src/main/java/es/ull/esit/app/local_search/complement/StopExecute.java	364
src/main/java/es/ull/esit/app/local_search/complement/StrategyType.java	364
src/main/java/es/ull/esit/app/local_search/complement/TabuSolutions.java	364
src/main/java/es/ull/esit/app/local_search/complement/UpdateParameter.java	365
src/main/java/es/ull/esit/app/metaheuristics/strategy/Strategy.java	389
src/main/java/es/ull/esit/app/metaheuristics/generators/DistributionEstimationAlgorithm.java	366
src/main/java/es/ull/esit/app/metaheuristics/generators/EvolutionStrategies.java	366
src/main/java/es/ull/esit/app/metaheuristics/generators/Generator.java	367
src/main/java/es/ull/esit/app/metaheuristics/generators/GeneratorType.java	368
src/main/java/es/ull/esit/app/metaheuristics/generators/GeneticAlgorithm.java	368
src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbing.java	369
src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbingRestart.java	369
src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceDE.java	370
src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceEE.java	371
src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceGA.java	372
src/main/java/es/ull/esit/app/metaheuristics/generators/LimitRoulette.java	372
src/main/java/es/ull/esit/app/metaheuristics/generators/LimitThreshold.java	373
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiCaseSimulatedAnnealing.java	373
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiGenerator.java	374
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingDistance.java	375
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingRestart.java	376
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveStochasticHillClimbing.java	376
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveTabuSearch.java	377
src/main/java/es/ull/esit/app/metaheuristics/generators/Particle.java	378
src/main/java/es/ull/esit/app/metaheuristics/generators/ParticleSwarmOptimization.java	379
src/main/java/es/ull/esit/app/metaheuristics/generators/RandomSearch.java	379
src/main/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java	380
src/main/java/es/ull/esit/app/metaheuristics/generators/TabuSearch.java	382

src/main/java/es/ull/esit/app/problem/definition/Codification.java	382
src/main/java/es/ull/esit/app/problem/definition/ObjetiveFunction.java	383
src/main/java/es/ull/esit/app/problem/definition/Operator.java	383
src/main/java/es/ull/esit/app/problem/definition/Problem.java	384
src/main/java/es/ull/esit/app/problem/definition/State.java	385
src/main/java/es/ull/esit/app/problem/extension/FactoresPonderados.java	385
src/main/java/es/ull/esit/app/problem/extension/MetricasMultiobjetivo.java	386
src/main/java/es/ull/esit/app/problem/extension/MultiObjetivoPuro.java	387
src/main/java/es/ull/esit/app/problem/extension/SolutionMethod.java	387
src/main/java/es/ull/esit/app/problem/extension/TypeSolutionMethod.java	388
src/main/java/es/ull/esit/app/problem_operators/MutationOperator.java	389
src/main/java/es/ull/esit/app/strategy/Strategy.java	390
src/test/java/es/ull/esit/app/config/tspdynamic/TSPStateTest.java	391
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/AIOMutationTest.java	391
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/GenerationalReplaceTest.java	392
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointCrossoberTest.java	393
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointMutationTest.java	394
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/ProbabilisticSamplingTest.java	394
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/ProbabilityTest.java	395
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/RangeTest.java	396
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/RouletteSelectionTest.java	396
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/SteadyStateReplaceTest.java	397
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/TowPointsMutationTest.java	398
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/TruncationSelection.java	337
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/UniformCrossverTest.java	398
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/UnivarariteTest.java	399
src/test/java/es/ull/esit/app/factory_interface/FactoryInterfacesTest.java	400
src/test/java/es/ull/esit/app/factory_method/FactoryAcceptCandidateTest.java	400
src/test/java/es/ull/esit/app/factory_method/FactoryCandidateTest.java	401
src/test/java/es/ull/esit/app/factory_method/FactoryCrossoverTest.java	402
src/test/java/es/ull/esit/app/factory_method/FactoryDamplingTest.java	402
src/test/java/es/ull/esit/app/factory_method/FactoryDistributionTest.java	403
src/test/java/es/ull/esit/app/factory_method/FactoryFatherSelectionTest.java	404
src/test/java/es/ull/esit/app/factory_method/FactoryGeneratorTest.java	404
src/test/java/es/ull/esit/app/factory_method/FactoryLoaderTest.java	405
src/test/java/es/ull/esit/app/factory_method/FactoryMutationTest.java	406
src/test/java/es/ull/esit/app/factory_method/FactoryReplaceTest.java	406
src/test/java/es/ull/esit/app/factory_method/FactorySolutioMethodTest.java	407
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadTest.java	408
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptableCandidateTest.java	408
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptAnyoneTest.java	409
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptBestTest.java	410
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptMulticaseTest.java	411
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadTTest.java	411
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadUTest.java	412
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominatedTabuTest.java	413
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominatedTest.java	413
src/test/java/es/ull/esit/app/local_search/acceptation_type/DominanceTest.java	414
src/test/java/es/ull/esit/app/local_search/candidate_type/CandidateValueTest.java	415
src/test/java/es/ull/esit/app/local_search/candidate_type/GreaterCandidateTest.java	415
src/test/java/es/ull/esit/app/local_search/candidate_type/NotDominatedCandidatedTest.java	416
src/test/java/es/ull/esit/app/local_search/candidate_type/RandomCandidateTest.java	417
src/test/java/es/ull/esit/app/local_search/candidate_type/SmallerCandidateTest.java	417
src/test/java/es/ull/esit/app/local_search/complement/StopExecuteTest.java	418
src/test/java/es/ull/esit/app/local_search/complement/TabuSolutionTest.java	419
src/test/java/es/ull/esit/app/local_search/complement/UpdateParameterTest.java	419
src/test/java/es/ull/esit/app/metaheuristics/strategy/StrategyTest.java	439
src/test/java/es/ull/esit/app/metaheuristics/generators/DistributionEstimationAlgorithmTest.java	420

src/test/java/es/ull/esit/app/metaheuristics/generators/	EvolutionStrategiesTest.java	421
src/test/java/es/ull/esit/app/metaheuristics/generators/	GeneratorTest.java	421
src/test/java/es/ull/esit/app/metaheuristics/generators/	GenericAlgorithmsTest.java	422
src/test/java/es/ull/esit/app/metaheuristics/generators/	HillClimbingRestartTest.java	423
src/test/java/es/ull/esit/app/metaheuristics/generators/	HillClimbingTest.java	423
src/test/java/es/ull/esit/app/metaheuristics/generators/	InstanceDETest.java	424
src/test/java/es/ull/esit/app/metaheuristics/generators/	InstanceEETest.java	425
src/test/java/es/ull/esit/app/metaheuristics/generators/	InstanceGATest.java	425
src/test/java/es/ull/esit/app/metaheuristics/generators/	LimitRouletteTest.java	426
src/test/java/es/ull/esit/app/metaheuristics/generators/	LimitThresholdTest.java	427
src/test/java/es/ull/esit/app/metaheuristics/generators/	MultiCaseSimulatedAnnealingTest.java	427
src/test/java/es/ull/esit/app/metaheuristics/generators/	MultiobjectiveClimbingRestartTet.java	428
src/test/java/es/ull/esit/app/metaheuristics/generators/	MultiobjectiveStochasticHillClimbingTest.java	429
src/test/java/es/ull/esit/app/metaheuristics/generators/	MultiobjectiveTabuSearchTest.java	429
src/test/java/es/ull/esit/app/metaheuristics/generators/	ParticleSwarmOptimizationTest.java	430
src/test/java/es/ull/esit/app/metaheuristics/generators/	ParticleTest.java	431
src/test/java/es/ull/esit/app/metaheuristics/generators/	RandomSearchTest.java	432
src/test/java/es/ull/esit/app/metaheuristics/generators/	SimulatedAnnealing.java	381
src/test/java/es/ull/esit/app/metaheuristics/generators/	TabuSearchTest.java	432
src/test/java/es/ull/esit/app/problem/definition/	CodificationTest.java	433
src/test/java/es/ull/esit/app/problem/definition/	ObjectiveFunctionTest.java	434
src/test/java/es/ull/esit/app/problem/definition/	OperatorTest.java	434
src/test/java/es/ull/esit/app/problem/definition/	ProblemTest.java	435
src/test/java/es/ull/esit/app/problem/definition/	StateTest.java	436
src/test/java/es/ull/esit/app/problem/extension/	FactoresPonderadosTest.java	436
src/test/java/es/ull/esit/app/problem/extension/	MetricasMultiobjetivoTest.java	437
src/test/java/es/ull/esit/app/problem/extension/	SolutionMethodTest.java	438
src/test/java/es/ull/esit/app/problem/extension/	TypeSolutionMethod.java	388
src/test/java/es/ull/esit/app/problem_operators/	MutationOperatorTest.java	438
src/test/java/es/ull/esit/app/strategy/	StrategyTest.java	440

Capítulo 5

Documentación de espacios de nombres

5.1. Paquete `es.ull.esit.app.config.tspdynamic`

Clases

- class [TSPState](#)
- class **TSPStateTest**

5.2. Paquete `es.ull.esit.app.evolutionary_algorithms.complement`

Clases

- class [AIOMutation](#)
- class **AIOMutationTest**
- class [Crossover](#)
- enum [CrossoverType](#)
- class [Distribution](#)
- enum [DistributionType](#)
- class [FatherSelection](#)
- class [GenerationalReplace](#)
- class **GenerationalReplaceTest**
- class [Mutation](#)
- enum [MutationType](#)
- class [OnePointCrossover](#)
- class **OnePointCrossoverTest**
- class [OnePointMutation](#)
- class **OnePointMutationTest**
- class [ProbabilisticSampling](#)
- class **ProbabilisticSamplingTest**
- class [Probability](#)
- class **ProbabilityTest**
- class [Range](#)
- class **RangeTest**
- class [Replace](#)
- enum [ReplaceType](#)
- class [RouletteSelection](#)

- class **RouletteSelectionTest**
- class [Sampling](#)
- enum [SamplingType](#)
- enum [SelectionType](#)
- class [SteadyStateReplace](#)
- class **SteadyStateReplaceTest**
- class [TowPointsMutation](#)
- class **TowPointsMutationTest**
- class [TruncationSelection](#)
- class **TruncationSelectionTest**
- class [UniformCrossover](#)
- class **UniformCrossoverTest**
- class [Univariate](#)
- class **UnivariateTest**

5.3. Paquete `es.ull.esit.app.factory_interface`

Classes

- class **FactoryInterfacesTest**
- interface [IFFactoryAcceptCandidate](#)
- interface [IFFactoryCandidate](#)
- interface [IFFactoryCrossover](#)
- interface [IFFactoryDistribution](#)
- interface [IFFactoryFatherSelection](#)
- interface [IFFactoryGenerator](#)
- interface [IFFactoryMutation](#)
- interface [IFFactoryReplace](#)
- interface [IFFactorySolutionMethod](#)
- interface [IFFSampling](#)

5.4. Paquete `es.ull.esit.app.factory_method`

Classes

- class [FactoryAcceptCandidate](#)
- class **FactoryAcceptCandidateTest**
- class [FactoryCandidate](#)
- class **FactoryCandidateTest**
- class [FactoryCrossover](#)
- class **FactoryCrossoverTest**
- class [FactoryDistribution](#)
- class **FactoryDistributionTest**
- class [FactoryFatherSelection](#)
- class **FactoryFatherSelectionTest**
- class [FactoryGenerator](#)
- class **FactoryGeneratorTest**
- class [FactoryLoader](#)
- class **FactoryLoaderTest**
- class [FactoryMutation](#)
- class **FactoryMutationTest**
- class [FactoryReplace](#)
- class **FactoryReplaceTest**
- class [FactorySampling](#)
- class **FactorySamplingTest**
- class [FactorySolutionMethod](#)
- class **FactorySolutionMethodTest**

5.5. Paquete es.ull.esit.app.local_search.acceptation_type

Clases

- class [AcceptableCandidate](#)
- class **AcceptableCandidateTest**
- class [AcceptAnyone](#)
- class **AcceptAnyoneTest**
- class [AcceptBest](#)
- class **AcceptBestTest**
- class [AcceptMulticase](#)
- class **AcceptMulticaseTest**
- class [AcceptNotBad](#)
- class [AcceptNotBadT](#)
- class **AcceptNotBadTest**
- class **AcceptNotBadTTest**
- class [AcceptNotBadU](#)
- class **AcceptNotBadUTest**
- class [AcceptNotDominated](#)
- class [AcceptNotDominatedTabu](#)
- class **AcceptNotDominatedTabuTest**
- class **AcceptNotDominatedTest**
- enum [AcceptType](#)
- class [Dominance](#)
- class **DominanceTest**

5.5.1. Descripción detallada

@(#) AcceptNoBadT.java

@(#) AcceptNoBadU.java

5.6. Paquete es.ull.esit.app.local_search.candidate_type

Clases

- enum [CandidateType](#)
- class [CandidateValue](#)
- class **CandidateValueTest**
- class [GreaterCandidate](#)
- class **GreaterCandidateTest**
- class [NotDominatedCandidate](#)
- class **NotDominatedCandidateTest**
- class [RandomCandidate](#)
- class **RandomCandidateTest**
- class [SearchCandidate](#)
- class [SmallerCandidate](#)
- class **SmallerCandidateTest**

5.6.1. Descripción detallada

@(#) [SmallerCandidate.java](#)

5.7. Paquete `es.ull.esit.app.local_search.complement`

Clases

- class [StopExecute](#)
- class **StopExecuteTest**
- enum [StrategyType](#)
- class [TabuSolutions](#)
- class **TabuSolutionsTest**
- class [UpdateParameter](#)
- class **UpdateParameterTest**

5.8. Paquete `es.ull.esit.app.metaheuristics.strategy`

Clases

- class [Strategy](#)
- class **StrategyTest**

5.9. Paquete `es.ull.esit.app.metaheuristics.generators`

Clases

- class [DistributionEstimationAlgorithm](#)
- class **DistributionEstimationAlgorithmTest**
- class [EvolutionStrategies](#)
- class **EvolutionStrategiesTest**
- class [Generator](#)
- class **GeneratorTest**
- enum [GeneratorType](#)
- class [GeneticAlgorithm](#)
- class **GeneticAlgorithmTest**
- class [HillClimbing](#)
- class [HillClimbingRestart](#)
- class **HillClimbingRestartTest**
- class **HillClimbingTest**
- class [InstanceDE](#)
- class **InstanceDETest**
- class [InstanceEE](#)
- class **InstanceEETest**
- class [InstanceGA](#)
- class **InstanceGATest**
- class [LimitRoulette](#)
- class **LimitRouletteTest**

- class [LimitThreshold](#)
- class **LimitThresholdTest**
- class [MultiCaseSimulatedAnnealing](#)
- class **MultiCaseSimulatedAnnealingTest**
- class [MultiGenerator](#)
- class [MultiobjectiveHillClimbingDistance](#)
- class [MultiobjectiveHillClimbingRestart](#)
- class **MultiobjectiveHillClimbingRestartTest**
- class [MultiobjectiveStochasticHillClimbing](#)
- class **MultiobjectiveStochasticHillClimbingTest**
- class [MultiobjectiveTabuSearch](#)
- class **MultiobjectiveTabuSearchTest**
- class [Particle](#)
- class [ParticleSwarmOptimization](#)
- class **ParticleSwarmOptimizationTest**
- class **ParticleTest**
- class [RandomSearch](#)
- class **RandomSearchTest**
- class [SimulatedAnnealing](#)
- class [TabuSearch](#)
- class **TabuSearchTest**

5.10. Paquete es.ull.esit.app.problem.definition

Classes

- class [Codification](#)
- class **CodificationTest**
- class [ObjetiveFunction](#)
- class **ObjetiveFunctionTest**
- class [Operator](#)
- class **OperatorTest**
- class [Problem](#)
- class **ProblemTest**
- class [State](#)
- class **StateTest**

5.11. Paquete es.ull.esit.app.problem.extension

Classes

- class [FactoresPonderados](#)
- class **FactoresPonderadosTest**
- class [MetricasMultiobjetivo](#)
- class **MetricasMultiobjetivoTest**
- class [MultiObjetivoPuro](#)
- class [SolutionMethod](#)
- class **SolutionMethodTest**
- enum [TypeSolutionMethod](#)
- class **TypeSolutionMethodTest**

5.12. Paquete es.ull.esit.app.problem_operators

Clases

- class [MutationOperator](#)
- class **MutationOperatorTest**

5.13. Paquete es.ull.esit.app.strategy

Clases

- class [Strategy](#)
- class **StrategyTest**

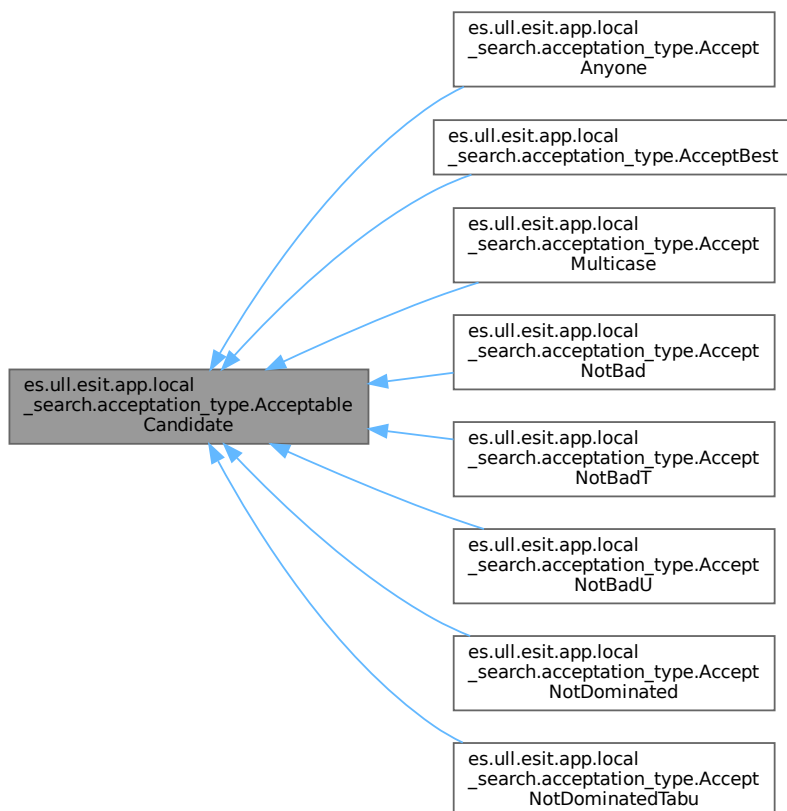
Capítulo 6

Documentación de clases

6.1. Referencia de la clase

es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate

Diagrama de herencia de es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate



Métodos públicos

- abstract Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.1.1. Descripción detallada

Abstract class representing an acceptable candidate strategy for candidate states.

6.1.2. Documentación de funciones miembro

6.1.2.1. [acceptCandidate\(\)](#)

```
abstract Boolean es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate.acceptCandidate (
    State stateCurrent,
    State stateCandidate ) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException [abstract]
```

Abstract method to determine if a candidate state is acceptable.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>stateCandidate</i>	[State] Candidate state to evaluate.

Devuelve

[Boolean] True if the candidate state is acceptable, false otherwise.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a specified class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a specified method cannot be found.

Reimplementado en [es.ull.esit.app.local_search.acceptation_type.AcceptAnyone](#), [es.ull.esit.app.local_search.acceptation_type.AcceptMulticase](#), [es.ull.esit.app.local_search.acceptation_type.AcceptNotBad](#), [es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT](#), [es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU](#), [es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated](#) y [es.ull.esit.app.local_search.acceptation_type.AcceptNotDominant](#)

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/local_search/acceptation_type/[AcceptableCandidate.java](#)

6.2. Referencia de la clase es.ull.esit.app.local_search.acceptation_type.AcceptAnyone

Diagrama de herencia de es.ull.esit.app.local_search.acceptation_type.AcceptAnyone

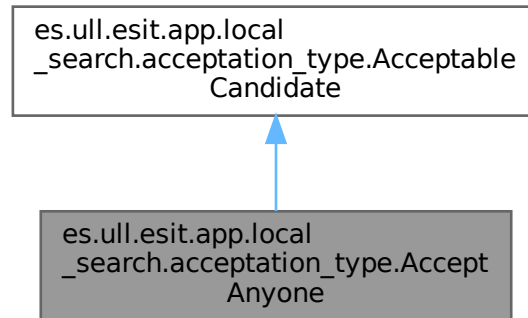
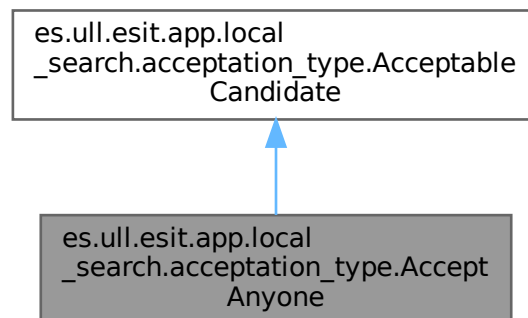


Diagrama de colaboración de es.ull.esit.app.local_search.acceptation_type.AcceptAnyone:



Métodos públicos

- Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate)

6.2.1. Descripción detallada

Class representing an accept-anyone strategy for candidate states.

6.2.2. Documentación de funciones miembro

6.2.2.1. acceptCandidate()

```
Boolean es.ull.esit.app.local_search.acceptation_type.AcceptAnyone.acceptCandidate (
    State stateCurrent,
    State stateCandidate )
```

Determines if a candidate state is acceptable.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>stateCandidate</i>	[State] Candidate state to evaluate.

Devuelve

[Boolean] Always returns true, indicating the candidate state is acceptable.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptAnyone.java](#)

6.3. Referencia de la clase

es.ull.esit.app.local_search.acceptation_type.AcceptBest

Diagrama de herencia de es.ull.esit.app.local_search.acceptation_type.AcceptBest

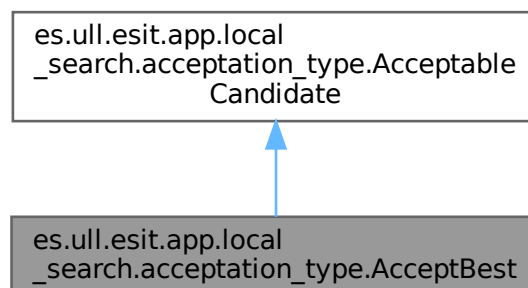
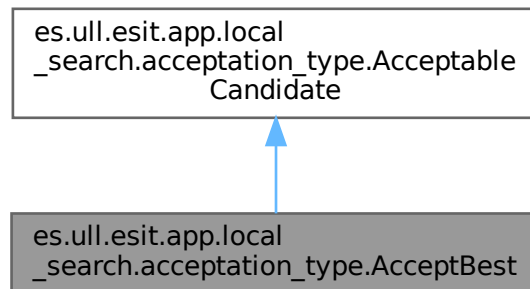


Diagrama de colaboración de es.ull.esit.app.local_search.acceptation_type.AcceptBest:



Métodos públicos

- Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.3.1. Descripción detallada

Class representing an accept-best strategy for candidate states.

6.3.2. Documentación de funciones miembro

6.3.2.1. acceptCandidate()

```

Boolean es.ull.esit.app.local_search.acceptation_type.AcceptBest.acceptCandidate (
    State stateCurrent,
    State stateCandidate ) throws IllegalArgumentException, SecurityException, Class↵
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
  
```

Determines if a candidate state is acceptable based on whether it is better than or equal to the current state.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>stateCandidate</i>	[State] Candidate state to evaluate.

Devuelve

[Boolean] True if the candidate state is better than or equal to the current state, false otherwise.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a specified class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a specified method cannot be found.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptBest.java`

6.4. Referencia de la clase

es.ull.esit.app.local_search.acceptation_type.AcceptMulticase

Diagrama de herencia de `es.ull.esit.app.local_search.acceptation_type.AcceptMulticase`

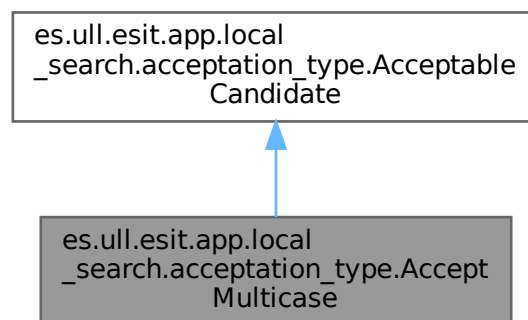
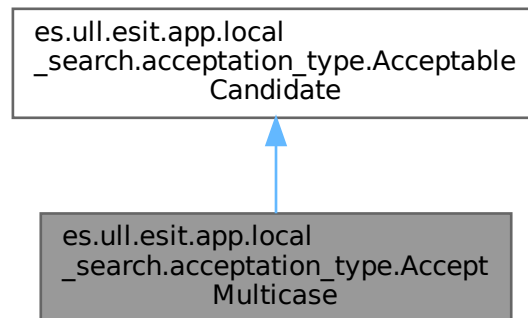


Diagrama de colaboración de `es.ull.esit.app.local_search.acceptation_type.AcceptMulticase`:



Métodos públicos

- Boolean [acceptCandidate](#) (`State stateCurrent`, `State stateCandidate`)

6.4.1. Descripción detallada

Class representing an accept-multicase strategy for candidate states.

6.4.2. Documentación de funciones miembro

6.4.2.1. `acceptCandidate()`

```
Boolean es.ull.esit.app.local_search.acceptation_type.AcceptMulticase.acceptCandidate (
    State stateCurrent,
    State stateCandidate )
```

Determines if a candidate state is acceptable based on multicase criteria.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>stateCandidate</i>	[State] Candidate state to evaluate.

Devuelve

[Boolean] True if the candidate state is acceptable, false otherwise.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/acceptation_type/`[AcceptMulticase.java](#)

6.5. Referencia de la clase

es.ull.esit.app.local_search.acceptation_type.AcceptNotBad

Diagrama de herencia de es.ull.esit.app.local_search.acceptation_type.AcceptNotBad

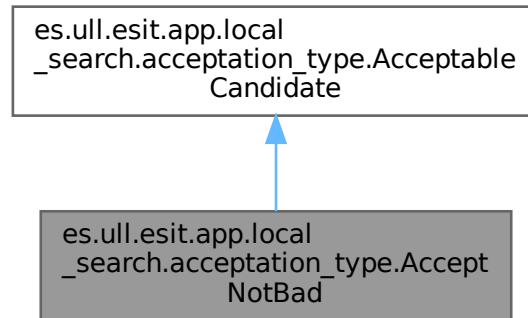
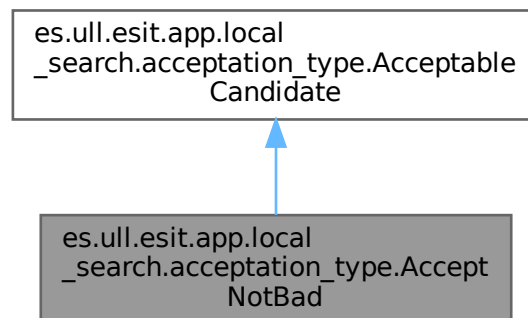


Diagrama de colaboración de es.ull.esit.app.local_search.acceptation_type.AcceptNotBad:



Métodos públicos

- Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate)

6.5.1. Descripción detallada

Class that implements the acceptance type "Accept Not Bad"

6.5.2. Documentación de funciones miembro

6.5.2.1. `acceptCandidate()`

```
Boolean es.ull.esit.app.local_search.acceptation_type.AcceptNotBad.acceptCandidate (
    State stateCurrent,
    State stateCandidate )
```

Decides whether to accept or not a candidate state.

Parámetros

<code>stateCurrent</code>	[State] Current state.
<code>stateCandidate</code>	[State] Candidate state.

Devuelve

[Boolean] True if the candidate state is accepted, false otherwise.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBad.java`

6.6. Referencia de la clase `es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT`

Diagrama de herencia de `es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT`

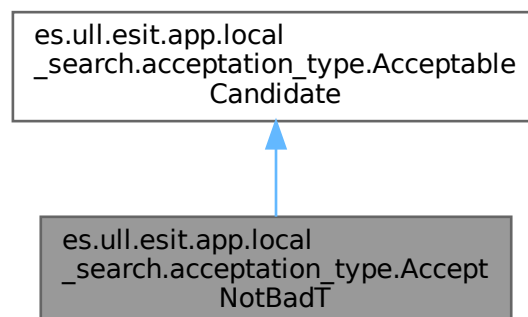
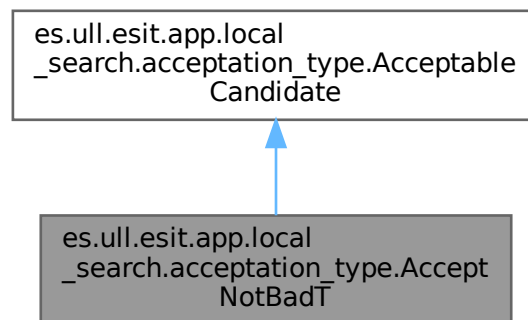


Diagrama de colaboración de `es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT`:



Métodos públicos

- Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.6.1. Descripción detallada

Class that implements the acceptance type "Accept Not Bad with Temperature"

6.6.2. Documentación de funciones miembro

6.6.2.1. acceptCandidate()

```

Boolean es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT.acceptCandidate (
    State stateCurrent,
    State stateCandidate ) throws IllegalArgumentException, SecurityException, Class↵
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
  
```

Decides whether to accept or not a candidate state.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>stateCandidate</i>	[State] Candidate state.

Devuelve

[Boolean] True if the candidate state is accepted, false otherwise.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadT.java`

6.7. Referencia de la clase es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU

Diagrama de herencia de es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU

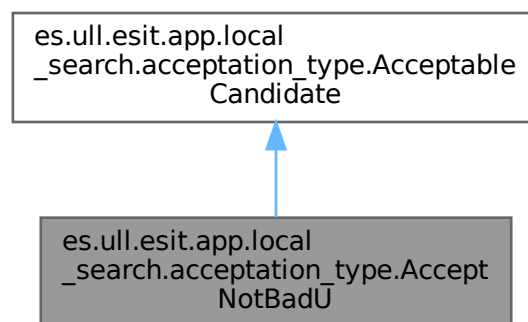
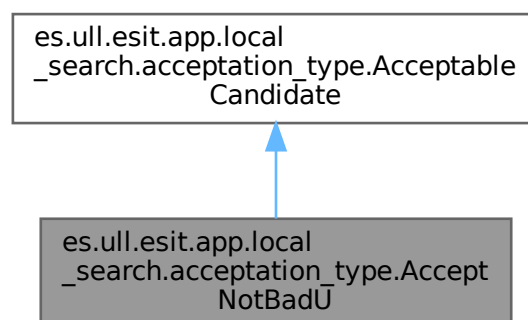


Diagrama de colaboración de es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU:



Métodos públicos

- Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.7.1. Descripción detallada

Class that implements the acceptance type "Accept Not Bad with Utility"

6.7.2. Documentación de funciones miembro

6.7.2.1. acceptCandidate()

```
Boolean es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU.acceptCandidate (
    State stateCurrent,
    State stateCandidate ) throws IllegalArgumentException, SecurityException, Class↵
NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Decides whether to accept or not a candidate state.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>stateCandidate</i>	[State] Candidate state.

Devuelve

[Boolean] True if the candidate state is accepted, false otherwise.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadU.java](#)

6.8. Referencia de la clase

es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated

Diagrama de herencia de es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated

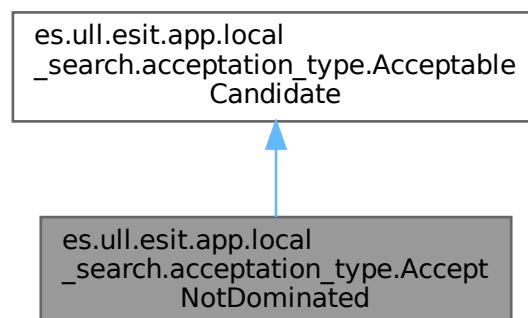
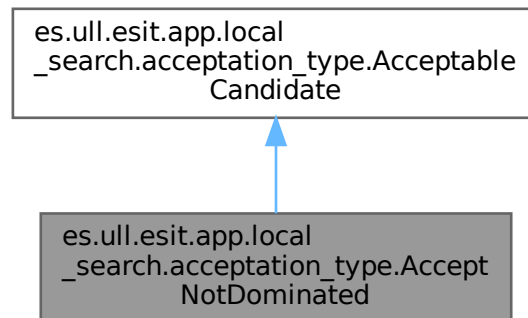


Diagrama de colaboración de es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated:



Métodos públicos

- Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate)

6.8.1. Descripción detallada

Class that implements the acceptance type "Accept Not Dominated".

6.8.2. Documentación de funciones miembro

6.8.2.1. acceptCandidate()

```
Boolean es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated.acceptCandidate (
    State stateCurrent,
    State stateCandidate )
```

Decides whether to accept or not a candidate state.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>stateCandidate</i>	[State] Candidate state.

Devuelve

[Boolean] True if the candidate state is accepted, false otherwise.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/local_search/acceptation_type/[AcceptNotDominated.java](#)

6.9. Referencia de la clase `es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu` ↩

Diagrama de herencia de `es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu`

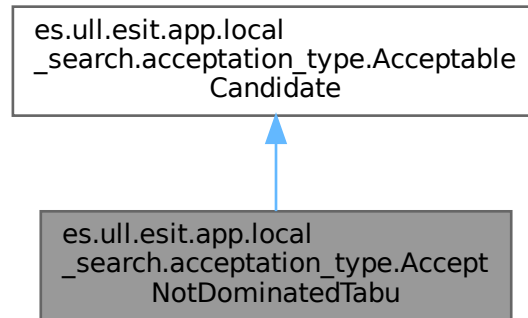
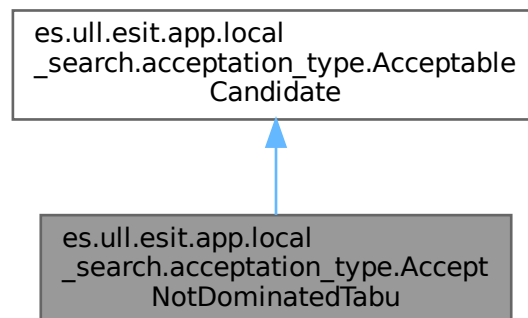


Diagrama de colaboración de `es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu`:



Métodos públicos

- Boolean [acceptCandidate](#) (State stateCurrent, State stateCandidate)

6.9.1. Descripción detallada

Class that implements the acceptance type "Accept Not Dominated Tabu".

6.9.2. Documentación de funciones miembro

6.9.2.1. `acceptCandidate()`

```
Boolean es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu.acceptCandidate (
    State stateCurrent,
    State stateCandidate )
```

Decides whether to accept or not a candidate state.

Parámetros

<code>stateCurrent</code>	[State] Current state.
<code>stateCandidate</code>	[State] Candidate state.

Devuelve

[Boolean] True if the candidate state is accepted, false otherwise.

Reimplementado de [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominatedTabu.java`

6.10. Referencia de la enumeración `es.ull.esit.app.local_search.acceptation_type.AcceptType`

Atributos públicos

- [AcceptBest](#)
- [AcceptAnyone](#)
- [AcceptNotBadT](#)
- [AcceptNotBadU](#)
- [AcceptNotDominated](#)
- [AcceptNotDominatedTabu](#)
- [AcceptNotBad](#)
- [AcceptMulticase](#)

6.10.1. Descripción detallada

Enum that defines the different types of acceptation strategies.

6.10.2. Documentación de datos miembro

6.10.2.1. `AcceptAnyone`

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptAnyone
```

Accept the best candidate solution

6.10.2.2. AcceptBest

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptBest
```

6.10.2.3. AcceptMulticase

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptMulticase
```

Accept not bad candidate solution

6.10.2.4. AcceptNotBad

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptNotBad
```

Accept not dominated candidate solution with tabu

6.10.2.5. AcceptNotBadT

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptNotBadT
```

Accept any candidate solution

6.10.2.6. AcceptNotBadU

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptNotBadU
```

Accept not bad candidate solution based on temperature

6.10.2.7. AcceptNotDominated

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptNotDominated
```

Accept not bad candidate solution based on utility

6.10.2.8. AcceptNotDominatedTabu

```
es.ull.esit.app.local_search.acceptation_type.AcceptType.AcceptNotDominatedTabu
```

Accept not dominated candidate solution

La documentación de esta enumeración está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/acceptation_type/`[AcceptType.java](#)

6.11. Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation

Diagrama de herencia de es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation

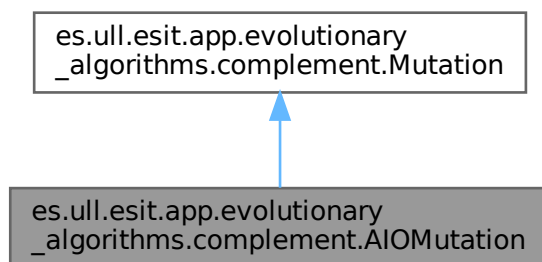
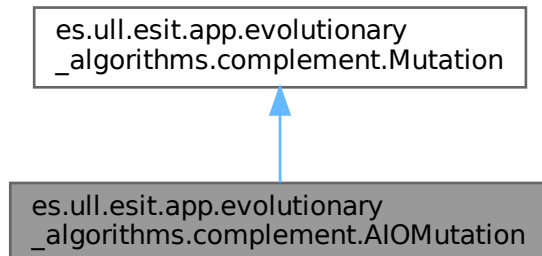


Diagrama de colaboración de es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation:



Métodos públicos

- State [mutation](#) (State state, double pm)
- void [sortedPathValue](#) (State state)

Métodos públicos estáticos

- static void [fillPath](#) ()

Atributos estáticos protegidos

- static final List< Object > [path](#) = new ArrayList<>()

6.11.1. Descripción detallada

Class that implements the AIO mutation operator.

6.11.2. Documentación de funciones miembro

6.11.2.1. fillPath()

```
static void es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation.fillPath ( ) [static]
```

Fills the path list with indices corresponding to the variable count of the problem's codification.

6.11.2.2. mutation()

```
State es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation.mutation (
    State state,
    double pm )
```

Applies the AIO mutation to the given state with the specified mutation probability.

Parámetros

<i>state</i>	[State] the state to mutate.
<i>pm</i>	[double] the mutation probability.

Devuelve

[State] the mutated state.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Mutation](#).

6.11.2.3. sortedPathValue()

```
void es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation.sortedPathValue (
    State state )
```

Sorts the states in the given state based on their values in ascending order.

Parámetros

<i>state</i>	[State] the state containing the codes to sort.
--------------	---

6.11.3. Documentación de datos miembro

6.11.3.1. `path`

```
final List<Object> es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation.path = new  
ArrayList<>() [static], [protected]
```

Path used for sorting states based on their values.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/AIOMutation.java`

6.12. Referencia de la enumeración `es.ull.esit.app.local_search.candidate_type.CandidateType`

Atributos públicos

- `SmallerCandidate`
- `GreaterCandidate`
- `RandomCandidate`
- `NotDominatedCandidate`

6.12.1. Descripción detallada

Enum that represents the different types of candidates that can be used in local search algorithms.

6.12.2. Documentación de datos miembro

6.12.2.1. `GreaterCandidate`

```
es.ull.esit.app.local_search.candidate_type.CandidateType.GreaterCandidate
```

Candidate with smaller objective function value

6.12.2.2. `NotDominatedCandidate`

```
es.ull.esit.app.local_search.candidate_type.CandidateType.NotDominatedCandidate
```

Candidate selected randomly

6.12.2.3. `RandomCandidate`

```
es.ull.esit.app.local_search.candidate_type.CandidateType.RandomCandidate
```

Candidate with greater objective function value

6.12.2.4. SmallerCandidate

`es.ull.esit.app.local_search.candidate_type.CandidateType.SmallerCandidate`

La documentación de esta enumeración está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/candidate_type/CandidateType.java`

6.13. Referencia de la clase

`es.ull.esit.app.local_search.candidate_type.CandidateValue`

Métodos públicos

- `CandidateValue ()`
- `CandidateValue (StrategyType strategy, IFFactoryCandidate ifFactory, CandidateType typecand, TabuSolutions tabusolution, SearchCandidate searchcandidate)`
- `SearchCandidate newSearchCandidate (CandidateType typecandidate)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- State `stateCandidate` (State stateCurrent, CandidateType typeCandidate, StrategyType strategy, Integer operatornumber, List< State > neighborhood) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- TabuSolutions `getTabusolution ()`
- void `setTabusolution (TabuSolutions tabusolution)`

6.13.1. Descripción detallada

Class that represents the candidate value in local search algorithms.

6.13.2. Documentación de constructores y destructores

6.13.2.1. CandidateValue() [1/2]

`es.ull.esit.app.local_search.candidate_type.CandidateValue.CandidateValue ()`

Constructor for CandidateValue

6.13.2.2. CandidateValue() [2/2]

```
es.ull.esit.app.local_search.candidate_type.CandidateValue.CandidateValue (
    StrategyType strategy,
    IFFactoryCandidate ifFactory,
    CandidateType typecand,
    TabuSolutions tabusolution,
    SearchCandidate searchcandidate )
```

Constructor for CandidateValue with parameters.

Parámetros

<i>strategy</i>	[StrategyType] Type of strategy used.
<i>ifFactory</i>	[IFFactoryCandidate] Factory interface for creating candidates.
<i>typecand</i>	[CandidateType] Type of candidate.
<i>tabusolution</i>	[TabuSolutions] Tabu solutions.
<i>searchcandidate</i>	[SearchCandidate] Search candidate.

6.13.3. Documentación de funciones miembro

6.13.3.1. getTabusolution()

```
TabuSolutions es.ull.esit.app.local_search.candidate_type.CandidateValue.getTabusolution ( )
```

Gets the tabu solutions.

Devuelve

[TabuSolutions] The tabu solutions.

6.13.3.2. newSearchCandidate()

```
SearchCandidate es.ull.esit.app.local_search.candidate_type.CandidateValue.newSearchCandidate
(
    CandidateType typecand ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
Exception, NoSuchMethodException
```

Creates a new SearchCandidate based on the provided CandidateType.

Parámetros

<i>typecandidate</i>	[CandidateType] The type of candidate to create.
----------------------	--

Devuelve

[SearchCandidate] The created search candidate.

Excepciones

<i>IllegalArgumentException</i>	If the argument is invalid.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to the class is illegal.
<i>InvocationTargetException</i>	If the method invocation fails.
<i>NoSuchMethodException</i>	If the method does not exist.

6.13.3.3. setTabusolution()

```
void es.ull.esit.app.local_search.candidate_type.CandidateValue.setTabusolution (
    TabuSolutions tabusolution )
```

Sets the tabu solutions.

Parámetros

<i>tabusolution</i>	[TabuSolutions] The tabu solutions to set.
---------------------	--

6.13.3.4. stateCandidate()

```
State es.ull.esit.app.local_search.candidate_type.CandidateValue.stateCandidate (
    State stateCurrent,
    CandidateType typeCandidate,
    StrategyType strategy,
    Integer operatornumber,
    List< State > neighborhood ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
Exception, NoSuchMethodException
```

Finds a candidate state based on the current state, candidate type, strategy, operator number, and neighborhood.

Parámetros

<i>stateCurrent</i>	[State] The current state.
<i>typeCandidate</i>	[CandidateType] The type of candidate.
<i>strategy</i>	[StrategyType] The strategy type.
<i>operatornumber</i>	[Integer] The operator number.
<i>neighborhood</i>	[List<State>] The neighborhood of states.

Devuelve

[State] The candidate state found.

Excepciones

<i>IllegalArgumentException</i>	If the argument is invalid.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to the class is illegal.
<i>InvocationTargetException</i>	If the method invocation fails.
<i>NoSuchMethodException</i>	If the method does not exist.

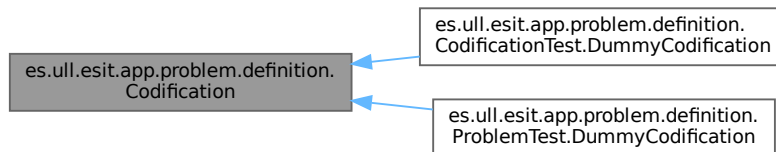
La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/local_search/candidate_type/[CandidateValue.java](#)

6.14. Referencia de la clase

es.ull.esit.app.problem.definition.Codification

Diagrama de herencia de es.ull.esit.app.problem.definition.Codification



Métodos públicos

- abstract boolean [validState](#) ([State](#) state)
- abstract Object [getVariableAleatoryValue](#) (int key)
- abstract int [getAleatoryKey](#) ()
- abstract int [getVariableCount](#) ()

6.14.1. Descripción detallada

Abstract class representing a codification of a problem.

6.14.2. Documentación de funciones miembro

6.14.2.1. [getAleatoryKey\(\)](#)

```
abstract int es.ull.esit.app.problem.definition.Codification.getAleatoryKey ( ) [abstract]
```

Retrieves the total number of aleatory keys in the codification.

Devuelve

[int] The number of aleatory keys.

6.14.2.2. [getVariableAleatoryValue\(\)](#)

```
abstract Object es.ull.esit.app.problem.definition.Codification.getVariableAleatoryValue (
    int key ) [abstract]
```

Generates a random valid state according to the codification rules.

Parámetros

<i>key</i>	[int] The key or index of the variable to generate.
------------	---

Devuelve

[State] A randomly generated valid state.

6.14.2.3. `getVariableCount()`

```
abstract int es.ull.esit.app.problem.definition.Codification.getVariableCount ( ) [abstract]
```

Retrieves the total number of variables in the codification.

Devuelve

[int] The number of variables.

6.14.2.4. `validState()`

```
abstract boolean es.ull.esit.app.problem.definition.Codification.validState (
    State state ) [abstract]
```

Checks if the given state is valid according to the codification rules.

Parámetros

<i>state</i>	[State] The state to be validated.
--------------	------------------------------------

Devuelve

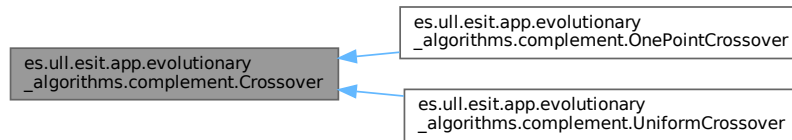
true if the state is valid, false otherwise.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/definition/Codification.java](#)

6.15. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.Crossover`

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.Crossover`



Métodos públicos

- abstract State `crossover` (State father1, State father2, double pc)

6.15.1. Descripción detallada

Abstract class that defines the crossover operator.

6.15.2. Documentación de funciones miembro

6.15.2.1. `crossover()`

```

abstract State es.ull.esit.app.evolutionary_algorithms.complement.Crossover.crossover (
    State father1,
    State father2,
    double pc ) [abstract]
  
```

Applies the crossover operation between two parent states with a given crossover probability.

Parámetros

<i>father1</i>	[State] the first parent state.
<i>father2</i>	[State] the second parent state.
<i>pc</i>	[double] the crossover probability.

Devuelve

[State] the resulting offspring state after crossover.

Reimplementado en [es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover](#) y [es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover](#)

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Crossover.java`

6.16. Referencia de la enumeración `es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType`

Atributos públicos

- [ONE_POINT_CROSSOVER](#)
- [UNIFORM_CROSSOVER](#)

6.16.1. Descripción detallada

Enum representing different types of crossover operations.

6.16.2. Documentación de datos miembro

6.16.2.1. ONE_POINT_CROSSOVER

```
es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType.ONE_POINT_CROSSOVER
```

6.16.2.2. UNIFORM_CROSSOVER

```
es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType.UNIFORM_CROSSOVER
```

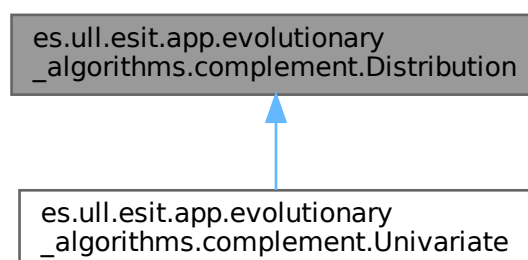
One-point crossover operation.

La documentación de esta enumeración está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/CrossoverType.java`

6.17. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.Distribution`

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.Distribution`



Métodos públicos

- abstract List< [Probability](#) > [distribution](#) (List< State > fathers)

6.17.1. Descripción detallada

Abstract class that defines the distribution operator.

6.17.2. Documentación de funciones miembro

6.17.2.1. [distribution\(\)](#)

```
abstract List< Probability > es.ull.esit.app.evolutionary_algorithms.complement.Distribution.↵
distribution (
    List< State > fathers ) [abstract]
```

Applies the distribution operation to a list of father states.

Parámetros

<i>fathers</i>	[List<State>] the list of father states.
----------------	--

Devuelve

[List<Probability>] the list of probabilities resulting from the distribution.

Reimplementado en [es.ull.esit.app.evolutionary_algorithms.complement.Univariate](#).

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/[Distribution.java](#)

6.18. Referencia de la clase es.ull.esit.app.metaheuristics.generators.↵ DistributionEstimationAlgorithm

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm

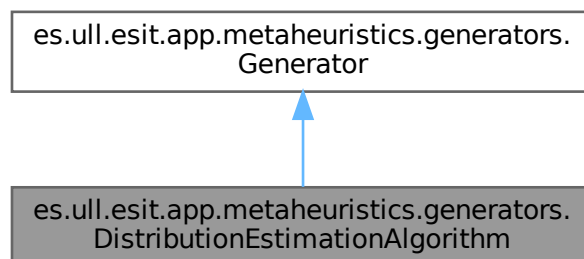
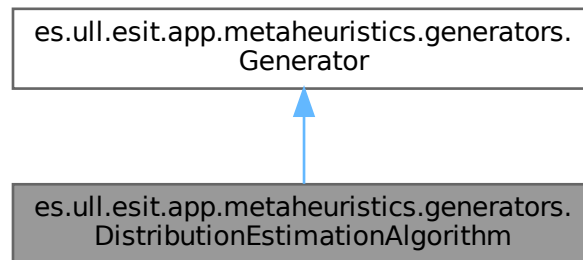


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm`:



Métodos públicos

- [DistributionEstimationAlgorithm](#) ()
- State [maxValue](#) (List< State > listInd)
- State [generate](#) (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- State [getReference](#) ()
- List< State > [getReferenceList](#) ()
- [GeneratorType](#) [getType](#) ()
- void [setInitialReference](#) (State stateInitialRef)
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- List< State > [getListStateRef](#) ()
- List< State > [getListReference](#) ()
- void [setListReference](#) (List< State > listReference)
- [GeneratorType](#) [getGeneratorType](#) ()
- void [setGeneratorType](#) ([GeneratorType](#) generatorType)
- List< State > [getfathersList](#) () throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- List< State > [getSonList](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- `DistributionType` [getDistributionType](#) ()
- void [setDistributionType](#) (`DistributionType` distributionType)
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()
- ReplaceType [getReplaceType](#) ()
- SelectionType [getSelectionType](#) ()

Métodos públicos estáticos

- static void `setReplaceType` (`ReplaceType` `replaceType`)
- static void `setSelectionType` (`SelectionType` `selectionType`)
- static int `getCountRef` ()
- static void `setCountRef` (int `countRef`)

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.18.1. Descripción detallada

Class that implements the Distribution Estimation Algorithm (DEA) generator.

6.18.2. Documentación de constructores y destructores

6.18.2.1. `DistributionEstimationAlgorithm()`

```
es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.DistributionEstimation↵
Algorithm ( )
```

Constructor method.

6.18.3. Documentación de funciones miembro

6.18.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.award↵
UpdateREF (
    State stateCandidate )
```

Awards an update to the reference list if the candidate state is already present.

Parámetros

<code>stateCandidate</code>	[State] Candidate state to check.
-----------------------------	-----------------------------------

Devuelve

[boolean] True if the candidate state is found in the reference list, false otherwise.

Reimplementado de `es.ull.esit.app.metaheuristics.generators.Generator`.

6.18.3.2. generate()

State es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.generate (Integer *operatornumber*) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

Generates a new individual using the Distribution Estimation Algorithm (DEA).

Parámetros

<i>operatornumber</i>	[Integer] Number of operators.
-----------------------	--------------------------------

Devuelve

[State] Generated individual.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If illegal access occurs.
<i>InvocationTargetException</i>	If an invocation target error occurs.
<i>NoSuchMethodException</i>	If a method is not found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.3. getCountRef()

static int es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getCountRef () [static]

Gets the counter for reference states.

Devuelve

[int] Counter for reference states.

6.18.3.4. getDistributionType()

DistributionType es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getDistributionType ()

Gets the distribution type used in the DEA.

Devuelve

[DistributionType] Distribution type used in the DEA.

6.18.3.5. `getfathersList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.↵
getfathersList ( ) throws IllegalArgumentException, SecurityException, ClassNotFoundException,
InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethod↵
Exception
```

Gets the list of father states selected for generating new individuals.

Devuelve

[List<State>] List of father states.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If an illegal access error occurs.
<i>InvocationTargetException</i>	If an invocation target error occurs.
<i>NoSuchMethodException</i>	If a method is not found.

6.18.3.6. `getGeneratorType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.get↵
GeneratorType ( )
```

Gets the type of generator.

Devuelve

[GeneratorType] Type of generator.

6.18.3.7. `getListCountBetterGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getListCount↵
BetterGender ( )
```

Gets the sampling type used in the DEA.

Devuelve

[SamplingType] Sampling type used in the DEA.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.8. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getListCountGender ( )
```

Gets the list of gender counts.

Devuelve

[int[]] List of gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.9. `getListReference()`

```
List< State > es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getListReference ( )
```

Gets the list of reference states.

Devuelve

[List<State>] List of reference states.

6.18.3.10. `getListStateRef()`

```
List< State > es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getListStateRef ( )
```

Gets the list of reference states for the DEA.

Devuelve

[List<State>] List of reference states.

6.18.3.11. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getReference ( )
```

Gets the reference state for the DEA.

Devuelve

[State] Reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.12. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.get↔  
ReferenceList ( )
```

Gets the list of reference states for the DEA.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.13. `getReplaceType()`

```
ReplaceType es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.get↔  
ReplaceType ( )
```

Gets the replacement type used in the DEA.

Devuelve

[ReplaceType] Replacement type used in the DEA.

6.18.3.14. `getSelectionType()`

```
SelectionType es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.get↔  
SelectionType ( )
```

Gets the selection type used in the DEA.

Devuelve

[SelectionType] Selection type used in the DEA.

6.18.3.15. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.get↔  
SonList ( )
```

Gets the list of son states generated by the DEA.

Devuelve

[List<State>] List of son states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.16. getTrace()

```
float[] es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getTrace ( )
```

Gets the trace of weight changes.

Devuelve

[float[]] Trace of weight changes.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.17. getType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.get↵
Type ( )
```

Abstract method to get the type of generator.

Devuelve

[GeneratorType] The type of generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.18. getWeight()

```
float es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.getWeight ( )
```

Gets the weight of the distribution estimation algorithm.

Devuelve

[float] Weight of the distribution estimation algorithm.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.19. maxValue()

```
State es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.maxValue (
    List< State > listInd )
```

Calculates the individual with the maximum evaluation from a list of individuals.

Parámetros

<i>listInd</i>	[List<State>] List of individuals.
----------------	------------------------------------

Devuelve

[State] Individual with the maximum evaluation.

6.18.3.20. `setCountRef()`

```
static void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.set↵
CountRef (
    int countRef ) [static]
```

Sets the counter for reference states.

Parámetros

<code>countRef</code>	[int] Counter for reference states.
-----------------------	-------------------------------------

6.18.3.21. `setDistributionType()`

```
void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.setDistribution↵
Type (
    DistributionType distributionType )
```

Sets the distribution type used in the DEA.

Parámetros

<code>distributionType</code>	[DistributionType] Distribution type used in the DEA.
-------------------------------	---

6.18.3.22. `setGeneratorType()`

```
void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.setGenerator↵
Type (
    GeneratorType generatorType )
```

Sets the type of generator.

Parámetros

<code>generatorType</code>	[GeneratorType] Type of generator.
----------------------------	------------------------------------

6.18.3.23. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.setInitial↵
Reference (
    State stateInitialRef )
```

Sets the initial reference state for the DEA.

Parámetros

<i>stateInitialRef</i>	[State] Initial reference state.
------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.24. setListReference()

```
void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.setListReference
(
    List< State > listReference )
```

Sets the list of reference states.

Parámetros

<i>listReference</i>	[List<State>] List of reference states.
----------------------	---

6.18.3.25. setReplaceType()

```
static void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.set↔
ReplaceType (
    ReplaceType replaceType ) [static]
```

Sets the replacement type used in the DEA.

Parámetros

<i>replaceType</i>	[ReplaceType] Replacement type used in the DEA.
--------------------	---

6.18.3.26. setSelectionType()

```
static void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.set↔
SelectionType (
    SelectionType selectionType ) [static]
```

Sets the selection type used in the DEA.

Parámetros

<i>selectionType</i>	[SelectionType] Selection type used in the DEA.
----------------------	---

6.18.3.27. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.setWeight (
    float weight )
```

Sets the weight of the distribution estimation algorithm.

Parámetros

<i>weight</i>	[float] Weight of the distribution estimation algorithm.
---------------	--

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.18.3.28. `updateReference()`

```
void es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm.updateReference
(
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Updates the reference list with a new candidate state.

Parámetros

<i>stateCandidate</i>	[State] Candidate state to update the reference list.
<i>countIterationsCurrent</i>	[Integer] Current iteration count.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/DistributionEstimationAlgorithm.java`

6.19. Referencia de la enumeración `es.ull.esit.app.evolutionary_algorithms.complement.DistributionType`

Atributos públicos

- [UNIVARIATE](#)

6.19.1. Descripción detallada

Enumeration representing different types of distribution methods.

6.19.2. Documentación de datos miembro

6.19.2.1. UNIVARIATE

```
es.ull.esit.app.evolutionary_algorithms.complement.DistributionType.UNIVARIATE
```

Univariate distribution method

La documentación de esta enumeración está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/DistributionType.java`

6.20. Referencia de la clase

es.ull.esit.app.local_search.acceptation_type.Dominance

Métodos públicos

- boolean [listDominance](#) (State solutionX, List< State > list)
- boolean [dominance](#) (State solutionX, State solutionY)

6.20.1. Descripción detallada

Class that implements dominance-based acceptance criteria for local search algorithms.

6.20.2. Documentación de funciones miembro

6.20.2.1. dominance()

```
boolean es.ull.esit.app.local_search.acceptation_type.Dominance.dominance (
    State solutionX,
    State solutionY )
```

Determines if solutionX dominates solutionY.

Parámetros

<i>solutionX</i>	[State] The candidate solution.
<i>solutionY</i>	[State] The solution to compare against.

Devuelve

[boolean] True if solutionX dominates solutionY, false otherwise.

6.20.2.2. listDominance()

```
boolean es.ull.esit.app.local_search.acceptation_type.Dominance.listDominance (
    State solutionX,
    List< State > list )
```

Determines if solutionX dominates any of the non-dominated solutions in a list.

Parámetros

<i>solutionX</i>	[State] The candidate solution.
<i>list</i>	[List<State>] The list of non-dominated solutions.

Devuelve

[boolean] True if solutionX was added to the list, false otherwise.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/local_search/acceptation_type/[Dominance.java](#)

6.21. Referencia de la clase es.ull.esit.app.metaheuristics.generators.EvolutionStrategies

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.EvolutionStrategies

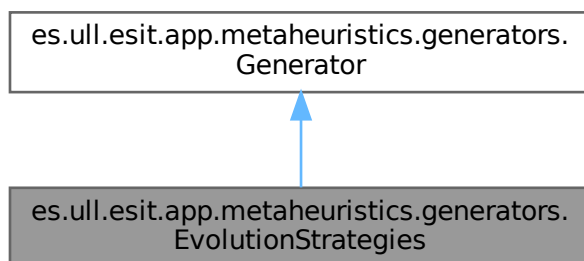
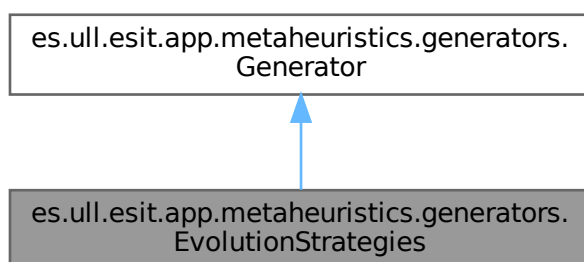


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.EvolutionStrategies:



Métodos públicos

- [EvolutionStrategies](#) ()
- State [generate](#) (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- State [getReference](#) ()
- void [setStateRef](#) (State stateRef)
- [GeneratorType](#) [getType](#) ()
- void [setInitialReference](#) (State stateInitialRef)
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- List< State > [getListStateRef](#) ()
- List< State > [getListStateReference](#) ()
- void [setListStateReference](#) (List< State > listStateReference)
- [GeneratorType](#) [getTypeGenerator](#) ()
- void [setTypeGenerator](#) ([GeneratorType](#) generatorType)
- List< State > [getReferenceList](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()
- List< State > [getSonList](#) ()

Métodos públicos estáticos

- static int [getCountRef](#) ()
- static void [setCountRef](#) (int countRef)

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.21.1. Descripción detallada

Class that implements the Evolution Strategies generator.

6.21.2. Documentación de constructores y destructores

6.21.2.1. EvolutionStrategies()

```
es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.EvolutionStrategies ( )
```

6.21.3. Documentación de funciones miembro

6.21.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.awardUpdateREF (
    State stateCandidate )
```

Awards an update to the reference state based on a candidate state.

Parámetros

<i>stateCandidate</i>	[State] The candidate state to consider for awarding an update.
-----------------------	---

Devuelve

[boolean] True if the update was awarded, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new state using the Evolution Strategies method.

Parámetros

<i>operatornumber</i>	The operator number (not used in this implementation).
-----------------------	--

Devuelve

A new generated state.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If illegal access occurs.
<i>InvocationTargetException</i>	If an invocation target error occurs.
<i>NoSuchMethodException</i>	If a method cannot be found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.3. getCountRef()

```
static int es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getCountRef ( ) [static]
```

Gets the count reference for the generator.

Devuelve

[int] The count reference for the generator.

6.21.3.4. `getListCountBetterGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getListCountBetterGender ( )
```

Gets the list of counts of better genders.

Devuelve

[int[]] The list of counts of better genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.5. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getListCountGender ( )
```

Gets the list of counts of genders.

Devuelve

[int[]] The list of counts of genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.6. `getListStateRef()`

```
List< State > es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getListStateRef ( )
```

Gets the list of reference states for the Evolution Strategies.

Devuelve

[List<State>] The list of reference states.

6.21.3.7. `getListStateReference()`

```
List< State > es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getListStateReference ( )
```

Gets the list of reference states.

Devuelve

[List<State>] The list of reference states.

6.21.3.8. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getReference ( )
```

Gets the reference state for the Evolution Strategies.

Devuelve

The reference state.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If illegal access occurs.
<i>InvocationTargetException</i>	If an invocation target error occurs.
<i>NoSuchMethodException</i>	If a method cannot be found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.9. getReferenceList()

```
List< State > es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getReferenceList (
)
```

Gets the list of reference states.

Devuelve

[List<State>] The list of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.10. getSonList()

```
List< State > es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getSonList ( )
```

Gets the list of son states.

Devuelve

[List<State>] The list of son states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.11. getTrace()

```
float[] es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getTrace ( )
```

Gets the trace of the generator.

Devuelve

[float[]] The trace of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.12. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getType ( )
```

Gets the type of generator.

Devuelve

[GeneratorType] The type of generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.13. `getTypeGenerator()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getTypeGenerator ( )
```

Gets the type of generator.

Devuelve

[GeneratorType] The type of generator.

6.21.3.14. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.getWeight ( )
```

Gets the weight of the generator.

Devuelve

[float] The weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.15. `setCountRef()`

```
static void es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.setCountRef (
    int countRef ) [static]
```

Sets the count reference for the generator.

Parámetros

<code>countRef</code>	[int] The count reference to set.
-----------------------	-----------------------------------

6.21.3.16. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state for the Evolution Strategies.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state to set.
------------------------	---

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.17. setListStateReference()

```
void es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.setListStateReference (
    List< State > listStateReference )
```

Sets the list of reference states.

Parámetros

<i>listStateReference</i>	[List<State>] The list of reference states to set.
---------------------------	--

6.21.3.18. setStateRef()

```
void es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.setStateRef (
    State stateRef )
```

Sets the reference state for the Evolution Strategies.

Parámetros

<i>stateRef</i>	[State] The reference state to set.
-----------------	-------------------------------------

6.21.3.19. setTypeGenerator()

```
void es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.setTypeGenerator (
    GeneratorType generatorType )
```

Sets the type of generator.

Parámetros

<i>generatorType</i>	[GeneratorType] The type of generator to set.
----------------------	---

6.21.3.20. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.setWeight (
    float weight )
```

Sets the weight of the generator.

Parámetros

<i>weight</i>	[float] The weight to set.
---------------	----------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.21.3.21. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.EvolutionStrategies.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Updates the reference list with a candidate state.

Parámetros

<i>stateCandidate</i>	[State] The candidate state to consider for updating the reference list.
<i>countIterationsCurrent</i>	[Integer] The current iteration count.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If illegal access occurs.
<i>InvocationTargetException</i>	If an invocation target error occurs.
<i>NoSuchMethodException</i>	If a method cannot be found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/EvolutionStrategies.java`

6.22. Referencia de la clase

es.ull.esit.app.problem.extension.FactoresPonderados

Diagrama de herencia de es.ull.esit.app.problem.extension.FactoresPonderados

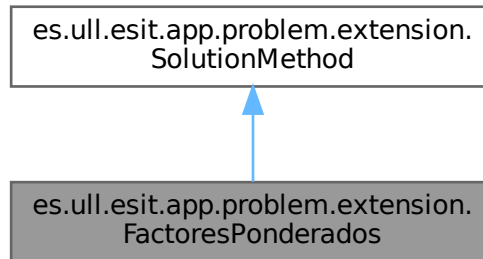
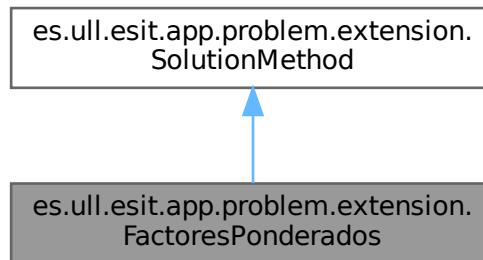


Diagrama de colaboración de es.ull.esit.app.problem.extension.FactoresPonderados:



Métodos públicos

- void [evaluationState](#) (State state)

6.22.1. Descripción detallada

Class that implements the weighted factors solution method.

6.22.2. Documentación de funciones miembro

6.22.2.1. evaluationState()

```
void es.ull.esit.app.problem.extension.FactoresPonderados.evaluationState (
    State state )
```

Constructor for the FactoresPonderados class.

Parámetros

<i>typeSolutionMethod</i>	[TypeSolutionMethod] Type of solution method.
---------------------------	---

Reimplementado de [es.ull.esit.app.problem.extension.SolutionMethod](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/extension/FactoresPonderados.java](#)

6.23. Referencia de la clase es.ull.esit.app.factory_method.FactoryAcceptCandidate

Diagrama de herencia de es.ull.esit.app.factory_method.FactoryAcceptCandidate

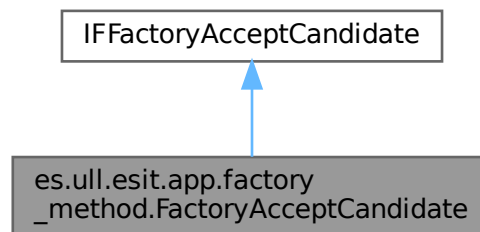
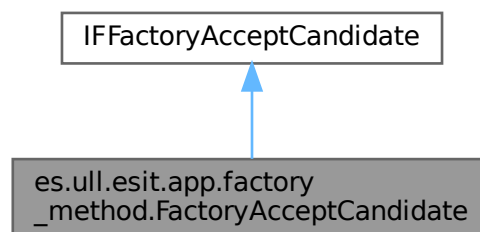


Diagrama de colaboración de es.ull.esit.app.factory_method.FactoryAcceptCandidate:



Métodos públicos

- AcceptableCandidate [createAcceptCandidate](#) (AcceptType typeacceptation) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.23.1. Descripción detallada

Class that implements the factory method for creating AcceptableCandidate instances.

6.23.2. Documentación de funciones miembro

6.23.2.1. createAcceptCandidate()

```
AcceptableCandidate es.ull.esit.app.factory_method.FactoryAcceptCandidate.createAcceptCandidate
(
    AcceptType typeacceptation ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵
Exception, NoSuchMethodException
```

Factory method to create an AcceptableCandidate based on the provided AcceptType.

Parámetros

<i>typeacceptation</i>	[AcceptType] The type of acceptance strategy to create.
------------------------	---

Devuelve

[AcceptableCandidate] An instance of the specified AcceptableCandidate type.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_method/FactoryAcceptCandidate.java`

6.24. Referencia de la clase es.ull.esit.app.factory_method.FactoryCandidate

Diagrama de herencia de es.ull.esit.app.factory_method.FactoryCandidate

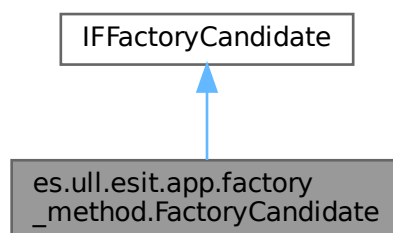
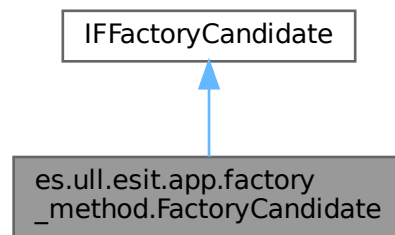


Diagrama de colaboración de es.ull.esit.app.factory_method.FactoryCandidate:



Métodos públicos

- SearchCandidate [createSearchCandidate](#) (CandidateType typeCandidate) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.24.1. Descripción detallada

Class that implements the factory method for creating SearchCandidate instances.

6.24.2. Documentación de funciones miembro

6.24.2.1. createSearchCandidate()

```
SearchCandidate es.ull.esit.app.factory_method.FactoryCandidate.createSearchCandidate (
    CandidateType typeCandidate ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↔
    Exception, NoSuchMethodException
```

Factory method to create a SearchCandidate based on the provided CandidateType.

Parámetros

<i>typeCandidate</i>	[CandidateType] The type of candidate strategy to create.
----------------------	---

Devuelve

[SearchCandidate] An instance of the specified SearchCandidate type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
---------------------------------	----------------------------------

Excepciones

<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_method/FactoryCandidate.java`

6.25. Referencia de la clase `es.ull.esit.app.factory_method.FactoryCrossover`

Diagrama de herencia de `es.ull.esit.app.factory_method.FactoryCrossover`

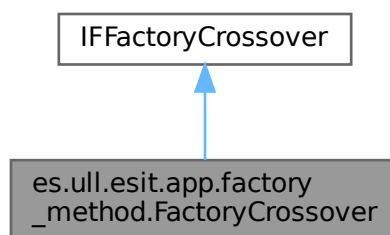
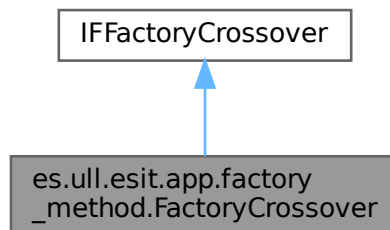


Diagrama de colaboración de `es.ull.esit.app.factory_method.FactoryCrossover`:



Métodos públicos

- Crossover [createCrossover](#) (CrossoverType crossovertype) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.25.1. Descripción detallada

Class that implements the factory method for creating Crossover instances.

6.25.2. Documentación de funciones miembro

6.25.2.1. createCrossover()

```
Crossover es.ull.esit.app.factory_method.FactoryCrossover.createCrossover (
    CrossoverType crossovertype ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetEx←
    ception, NoSuchMethodException
```

Factory method to create a Crossover based on the provided CrossoverType.

Parámetros

<i>crossovertype</i>	[CrossoverType] The type of crossover strategy to create.
----------------------	---

Devuelve

[Crossover] An instance of the specified Crossover type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_method/FactoryCrossover.java`

6.26. Referencia de la clase

es.ull.esit.app.factory_method.FactoryDistribution

Diagrama de herencia de es.ull.esit.app.factory_method.FactoryDistribution

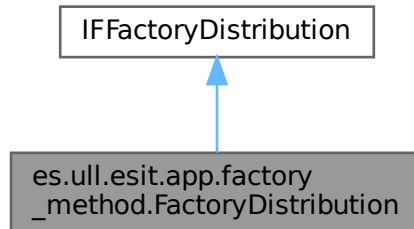
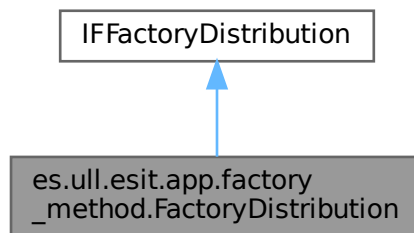


Diagrama de colaboración de es.ull.esit.app.factory_method.FactoryDistribution:



Métodos públicos

- Distribution [createDistribution](#) (DistributionType distributiontype) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.26.1. Descripción detallada

Class that implements the factory method for creating Distribution instances.

6.26.2. Documentación de funciones miembro

6.26.2.1. createDistribution()

```
Distribution es.ull.esit.app.factory_method.FactoryDistribution.createDistribution (
    DistributionType distributiontype ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Factory method to create a Distribution based on the provided DistributionType.

Parámetros

<i>distributiontype</i>	[DistributionType] The type of distribution strategy to create.
-------------------------	---

Devuelve

[Distribution] An instance of the specified Distribution type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_method/[FactoryDistribution.java](#)

6.27. Referencia de la clase

es.ull.esit.app.factory_method.FactoryFatherSelection

Diagrama de herencia de es.ull.esit.app.factory_method.FactoryFatherSelection

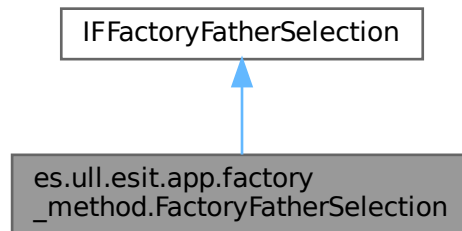
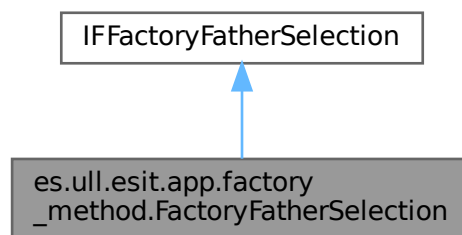


Diagrama de colaboración de es.ull.esit.app.factory_method.FactoryFatherSelection:



Métodos públicos

- `FatherSelection` [createSelectFather](#) (`SelectionType` selectionType) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.27.1. Descripción detallada

Class that implements the factory method for creating `FatherSelection` instances.

6.27.2. Documentación de funciones miembro

6.27.2.1. createSelectFather()

```
FatherSelection es.ull.esit.app.factory_method.FactoryFatherSelection.createSelectFather (
    SelectionType selectionType ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Factory method to create a FatherSelection based on the provided SelectionType.

Parámetros

<i>selectionType</i>	[SelectionType] The type of selection strategy to create.
----------------------	---

Devuelve

[FatherSelection] An instance of the specified FatherSelection type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_method/[FactoryFatherSelection.java](#)

6.28. Referencia de la clase

es.ull.esit.app.factory_method.FactoryGenerator

Diagrama de herencia de es.ull.esit.app.factory_method.FactoryGenerator

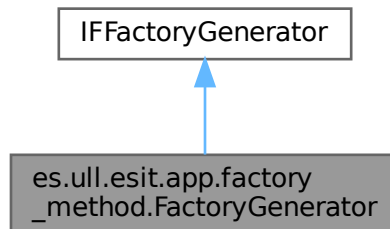
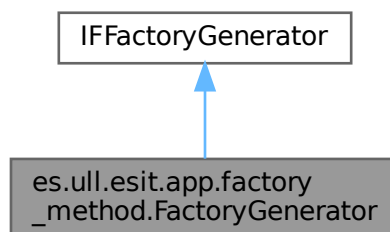


Diagrama de colaboración de es.ull.esit.app.factory_method.FactoryGenerator:



Métodos públicos

- Generator `createGenerator` (GeneratorType generatorType) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.28.1. Descripción detallada

Class that implements the factory method for creating Generator instances.

6.28.2. Documentación de funciones miembro

6.28.2.1. createGenerator()

```
Generator es.ull.esit.app.factory_method.FactoryGenerator.createGenerator (
    GeneratorType generatorType ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Factory method to create a Generator based on the provided GeneratorType.

Parámetros

<i>generatorType</i>	[GeneratorType] The type of generator strategy to create.
----------------------	---

Devuelve

[Generator] An instance of the specified Generator type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_method/[FactoryGenerator.java](#)

6.29. Referencia de la clase es.ull.esit.app.factory_method.FactoryLoader

Métodos públicos estáticos

- static Object [getInstance](#) (String className) throws ClassNotFoundException, IllegalAccessException, InstantiationException, InvocationTargetException, NoSuchMethodException

6.29.1. Descripción detallada

Class that provides a method to load and instantiate classes by name.

6.29.2. Documentación de funciones miembro

6.29.2.1. getInstance()

```
static Object es.ull.esit.app.factory_method.FactoryLoader.getInstance (
    String className ) throws ClassNotFoundException, IllegalAccessException, InstantiationException, InvocationTargetException, NoSuchMethodException [static]
```

Factory method to get an instance of a class given its name.

Parámetros

<i>className</i>	[String] The fully qualified name of the class to instantiate.
------------------	--

Devuelve

[Object] An instance of the specified class.

Excepciones

<i>ClassNotFoundException</i>	If the class cannot be located.
<i>IllegalAccessException</i>	If the class or its nullary constructor is not accessible.
<i>InstantiationException</i>	If the class represents an abstract class, an interface, an array class, a primitive type, or void; or if the class has no nullary constructor; or if the instantiation fails for some other reason.
<i>InvocationTargetException</i>	If the underlying constructor throws an exception.
<i>NoSuchMethodException</i>	If the class does not have a nullary constructor.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_method/[FactoryLoader.java](#)

6.30. Referencia de la clase

es.ull.esit.app.factory_method.FactoryMutation

Diagrama de herencia de es.ull.esit.app.factory_method.FactoryMutation

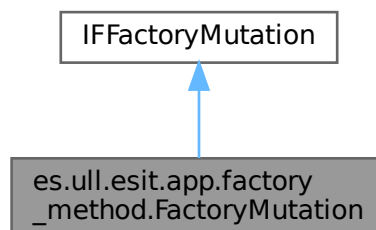
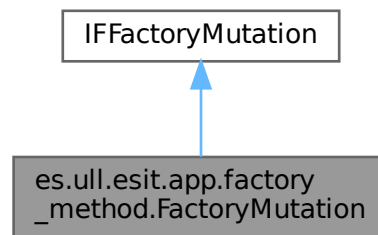


Diagrama de colaboración de es.ull.esit.app.factory_method.FactoryMutation:



Métodos públicos

- Mutation [createMutation](#) (MutationType typeMutation) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchElementException

6.30.1. Descripción detallada

Class that implements the factory method for creating Mutation instances.

6.30.2. Documentación de funciones miembro

6.30.2.1. createMutation()

```

Mutation es.ull.esit.app.factory_method.FactoryMutation.createMutation (
    MutationType typeMutation ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetEx←
    ception, NoSuchElementException
  
```

Factory method to create a Mutation based on the provided MutationType.

Parámetros

<i>typeMutation</i>	[MutationType] The type of mutation strategy to create.
---------------------	---

Devuelve

[Mutation] An instance of the specified Mutation type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
---------------------------------	----------------------------------

Excepciones

<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_method/FactoryMutation.java`

6.31. Referencia de la clase **es.ull.esit.app.factory_method.FactoryReplace**

Diagrama de herencia de `es.ull.esit.app.factory_method.FactoryReplace`

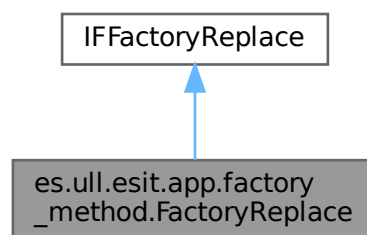
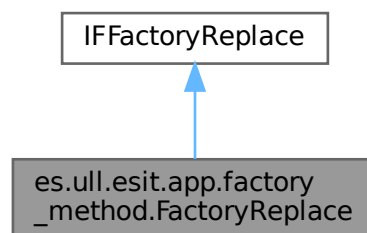


Diagrama de colaboración de `es.ull.esit.app.factory_method.FactoryReplace`:



Métodos públicos

- Replace [createReplace](#) (ReplaceType typereplace) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.31.1. Descripción detallada

Class that implements the factory method for creating Replace instances.

6.31.2. Documentación de funciones miembro

6.31.2.1. createReplace()

```
Replace es.ull.esit.app.factory_method.FactoryReplace.createReplace (
    ReplaceType typereplace ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Factory method to create a Replace based on the provided ReplaceType.

Parámetros

<i>typereplace</i>	[ReplaceType] The type of replace strategy to create.
--------------------	---

Devuelve

[Replace] An instance of the specified Replace type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_method/[FactoryReplace.java](#)

6.32. Referencia de la clase

es.ull.esit.app.factory_method.FactorySampling

Diagrama de herencia de es.ull.esit.app.factory_method.FactorySampling

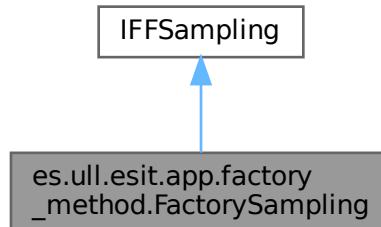
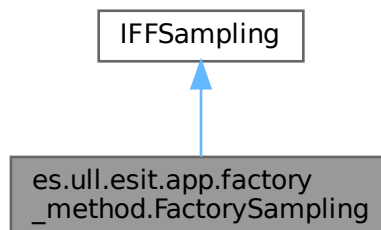


Diagrama de colaboración de es.ull.esit.app.factory_method.FactorySampling:



Métodos públicos

- Sampling [createSampling](#) (SamplingType typesampling) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.32.1. Descripción detallada

Class that implements the factory method for creating Sampling instances.

6.32.2. Documentación de funciones miembro

6.32.2.1. createSampling()

Sampling es.ull.esit.app.factory_method.FactorySampling.createSampling (SamplingType *typesampling*) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

Factory method to create a Sampling based on the provided SamplingType.

Parámetros

<i>typesampling</i>	[SamplingType] The type of sampling strategy to create.
---------------------	---

Devuelve

[Sampling] An instance of the specified Sampling type.

Excepciones

<i>IllegalArgumentException</i>	If the provided type is invalid.
<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_method/[FactorySampling.java](#)

6.33. Referencia de la clase

es.ull.esit.app.factory_method.FactorySolutionMethod

Diagrama de herencia de es.ull.esit.app.factory_method.FactorySolutionMethod

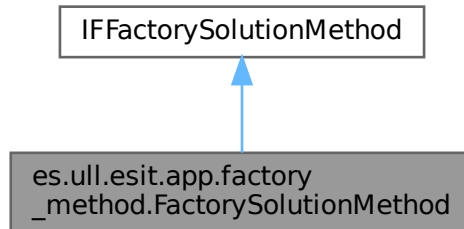
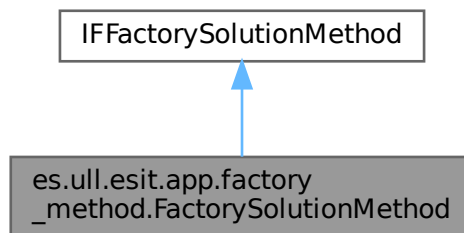


Diagrama de colaboración de es.ull.esit.app.factory_method.FactorySolutionMethod:



Métodos públicos

- `SolutionMethod` [createdSolutionMethod](#) (`TypeSolutionMethod` method) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.33.1. Descripción detallada

Class that implements the factory method for creating `SolutionMethod` instances.

6.33.2. Documentación de funciones miembro

6.33.2.1. createdSolutionMethod()

`SolutionMethod es.ull.esit.app.factory_method.FactorySolutionMethod.createdSolutionMethod (TypeSolutionMethod method)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

Factory method to create a `SolutionMethod` based on the provided `TypeSolutionMethod`.

Parámetros

<i>method</i>	[<code>TypeSolutionMethod</code>] The type of solution method strategy to create.
---------------	---

Devuelve

[`SolutionMethod`] An instance of the specified `SolutionMethod` type.

Excepciones

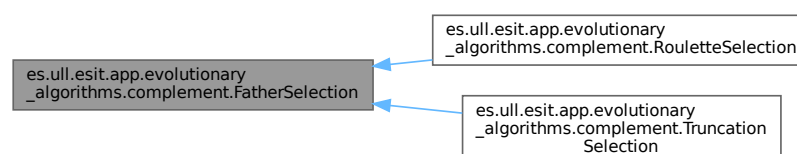
<i>IllegalArgumentException</i>	If the provided type is invalid.
<i>SecurityException</i>	If there is a security violation during instantiation.
<i>ClassNotFoundException</i>	If the class corresponding to the type is not found.
<i>InstantiationException</i>	If there is an error during instantiation.
<i>IllegalAccessException</i>	If there is an illegal access during instantiation.
<i>InvocationTargetException</i>	If the constructor throws an exception.
<i>NoSuchMethodException</i>	If the constructor is not found.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_method/[FactorySolutionMethod.java](#)

6.34. Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection

Diagrama de herencia de es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection



Métodos públicos

- `abstract List< State > selection (List< State > listState, int truncation)`

6.34.1. Descripción detallada

Abstract class that defines the father selection operator.

6.34.2. Documentación de funciones miembro

6.34.2.1. `selection()`

```
abstract List< State > es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection.↵
selection (
    List< State > listState,
    int truncation ) [abstract]
```

Selects a list of father states from the given list of states using truncation selection.

Parámetros

<i>listState</i>	[List<State>] the list of states to select from.
<i>truncation</i>	[int] the truncation parameter.

Devuelve

[List<State>] the list of selected father states.

Reimplementado en [es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection](#) y [es.ull.esit.app.evolutionary_algorithms.c](#)

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/FatherSelection.java`

6.35. Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace

Diagrama de herencia de es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace

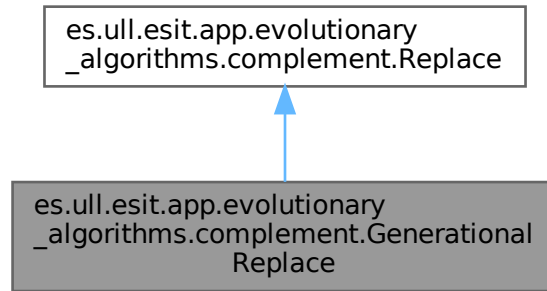
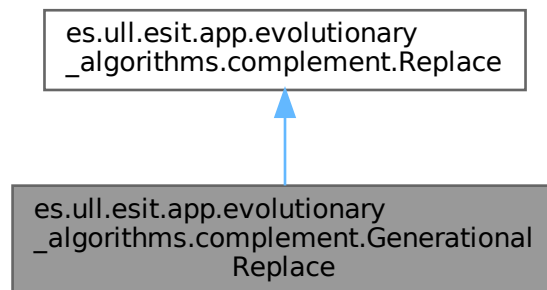


Diagrama de colaboración de es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace:



Métodos públicos

- `List< State > replace (State stateCandidate, List< State > listState)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.35.1. Descripción detallada

Class that implements the generational replacement strategy.

6.35.2. Documentación de funciones miembro

6.35.2.1. replace()

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace.replace (
    State stateCandidate,
    List< State > listState ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
Exception, NoSuchMethodException
```

Replaces the worst state in the population with the candidate state.

Parámetros

<i>stateCandidate</i>	[State] the candidate state to be added to the population.
<i>listState</i>	[List<State>] the current population of states.

Devuelve

[List<State>] the updated population of states after replacement.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Replace](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/GenerationalReplace.java`

6.36. Referencia de la clase es.ull.esit.app.metaheuristics.generators.Generator

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.Generator



Métodos públicos

- abstract State `generate` (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

- abstract void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- abstract State [getReference](#) ()
- abstract void [setInitialReference](#) (State stateInitialRef)
- abstract [GeneratorType](#) [getType](#) ()
- abstract List< State > [getReferenceList](#) ()
- abstract List< State > [getSonList](#) ()
- abstract boolean [awardUpdateREF](#) (State stateCandidate)
- abstract void [setWeight](#) (float weight)
- abstract float [getWeight](#) ()
- abstract float[] [getTrace](#) ()
- abstract int[] [getListCountBetterGender](#) ()
- abstract int[] [getListCountGender](#) ()

Atributos públicos

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.36.1. Descripción detallada

Abstract class for all the generators.

6.36.2. Documentación de funciones miembro

6.36.2.1. [awardUpdateREF\(\)](#)

```
abstract boolean es.ull.esit.app.metaheuristics.generators.Generator.awardUpdateREF (
    State stateCandidate ) [abstract]
```

Abstract method to decide whether to award an update to the reference state.

Parámetros

<i>stateCandidate</i>	[State] The candidate state.
-----------------------	------------------------------

Devuelve

[boolean] True if the update is awarded, false otherwise.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.2. generate()

```
abstract State es.ull.esit.app.metaheuristics.generators.Generator.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    Exception, NoSuchMethodException [abstract]
```

Abstract method to generate a new state.

Parámetros

<i>operatornumber</i>	[Integer] Number of the operator to be used.
-----------------------	--

Excepciones

<i>NoSuchMethodException</i>	If the method does not exist.
<i>InvocationTargetException</i>	If the method cannot be invoked.
<i>IllegalAccessException</i>	If there is no access to the method.
<i>InstantiationException</i>	If there is an error during the instantiation.
<i>ClassNotFoundException</i>	If the class is not found.
<i>SecurityException</i>	If a security violation occurs.
<i>IllegalArgumentException</i>	If an illegal argument is provided.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.3. getListCountBetterGender()

```
abstract int[] es.ull.esit.app.metaheuristics.generators.Generator.getListCountBetterGender (
) [abstract]
```

List of counts for better gender statistics.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.4. `getListCountGender()`

```
abstract int[] es.ull.esit.app.metaheuristics.generators.Generator.getListCountGender ( )
[abstract]
```

List of counts for gender statistics.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.5. `getReference()`

```
abstract State es.ull.esit.app.metaheuristics.generators.Generator.getReference ( ) [abstract]
```

Abstract method to get the reference state.

Devuelve

[State] The reference state.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.6. `getReferenceList()`

```
abstract List< State > es.ull.esit.app.metaheuristics.generators.Generator.getReferenceList (
) [abstract]
```

Abstract method to get the list of reference states.

Devuelve

[List<State>] The list of reference states.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.7. `getSonList()`

```
abstract List< State > es.ull.esit.app.metaheuristics.generators.Generator.getSonList ( )
[abstract]
```

Abstract method to get the list of son states.

Devuelve

[List<State>] The list of son states.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.8. `getTrace()`

```
abstract float[] es.ull.esit.app.metaheuristics.generators.Generator.getTrace ( ) [abstract]
```

Abstract method to get the trace of the generator.

Devuelve

[float[]] The trace of the generator.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.9. `getType()`

```
abstract GeneratorType es.ull.esit.app.metaheuristics.generators.Generator.getType ( ) [abstract]
```

Abstract method to get the type of generator.

Devuelve

[GeneratorType] The type of generator.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.10. getWeight()

```
abstract float es.ull.esit.app.metaheuristics.generators.Generator.getWeight ( ) [abstract]
```

Abstract method to get the weight of the generator.

Devuelve

[float] The weight of the generator.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveParticle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.11. setInitialReference()

```
abstract void es.ull.esit.app.metaheuristics.generators.Generator.setInitialReference (
    State stateInitialRef ) [abstract]
```

Abstract method to set the initial reference state.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state.
------------------------	--------------------------------------

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveParticle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.12. setWeight()

```
abstract void es.ull.esit.app.metaheuristics.generators.Generator.setWeight (
    float weight ) [abstract]
```

Abstract method to decide whether to award an update to the son state.

Parámetros

<i>stateCandidate</i>	[State] The candidate state.
-----------------------	------------------------------

Devuelve

[boolean] True if the update is awarded, false otherwise.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.2.13. updateReference()

```
abstract void es.ull.esit.app.metaheuristics.generators.Generator.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException [abstract]
```

Abstract method to update the reference state.

Parámetros

<i>operatornumber</i>	[Integer] Number of the operator to be used.
-----------------------	--

Excepciones

<i>NoSuchMethodException</i>	If the method does not exist.
<i>InvocationTargetException</i>	If the method cannot be invoked.
<i>IllegalAccessException</i>	If there is no access to the method.
<i>InstantiationException</i>	If there is an error during the instantiation.
<i>ClassNotFoundException</i>	If the class is not found.
<i>SecurityException</i>	If a security violation occurs.
<i>IllegalArgumentException</i>	If an illegal argument is provided.

Reimplementado en [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#), [es.ull.esit.app.metaheuristics.generators.HillClimbing](#), [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#), [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#), [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#), [es.ull.esit.app.metaheuristics.generators.Particle](#), [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#), [es.ull.esit.app.metaheuristics.generators.RandomSearch](#), [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#), [es.ull.esit.app.metaheuristics.generators.TabuSearch](#) y [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#).

6.36.3. Documentación de datos miembro

6.36.3.1. countBetterGender

```
int es.ull.esit.app.metaheuristics.generators.Generator.countBetterGender
```

Counter for better gender statistics.

6.36.3.2. countGender

```
int es.ull.esit.app.metaheuristics.generators.Generator.countGender
```

Counter for gender statistics.

6.36.3.3. listCountBetterGender

```
int [] es.ull.esit.app.metaheuristics.generators.Generator.listCountBetterGender
```

List to hold counts of gender statistics.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/Generator.java`

6.37. Referencia de la enumeración `es.ull.esit.app.metaheuristics.generators.GeneratorType`

Atributos públicos

- `TabuSearch`
- `SimulatedAnnealing`
- `StochasticHillClimbing`
- `RandomSearch`
- `LimitThreshold`
- `HillClimbingRestart`
- `HillClimbingDistance`
- `GeneticAlgorithm`
- `EvolutionStrategies`
- `DistributionEstimationAlgorithm`
- `ParticleSwarmOptimization`
- `MultiGenerator`
- `MultiobjectiveTabuSearch`
- `MultiobjectiveStochasticHillClimbing`
- `MultiCaseSimulatedAnnealing`
- `MultiobjectiveHillClimbingRestart`
- `MultiobjectiveHillClimbingDistance`
- `HillClimbing`

6.37.1. Descripción detallada

Enumeration of the different types of generators.

6.37.2. Documentación de datos miembro

6.37.2.1. DistributionEstimationAlgorithm

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.DistributionEstimationAlgorithm
```

Evolution Strategies generator type.

6.37.2.2. EvolutionStrategies

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.EvolutionStrategies
```

Genetic Algorithm generator type.

6.37.2.3. GeneticAlgorithm

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.GeneticAlgorithm
```

Hill Climbing with Distance generator type.

6.37.2.4. HillClimbing

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.HillClimbing
```

Multi-objective Hill Climbing with Distance generator type.

6.37.2.5. HillClimbingDistance

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.HillClimbingDistance
```

Hill Climbing with Restart generator type.

6.37.2.6. HillClimbingRestart

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.HillClimbingRestart
```

Limit Threshold generator type.

6.37.2.7. LimitThreshold

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.LimitThreshold
```

Random Search generator type.

6.37.2.8. MultiCaseSimulatedAnnealing

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.MultiCaseSimulatedAnnealing
```

Multi-objective Stochastic Hill Climbing generator type.

6.37.2.9. MultiGenerator

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.MultiGenerator
```

Particle Swarm Optimization generator type.

6.37.2.10. MultiobjectiveHillClimbingDistance

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.MultiobjectiveHillClimbingDistance
```

Multi-objective Hill Climbing with Restart generator type.

6.37.2.11. MultiobjectiveHillClimbingRestart

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.MultiobjectiveHillClimbingRestart
```

Multi-case Simulated Annealing generator type.

6.37.2.12. MultiobjectiveStochasticHillClimbing

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.MultiobjectiveStochasticHillClimbing
```

Multi-objective Tabu Search generator type.

6.37.2.13. MultiobjectiveTabuSearch

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.MultiobjectiveTabuSearch
```

Multi-generator type.

6.37.2.14. ParticleSwarmOptimization

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.ParticleSwarmOptimization
```

Distribution Estimation Algorithm generator type.

6.37.2.15. RandomSearch

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.RandomSearch
```

Stochastic Hill Climbing generator type.

6.37.2.16. SimulatedAnnealing

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.SimulatedAnnealing
```

Tabu Search generator type.

6.37.2.17. StochasticHillClimbing

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.StochasticHillClimbing
```

Simulated Annealing generator type.

6.37.2.18. TabuSearch

```
es.ull.esit.app.metaheuristics.generators.GeneratorType.TabuSearch
```

La documentación de esta enumeración está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/GeneratorType.java`

6.38. Referencia de la clase `es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm`

Diagrama de herencia de `es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm`

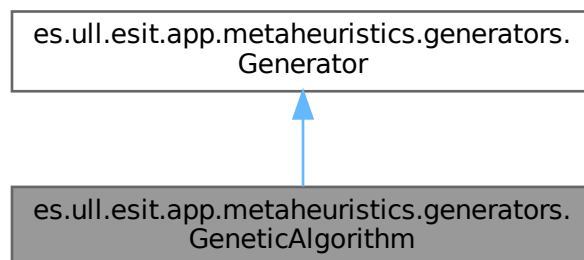
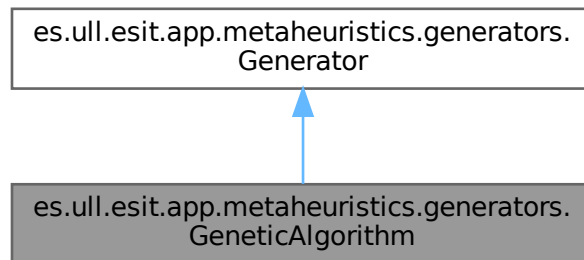


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm`:



Métodos públicos

- `GeneticAlgorithm ()`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `State getReference ()`
- `void setStateRef (State stateRef)`
- `void setInitialReference (State stateInitialRef)`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `List< State > getListState ()`
- `void setListState (List< State > listState)`
- `List< State > getListStateRef ()`
- `GeneratorType getGeneratorType ()`
- `void setGeneratorType (GeneratorType generatorType)`
- `GeneratorType getType ()`
- `List< State > getReferenceList ()`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `float getWeight ()`
- `void setWeight (float weight)`
- `int[] getListCountBetterGender ()`
- `int[] getListCountGender ()`
- `float[] getTrace ()`

Métodos públicos estáticos

- `static int getCountRef ()`
- `static void setCountRef (int countRef)`

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- `int` `countGender`
- `int` `countBetterGender`
- `int[]` `listCountBetterGender`

6.38.1. Descripción detallada

Genetic Algorithm generator implementation.

6.38.2. Documentación de constructores y destructores

6.38.2.1. `GeneticAlgorithm()`

```
es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.GeneticAlgorithm ( )
```

Constructor

6.38.3. Documentación de funciones miembro

6.38.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.awardUpdateREF (
    State stateCandidate )
```

Awards an update to the reference state with a candidate state.

Parámetros

<code>stateCandidate</code>	[State] The candidate state.
-----------------------------	------------------------------

Devuelve

[boolean] True if the update is awarded, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.2. `generate()`

```
State es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new state using the Genetic Algorithm.

Parámetros

<i>operatornumber</i>	[Integer] The operator number.
-----------------------	--------------------------------

Devuelve

[State] The generated state.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class is not found.
<i>InstantiationException</i>	If there is an error during the instantiation.
<i>IllegalAccessException</i>	If there is no access to the method.
<i>InvocationTargetException</i>	If the method cannot be invoked.
<i>NoSuchMethodException</i>	If the method does not exist.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.3. getCountRef()

```
static int es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getCountRef ( ) [static]
```

Gets the reference count for the Genetic Algorithm.

Devuelve

[int] The reference count.

6.38.3.4. getGeneratorType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getGeneratorType ( )
```

Gets the generator type.

Devuelve

[GeneratorType] The generator type.

6.38.3.5. getListCountBetterGender()

```
int [ ] es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getListCountBetterGender ( )
```

Gets the list of better gender counts.

Devuelve

[int[]] The list of better gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.6. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getListCountGender ( )
```

Gets the list of gender counts.

Devuelve

[int[]] The list of gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.7. `getListState()`

```
List< State > es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getListState ( )
```

Gets the list of states in the Genetic Algorithm.

Devuelve

[List<State>] The list of states.

6.38.3.8. `getListStateRef()`

```
List< State > es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getListStateRef ( )
```

Gets the reference list of states from the strategy.

Devuelve

[List<State>] The reference list of states.

6.38.3.9. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getReference ( )
```

Gets the reference state for the Genetic Algorithm.

Devuelve

[State] The reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.10. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getReferenceList ( )
```

Gets the list of reference states.

Devuelve

[List<State>] The list of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.11. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getSonList ( )
```

Gets the list of son states.

Devuelve

[List<State>] The list of son states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.12. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getTrace ( )
```

Gets the list of trace values.

Devuelve

[float[]] The list of trace values.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.13. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getType ( )
```

Gets the type of generator.

Devuelve

[GeneratorType] The type of generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.14. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.getWeight ( )
```

Gets the weight parameter for the Genetic Algorithm.

Devuelve

[float] The weight parameter.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.15. `setCountRef()`

```
static void es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.setCountRef (
    int countRef ) [static]
```

Sets the reference count for the Genetic Algorithm.

Parámetros

<i>countRef</i>	[int] The reference count to set.
-----------------	-----------------------------------

6.38.3.16. setGeneratorType()

```
void es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.setGeneratorType (
    GeneratorType generatorType )
```

Sets the generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] The generator type to set.
----------------------	--

6.38.3.17. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state for the Genetic Algorithm.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state.
------------------------	--------------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.18. setListState()

```
void es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.setListState (
    List< State > listState )
```

Sets the list of states in the Genetic Algorithm.

Parámetros

<i>listState</i>	[List<State>] The list of states to set.
------------------	--

6.38.3.19. setStateRef()

```
void es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.setStateRef (
    State stateRef )
```

Sets the reference state for the Genetic Algorithm.

Parámetros

<i>stateRef</i>	[State] The reference state to set.
-----------------	-------------------------------------

6.38.3.20. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.setWeight (
    float weight )
```

Sets the weight parameter for the Genetic Algorithm.

Parámetros

<i>weight</i>	[float] The weight parameter to set.
---------------	--------------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.38.3.21. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Updates the reference state with a candidate state.

Parámetros

<i>stateCandidate</i>	[State] The candidate state.
<i>countIterationsCurrent</i>	[Integer] The current iteration count.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class is not found.
<i>InstantiationException</i>	If there is an error during the instantiation.
<i>IllegalAccessException</i>	If there is no access to the method.
<i>InvocationTargetException</i>	If the method cannot be invoked.
<i>NoSuchMethodException</i>	If the method does not exist.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/GeneticAlgorithm.java`

6.39. Referencia de la clase

es.ull.esit.app.local_search.candidate_type.GreaterCandidate

Diagrama de herencia de es.ull.esit.app.local_search.candidate_type.GreaterCandidate

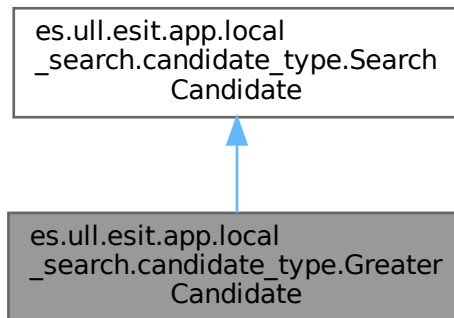
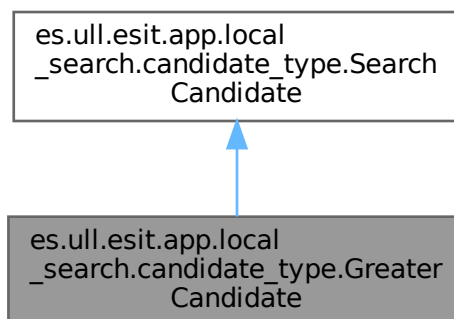


Diagrama de colaboración de es.ull.esit.app.local_search.candidate_type.GreaterCandidate:



Métodos públicos

- State [stateSearch](#) (List< State > listNeighborhood) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.39.1. Descripción detallada

Class that represents a candidate with a greater objective function value in local search algorithms.

6.39.2. Documentación de funciones miembro

6.39.2.1. `stateSearch()`

```
State es.ull.esit.app.local_search.candidate_type.GreaterCandidate.stateSearch (
    List< State > listNeighborhood ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Creates a new State from the list of neighborhood states with the greatest objective function value.

Parámetros

<i>listNeighborhood</i>	[List<State>] List of neighborhood states.
-------------------------	--

Devuelve

[State] State with the greatest objective function value.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If there is an illegal access.
<i>InvocationTargetException</i>	If the method invocation fails.
<i>NoSuchMethodException</i>	If the method is not found.

Reimplementado de [es.ull.esit.app.local_search.candidate_type.SearchCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/candidate_type/GreaterCandidate.java`

6.40. Referencia de la clase

es.ull.esit.app.metaheuristics.generators.HillClimbing

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.HillClimbing

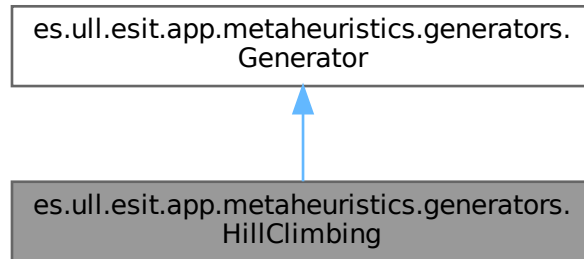
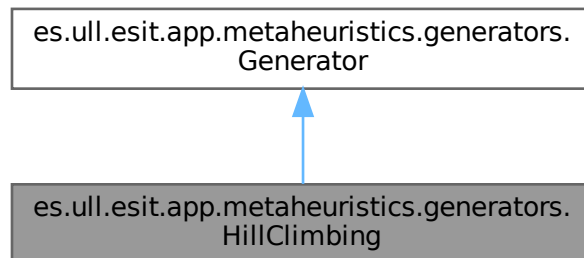


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.HillClimbing:



Métodos públicos

- [HillClimbing](#) ()
- State [generate](#) (Integer operatornumber) throws IllegalArgumentException, SecurityException, Class↔NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuch↔MethodException
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException↔Exception, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- List< State > [getReferenceList](#) ()
- State [getReference](#) ()
- void [setStateRef](#) (State stateRef)
- void [setInitialReference](#) (State stateInitialRef)
- [GeneratorType](#) [getGeneratorType](#) ()

- void `setGeneratorType` (`GeneratorType` generatorType)
- `GeneratorType` `getType` ()
- List< State > `getSonList` ()
- void `setTypeCandidate` (`CandidateType` typeCandidate)
- boolean `awardUpdateREF` (`State` stateCandidate)
- float `getWeight` ()
- void `setWeight` (float weight)
- int[] `getListCountBetterGender` ()
- int[] `getListCountGender` ()
- float[] `getTrace` ()

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.40.1. Descripción detallada

Class that implements the Hill Climbing generator.

6.40.2. Documentación de constructores y destructores

6.40.2.1. `HillClimbing()`

```
es.ull.esit.app.metaheuristics.generators.HillClimbing.HillClimbing ( )
```

Default constructor. Initializes parameters following the pattern of other generators.

6.40.3. Documentación de funciones miembro

6.40.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.HillClimbing.awardUpdateREF (
    State stateCandidate )
```

Indicates if the candidate improves the reference, according to the problem type. This is used as a "reward" in portfolio generators.

Parámetros

<code>stateCandidate</code>	[State] The candidate state to evaluate.
-----------------------------	--

Devuelve

[boolean] True if the candidate improves the reference, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.HillClimbing.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new candidate state from the neighborhood of the reference state.

Parámetros

<i>operatornumber</i>	[Integer] The operator number to use for generating neighbors.
-----------------------	--

Devuelve

[State] The generated candidate state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.3. getGeneratorType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.HillClimbing.getGeneratorType ( )
```

Gets the specific generator type.

Devuelve

[GeneratorType] The specific generator type.

6.40.3.4. getListCountBetterGender()

```
int[] es.ull.esit.app.metaheuristics.generators.HillClimbing.getListCountBetterGender ( )
```

Returns the array of "better gender" counters (for 10 periods).

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.5. getListCountGender()

```
int[] es.ull.esit.app.metaheuristics.generators.HillClimbing.getListCountGender ( )
```

Returns the array of "gender" counters (for 10 periods).

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.6. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.HillClimbing.getReference ( )
```

Gets the current reference state.

Devuelve

[State] The current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.7. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.HillClimbing.getReferenceList ( )
```

Gets the list of reference states (best solutions found).

Devuelve

[List<State>] The list of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.8. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.HillClimbing.getSonList ( )
```

Gets the list of son states (not used in Hill Climbing).

Devuelve

[List<State>] An empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.9. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.HillClimbing.getTrace ( )
```

Gets the trace array.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.10. getType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.HillClimbing.getType ( )
```

Gets the type of the generator.

Devuelve

[GeneratorType] The type of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.11. getWeight()

```
float es.ull.esit.app.metaheuristics.generators.HillClimbing.getWeight ( )
```

Gets the weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.12. setGeneratorType()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbing.setGeneratorType (
    GeneratorType generatorType )
```

Sets the specific generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] The generator type to set.
----------------------	--

6.40.3.13. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbing.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state to set.
------------------------	---

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.14. setStateRef()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbing.setStateRef (
    State stateRef )
```

Sets the reference state explicitly.

Parámetros

<i>stateRef</i>	[State] The reference state to set.
-----------------	-------------------------------------

6.40.3.15. setTypeCandidate()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbing.setTypeCandidate (
    CandidateType typeCandidate )
```

Allows changing the type of candidate used (greater, lesser, etc.).

6.40.3.16. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbing.setWeight (
    float weight )
```

Sets the weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.40.3.17. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, Security↔
Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocation↔
TargetException, NoSuchMethodException
```

Updates the reference state using the configured acceptance policy.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbing.java`

6.41. Referencia de la clase

es.ull.esit.app.metaheuristics.generators.HillClimbingRestart

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.HillClimbingRestart

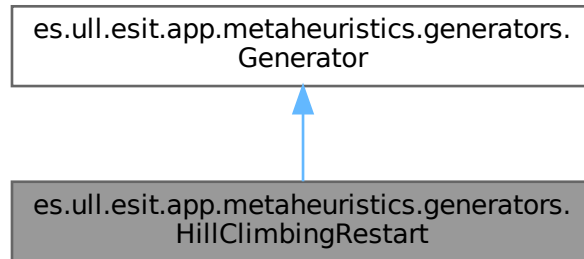
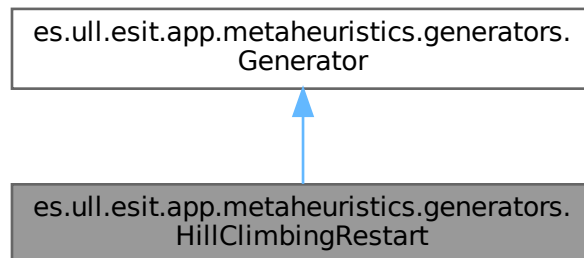


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.HillClimbingRestart:



Métodos públicos

- `HillClimbingRestart ()`
- State `generate` (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- void `updateReference` (State stateCandidate, Integer countIterationsCurrent) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- List< State > `getReferenceList` ()
- State `getReference` ()
- void `setStateRef` (State stateRef)
- void `setInitialReference` (State stateInitialRef)
- `GeneratorType getGeneratorType` ()

- void `setGeneratorType` (`GeneratorType` generatorType)
- `GeneratorType` `getType` ()
- List< State > `getSonList` ()
- void `setTypeCandidate` (`CandidateType` typeCandidate)
- boolean `awardUpdateREF` (`State` stateCandidate)
- float `getWeight` ()
- void `setWeight` (float weight)
- int[] `getListCountBetterGender` ()
- int[] `getListCountGender` ()
- float[] `getTrace` ()
- int `getCount` ()
- void `setCount` (int countValue)
- int `getCountCurrent` ()
- void `setCountCurrent` (int countCurrentValue)
- int `getCountGender` ()
- void `setCountGender` (int countGenderValue)
- int `getCountBetterGender` ()
- void `setCountBetterGender` (int countBetterGenderValue)

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.41.1. Descripción detallada

Hill class that implements a Hill Climbing with Restart generator.

6.41.2. Documentación de constructores y destructores

6.41.2.1. `HillClimbingRestart()`

```
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.HillClimbingRestart ( )
```

Default constructor. Initializes acceptance/candidate types and statistics structures.

6.41.3. Documentación de funciones miembro

6.41.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.awardUpdateREF (
    State stateCandidate )
```

Decides whether the reference should be updated based on the evaluations of the candidate and current reference, and the problem type (Maximize / Minimize).

Parámetros

<i>stateCandidate</i>	[State] The candidate state to evaluate.
-----------------------	--

Devuelve

true if the candidate should replace the reference, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Generates a new candidate state. When the restart condition is met, the current reference is stored and a new random reference state is created.

Parámetros

<i>operatornumber</i>	operator index used by the neighborhood generator.
-----------------------	--

Devuelve

the new candidate state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.3. getCount()

```
int es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getCount ( )
```

Static counter for restarts.

Devuelve

[int] The current value of the restart counter.

6.41.3.4. getCountBetterGender()

```
int es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getCountBetterGender ( )
```

Gets the count of better genders.

Devuelve

[int] The current count of better genders for this generator.

6.41.3.5. `getCountCurrent()`

```
int es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getCountCurrent ( )
```

Gets the current step count for restarts.

Devuelve

[int] The current step count for restarts.

6.41.3.6. `getCountGender()`

```
int es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getCountGender ( )
```

Gets the count of genders.

Devuelve

[int] The current count of genders for this generator.

6.41.3.7. `getGeneratorType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getGeneratorType ( )
```

Gets the generator type.

Devuelve

[GeneratorType] The type of the generator.

6.41.3.8. `getListCountBetterGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getListCountBetterGender ( )
```

Gets the list of better gender counts for this generator.

Devuelve

[int[]] The list of better gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.9. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getListCountGender ( )
```

Gets the list of gender counts for this generator.

Devuelve

[int[]] The list of gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.10. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getReference ( )
```

Abstract method to get the reference state.

Devuelve

[State] The reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.11. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getReferenceList ( )
```

Returns a copy of all reference states handled by this generator: the current reference (if any) plus all stored references from restarts.

Devuelve

[List<State>] list of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.12. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getSonList ( )
```

HillClimbingRestart does not maintain a separate list of sons, so an empty list is returned to avoid null handling issues.

Devuelve

[List<State>] An empty list of son states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.13. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getTrace ( )
```

Gets the trace of this generator.

Devuelve

`[float[]]` The trace of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.14. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getType ( )
```

Gets the generator type.

Devuelve

`[GeneratorType]` The type of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.15. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.getWeight ( )
```

Gets the weight of this generator.

Devuelve

`[float]` The weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.16. `setCount()`

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setCount (
    int countValue )
```

Sets the static counter for restarts.

Parámetros

<code>countValue</code>	<code>[int]</code> The new value for the restart counter.
-------------------------	---

6.41.3.17. setCountBetterGender()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setCountBetterGender (
    int countBetterGenderValue )
```

Sets the count of better genders.

Parámetros

<i>countBetterGenderValue</i>	[int] The new count of better genders for this generator.
-------------------------------	---

6.41.3.18. setCountCurrent()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setCountCurrent (
    int countCurrentValue )
```

Sets the current step count for restarts.

Parámetros

<i>countCurrentValue</i>	[int] The new value for the current step count.
--------------------------	---

6.41.3.19. setCountGender()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setCountGender (
    int countGenderValue )
```

Sets the count of genders.

Parámetros

<i>countGenderValue</i>	[int] The new count of genders for this generator.
-------------------------	--

6.41.3.20. setGeneratorType()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setGeneratorType (
    GeneratorType generatorType )
```

Sets the generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] The type of the generator.
----------------------	--

6.41.3.21. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state for hill climbing.

Parámetros

<code>stateInitialRef</code>	[State] Initial reference state.
------------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.22. `setStateRef()`

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setStateRef (
    State stateRef )
```

Sets the current reference state.

Parámetros

<code>stateRef</code>	[State] New reference state.
-----------------------	------------------------------

6.41.3.23. `setTypeCandidate()`

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setTypeCandidate (
    CandidateType typeCandidate )
```

Sets the candidate comparison type.

Parámetros

<code>typeCandidate</code>	[CandidateType] The candidate comparison type.
----------------------------	--

6.41.3.24. `setWeight()`

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.setWeight (
    float weight )
```

Sets the weight of this generator.

Parámetros

<code>weight</code>	[float] The new weight of the generator.
---------------------	--

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.41.3.25. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.HillClimbingRestart.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Updates the reference state if the candidate is accepted according to the acceptance policy.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbingRestart.java](#)

6.42. Referencia de la interface es.ull.esit.app.factory_interface.IFFactoryAcceptCandidate

Métodos públicos

- AcceptableCandidate [createAcceptCandidate](#) (AcceptType typeacceptation) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.42.1. Descripción detallada

Interface for the Factory of AcceptCandidate objects

6.42.2. Documentación de funciones miembro

6.42.2.1. createAcceptCandidate()

```
AcceptableCandidate es.ull.esit.app.factory_interface.IFFactoryAcceptCandidate.createAcceptCandidate (
    AcceptType typeacceptation ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Method to create an AcceptCandidate object based on the type of acceptance.

Parámetros

<i>typeacceptation</i>	[AcceptType] The type of acceptance.
------------------------	--------------------------------------

Devuelve

[AcceptableCandidate] The created AcceptCandidate object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_interface/[IFFactoryAcceptCandidate.java](#)

6.43. Referencia de la interface es.ull.esit.app.factory_interface.IFFactoryCandidate

Métodos públicos

- SearchCandidate [createSearchCandidate](#) (CandidateType typeCandidate) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.43.1. Descripción detallada

Interface for the Factory of Candidate objects.

6.43.2. Documentación de funciones miembro**6.43.2.1. createSearchCandidate()**

```
SearchCandidate es.ull.esit.app.factory_interface.IFFactoryCandidate.createSearchCandidate (
    CandidateType typeCandidate ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵
    Exception, NoSuchMethodException
```

Method to create a Candidate object based on the type of candidate.

Parámetros

<i>typeCandidate</i>	[CandidateType] The type of candidate.
----------------------	--

Devuelve

[SearchCandidate] The created Candidate object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_interface/IFactoryCandidate.java`

6.44. Referencia de la interface es.ull.esit.app.factory_interface.IFactoryCrossover

Métodos públicos

- Crossover [createCrossover](#) (CrossoverType crossovertype) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)

6.44.1. Descripción detallada

Interface for the Factory of Crossover objects.

6.44.2. Documentación de funciones miembro**6.44.2.1. createCrossover()**

```
Crossover es.ull.esit.app.factory_interface.IFactoryCrossover.createCrossover (
    CrossoverType crossovertype ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetEx←
    ception, NoSuchMethodException
```

Method to create a Crossover object based on the type of crossover.

Parámetros

<i>crossovertype</i>	[CrossoverType] The type of crossover.
----------------------	--

Devuelve

[Crossover] The created Crossover object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_interface/IFFactoryCrossover.java`

6.45. Referencia de la interface es.ull.esit.app.factory_interface.IFFactoryDistribution

Métodos públicos

- Distribution [createDistribution](#) (DistributionType typedistribution) throws *IllegalArgumentException*, *SecurityException*, *ClassNotFoundException*, *InstantiationException*, *IllegalAccessException*, *InvocationTargetException*, *NoSuchMethodException*

6.45.1. Descripción detallada

Interface for the Factory of Distribution objects.

6.45.2. Documentación de funciones miembro**6.45.2.1. createDistribution()**

```
Distribution es.ull.esit.app.factory_interface.IFFactoryDistribution.createDistribution (
    DistributionType typedistribution ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Method to create a Distribution object based on the type of distribution.

Parámetros

<i>typedistribution</i>	[DistributionType] The type of distribution.
-------------------------	--

Devuelve

[Distribution] The created Distribution object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_interface/IFactoryDistribution.java`

6.46. Referencia de la interface es.ull.esit.app.factory_interface.IFactoryFatherSelection

Métodos públicos

- FatherSelection [createSelectFather](#) (SelectionType selectionType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.46.1. Descripción detallada

Interface for the Factory of FatherSelection objects.

6.46.2. Documentación de funciones miembro**6.46.2.1. createSelectFather()**

```
FatherSelection es.ull.esit.app.factory_interface.IFactoryFatherSelection.createSelectFather
(
    SelectionType selectionType ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵
Exception, NoSuchMethodException
```

Method to create a FatherSelection object based on the type of selection.

Parámetros

<i>selectionType</i>	[SelectionType] The type of selection.
----------------------	--

Devuelve

[FatherSelection] The created FatherSelection object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_interface/IFFactoryFatherSelection.java`

6.47. Referencia de la interface `es.ull.esit.app.factory_interface.IFFactoryGenerator`

Métodos públicos

- Generator `createGenerator` (GeneratorType generatortype) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.47.1. Descripción detallada

Interface for the Factory of Generator objects.

6.47.2. Documentación de funciones miembro**6.47.2.1. `createGenerator()`**

```
Generator es.ull.esit.app.factory_interface.IFFactoryGenerator.createGenerator (
    GeneratorType generatortype ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵
    Exception, NoSuchMethodException
```

Method to create a Generator object based on the type of generator.

Parámetros

<i>generatortype</i>	[GeneratorType] The type of generator.
----------------------	--

Devuelve

[Generator] The created Generator object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_interface/IFactoryGenerator.java`

6.48. Referencia de la interface es.ull.esit.app.factory_interface.IFactoryMutation

Métodos públicos

- Mutation [createMutation](#) (MutationType typeMutation) throws *IllegalArgumentException*, *SecurityException*, *ClassNotFoundException*, *InstantiationException*, *IllegalAccessException*, *InvocationTargetException*, *NoSuchMethodException*

6.48.1. Descripción detallada

Interface for the Factory of Mutation objects.

6.48.2. Documentación de funciones miembro**6.48.2.1. createMutation()**

```
Mutation es.ull.esit.app.factory_interface.IFactoryMutation.createMutation (
    MutationType typeMutation ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetEx←
ception, NoSuchMethodException
```

Method to create a Mutation object based on the type of mutation.

Parámetros

<i>typeMutation</i>	[MutationType] The type of mutation.
---------------------	--------------------------------------

Devuelve

[Mutation] The created Mutation object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/factory_interface/[IFFactoryMutation.java](#)

6.49. Referencia de la interface es.ull.esit.app.factory_interface.IFFactoryReplace

Métodos públicos

- Replace [createReplace](#) (ReplaceType typereplace) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.49.1. Descripción detallada

Interface for the Factory of Replace objects.

6.49.2. Documentación de funciones miembro**6.49.2.1. createReplace()**

```
Replace es.ull.esit.app.factory_interface.IFFactoryReplace.createReplace (
    ReplaceType typereplace ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetEx←
    ception, NoSuchMethodException
```

Method to create a Replace object based on the type of replace.

Parámetros

<i>typereplace</i>	[ReplaceType] The type of replace.
--------------------	------------------------------------

Devuelve

[Replace] The created Replace object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_interface/IFactoryReplace.java`

6.50. Referencia de la interface

es.ull.esit.app.factory_interface.IFactorySolutionMethod

Métodos públicos

- `SolutionMethod` [createdSolutionMethod](#) (`TypeSolutionMethod method`) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.50.1. Descripción detallada

Interface for the Factory of SolutionMethod objects.

6.50.2. Documentación de funciones miembro**6.50.2.1. createdSolutionMethod()**

```
SolutionMethod es.ull.esit.app.factory_interface.IFactorySolutionMethod.createdSolutionMethod
(
    TypeSolutionMethod method ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetEx←
ception, NoSuchMethodException
```

Method to create a SolutionMethod object based on the type of method.

Parámetros

<i>method</i>	[TypeSolutionMethod] The type of solution method.
---------------	---

Devuelve

[SolutionMethod] The created SolutionMethod object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/factory_interface/IFFFactorySolutionMethod.java](#)

6.51. Referencia de la interface es.ull.esit.app.factory_interface.IFFSampling

Métodos públicos

- Sampling [createSampling](#) (SamplingType typesampling) throws *IllegalArgumentException*, *SecurityException*, *ClassNotFoundException*, *InstantiationException*, *IllegalAccessException*, *InvocationTargetException*, *NoSuchMethodException*

6.51.1. Descripción detallada

Interface for the Factory of Sampling objects.

6.51.2. Documentación de funciones miembro**6.51.2.1. createSampling()**

```
Sampling es.ull.esit.app.factory_interface.IFFSampling.createSampling (
    SamplingType typesampling ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetEx←
    ception, NoSuchMethodException
```

Method to create a Sampling object based on the type of sampling.

Parámetros

<i>typesampling</i>	[SamplingType] The type of sampling.
---------------------	--------------------------------------

Devuelve

[Sampling] The created Sampling object.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

La documentación de esta interface está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/factory_interface/IFFSampling.java`

6.52. Referencia de la clase

es.ull.esit.app.metaheuristics.generators.InstanceDE

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.InstanceDE

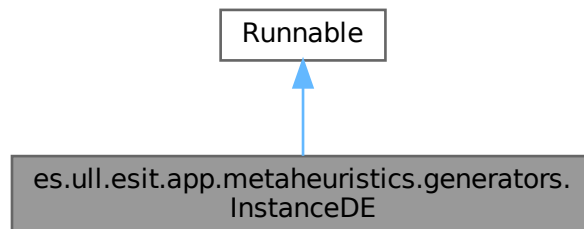
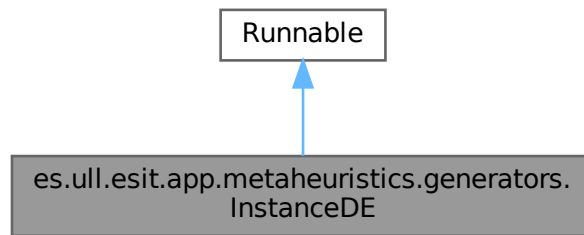


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.InstanceDE`:



Métodos públicos

- void `run()`
- boolean `isTerminate()`
- void `setTerminate(boolean terminate)`

6.52.1. Descripción detallada

Runnable that creates a Distribution Estimation Algorithm (DEA) generator in a separate thread and updates the MultiGenerator list.

6.52.2. Documentación de funciones miembro

6.52.2.1. `isTerminate()`

```
boolean es.ull.esit.app.metaheuristics.generators.InstanceDE.isTerminate ( )
```

Indicates whether this task has terminated.

Devuelve

true if the task has finished; false otherwise.

6.52.2.2. `run()`

```
void es.ull.esit.app.metaheuristics.generators.InstanceDE.run ( )
```

Creates a DEA generator using FactoryGenerator and replaces the existing DEA entry in MultiGenerator, if found.

6.52.2.3. `setTerminate()`

```
void es.ull.esit.app.metaheuristics.generators.InstanceDE.setTerminate (
    boolean terminate )
```

Sets the termination flag. Intended mainly for testing.

Parámetros

<i>terminate</i>	true if the task should be marked as finished; false otherwise.
------------------	---

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceDE.java`

6.53. Referencia de la clase

es.ull.esit.app.metaheuristics.generators.InstanceEE

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.InstanceEE

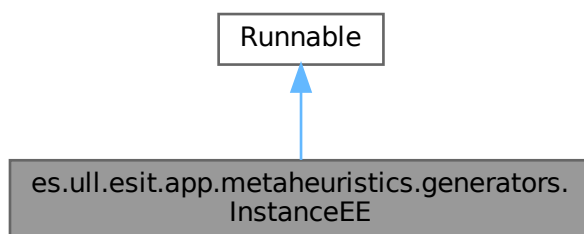
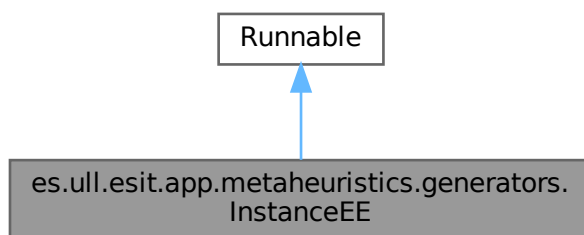


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.InstanceEE:

**Métodos públicos**

- void `run` ()
- boolean `isTerminate` ()
- void `setTerminate` (boolean terminate)

6.53.1. Descripción detallada

Runnable that creates an Evolution Strategies (ES) generator in a separate thread and updates the MultiGenerator list.

6.53.2. Documentación de funciones miembro

6.53.2.1. `isTerminate()`

```
boolean es.ull.esit.app.metaheuristics.generators.InstanceEE.isTerminate ( )
```

Indicates whether this task has terminated.

Devuelve

true if the task has finished; false otherwise.

6.53.2.2. `run()`

```
void es.ull.esit.app.metaheuristics.generators.InstanceEE.run ( )
```

Creates an EvolutionStrategies generator using FactoryGenerator and replaces the existing ES entry in MultiGenerator, if found.

6.53.2.3. `setTerminate()`

```
void es.ull.esit.app.metaheuristics.generators.InstanceEE.setTerminate (
    boolean terminate )
```

Sets the termination flag. Intended mainly for testing.

Parámetros

<i>terminate</i>	true if the task should be marked as finished; false otherwise.
------------------	---

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceEE.java`

6.54. Referencia de la clase

es.ull.esit.app.metaheuristics.generators.InstanceGA

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.InstanceGA

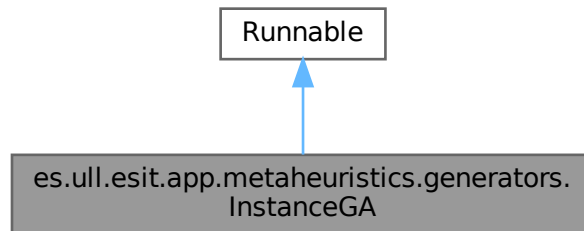
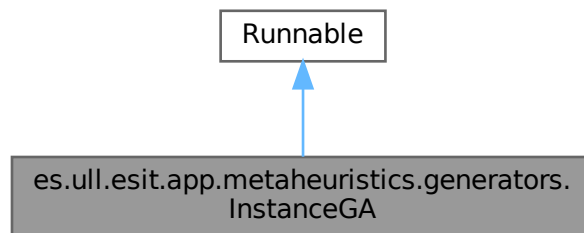


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.InstanceGA:



Métodos públicos

- void [run](#) ()
- boolean [isTerminate](#) ()
- void [setTerminate](#) (boolean terminate)

6.54.1. Descripción detallada

Runnable that creates a Genetic Algorithm (GA) generator in a separate thread and updates the MultiGenerator list.

6.54.2. Documentación de funciones miembro

6.54.2.1. `isTerminate()`

```
boolean es.ull.esit.app.metaheuristics.generators.InstanceGA.isTerminate ( )
```

Indicates whether this task has terminated.

Devuelve

true if the task has finished; false otherwise.

6.54.2.2. `run()`

```
void es.ull.esit.app.metaheuristics.generators.InstanceGA.run ( )
```

Creates a GeneticAlgorithm generator using FactoryGenerator and replaces the existing GA entry in MultiGenerator, if found.

6.54.2.3. `setTerminate()`

```
void es.ull.esit.app.metaheuristics.generators.InstanceGA.setTerminate (
    boolean terminate )
```

Sets the termination flag. Intended mainly for testing.

Parámetros

<i>terminate</i>	true if the task should be marked as finished; false otherwise.
------------------	---

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceGA.java`

6.55. Referencia de la clase `es.ull.esit.app.metaheuristics.generators.LimitRoulette`

Métodos públicos

- `Generator` `getGenerator` ()
- void `setGenerator` (`Generator` generator)
- float `getLimitHigh` ()
- void `setLimitHigh` (float limitHigh)
- float `getLimitLow` ()
- void `setLimitLow` (float limitLow)

6.55.1. Descripción detallada

Class to define a limit roulette for a generator.

6.55.2. Documentación de funciones miembro

6.55.2.1. `getGenerator()`

```
Generator es.ull.esit.app.metaheuristics.generators.LimitRoulette.getGenerator ( )
```

Gets the generator associated to the limit roulette.

Devuelve

[Generator] Generator associated to the limit roulette.

6.55.2.2. `getLimitHigh()`

```
float es.ull.esit.app.metaheuristics.generators.LimitRoulette.getLimitHigh ( )
```

Gets the high limit of the roulette.

Devuelve

[float] High limit of the roulette.

6.55.2.3. `getLimitLow()`

```
float es.ull.esit.app.metaheuristics.generators.LimitRoulette.getLimitLow ( )
```

Gets the low limit of the roulette.

Devuelve

[float] Low limit of the roulette.

6.55.2.4. `setGenerator()`

```
void es.ull.esit.app.metaheuristics.generators.LimitRoulette.setGenerator (
    Generator generator )
```

Sets the generator associated to the limit roulette.

Parámetros

<i>generator</i>	[Generator] Generator to be associated to the limit roulette.
------------------	---

6.55.2.5. `setLimitHigh()`

```
void es.ull.esit.app.metaheuristics.generators.LimitRoulette.setLimitHigh (
    float limitHigh )
```

Sets the high limit of the roulette.

Parámetros

<i>limitHigh</i>	[float] High limit of the roulette.
------------------	-------------------------------------

6.55.2.6. `setLimitLow()`

```
void es.ull.esit.app.metaheuristics.generators.LimitRoulette.setLimitLow (
    float limitLow )
```

Sets the low limit of the roulette.

Parámetros

<i>limitLow</i>	[float] Low limit of the roulette.
-----------------	------------------------------------

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/LimitRoulette.java`

6.56. Referencia de la clase `es.ull.esit.app.metaheuristics.generators.LimitThreshold`

Diagrama de herencia de `es.ull.esit.app.metaheuristics.generators.LimitThreshold`

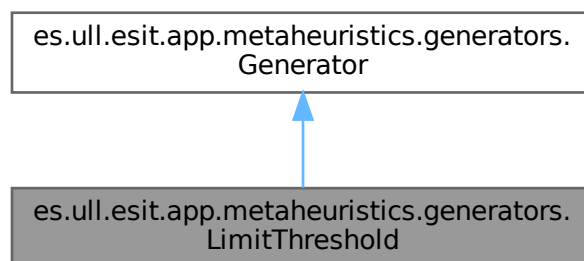
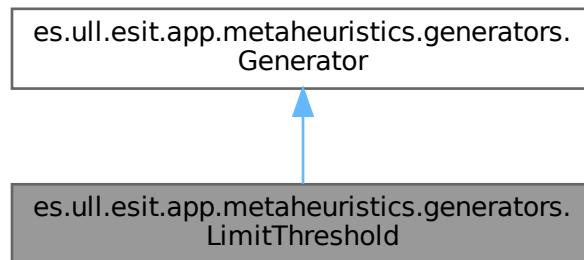


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.LimitThreshold`:



Métodos públicos

- `LimitThreshold ()`
- State `generate` (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- void `updateReference` (State stateCandidate, Integer countIterationsCurrent) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- State `getReference ()`
- void `setStateRef` (State stateRef)
- void `setInitialReference` (State stateInitialRef)
- `GeneratorType` `getGeneratorType ()`
- void `setGeneratorType` (`GeneratorType` generatorType)
- `GeneratorType` `getType ()`
- List< State > `getReferenceList ()`
- List< State > `getSonList ()`
- void `setTypeCandidate` (CandidateType typeCandidate)
- boolean `awardUpdateREF` (State stateCandidate)
- float `getWeight ()`
- void `setWeight` (float weight)
- int[] `getListCountBetterGender ()`
- int[] `getListCountGender ()`
- float[] `getTrace ()`

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.56.1. Descripción detallada

Limit Threshold local-search generator.

It generates a candidate in the neighbourhood of a reference state and decides whether to update that reference according to an acceptance rule.

6.56.2. Documentación de constructores y destructores

6.56.2.1. `LimitThreshold()`

```
es.ull.esit.app.metaheuristics.generators.LimitThreshold.LimitThreshold ( )
```

Default constructor.

It initialises the candidate comparison mode from the problem type and sets reasonable defaults for strategy, acceptance rule and weight.

6.56.3. Documentación de funciones miembro

6.56.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.LimitThreshold.awardUpdateREF (
    State stateCandidate )
```

Decides whether the candidate should be considered an "improvement" with respect to the current reference, based purely on evaluations.

This is independent from the acceptance rule used in `updateReference`.

Parámetros

<i>stateCandidate</i>	[State] Candidate state to compare.
-----------------------	-------------------------------------

Devuelve

[boolean] true if the candidate is better than the reference.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.2. `generate()`

```
State es.ull.esit.app.metaheuristics.generators.LimitThreshold.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    Exception, NoSuchMethodException
```

Generates a new candidate in the neighbourhood of the current reference using the configured candidate selection strategy.

Parámetros

<i>operatornumber</i>	[Integer] Operator to use for generating new states.
-----------------------	--

Devuelve

[State] Generated candidate state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.3. getGeneratorType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.LimitThreshold.getGeneratorType ( )
```

Returns the internal generator type.

Devuelve

[GeneratorType] Internal generator type.

6.56.3.4. getListCountBetterGender()

```
int[] es.ull.esit.app.metaheuristics.generators.LimitThreshold.getListCountBetterGender ( )
```

Returns the "better gender" counter array for this generator.

Devuelve

[int[]] Better gender counter array.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.5. getListCountGender()

```
int[] es.ull.esit.app.metaheuristics.generators.LimitThreshold.getListCountGender ( )
```

Returns the gender counter array for this generator.

Devuelve

[int[]] Gender counter array.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.6. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.LimitThreshold.getReference ( )
```

Returns the current reference state.

Devuelve

[State] Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.7. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.LimitThreshold.getReferenceList ( )
```

Returns a copy of the reference list maintained by this generator.

Currently this generator maintains a simple list that can be used for logging; each call appends the current reference (if not null) and returns a defensive copy.

Devuelve

[List<State>] Copy of the reference list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.8. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.LimitThreshold.getSonList ( )
```

This local-search generator does not keep a separate son list.

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.9. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.LimitThreshold.getTrace ( )
```

Returns the trace array used to store the weight evolution.

Devuelve

[float[]] Trace array.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.10. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.LimitThreshold.getType ( )
```

Returns the type of this generator.

Devuelve

[GeneratorType] Type of this generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.11. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.LimitThreshold.getWeight ( )
```

Returns the current generator weight.

Devuelve

[float] Current generator weight.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.12. `setGeneratorType()`

```
void es.ull.esit.app.metaheuristics.generators.LimitThreshold.setGeneratorType (
    GeneratorType generatorType )
```

Sets the internal generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] Internal generator type to set.
----------------------	---

6.56.3.13. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.LimitThreshold.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state.

Parámetros

<i>stateInitialRef</i>	[State] Initial reference state.
------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.14. `setStateRef()`

```
void es.ull.esit.app.metaheuristics.generators.LimitThreshold.setStateRef (
    State stateRef )
```

Sets the reference state (alias for `setInitialReference` for tests/clients).

Parámetros

<i>stateRef</i>	[State] Reference state to set.
-----------------	---------------------------------

6.56.3.15. `setTypeCandidate()`

```
void es.ull.esit.app.metaheuristics.generators.LimitThreshold.setTypeCandidate (
    CandidateType typeCandidate )
```

Allows changing the candidate comparison mode (greater/smaller).

Parámetros

<i>typeCandidate</i>	[CandidateType] Candidate comparison mode.
----------------------	--

6.56.3.16. `setWeight()`

```
void es.ull.esit.app.metaheuristics.generators.LimitThreshold.setWeight (
    float weight )
```

Sets the current generator weight.

Parámetros

<i>weight</i>	[float] Current generator weight to set.
---------------	--

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.56.3.17. `updateReference()`

```
void es.ull.esit.app.metaheuristics.generators.LimitThreshold.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Updates the reference state according to the configured acceptance rule.

Parámetros

<i>stateCandidate</i>	[State] Candidate state to consider for reference update.
<i>countIterationsCurrent</i>	[Integer] Current iteration count (of the metaheuristic).

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/metaheuristics/generators/LimitThreshold.java](#)

6.57. Referencia de la clase

es.ull.esit.app.problem.extension.MetricasMultiobjetivo

Métodos públicos

- double [calcularTasaError](#) (List< State > solutionsFPcurrent, List< State > solutionsFPtrue) throws BiffException, IOException
- double [calcularDistanciaGeneracional](#) (List< State > solutionsFPcurrent, List< State > solutionsFPtrue) throws BiffException, IOException
- double [calcularDispersion](#) (List< State > solutions) throws BiffException, IOException
- double [calcularMin](#) (List< Double > allMetrics)
- double [calcularMax](#) (List< Double > allMetrics)
- double [calcularMedia](#) (List< Double > allMetrics)

6.57.1. Descripción detallada

Class that implements multi-objective metrics.

6.57.2. Documentación de funciones miembro

6.57.2.1. [calcularDispersion\(\)](#)

```
double es.ull.esit.app.problem.extension.MetricasMultiobjetivo.calcularDispersion (
    List< State > solutions ) throws BiffException, IOException
```

Calculates the dispersion of solutions in the Pareto front.

Parámetros

<i>solutions</i>	[List<State>] List of solutions.
------------------	----------------------------------

Devuelve

[double] Dispersion value.

Excepciones

<i>BiffException</i>	If an error occurs while reading the Excel file.
<i>IOException</i>	If an I/O error occurs.

6.57.2.2. `calcularDistanciaGeneracional()`

```
double es.ull.esit.app.problem.extension.MetricasMultiobjetivo.calcularDistanciaGeneracional (
    List< State > solutionsFPcurrent,
    List< State > solutionsFPtrue ) throws BiffException, IOException
```

Calculates the generational distance between current and true Pareto front solutions.

Parámetros

<i>solutionsFPcurrent</i>	[List<State>] Current Pareto front solutions.
<i>solutionsFPtrue</i>	[List<State>] True Pareto front solutions.

Devuelve

[double] Generational distance.

Excepciones

<i>BiffException</i>	If an error occurs while reading the Excel file.
<i>IOException</i>	If an I/O error occurs.

6.57.2.3. `calcularMax()`

```
double es.ull.esit.app.problem.extension.MetricasMultiobjetivo.calcularMax (
    List< Double > allMetrics )
```

Calculates the maximum value from a list of metrics.

Parámetros

<i>allMetrics</i>	[List<Double>] List of metric values.
-------------------	---------------------------------------

Devuelve

[double] Maximum value.

6.57.2.4. `calcularMedia()`

```
double es.ull.esit.app.problem.extension.MetricasMultiobjetivo.calcularMedia (
    List< Double > allMetrics )
```

Calculates the average value from a list of metrics.

Parámetros

<i>allMetrics</i>	[List<Double>] List of metric values.
-------------------	---------------------------------------

Devuelve

[double] Average value.

6.57.2.5. calcularMin()

```
double es.ull.esit.app.problem.extension.MetricasMultiobjetivo.calcularMin (
    List< Double > allMetrics )
```

Calculates the minimum value from a list of metrics.

Parámetros

<i>allMetrics</i>	[List<Double>] List of metric values.
-------------------	---------------------------------------

Devuelve

[double] Minimum value.

6.57.2.6. calcularTasaError()

```
double es.ull.esit.app.problem.extension.MetricasMultiobjetivo.calcularTasaError (
    List< State > solutionsFPcurrent,
    List< State > solutionsFPtrue ) throws BiffException, IOException
```

Calculates the error rate between current and true Pareto front solutions.

Parámetros

<i>solutionsFPcurrent</i>	[List<State>] Current Pareto front solutions.
<i>solutionsFPtrue</i>	[List<State>] True Pareto front solutions.

Devuelve

[double] Error rate.

Excepciones

<i>BiffException</i>	If an error occurs while reading the Excel file.
<i>IOException</i>	If an I/O error occurs.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/extension/MetricasMultiobjetivo.java](#)

6.58. Referencia de la clase es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing ↩

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing

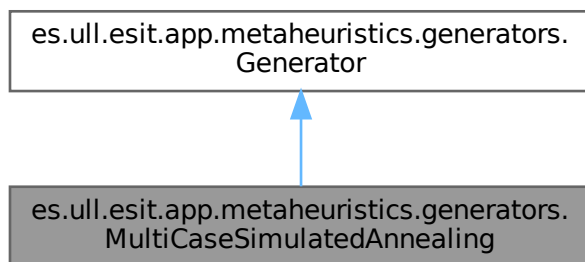
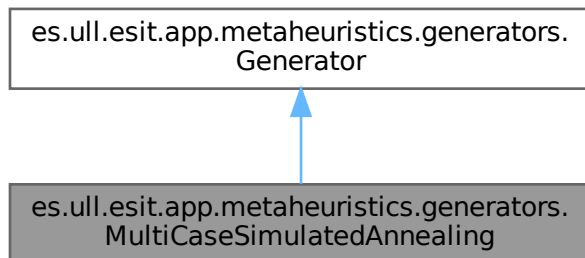


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing:



Métodos públicos

- [GeneratorType](#) [getTypeGenerator](#) ()
- void [setTypeGenerator](#) ([GeneratorType](#) typeGenerator)
- [MultiCaseSimulatedAnnealing](#) ()
- State [generate](#) (Integer operatornumber) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)
- State [getReference](#) ()
- void [setStateRef](#) (State stateRef)
- void [setInitialReference](#) (State stateInitialRef)
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)

- [GeneratorType](#) [getType](#) ()
- List< State > [getReferenceList](#) ()
- List< State > [getSonList](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()

Métodos públicos estáticos

- static double [tfinal](#) ()
- static void [setTfinal](#) (double finalTemperature)
- static double [getTinitial](#) ()
- static void [setTinitial](#) (double initialTemperature)

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.58.1. Descripción detallada

Generator that implements the Multi-case Simulated Annealing (MC-SA) algorithm.

6.58.2. Documentación de constructores y destructores

6.58.2.1. [MultiCaseSimulatedAnnealing\(\)](#)

```
es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.MultiCaseSimulated↵
Annealing ( )
```

Default constructor. It initialises the internal configuration for Multi-case Simulated Annealing and basic statistics arrays.

6.58.3. Documentación de funciones miembro

6.58.3.1. [awardUpdateREF\(\)](#)

```
boolean es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.awardUpdateREF (
    State stateCandidate )
```

Decides if the given candidate should be considered as an improvement over the current reference state when updating statistics.

Parámetros

<code>stateCandidate</code>	[State] Candidate state.
-----------------------------	--------------------------

Devuelve

[boolean] true if the candidate should be considered an improvement, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new candidate state using the current reference state and the neighbourhood defined by the given operator.

Parámetros

<code>operatornumber</code>	[Integer] the operator index.
-----------------------------	-------------------------------

Devuelve

[State] the generated candidate state.

Excepciones

<i>IllegalArgumentException</i>	if an illegal argument is provided.
<i>SecurityException</i>	if a security violation occurs.
<i>ClassNotFoundException</i>	if a required class cannot be found.
<i>InstantiationException</i>	if a class cannot be instantiated.
<i>IllegalAccessException</i>	if there is an illegal access.
<i>InvocationTargetException</i>	if a reflective invocation fails.
<i>NoSuchMethodException</i>	if a required method cannot be found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.3. getListCountBetterGender()

```
int[] es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getListCountBetterGender ( )
```

Returns the internal array with the number of times this generator has produced a better gender (for dynamic problems).

Devuelve

[int[]] Counts of better genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.4. getListCountGender()

```
int[] es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getListCountGender  
( )
```

Returns the internal array with the number of times this generator has produced genders (for dynamic problems).

Devuelve

[int[]] Counts of genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.5. getReference()

```
State es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getReference ( )
```

Returns the current reference state.

Devuelve

[State] The current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.6. getReferenceList()

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.get↔  
ReferenceList ( )
```

Returns the list of reference states visited so far. Each call stores a copy of the current reference state in the internal list and then returns it.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.7. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getSonList
( )
```

Multi-case Simulated Annealing does not maintain an explicit son list.

Devuelve

[List<State>] An empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.8. `getTinitial()`

```
static double es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.get←
Tinitial ( ) [static]
```

Gets the initial temperature used in the simulated annealing process.

Devuelve

[double] The initial temperature.

6.58.3.9. `getTrace()`

```
float[ ] es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getTrace ( )
```

Returns the trace of weight changes over time.

Devuelve

[float[]] Trace values.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.10. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getType (
)
```

Gets the generator type.

Devuelve

[GeneratorType] The generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.11. `getTypeGenerator()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getTypeGenerator ( )
```

Gets the type of generator.

Devuelve

[GeneratorType] The type of generator.

6.58.3.12. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.getWeight ( )
```

Gets the weight of this generator.

Devuelve

[float] Weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.13. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state.
------------------------	--------------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.14. `setStateRef()`

```
void es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.setStateRef (
    State stateRef )
```

Sets the current reference state.

Parámetros

<i>stateRef</i>	[State] The new reference state.
-----------------	----------------------------------

6.58.3.15. `setTfinal()`

```
static void es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.setTfinal (
    double finalTemperature ) [static]
```

Sets the final temperature used in the simulated annealing process.

Parámetros

<i>finalTemperature</i>	[double] The final temperature to set.
-------------------------	--

6.58.3.16. `setTinitial()`

```
static void es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.setTinitial
(
    double initialTemperature ) [static]
```

Sets the initial temperature used in the simulated annealing process.

Parámetros

<i>initialTemperature</i>	[double] The initial temperature to set.
---------------------------	--

6.58.3.17. `setTypeGenerator()`

```
void es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.setTypeGenerator (
    GeneratorType typeGenerator )
```

Sets the type of generator.

Parámetros

<i>typeGenerator</i>	[GeneratorType] The type of generator.
----------------------	--

6.58.3.18. `setWeight()`

```
void es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.setWeight (
    float weight )
```

Sets the weight of this generator.

Parámetros

<i>weight</i>	[float] New weight.
---------------	---------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.58.3.19. tfinal()

```
static double es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.tfinal ( )
[static]
```

Gets the final temperature used in the simulated annealing process.

Devuelve

[double] The final temperature.

6.58.3.20. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Updates the reference state according to the Multi-case SA acceptance rule and updates the temperature when required.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
<i>countIterationsCurrent</i>	[Integer] Current iteration count.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a required class cannot be found.
<i>InstantiationException</i>	If a class cannot be instantiated.
<i>IllegalAccessException</i>	If there is an illegal access.
<i>InvocationTargetException</i>	If a reflective invocation fails.
<i>NoSuchMethodException</i>	If a required method cannot be found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/metaheuristics/generators/MultiCaseSimulatedAnnealing.java](#)

6.59. Referencia de la clase es.ull.esit.app.metaheuristics.generators.MultiGenerator

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.MultiGenerator

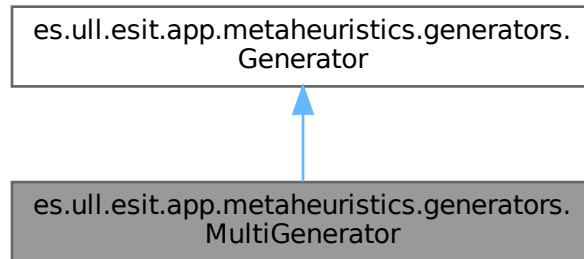
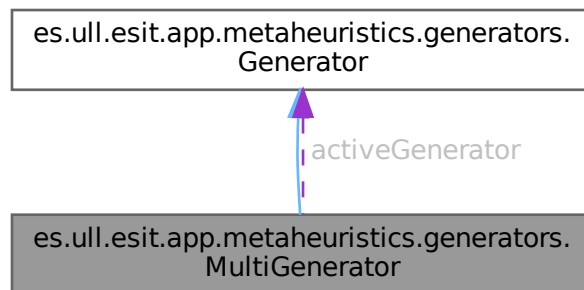


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.MultiGenerator:



Métodos públicos

- void `setGeneratortype` (`GeneratorType` generatortype)
- `MultiGenerator` ()
- State `generate` (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- State `getReference` ()
- List< State > `getReferenceList` ()
- List< State > `getSonList` ()
- `GeneratorType` `getType` ()
- void `setInitialReference` (State stateInitialRef)

- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- void [updateWeight](#) (State stateCandidate)
- boolean [searchState](#) (State stateCandidate)
- float [getWeight](#) ()
- [Generator](#) [roulette](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- void [updateAwardSC](#) ()
- void [updateAwardImp](#) ()
- void [setWeight](#) (float weight)
- float[] [getTrace](#) ()
- void [tournament](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- [MultiGenerator](#) ([MultiGenerator](#) other)
- [MultiGenerator](#) [copy](#) ()
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()

Métodos públicos estáticos

- static void [destroyMultiGenerator](#) ()
- static void [initializeListGenerator](#) () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- static void [initializeGenerators](#) () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- static void [createInstanceGeneratorsBPP](#) ()
- static [Generator](#)[] [getListGenerators](#) ()
- static void [setListGenerators](#) ([Generator](#)[] listGenerators)
- static [Generator](#) [getActiveGenerator](#) ()
- static void [setActiveGenerator](#) ([Generator](#) activeGenerator)
- static void [setListGeneratedPP](#) (List< State > [listGeneratedPP](#))

Atributos públicos estáticos

- static List< State > [listGeneratedPP](#) = new ArrayList<State> ()
- static [Generator](#) [activeGenerator](#)
- static List< State > [listStateReference](#) = new ArrayList<State>()

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.59.1. Documentación de constructores y destructores

6.59.1.1. `MultiGenerator()` [1/2]

```
es.ull.esit.app.metaheuristics.generators.MultiGenerator.MultiGenerator ( )
```

6.59.1.2. `MultiGenerator()` [2/2]

```
es.ull.esit.app.metaheuristics.generators.MultiGenerator.MultiGenerator (
    MultiGenerator other )
```

6.59.2. Documentación de funciones miembro

6.59.2.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.MultiGenerator.awardUpdateREF (
    State stateCandidate )
```

Abstract method to decide whether to award an update to the reference state.

Parámetros

<code>stateCandidate</code>	[State] The candidate state.
-----------------------------	------------------------------

Devuelve

[boolean] True if the update is awarded, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.2. `copy()`

```
MultiGenerator es.ull.esit.app.metaheuristics.generators.MultiGenerator.copy ( )
```

6.59.2.3. `createInstanceGeneratorsBPP()`

```
static void es.ull.esit.app.metaheuristics.generators.MultiGenerator.createInstanceGenerators↵
BPP ( ) [static]
```

6.59.2.4. `destroyMultiGenerator()`

```
static void es.ull.esit.app.metaheuristics.generators.MultiGenerator.destroyMultiGenerator ( )
[static]
```

6.59.2.5. generate()

State `es.ull.esit.app.metaheuristics.generators.MultiGenerator.generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

Abstract method to generate a new state.

Parámetros

<i>operatornumber</i>	[Integer] Number of the operator to be used.
-----------------------	--

Excepciones

<i>NoSuchMethodException</i>	If the method does not exist.
<i>InvocationTargetException</i>	If the method cannot be invoked.
<i>IllegalAccessException</i>	If there is no access to the method.
<i>InstantiationException</i>	If there is an error during the instantiation.
<i>ClassNotFoundException</i>	If the class is not found.
<i>SecurityException</i>	If a security violation occurs.
<i>IllegalArgumentException</i>	If an illegal argument is provided.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.6. getActiveGenerator()

static [Generator](#) `es.ull.esit.app.metaheuristics.generators.MultiGenerator.getActiveGenerator ()` [static]

6.59.2.7. getListCountBetterGender()

`int[] es.ull.esit.app.metaheuristics.generators.MultiGenerator.getListCountBetterGender ()`

List of counts for better gender statistics.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.8. getListCountGender()

`int[] es.ull.esit.app.metaheuristics.generators.MultiGenerator.getListCountGender ()`

List of counts for gender statistics.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.9. `getListGenerators()`

```
static Generator[] es.ull.esit.app.metaheuristics.generators.MultiGenerator.getListGenerators  
( ) [static]
```

6.59.2.10. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.MultiGenerator.getReference ( )
```

Abstract method to get the reference state.

Devuelve

[State] The reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.11. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiGenerator.getReferenceList ( )
```

Abstract method to get the list of reference states.

Devuelve

[List<State>] The list of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.12. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiGenerator.getSonList ( )
```

Abstract method to get the list of son states.

Devuelve

[List<State>] The list of son states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.13. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.MultiGenerator.getTrace ( )
```

Abstract method to get the trace of the generator.

Devuelve

[float[]] The trace of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.14. getType()

`GeneratorType` `es.ull.esit.app.metaheuristics.generators.MultiGenerator.getType ()`

Abstract method to get the type of generator.

Devuelve

[`GeneratorType`] The type of generator.

Reimplementado de `es.ull.esit.app.metaheuristics.generators.Generator`.

6.59.2.15. getWeight()

`float` `es.ull.esit.app.metaheuristics.generators.MultiGenerator.getWeight ()`

Abstract method to get the weight of the generator.

Devuelve

[`float`] The weight of the generator.

Reimplementado de `es.ull.esit.app.metaheuristics.generators.Generator`.

6.59.2.16. initializeGenerators()

```
static void es.ull.esit.app.metaheuristics.generators.MultiGenerator.initializeGenerators (
) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException,
IllegalAccessException, InvocationTargetException, NoSuchMethodException [static]
```

6.59.2.17. initializeListGenerator()

```
static void es.ull.esit.app.metaheuristics.generators.MultiGenerator.initializeListGenerator (
) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException,
IllegalAccessException, InvocationTargetException, NoSuchMethodException [static]
```

6.59.2.18. roulette()

`Generator` `es.ull.esit.app.metaheuristics.generators.MultiGenerator.roulette ()`

6.59.2.19. searchState()

```
boolean es.ull.esit.app.metaheuristics.generators.MultiGenerator.searchState (
    State stateCandidate )
```


6.59.2.20. setActiveGenerator()

```
static void es.ull.esit.app.metaheuristics.generators.MultiGenerator.setActiveGenerator (
    Generator activeGenerator ) [static]
```

6.59.2.21. setGeneratortype()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.setGeneratortype (
    GeneratorType generatortype )
```

6.59.2.22. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.setInitialReference (
    State stateInitialRef )
```

Abstract method to set the initial reference state.

Parámetros

<code>stateInitialRef</code>	[State] The initial reference state.
------------------------------	--------------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.23. setListGeneratedPP()

```
static void es.ull.esit.app.metaheuristics.generators.MultiGenerator.setListGeneratedPP (
    List< State > listGeneratedPP ) [static]
```

6.59.2.24. setListGenerators()

```
static void es.ull.esit.app.metaheuristics.generators.MultiGenerator.setListGenerators (
    Generator[] listGenerators ) [static]
```

6.59.2.25. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.setWeight (
    float weight )
```

Abstract method to decide whether to award an update to the son state.

Parámetros

<code>stateCandidate</code>	[State] The candidate state.
-----------------------------	------------------------------

Devuelve

[boolean] True if the update is awarded, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.26. tournament()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.tournament (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, Security↔
Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocation↔
TargetException, NoSuchMethodException
```

6.59.2.27. updateAwardImp()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.updateAwardImp ( )
```

6.59.2.28. updateAwardSC()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.updateAwardSC ( )
```

6.59.2.29. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, Security↔
Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocation↔
TargetException, NoSuchMethodException
```

Abstract method to update the reference state.

Parámetros

<i>operatornumber</i>	[Integer] Number of the operator to be used.
-----------------------	--

Excepciones

<i>NoSuchMethodException</i>	If the method does not exist.
<i>InvocationTargetException</i>	If the method cannot be invoked.
<i>IllegalAccessException</i>	If there is no access to the method.
<i>InstantiationException</i>	If there is an error during the instantiation.
<i>ClassNotFoundException</i>	If the class is not found.
<i>SecurityException</i>	If a security violation occurs.
<i>IllegalArgumentException</i>	If an illegal argument is provided.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.59.2.30. updateWeight()

```
void es.ull.esit.app.metaheuristics.generators.MultiGenerator.updateWeight (
    State stateCandidate )
```

6.59.3. Documentación de datos miembro

6.59.3.1. activeGenerator

```
Generator es.ull.esit.app.metaheuristics.generators.MultiGenerator.activeGenerator [static]
```

6.59.3.2. listGeneratedPP

```
List<State> es.ull.esit.app.metaheuristics.generators.MultiGenerator.listGeneratedPP = new
ArrayList<State> () [static]
```

6.59.3.3. listStateReference

```
List<State> es.ull.esit.app.metaheuristics.generators.MultiGenerator.listStateReference = new
ArrayList<State> () [static]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/metaheuristics/generators/MultiGenerator.java](#)

6.60. Referencia de la clase es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance ↩

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance

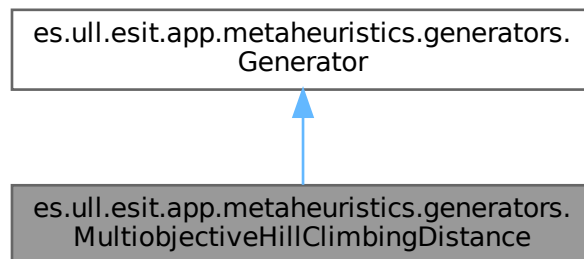
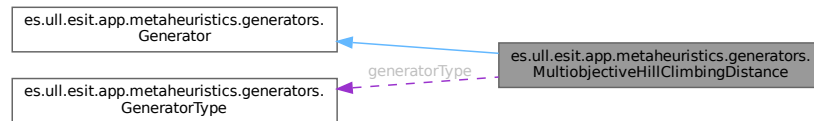


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance`:



Métodos públicos

- `MultiobjectiveHillClimbingDistance` ()
- State `generate` (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- void `updateReference` (State stateCandidate, Integer countIterationsCurrent) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- List< State > `getReferenceList` ()
- State `getReference` ()
- void `setStateRef` (State stateRef)
- void `setInitialReference` (State stateInitialRef)
- `GeneratorType` `getGeneratorType` ()
- void `setGeneratorType` (`GeneratorType` generatorType)
- `GeneratorType` `getType` ()
- List< State > `getSonList` ()
- boolean `awardUpdateREF` (State stateCandidate)
- float `getWeight` ()
- void `setWeight` (float weight)
- int[] `getListCountBetterGender` ()
- int[] `getListCountGender` ()
- float[] `getTrace` ()

Métodos públicos estáticos

- static List< Double > `distanceCalculateAdd` (List< State > solution)

Atributos públicos estáticos

- static int `sizeNeighbors`
- static List< Double > `distanceSolution` = new ArrayList<Double>()

Atributos protegidos

- CandidateValue `candidatevalue`
- AcceptType `typeAcceptation`
- StrategyType `strategy`
- CandidateType `typeCandidate`
- State `stateReferenceHC`
- IFFactoryAcceptCandidate `ifacceptCandidate`
- `GeneratorType` `generatorType`
- List< State > `listStateReference` = new ArrayList<State>()
- float `weight`
- List< Float > `listTrace` = new ArrayList<Float>()

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- `int` `countGender`
- `int` `countBetterGender`
- `int[]` `listCountBetterGender`

6.60.1. Documentación de constructores y destructores

6.60.1.1. `MultiobjectiveHillClimbingDistance()`

```
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.MultiobjectiveHillClimbingDistance ( )
```

6.60.2. Documentación de funciones miembro

6.60.2.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.awardUpdateREF (
    State stateCandidate )
```

Abstract method to decide whether to award an update to the reference state.

Parámetros

<code>stateCandidate</code>	[State] The candidate state.
-----------------------------	------------------------------

Devuelve

[boolean] True if the update is awarded, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.2. `distanceCalculateAdd()`

```
static List< Double > es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.distanceCalculateAdd (
    List< State > solution ) [static]
```

6.60.2.3. `generate()`

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Abstract method to generate a new state.

Parámetros

<i>operatornumber</i>	[Integer] Number of the operator to be used.
-----------------------	--

Excepciones

<i>NoSuchMethodException</i>	If the method does not exist.
<i>InvocationTargetException</i>	If the method cannot be invoked.
<i>IllegalAccessException</i>	If there is no access to the method.
<i>InstantiationException</i>	If there is an error during the instantiation.
<i>ClassNotFoundException</i>	If the class is not found.
<i>SecurityException</i>	If a security violation occurs.
<i>IllegalArgumentException</i>	If an illegal argument is provided.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.4. getGeneratorType()

[GeneratorType](#) es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.
getGeneratorType ()

6.60.2.5. getListCountBetterGender()

int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.getListCountBetterGender ()

List of counts for better gender statistics.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.6. getListCountGender()

int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.getListCountGender ()

List of counts for gender statistics.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.7. getReference()

State es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.getReference ()

Abstract method to get the reference state.

Devuelve

[State] The reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.8. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
getReferenceList ( )
```

Abstract method to get the list of reference states.

Devuelve

[List<State>] The list of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.9. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
getSonList ( )
```

Abstract method to get the list of son states.

Devuelve

[List<State>] The list of son states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.10. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.  
( )
```

Abstract method to get the trace of the generator.

Devuelve

[float[]] The trace of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.11. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
getType ( )
```

Abstract method to get the type of generator.

Devuelve

[GeneratorType] The type of generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.12. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.getWeight (
)
```

Abstract method to get the weight of the generator.

Devuelve

[float] The weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.13. `setGeneratorType()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.setGenerator↵
Type (
    GeneratorType generatorType )
```

6.60.2.14. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.setInitial↵
Reference (
    State stateInitialRef )
```

Abstract method to set the initial reference state.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state.
------------------------	--------------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.15. `setStateRef()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.setStateRef
(
    State stateRef )
```

6.60.2.16. `setWeight()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.setWeight (
    float weight )
```

Abstract method to decide whether to award an update to the son state.

Parámetros

<code>stateCandidate</code>	[State] The candidate state.
-----------------------------	------------------------------

Devuelve

[boolean] True if the update is awarded, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.2.17. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Abstract method to update the reference state.

Parámetros

<code>operatornumber</code>	[Integer] Number of the operator to be used.
-----------------------------	--

Excepciones

<code>NoSuchMethodException</code>	If the method does not exist.
<code>InvocationTargetException</code>	If the method cannot be invoked.
<code>IllegalAccessException</code>	If there is no access to the method.
<code>InstantiationException</code>	If there is an error during the instantiation.
<code>ClassNotFoundException</code>	If the class is not found.
<code>SecurityException</code>	If a security violation occurs.
<code>IllegalArgumentException</code>	If an illegal argument is provided.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.60.3. Documentación de datos miembro**6.60.3.1. candidatevalue**

```
CandidateValue es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.candidatevalue [protected]
```

6.60.3.2. distanceSolution

```
List<Double> es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.distanceSolution = new ArrayList<Double>() [static]
```

6.60.3.3. generatorType

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
generatorType [protected]
```

6.60.3.4. ifacceptCandidate

```
IFFactoryAcceptCandidate es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbing↔  
Distance.ifacceptCandidate [protected]
```

6.60.3.5. listStateReference

```
List<State> es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
listStateReference = new ArrayList<State>() [protected]
```

6.60.3.6. listTrace

```
List<Float> es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
listTrace = new ArrayList<Float>() [protected]
```

6.60.3.7. sizeNeighbors

```
int es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.sizeNeighbors  
[static]
```

6.60.3.8. stateReferenceHC

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.state↔  
ReferenceHC [protected]
```

6.60.3.9. strategy

```
StrategyType es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
strategy [protected]
```

6.60.3.10. typeAcceptation

```
AcceptType es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.type↔  
Acceptation [protected]
```

6.60.3.11. typeCandidate

```
CandidateType es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.↔  
typeCandidate [protected]
```

6.60.3.12. weight

```
float es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance.weight  
[protected]
```

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingDistance.java](#)

6.61. Referencia de la clase es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart ↩

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart

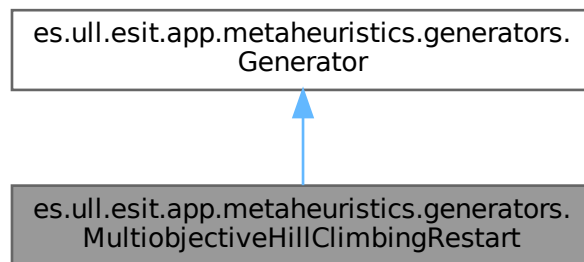
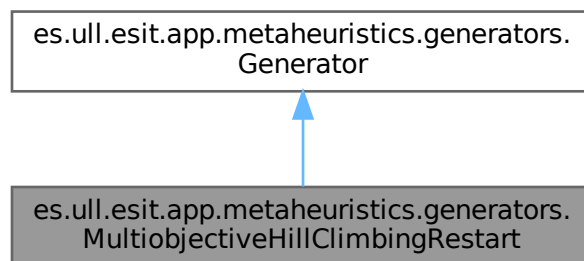


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart:



Métodos públicos

- [MultiobjectiveHillClimbingRestart](#) ()
- State [generate](#) (Integer operatornumber) throws IllegalArgumentException, SecurityException, Class↵NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuch↵MethodException
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException↵Exception, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- List< State > [getReferenceList](#) ()
- State [getReference](#) ()
- void [setStateRef](#) (State stateRef)
- void [setInitialReference](#) (State stateInitialRef)
- [GeneratorType](#) [getGeneratorType](#) ()
- void [setGeneratorType](#) ([GeneratorType](#) generatorType)
- [GeneratorType](#) [getType](#) ()
- List< State > [getSonList](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()

Métodos públicos estáticos

- static int [getSizeNeighbors](#) ()
- static void [setSizeNeighbors](#) (int sizeNeighbors)

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.61.1. Descripción detallada

Generator class that implements a multi-objective Hill Climbing with restart.

6.61.2. Documentación de constructores y destructores

6.61.2.1. [MultiobjectiveHillClimbingRestart\(\)](#)

```
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.Multiobjective↵
HillClimbingRestart ( )
```

Default constructor. It initialises the internal configuration and statistics arrays for the multiobjective Hill Climbing with restart.

6.61.3. Documentación de funciones miembro

6.61.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.award↵
UpdateREF (
    State stateCandidate )
```

Decides if the candidate should be considered an improvement for statistics purposes. For multiobjective HC, we simply require the candidate to be different from the current reference.

Parámetros

<code>stateCandidate</code>	[State] Candidate state.
-----------------------------	--------------------------

Devuelve

[boolean] true if considered an improvement, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.2. `generate()`

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵
Exception, NoSuchMethodException
```

Generates a new candidate state from the current reference using the neighbourhood defined by the given operator.

Parámetros

<code>operatornumber</code>	[Integer] Operator index.
-----------------------------	---------------------------

Devuelve

[State] Generated candidate state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.3. `getGeneratorType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.↵
getGeneratorType ( )
```

Returns the generator type.

Devuelve

[GeneratorType] Generator type.

6.61.3.4. `getListCountBetterGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.getListCountBetterGender ( )
```

Returns the internal statistics array of “better gender” counts.

Devuelve

[int[]] Counts of better genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.5. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.getListCountGender ( )
```

Returns the internal statistics array of gender counts.

Devuelve

[int[]] Counts of genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.6. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.getReference ( )
```

Returns the current reference state.

Devuelve

[State] Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.7. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.getReferenceList ( )
```

Returns the list of reference states visited so far. Each call appends a copy of the current reference state.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.8. `getSizeNeighbors()`

```
static int es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.get↵
SizeNeighbors ( ) [static]
```

Returns the neighbourhood size used in the restart logic.

Devuelve

[int] Neighbourhood size.

6.61.3.9. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.↵
getSonList ( )
```

Multiobjective Hill Climbing with restart does not maintain an explicit son list.

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.10. `getTrace()`

```
float [ ] es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.getTrace (
)
```

Returns the trace of weight changes.

Devuelve

[float[]] Trace values.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.11. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.↵
getType ( )
```

Returns the generator type (required by abstract base class).

Devuelve

[GeneratorType] Generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.12. getWeight()

```
float es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.getWeight (
)
```

Returns the weight of this generator.

Devuelve

[float] Weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.13. setGeneratorType()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.setGenerator↵
Type (
    GeneratorType generatorType )
```

Sets the generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] Generator type.
----------------------	---------------------------------

6.61.3.14. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.setInitial↵
Reference (
    State stateInitialRef )
```

Sets the initial reference state.

Parámetros

<i>stateInitialRef</i>	[State] Initial reference state.
------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.15. setSizeNeighbors()

```
static void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.set↵
SizeNeighbors (
    int sizeNeighbors ) [static]
```

Sets the neighbourhood size used in the restart logic.

Parámetros

<i>sizeNeighbors</i>	[int] Neighbourhood size.
----------------------	---------------------------

6.61.3.16. `setStateRef()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.setStateRef (
    State stateRef )
```

Sets the current reference state.

Parámetros

<i>stateRef</i>	[State] New reference state.
-----------------	------------------------------

6.61.3.17. `setWeight()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.setWeight (
    float weight )
```

Sets the weight of this generator.

Parámetros

<i>weight</i>	[float] New weight.
---------------	---------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.61.3.18. `updateReference()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart.update↵
Reference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, Security↵
Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocation↵
TargetException, NoSuchMethodException
```

Updates the reference solution according to the multiobjective acceptance rule and restart logic.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
<i>countIterationsCurrent</i>	[Integer] Current iteration.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingRestart.java](#)

6.62. Referencia de la clase `es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing` ↩

Diagrama de herencia de `es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing`

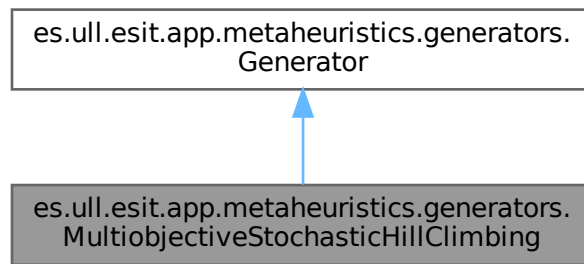
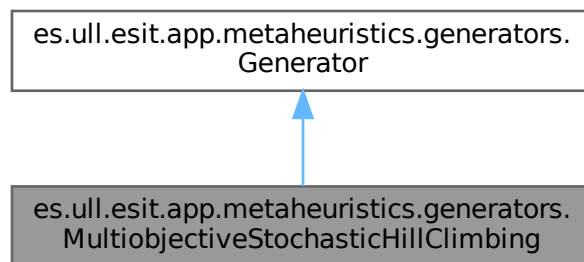


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing`:



Métodos públicos

- [MultiobjectiveStochasticHillClimbing](#) ()
- State [generate](#) (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- List< State > [getReferenceList](#) ()

- State `getReference ()`
- void `setStateRef (State stateRef)`
- void `setInitialReference (State stateInitialRef)`
- `GeneratorType` `getGeneratorType ()`
- void `setGeneratorType (GeneratorType generatorType)`
- `GeneratorType` `getType ()`
- List< State > `getSonList ()`
- boolean `awardUpdateREF (State stateCandidate)`
- float `getWeight ()`
- void `setWeight (float weight)`
- int[] `getListCountBetterGender ()`
- int[] `getListCountGender ()`
- float[] `getTrace ()`

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.62.1. Descripción detallada

Generator that implements a multi-objective Stochastic Hill Climbing.

6.62.2. Documentación de constructores y destructores

6.62.2.1. `MultiobjectiveStochasticHillClimbing()`

```
es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.Multiobjective↵
StochasticHillClimbing ( )
```

Default constructor. It initialises the internal configuration and statistics arrays for the multiobjective Stochastic Hill Climbing.

6.62.3. Documentación de funciones miembro

6.62.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.award↵
UpdateREF (
    State stateCandidate )
```

Decides if the candidate should be considered an improvement for statistics purposes. Here we use a simple comparison on the first objective.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
-----------------------	--------------------------

Devuelve

[boolean] true if considered an improvement, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.generate
(
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵
Exception, NoSuchMethodException
```

Generates a new candidate state from the current reference using the neighbourhood defined by the given operator.

Parámetros

<i>operatornumber</i>	[Integer] Operator index.
-----------------------	---------------------------

Devuelve

[State] Generated candidate state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.3. getGeneratorType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.↵
getGeneratorType ( )
```

Returns the generator type.

Devuelve

[GeneratorType] Generator type.

6.62.3.4. getListCountBetterGender()

```
int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getList↵
CountBetterGender ( )
```

Returns the internal statistics array of “better gender” counts.

Devuelve

[int[]] Counts of better genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.5. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getListCountGender ( )
```

Returns the internal statistics array of gender counts.

Devuelve

[int[]] Counts of genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.6. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getReference ( )
```

Returns the current reference state.

Devuelve

[State] Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.7. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getReferenceList ( )
```

Returns the list of reference states visited so far. Each call appends a copy of the current reference state.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.8. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getSonList ( )
```

Multiobjective Stochastic Hill Climbing does not maintain an explicit son list.

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.9. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getTrace ( )
```

Returns the trace of weight changes.

Devuelve

[float[]] Trace values.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.10. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getType ( )
```

Returns the generator type (required by abstract base class).

Devuelve

[GeneratorType] Generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.11. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.getWeight ( )
```

Returns the weight of this generator.

Devuelve

[float] Weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.12. `setGeneratorType()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.setGeneratorType ( GeneratorType generatorType )
```

Sets the generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] Generator type.
----------------------	---------------------------------

6.62.3.13. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.set↵
InitialReference (
    State stateInitialRef )
```

Sets the initial reference state.

Parámetros

<i>stateInitialRef</i>	[State] Initial reference state.
------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.14. setStateRef()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.setState↵
Ref (
    State stateRef )
```

Sets the current reference state.

Parámetros

<i>stateRef</i>	[State] New reference state.
-----------------	------------------------------

6.62.3.15. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.setWeight
(
    float weight )
```

Sets the weight of this generator.

Parámetros

<i>weight</i>	[float] New weight.
---------------	---------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.62.3.16. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing.update↵
```

```

Reference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, Security↵
Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocation↵
TargetException, NoSuchMethodException

```

Updates the reference solution according to the multiobjective acceptance rule.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
<i>countIterationsCurrent</i>	[Integer] Current iteration.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveStochasticHillClimbing.java](#)

6.63. Referencia de la clase [es.ull.esit.app.metaheuristics.generators.↵](#) MultiobjectiveTabuSearch

Diagrama de herencia de [es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch](#)

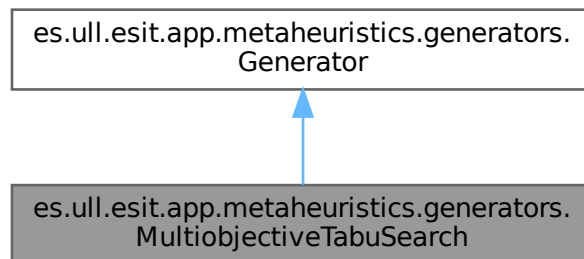
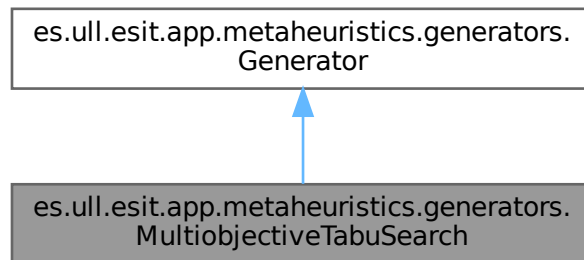


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch`:



Métodos públicos

- `MultiobjectiveTabuSearch ()`
- State `getStateReferenceTS ()`
- void `setStateReferenceTS (State stateReferenceTS)`
- `GeneratorType getTypeGenerator ()`
- void `setTypeGenerator (GeneratorType typeGenerator)`
- State `generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- void `updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `GeneratorType getType ()`
- List< State > `getReferenceList ()`
- State `getReference ()`
- void `setInitialReference (State stateInitialRef)`
- void `setStateRef (State stateRef)`
- List< State > `getSonList ()`
- void `setTypeCandidate (CandidateType typeCandidate)`
- boolean `awardUpdateREF (State stateCandidate)`
- float `getWeight ()`
- void `setWeight (float weight)`
- int[] `getListCountBetterGender ()`
- int[] `getListCountGender ()`
- float[] `getTrace ()`

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.63.1. Descripción detallada

Generator that implements a multi-objective Tabu Search algorithm.

6.63.2. Documentación de constructores y destructores

6.63.2.1. MultiobjectiveTabuSearch()

```
es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.MultiobjectiveTabuSearch (
)
```

Default constructor. It initialises the internal configuration and statistics arrays for the multiobjective Tabu Search.

6.63.3. Documentación de funciones miembro

6.63.3.1. awardUpdateREF()

```
boolean es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.awardUpdateREF (
    State stateCandidate )
```

Decides if the candidate should be considered an improvement for statistics purposes. For Tabu Search, we consider any different state as a potential update.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
-----------------------	--------------------------

Devuelve

[boolean] true if considered an improvement, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    Exception, NoSuchMethodException
```

Generates a new candidate state from the current reference using the neighbourhood defined by the given operator. Tabu logic is handled in the acceptance rule (AcceptNotDominatedTabu).

Parámetros

<i>operatornumber</i>	[Integer] Operator index.
-----------------------	---------------------------

Devuelve

[State] Generated candidate state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.3. `getListCountBetterGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getListCountBetterGender ( )
```

Returns the internal statistics array of “better gender” counts.

Devuelve

[int[]] Counts of better genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.4. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getListCountGender ( )
```

Returns the internal statistics array of gender counts.

Devuelve

[int[]] Counts of genders.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.5. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getReference ( )
```

Returns the current reference state.

Devuelve

[State] Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.6. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getReferenceList ( )
```

Returns the list of reference states visited so far. Each call appends the current reference state.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.7. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getSonList ( )
```

Multiobjective Tabu Search does not maintain an explicit son list.

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.8. `getStateReferenceTS()`

```
State es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getStateReferenceTS ( )
```

Returns the current reference state of the Tabu Search.

Devuelve

[State] Current reference state.

6.63.3.9. `getTrace()`

```
float [ ] es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getTrace ( )
```

Returns the trace of weight changes.

Devuelve

[float[]] Trace values.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.10. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getType ( )
```

Returns the generator type (required by abstract base class).

Devuelve

[GeneratorType] Generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.11. `getTypeGenerator()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getTypeGenerator ( )
```

Returns the internal generator type.

Devuelve

[GeneratorType] Generator type.

6.63.3.12. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.getWeight ( )
```

Returns the weight of this generator.

Devuelve

[float] Weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.13. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.setInitialReference ( State stateInitialRef )
```

Sets the initial reference state.

Parámetros

<code>stateInitialRef</code>	[State] Initial reference state.
------------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.14. setStateRef()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.setStateRef (
    State stateRef )
```

Sets the current reference state.

Parámetros

<i>stateRef</i>	[State] New reference state.
-----------------	------------------------------

6.63.3.15. setStateReferenceTS()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.setStateReferenceTS (
    State stateReferenceTS )
```

Sets the current reference state of the Tabu Search.

Parámetros

<i>stateReferenceTS</i>	[State] New reference state.
-------------------------	------------------------------

6.63.3.16. setTypeCandidate()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.setTypeCandidate (
    CandidateType typeCandidate )
```

Sets the candidate selection rule.

Parámetros

<i>typeCandidate</i>	[CandidateType] New candidate rule.
----------------------	-------------------------------------

6.63.3.17. setTypeGenerator()

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.setTypeGenerator (
    GeneratorType typeGenerator )
```

Sets the internal generator type.

Parámetros

<i>typeGenerator</i>	[GeneratorType] Generator type.
----------------------	---------------------------------

6.63.3.18. `setWeight()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.setWeight (
    float weight )
```

Sets the weight of this generator.

Parámetros

<i>weight</i>	[float] New weight.
---------------	---------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.63.3.19. `updateReference()`

```
void es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Updates the reference solution according to the multiobjective Tabu acceptance rule and maintains the tabu list.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
<i>countIterationsCurrent</i>	[Integer] Current iteration (unused here).

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveTabuSearch.java`

6.64. Referencia de la clase

es.ull.esit.app.problem.extension.MultiObjetivoPuro

Diagrama de herencia de es.ull.esit.app.problem.extension.MultiObjetivoPuro

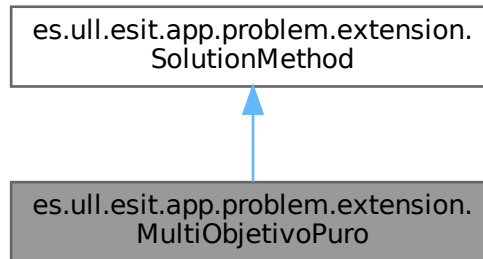
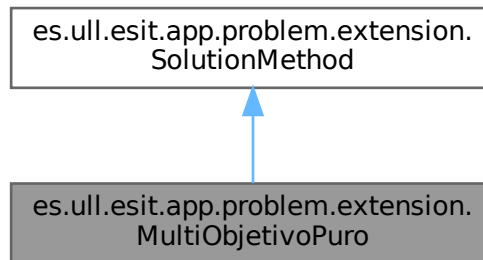


Diagrama de colaboración de es.ull.esit.app.problem.extension.MultiObjetivoPuro:



Métodos públicos

- void [evaluationState](#) (State state)

6.64.1. Descripción detallada

Class that implements the pure multi-objective solution method.

6.64.2. Documentación de funciones miembro

6.64.2.1. evaluationState()

```
void es.ull.esit.app.problem.extension.MultiObjetivoPuro.evaluationState (
    State state )
```

Constructor.

Parámetros

<i>name</i>	[String] Name of the solution method.
-------------	---------------------------------------

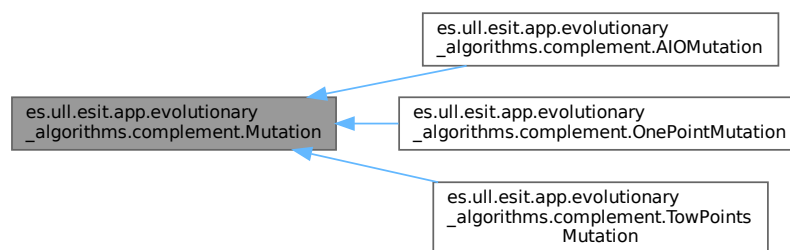
Reimplementado de [es.ull.esit.app.problem.extension.SolutionMethod](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/extension/MultiObjetivoPuro.java](#)

6.65. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.Mutation`

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.Mutation`



Métodos públicos

- abstract State [mutation](#) (State state, double pm)

6.65.1. Descripción detallada

Abstract class that defines the mutation operator.

6.65.2. Documentación de funciones miembro

6.65.2.1. `mutation()`

```

abstract State es.ull.esit.app.evolutionary_algorithms.complement.Mutation.mutation (
    State state,
    double pm ) [abstract]
  
```

Applies mutation to a given state with a specified mutation probability.

Parámetros

<i>state</i>	[State] the state to be mutated
<i>pm</i>	[double] the mutation probability

Devuelve

[State] the mutated state

Reimplementado en [es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation](#), [es.ull.esit.app.evolutionary_algorithms.c](#) y [es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation.java](#)

6.66. Referencia de la clase

es.ull.esit.app.problem_operators.MutationOperator

Diagrama de herencia de es.ull.esit.app.problem_operators.MutationOperator

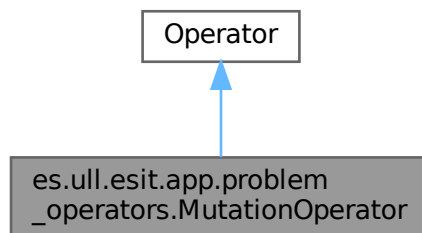
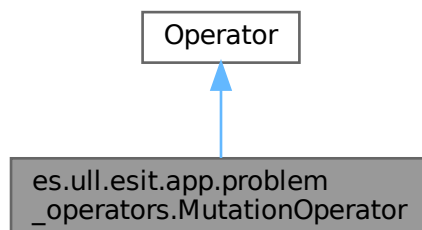


Diagrama de colaboración de es.ull.esit.app.problem_operators.MutationOperator:



Métodos públicos

- List< State > [generatedNewState](#) (State stateCurrent, Integer operatornumber)
- List< State > [generateRandomState](#) (Integer operatornumber)

6.66.1. Descripción detallada

Class that implements the mutation operator.

6.66.2. Documentación de funciones miembro**6.66.2.1. generatedNewState()**

```
List< State > es.ull.esit.app.problem_operators.MutationOperator.generatedNewState (
    State stateCurrent,
    Integer operatornumber )
```

Generates new states by applying the mutation operator.

Parámetros

<i>stateCurrent</i>	[State] Current state.
<i>operatornumber</i>	[Integer] Number of new states to generate.

Devuelve

listNeighborhood [List<State>] List of new states generated.

6.66.2.2. generateRandomState()

```
List< State > es.ull.esit.app.problem_operators.MutationOperator.generateRandomState (
    Integer operatornumber )
```

Generates random states.

Parámetros

<i>operatornumber</i>	[Integer] Number of random states to generate.
-----------------------	--

Devuelve

listRandomStates [List<State>] List of random states generated.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/problem_operators/[MutationOperator.java](#)

6.67. Referencia de la enumeración

`es.ull.esit.app.evolutionary_algorithms.complement.MutationType`

Atributos públicos

- [TOW_POINTS_MUTATION](#)
- [ONE_POINT_MUTATION](#)
- [AIO_MUTATION](#)

6.67.1. Descripción detallada

Enumeration representing different types of mutation methods.

6.67.2. Documentación de datos miembro

6.67.2.1. AIO_MUTATION

```
es.ull.esit.app.evolutionary_algorithms.complement.MutationType.AIO_MUTATION
```

One-point mutation method

6.67.2.2. ONE_POINT_MUTATION

```
es.ull.esit.app.evolutionary_algorithms.complement.MutationType.ONE_POINT_MUTATION
```

Two-point mutation method

6.67.2.3. TOW_POINTS_MUTATION

```
es.ull.esit.app.evolutionary_algorithms.complement.MutationType.TOW_POINTS_MUTATION
```

La documentación de esta enumeración está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/MutationType.java`

6.68. Referencia de la clase es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate

Diagrama de herencia de es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate

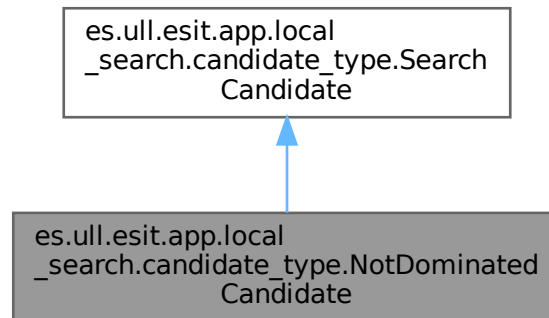
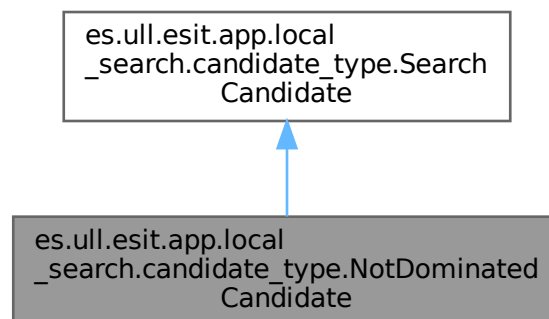


Diagrama de colaboración de es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate:



Métodos públicos

- State [stateSearch](#) (List< State > listNeighborhood) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

6.68.1. Descripción detallada

Class that represents a candidate not dominated in local search algorithms.

6.68.2. Documentación de funciones miembro

6.68.2.1. stateSearch()

```
State es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate.stateSearch (
    List< State > listNeighborhood ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Creates a new State from the list of neighborhood states that is not dominated by any other.

Parámetros

<i>listNeighborhood</i>	[List<State>] List of neighborhood states.
-------------------------	--

Devuelve

[State] State not dominated by any other.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If there is an illegal access.
<i>InvocationTargetException</i>	If the method invocation fails.
<i>NoSuchMethodException</i>	If the method is not found.

Reimplementado de [es.ull.esit.app.local_search.candidate_type.SearchCandidate](#).

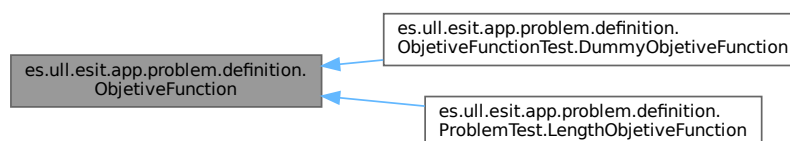
La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/local_search/candidate_type/NotDominatedCandidate.java](#)

6.69. Referencia de la clase

es.ull.esit.app.problem.definition.ObjetivoFunction

Diagrama de herencia de es.ull.esit.app.problem.definition.ObjetivoFunction



Métodos públicos

- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- ProblemType [getTypeProblem](#) ()
- void [setTypeProblem](#) (ProblemType typeProblem)
- abstract Double [evaluation](#) ([State](#) state)

6.69.1. Descripción detallada

Abstract class representing an objective function for evaluating states.

6.69.2. Documentación de funciones miembro**6.69.2.1. [evaluation\(\)](#)**

```
abstract Double es.ull.esit.app.problem.definition.ObjectiveFunction.evaluation (
    State state ) [abstract]
```

Evaluates the given state and returns its objective value.

Parámetros

<i>state</i>	[State] The state to evaluate.
--------------	--------------------------------

Devuelve

[Double] The objective value of the state.

6.69.2.2. [getTypeProblem\(\)](#)

```
ProblemType es.ull.esit.app.problem.definition.ObjectiveFunction.getTypeProblem ( )
```

Gets the type of problem associated with this objective function.

Devuelve

[ProblemType] The type of problem.

6.69.2.3. [getWeight\(\)](#)

```
float es.ull.esit.app.problem.definition.ObjectiveFunction.getWeight ( )
```

Gets the weight of the objective function.

Devuelve

[float] The weight of the objective function.

6.69.2.4. setTypeProblem()

```
void es.ull.esit.app.problem.definition.ObjectiveFunction.setTypeProblem (
    ProblemType typeProblem )
```

Sets the type of problem associated with this objective function.

Parámetros

<i>typeProblem</i>	[ProblemType] The type of problem to set.
--------------------	---

6.69.2.5. setWeight()

```
void es.ull.esit.app.problem.definition.ObjectiveFunction.setWeight (
    float weight )
```

Sets the weight of the objective function.

Parámetros

<i>weight</i>	[float] The weight to set.
---------------	----------------------------

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/definition/ObjectiveFunction.java](#)

6.70. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover` ↩

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover`

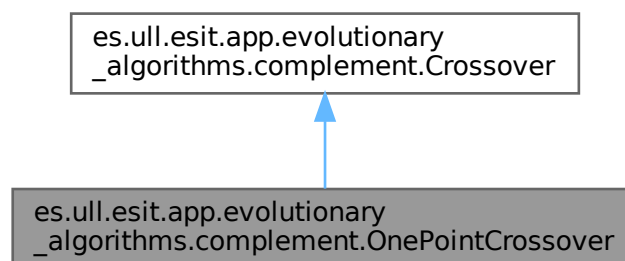
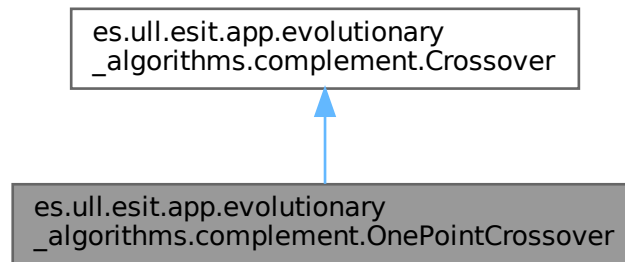


Diagrama de colaboración de `es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover`:



Métodos públicos

- State [crossover](#) (State father1, State father2, double pc)

6.70.1. Descripción detallada

Class that implements the one-point crossover operator.

6.70.2. Documentación de funciones miembro

6.70.2.1. `crossover()`

```

State es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover.crossover (
    State father1,
    State father2,
    double pc )
  
```

Applies the one-point crossover operation between two parent states with a given crossover probability.

Parámetros

<i>father1</i>	[State] the first parent state.
<i>father2</i>	[State] the second parent state.
<i>pc</i>	[double] the crossover probability.

Devuelve

[State] the resulting offspring state after crossover.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Crossover](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointCrossover.java`

6.71. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation`.↔

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation`

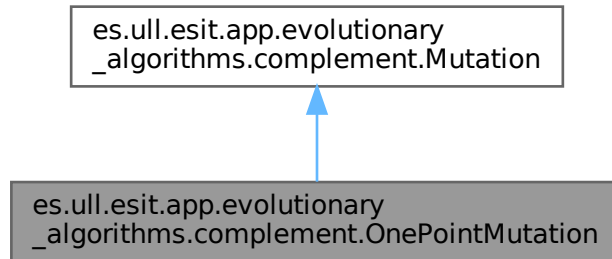
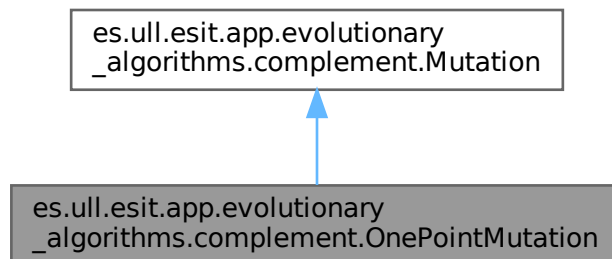


Diagrama de colaboración de `es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation`:



Métodos públicos

- State [mutation](#) (State state, double pm)

6.71.1. Descripción detallada

Class that implements the one-point mutation operator.

6.71.2. Documentación de funciones miembro

6.71.2.1. `mutation()`

```
State es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation.mutation (
    State state,
    double pm )
```

Applies the one-point mutation operation on a given state with a specified mutation probability.

Parámetros

<i>state</i>	[State] the state to be mutated.
<i>pm</i>	[double] the mutation probability.

Devuelve

[State] the mutated state.

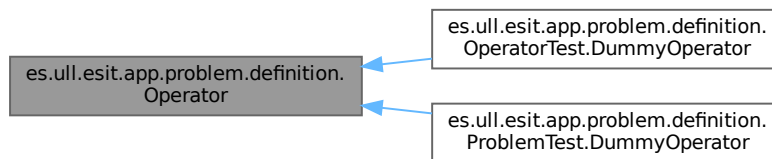
Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Mutation](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointMutation.java](#)

6.72. Referencia de la clase `es.ull.esit.app.problem.definition.Operator`

Diagrama de herencia de `es.ull.esit.app.problem.definition.Operator`

**Métodos públicos**

- `abstract List< State > generatedNewState (State stateCurrent, Integer operatornumber)`
- `abstract List< State > generateRandomState (Integer operatornumber)`

6.72.1. Descripción detallada

Abstract class that defines the operators of the problem.

6.72.2. Documentación de funciones miembro

6.72.2.1. `generatedNewState()`

```

abstract List< State > es.ull.esit.app.problem.definition.Operator.generatedNewState (
    State stateCurrent,
    Integer operatornumber ) [abstract]
  
```

Generates new states from the current state using the specified operator number.

Parámetros

<i>stateCurrent</i>	[State] Current state from which new states will be generated.
<i>operatornumber</i>	[Integer] The operator number to be applied.

Devuelve

[List<State>] A list of newly generated states.

6.72.2.2. generateRandomState()

```
abstract List< State > es.ull.esit.app.problem.definition.Operator.generateRandomState (
    Integer operatornumber ) [abstract]
```

Generates random states using the specified operator number.

Parámetros

<i>operatornumber</i>	[Integer] The operator number to be applied.
-----------------------	--

Devuelve

[List<State>] A list of randomly generated states.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/definition/Operator.java](#)

6.73. Referencia de la clase es.ull.esit.app.metaheuristics.generators.Particle

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.Particle

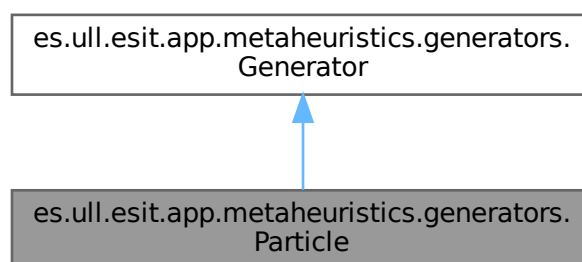
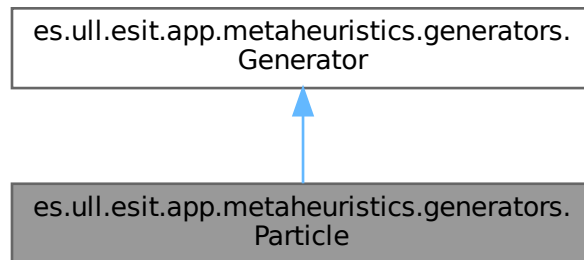


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.Particle`:



Métodos públicos

- `Particle ()`
- `Particle (State statePBest, State stateActual, List< Object > velocity)`
- `List< Object > getVelocity ()`
- `void setVelocity (List< Object > velocity)`
- `State getStatePBest ()`
- `void setStatePBest (State statePBest)`
- `State getStateActual ()`
- `void setStateActual (State stateActual)`
- `State generate (Integer operatornumber)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `void updateReference (State stateCandidate, Integer countIterationsCurrent)` throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `State getReference ()`
- `void setInitialReference (State stateInitialRef)`
- `GeneratorType getType ()`
- `List< State > getReferenceList ()`
- `List< State > getSonList ()`
- `boolean awardUpdateREF (State stateCandidate)`
- `void setWeight (float weight)`
- `float getWeight ()`
- `float[] getTrace ()`
- `int[] getListCountBetterGender ()`
- `int[] getListCountGender ()`

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- `int countGender`
- `int countBetterGender`
- `int[] listCountBetterGender`

6.73.1. Descripción detallada

Class that represents a particle in Particle Swarm Optimization (PSO). Each particle has:

- A current state (position).
- A personal best state (pBest).
- A velocity vector.

6.73.2. Documentación de constructores y destructores

6.73.2.1. Particle() [1/2]

```
es.ull.esit.app.metaheuristics.generators.Particle.Particle ( )
```

Default constructor. Creates empty states and an empty velocity vector, and initialises statistic arrays.

6.73.2.2. Particle() [2/2]

```
es.ull.esit.app.metaheuristics.generators.Particle.Particle (
    State statePBest,
    State stateActual,
    List< Object > velocity )
```

Full constructor.

Parámetros

<i>statePBest</i>	[State] Personal best state.
<i>stateActual</i>	[State] Current state.
<i>velocity</i>	[List<Object>] Velocity vector.

6.73.3. Documentación de funciones miembro

6.73.3.1. awardUpdateREF()

```
boolean es.ull.esit.app.metaheuristics.generators.Particle.awardUpdateREF (
    State stateCandidate )
```

Decides whether a candidate should be considered an improvement over the personal best; useful for statistics.

Parámetros

<i>stateCandidate</i>	[State] Candidate state to compare against pBest.
-----------------------	---

Devuelve

[boolean] True if candidate is better than pBest, false otherwise.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.Particle.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Updates the velocity and position of the particle according to PSO equations and returns the updated current state.

Parámetros

<i>operatornumber</i>	[Integer] Not used in PSO.
-----------------------	----------------------------

Devuelve

[State] Updated current state of the particle.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.3. getListCountBetterGender()

```
int[] es.ull.esit.app.metaheuristics.generators.Particle.getListCountBetterGender ( )
```

Returns the local counters for “better gender” statistics.

Devuelve

[int[]] The array of counters.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.4. getListCountGender()

```
int[] es.ull.esit.app.metaheuristics.generators.Particle.getListCountGender ( )
```

Returns the local counters for gender statistics.

Devuelve

[int[]] The array of counters.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.5. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.Particle.getReference ( )
```

Returns the current personal best as reference.

Devuelve

[State] The personal best state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.6. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.Particle.getReferenceList ( )
```

Returns the list of stored personal bests. Each call appends the current pBest (if not null) and returns a copy.

Devuelve

[List<State>] A new list containing the history of personal best states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.7. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.Particle.getSonList ( )
```

PSO does not maintain an explicit list of “sons”; we return the current state as a singleton list if present.

Devuelve

[List<State>] A list containing the current state, or empty if null.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.8. `getStateActual()`

```
State es.ull.esit.app.metaheuristics.generators.Particle.getStateActual ( )
```

Returns the current state.

Devuelve

[State] The current state.

6.73.3.9. getStatePBest()

```
State es.ull.esit.app.metaheuristics.generators.Particle.getStatePBest ( )
```

Returns the personal best state.

Devuelve

[State] The personal best state.

6.73.3.10. getTrace()

```
float[] es.ull.esit.app.metaheuristics.generators.Particle.getTrace ( )
```

Returns the trace of weight values.

Devuelve

[float[]] The trace array.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.11. getType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.Particle.getType ( )
```

Returns the type of the generator.

Devuelve

[GeneratorType] The generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.12. getVelocity()

```
List< Object > es.ull.esit.app.metaheuristics.generators.Particle.getVelocity ( )
```

Returns a defensive copy of the velocity vector.

Devuelve

[List<Object>] A new list containing the velocity elements.

6.73.3.13. getWeight()

```
float es.ull.esit.app.metaheuristics.generators.Particle.getWeight ( )
```

Returns the weight of the particle.

Devuelve

[float] The current weight.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.14. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.Particle.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference; for a particle we initialise both current state and personal best with the same state.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state.
------------------------	--------------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.15. setStateActual()

```
void es.ull.esit.app.metaheuristics.generators.Particle.setStateActual (
    State stateActual )
```

Sets the current state.

Parámetros

<i>stateActual</i>	[State] The new current state.
--------------------	--------------------------------

6.73.3.16. setStatePBest()

```
void es.ull.esit.app.metaheuristics.generators.Particle.setStatePBest (
    State statePBest )
```

Sets the personal best state.

Parámetros

<i>statePBest</i>	[State] The new personal best state.
-------------------	--------------------------------------

6.73.3.17. setVelocity()

```
void es.ull.esit.app.metaheuristics.generators.Particle.setVelocity (
    List< Object > velocity )
```

Sets the velocity vector.

Parámetros

<i>velocity</i>	The new velocity vector.
-----------------	--------------------------

6.73.3.18. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.Particle.setWeight (
    float weight )
```

Sets the weight of the particle.

Parámetros

<i>weight</i>	[float] The weight to set.
---------------	----------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.73.3.19. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.Particle.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Updates the personal best (pBest) according to the problem type (maximisation or minimisation).

Parámetros

<i>stateCandidate</i>	[State] Candidate state to consider for pBest update.
<i>countIterationsCurrent</i>	[Integer] Current iteration count (not used directly here).

Excepciones

<i>IllegalArgumentException</i>	If arguments are invalid.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class cannot be found.
<i>InstantiationException</i>	If an object cannot be instantiated.
<i>IllegalAccessException</i>	If access to a class or method is illegal.
<i>InvocationTargetException</i>	If a method invocation fails.
<i>NoSuchMethodException</i>	If a method cannot be found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/Particle.java`

6.74. Referencia de la clase es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization ↩

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization

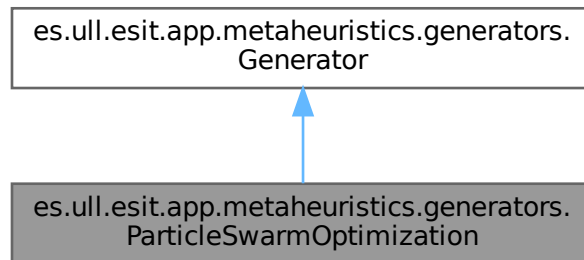
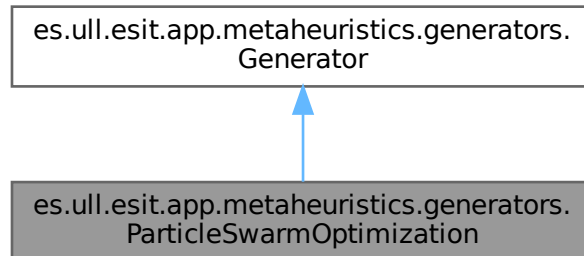


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization:



Métodos públicos

- [ParticleSwarmOptimization](#) ()
- State [generate](#) (Integer operatornumber) throws IllegalArgumentException, SecurityException, Class↩NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuch↩MethodException
- void [inicialiceLBest](#) ()
- State [getReference](#) ()
- State [getStateReferencePSO](#) ()
- void [setStateReferencePSO](#) (State stateReferencePSO)
- List< [Particle](#) > [getListStateReference](#) ()
- void [setListStateReference](#) (List< State > listStateReference)
- List< [Particle](#) > [getListParticle](#) ()
- List< [Particle](#) > [setListParticle](#) (List< [Particle](#) > listParticle)

- [GeneratorType](#) [getGeneratorType](#) ()
- void [setGeneratorType](#) ([GeneratorType](#) generatorType)
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)
- State [gBestInicial](#) ()
- void [setInitialReference](#) (State stateInitialRef)
- [GeneratorType](#) [getType](#) ()
- List< State > [getReferenceList](#) ()
- List< State > [getSonList](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- void [setWeight](#) (float weight)
- float [getWeight](#) ()
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()
- State [getGBest](#) ()
- void [setGBest](#) (State gBest)

Métodos públicos estáticos

- static int [getCountParticle](#) ()
- static int [getCountParticleBySwarm](#) ()
- static State[] [getLBest](#) ()
- static double [getConstriction](#) ()
- static void [setConstriction](#) (double constriction)
- static int [getCountCurrentIterPSO](#) ()
- static void [setCountCurrentIterPSO](#) (int countCurrentIterPSO)
- static double [getWmax](#) ()
- static void [setWmax](#) (double wmax)
- static double [getWmin](#) ()
- static void [setWmin](#) (double wmin)
- static int [getLearning1](#) ()
- static void [setLearning1](#) (int learning1)
- static int [getLearning2](#) ()
- static void [setLearning2](#) (int learning2)
- static boolean [isBinary](#) ()
- static void [setBinary](#) (boolean binary)
- static int [getCountRef](#) ()
- static void [setCountRef](#) (int countRef)

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.74.1. Descripción detallada

Class that implements the Particle Swarm Optimization (PSO) generator. It manages:

- A list of particles.
- Local best per swarm (lBest).
- Global best (gBest).

6.74.2. Documentación de constructores y destructores

6.74.2.1. `ParticleSwarmOptimization()`

```
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.ParticleSwarmOptimization
( )
```

Default constructor. Initialises the PSO generator with particles from RandomSearch reference states or reuses an existing particle list from the strategy map.

Excepciones

<i>IllegalArgumentException</i>	If an error occurs during initialisation.
<i>SecurityException</i>	If an error occurs during initialisation.
<i>ClassNotFoundException</i>	If an error occurs during initialisation.
<i>InstantiationException</i>	If an error occurs during initialisation.
<i>IllegalAccessException</i>	If an error occurs during initialisation.
<i>InvocationTargetException</i>	If an error occurs during initialisation.
<i>NoSuchMethodException</i>	If an error occurs during initialisation.

6.74.3. Documentación de funciones miembro

6.74.3.1. `awardUpdateREF()`

```
boolean es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.awardUpdateREF (
    State stateCandidate )
```

Returns true if the candidate is better than the current gBest (according to the problem type).

Parámetros

<i>stateCandidate</i>	[State] The candidate state to compare.
-----------------------	---

Devuelve

[boolean] True if candidate is better than gBest.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.2. gBestInicial()

```
State es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.gBestInicial ( )
```

Computes the initial global best state (gBest) from lBest array.

Devuelve

[State] The initial gBest state.

6.74.3.3. generate()

```
State es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Moves one particle using its internal PSO update and returns its new state.

Parámetros

<i>operatornumber</i>	[Integer] Not used in PSO.
-----------------------	----------------------------

Devuelve

[State] The new state of the current particle after movement.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.4. getConstriction()

```
static double es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getConstriction
( ) [static]
```

Convenience getters/setters for static fields (used by Particle).

Devuelve

[double] The constriction factor.

6.74.3.5. getCountCurrentIterPSO()

```
static int es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getCountCurrentIterPSO
( ) [static]
```

Gets the count of the current iteration in PSO.

Devuelve

[int] The current iteration count.

6.74.3.6. getCountParticle()

```
static int es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getCount←  
Particle ( ) [static]
```

Helper getters for static fields used by Particle.

Devuelve

[int] The total number of particles.

6.74.3.7. getCountParticleBySwarm()

```
static int es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getCount←  
ParticleBySwarm ( ) [static]
```

Returns the number of swarms.

Devuelve

[int] The number of swarms.

6.74.3.8. getCountRef()

```
static int es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getCountRef ( )  
[static]
```

Gets the total number of particles (countRef).

Devuelve

[int] The total number of particles.

6.74.3.9. getGBest()

```
State es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getGBest ( )
```

Gets the global best state (gBest).

Devuelve

[State] The global best state.

6.74.3.10. getGeneratorType()

```
GeneratorType es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getGeneratorType ( )
```

Gets the generator type.

Devuelve

[GeneratorType] The generator type.

6.74.3.11. getLBest()

```
static State[] es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getLBest ( ) [static]
```

Returns the local best array (lBest).

Devuelve

[State[]] The local best array.

6.74.3.12. getLearning1()

```
static int es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getLearning1 ( ) [static]
```

Gets the cognitive learning factor.

Devuelve

[int] The cognitive learning factor.

6.74.3.13. getLearning2()

```
static int es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getLearning2 ( ) [static]
```

Gets the social learning factor.

Devuelve

[int] The social learning factor.

6.74.3.14. `getListCountBetterGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getListCountBetterGender ( )
```

Returns the “better gender” statistics array.

Devuelve

[int[]] The array of better gender counters.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.15. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getListCountGender ( )
```

Returns the gender statistics array.

Devuelve

[int[]] The array of gender counters.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.16. `getListParticle()`

```
List< Particle > es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getListParticle ( )
```

Returns the particle list.

Devuelve

[List<Particle>] The particle list.

6.74.3.17. `getListStateReference()`

```
List< Particle > es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getListStateReference ( )
```

Returns the particle list (for tests / external inspection).

Devuelve

[List<Particle>] The particle list.

6.74.3.18. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getReference ( )
```

Returns the current reference state. For PSO, this is the global best if available, otherwise the internal reference.

Devuelve

[State] The current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.19. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getReference↵  
List ( )
```

Returns the list of reference states (gBest history).

Devuelve

[List<State>] The list of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.20. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getSonList ( )
```

PSO does not keep a separate list of sons; we return current particle positions as “sons”.

Devuelve

[List<State>] The list of current particle states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.21. `getStateReferencePSO()`

```
State es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getStateReferencePSO  
( )
```

Gets the internal PSO reference state.

Devuelve

[State] The internal PSO reference state.

6.74.3.22. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getTrace ( )
```

Returns the trace of weight values.

Devuelve

[float[]] The trace array.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.23. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getType ( )
```

Abstract method to get the type of generator.

Devuelve

[GeneratorType] The type of generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.24. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getWeight ( )
```

Gets the weight of this generator.

Devuelve

[float] The current weight.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.25. `getWmax()`

```
static double es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getWmax ( )  
[static]
```

Maximum inertia weight accessors.

Devuelve

[double] The maximum inertia weight.

6.74.3.26. getWmin()

```
static double es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.getWmin ( )  
[static]
```

Minimum inertia weight accessors.

Devuelve

[double] The minimum inertia weight.

6.74.3.27. inicialiceLBest()

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.inicialiceLBest ( )
```

Initialises the local best (lBest) for each swarm from the personal best of the particles.

6.74.3.28. isBinary()

```
static boolean es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.isBinary ( )  
[static]
```

Binary mode flag (both isBinary and getBinary for convenience).

Devuelve

[boolean] True if PSO is in binary mode, false otherwise.

6.74.3.29. setBinary()

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setBinary (   
    boolean binary ) [static]
```

Sets the binary mode flag.

Parámetros

<i>binary</i>	[boolean] The new binary mode flag.
---------------	-------------------------------------

6.74.3.30. setConstriction()

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setConstriction (   
    double constriction ) [static]
```

Sets the constriction factor.

Parámetros

<i>constriction</i>	[double] The new constriction factor.
---------------------	---------------------------------------

6.74.3.31. `setCountCurrentIterPSO()`

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setCountCurrentIterPSO (
    int countCurrentIterPSO ) [static]
```

Sets the count of the current iteration in PSO.

Parámetros

<i>countCurrentIterPSO</i>	[int] The new current iteration count.
----------------------------	--

6.74.3.32. `setCountRef()`

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setCountRef (
    int countRef ) [static]
```

Sets the total number of particles (countRef).

Parámetros

<i>countRef</i>	[int] The new total number of particles.
-----------------	--

6.74.3.33. `setGBest()`

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setGBest (
    State gBest )
```

Sets the global best state (gBest).

Parámetros

<i>gBest</i>	[State] The new global best state.
--------------	------------------------------------

6.74.3.34. `setGeneratorType()`

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setGeneratorType (
    GeneratorType generatorType )
```

Sets the generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] The new generator type.
----------------------	---

6.74.3.35. setInitialReference()

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state. For PSO we may use it as an initial gBest.

Parámetros

<i>stateInitialRef</i>	[State] The initial reference state.
------------------------	--------------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.36. setLearning1()

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setLearning1 (
    int learning1 ) [static]
```

Sets the cognitive learning factor.

Parámetros

<i>learning1</i>	[int] The new cognitive learning factor.
------------------	--

6.74.3.37. setLearning2()

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setLearning2 (
    int learning2 ) [static]
```

Sets the social learning factor.

Parámetros

<i>learning2</i>	[int] The new social learning factor.
------------------	---------------------------------------

6.74.3.38. setListParticle()

```
List< Particle > es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setList↵
Particle (
    List< Particle > listParticle )
```

Sets the particle list.

Parámetros

<i>listParticle</i>	[List<Particle>] The new particle list.
---------------------	---

Devuelve

[List<Particle>] The same list, for chaining.

6.74.3.39. setListStateReference()

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setListStateReference
(
    List< State > listStateReference )
```

Sets the internal list of reference states (gBest history).

Parámetros

<i>listStateReference</i>	[List<State>] The new list of reference states.
---------------------------	---

6.74.3.40. setStateReferencePSO()

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setStateReferencePSO
(
    State stateReferencePSO )
```

Sets the internal PSO reference state.

Parámetros

<i>stateReferencePSO</i>	[State] The new internal PSO reference state.
--------------------------	---

6.74.3.41. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setWeight (
    float weight )
```

Sets the weight of this generator.

Parámetros

<i>weight</i>	[float] The new weight.
---------------	-------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.74.3.42. setWmax()

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setWmax (
    double wmax ) [static]
```

Sets the maximum inertia weight.

Parámetros

<i>wmax</i>	[double] The new maximum inertia weight.
-------------	--

6.74.3.43. setWmin()

```
static void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.setWmin (
    double wmin ) [static]
```

Sets the minimum inertia weight.

Parámetros

<i>wmin</i>	[double] The new minimum inertia weight.
-------------	--

6.74.3.44. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Updates lBest and gBest based on the current particle personal best and the problem type (maximisation/minimisation).

Parámetros

<i>stateCandidate</i>	[State] The candidate state (not used directly here).
<i>countIterationsCurrent</i>	[Integer] The current iteration count.

Excepciones

<i>IllegalArgumentException</i>	If an error occurs during update.
<i>SecurityException</i>	If an error occurs during update.
<i>ClassNotFoundException</i>	If an error occurs during update.
<i>InstantiationException</i>	If an error occurs during update.
<i>IllegalAccessException</i>	If an error occurs during update.
<i>InvocationTargetException</i>	If an error occurs during update.
<i>NoSuchMethodException</i>	If an error occurs during update.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/ParticleSwarmOptimization.java`

6.75. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling`↔

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling`

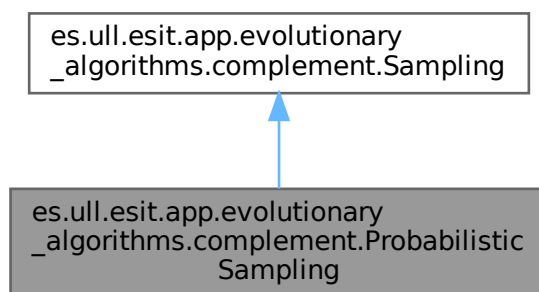
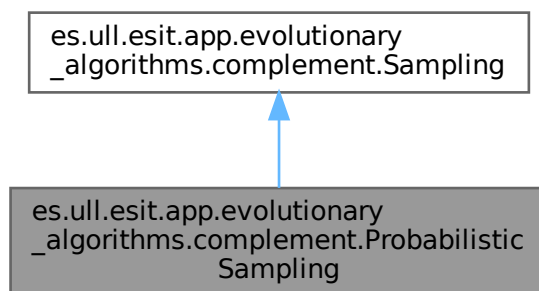


Diagrama de colaboración de `es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling`:



Métodos públicos

- `List< State > sampling` (`List< State > fathers`, `int countInd`)
- `List< State > listState` (`int countInd`)

6.75.1. Descripción detallada

Class that implements the probabilistic sampling operator.

6.75.2. Documentación de funciones miembro

6.75.2.1. `listState()`

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling.list↔
State (
    int countInd )
```

Generates a list of new State objects.

Parámetros

<i>countInd</i>	[int] the number of State objects to generate.
-----------------	--

Devuelve

[List<State>] the list of generated State objects.

6.75.2.2. `sampling()`

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling.↔
sampling (
    List< State > fathers,
    int countInd )
```

Applies the probabilistic sampling operation to generate new states based on a list of parent states.

Parámetros

<i>fathers</i>	[List<State>] the list of parent states.
<i>countInd</i>	[int] the number of new states to generate.

Devuelve

[List<State>] the list of newly generated states.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Sampling](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/ProbabilisticSampling.java`

6.76. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.Probability`

Métodos públicos

- float `getProbability` ()
- void `setProbability` (float probability)
- Object `getKey` ()
- void `setKey` (Object key)
- Object `getValue` ()
- void `setValue` (Object value)

6.76.1. Descripción detallada

Class that represents a probability with an associated key and value.

6.76.2. Documentación de funciones miembro

6.76.2.1. `getKey()`

```
Object es.ull.esit.app.evolutionary_algorithms.complement.Probability.getKey ( )
```

Gets the key associated with this probability.

Devuelve

[Object] The key associated with this probability.

6.76.2.2. `getProbability()`

```
float es.ull.esit.app.evolutionary_algorithms.complement.Probability.getProbability ( )
```

Gets the probability value.

Devuelve

[float] The probability value.

6.76.2.3. `getValue()`

```
Object es.ull.esit.app.evolutionary_algorithms.complement.Probability.getValue ( )
```

Gets the value associated with this probability.

Devuelve

[Object] The value associated with this probability.

6.76.2.4. `setKey()`

```
void es.ull.esit.app.evolutionary_algorithms.complement.Probability.setKey (
    Object key )
```

Sets the key associated with this probability.

Parámetros

<i>key</i>	[Object] The key to set.
------------	--------------------------

6.76.2.5. setProbability()

```
void es.ull.esit.app.evolutionary_algorithms.complement.Probability.setProbability (
    float probability )
```

Sets the probability value.

Parámetros

<i>probability</i>	[float] The probability value to set.
--------------------	---------------------------------------

6.76.2.6. setValue()

```
void es.ull.esit.app.evolutionary_algorithms.complement.Probability.setValue (
    Object value )
```

Sets the value associated with this probability.

Parámetros

<i>value</i>	[Object] The value to set.
--------------	----------------------------

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Probability.java`

6.77. Referencia de la clase es.ull.esit.app.problem.definition.Problem**Clases**

- enum [ProblemType](#)

Métodos públicos

- [Problem](#) ()
- List< [ObjectiveFunction](#) > [getFunction](#) ()
- void [setFunction](#) (List< [ObjectiveFunction](#) > function)
- [State](#) [getState](#) ()
- void [setState](#) ([State](#) state)
- [ProblemType](#) [getTypeProblem](#) ()
- void [setTypeProblem](#) ([ProblemType](#) typeProblem)

- [Codification](#) [getCodification](#) ()
- void [setCodification](#) ([Codification](#) codification)
- [Operator](#) [getOperator](#) ()
- void [setOperator](#) ([Operator](#) operator)
- int [getPossibleValue](#) ()
- void [setPossibleValue](#) (int possibleValue)
- void [evaluate](#) ([State](#) state) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)
- [TypeSolutionMethod](#) [getTypeSolutionMethod](#) ()
- void [setTypeSolutionMethod](#) ([TypeSolutionMethod](#) typeSolutionMethod)
- [IFFactorySolutionMethod](#) [getFactorySolutionMethod](#) ()
- void [setFactorySolutionMethod](#) ([IFFactorySolutionMethod](#) factorySolutionMethod)
- [SolutionMethod](#) [newSolutionMethod](#) ([TypeSolutionMethod](#) typeSolutionMethod) throws [IllegalArgumentException](#), [SecurityException](#), [ClassNotFoundException](#), [InstantiationException](#), [IllegalAccessException](#), [InvocationTargetException](#), [NoSuchMethodException](#)

6.77.1. Descripción detallada

Class that defines the problem to be solved.

6.77.2. Documentación de constructores y destructores

6.77.2.1. Problem()

```
es.ull.esit.app.problem.definition.Problem.Problem ( )
```

Constructor for the Problem class.

6.77.3. Documentación de funciones miembro

6.77.3.1. evaluate()

```
void es.ull.esit.app.problem.definition.Problem.evaluate (
    State state ) throws IllegalArgumentException, SecurityException, ClassNotFoundException,
InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Evaluates the given state using the defined objective functions or solution method.

Parámetros

<i>state</i>	[State] The state to be evaluated.
--------------	------------------------------------

Excepciones

<i>IllegalArgumentException</i>	If the argument is invalid.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class cannot be found.

Excepciones

<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method cannot be found.

6.77.3.2. getCodification()

`Codification` `es.ull.esit.app.problem.definition.Problem.getCodification ()`

Gets the codification method used in the problem.

Devuelve

[Codification] The codification method.

6.77.3.3. getFactorySolutionMethod()

`IFFactorySolutionMethod` `es.ull.esit.app.problem.definition.Problem.getFactorySolutionMethod ()`

Gets the factory for creating solution methods.

Devuelve

[IFFactorySolutionMethod] The factory for solution methods.

6.77.3.4. getFunction()

`List< ObjectiveFunction >` `es.ull.esit.app.problem.definition.Problem.getFunction ()`

Gets the list of objective functions.

Devuelve

List of objective functions.

6.77.3.5. getOperator()

`Operator` `es.ull.esit.app.problem.definition.Problem.getOperator ()`

Gets the operator used in the problem.

Devuelve

[Operator] The operator.

6.77.3.6. `getPossibleValue()`

```
int es.ull.esit.app.problem.definition.Problem.getPossibleValue ( )
```

Gets the possible values for the problem.

Devuelve

[int] The possible values.

6.77.3.7. `getState()`

```
State es.ull.esit.app.problem.definition.Problem.getState ( )
```

Gets the current state of the problem.

Devuelve

[State] The current state.

6.77.3.8. `getTypeProblem()`

```
ProblemType es.ull.esit.app.problem.definition.Problem.getTypeProblem ( )
```

Gets the type of the problem.

Devuelve

[ProblemType] The type of the problem.

6.77.3.9. `getTypeSolutionMethod()`

```
TypeSolutionMethod es.ull.esit.app.problem.definition.Problem.getTypeSolutionMethod ( )
```

Gets the type of solution method used in the problem.

Devuelve

[TypeSolutionMethod] The type of solution method.

6.77.3.10. `newSolutionMethod()`

```
SolutionMethod es.ull.esit.app.problem.definition.Problem.newSolutionMethod (
    TypeSolutionMethod typeSolutionMethod ) throws IllegalArgumentException, SecurityEx-
ception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invoca-
tionTargetException, NoSuchMethodException
```

Creates a new solution method based on the specified type.

Parámetros

<i>typeSolutionMethod</i>	[TypeSolutionMethod] The type of solution method to create.
---------------------------	---

Devuelve

[SolutionMethod] The newly created solution method.

Excepciones

<i>IllegalArgumentException</i>	If the argument is invalid.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method cannot be found.

6.77.3.11. setCodification()

```
void es.ull.esit.app.problem.definition.Problem.setCodification (
    Codification codification )
```

Sets the codification method used in the problem.

Parámetros

<i>codification</i>	[Codification] The codification method to set.
---------------------	--

6.77.3.12. setFactorySolutionMethod()

```
void es.ull.esit.app.problem.definition.Problem.setFactorySolutionMethod (
    IFFactorySolutionMethod factorySolutionMethod )
```

Sets the factory for creating solution methods.

Parámetros

<i>factorySolutionMethod</i>	[IFFactorySolutionMethod] The factory to set.
------------------------------	---

6.77.3.13. setFunction()

```
void es.ull.esit.app.problem.definition.Problem.setFunction (
    List< ObjectiveFunction > function )
```

Sets the list of objective functions.

Parámetros

<i>function</i>	[List<ObjectiveFunction>] List of objective functions to set.
-----------------	---

6.77.3.14. setOperator()

```
void es.ull.esit.app.problem.definition.Problem.setOperator (  
    Operator operator )
```

Sets the operator used in the problem.

Parámetros

<i>operator</i>	[Operator] The operator to set.
-----------------	---------------------------------

6.77.3.15. setPossibleValue()

```
void es.ull.esit.app.problem.definition.Problem.setPossibleValue (  
    int possibleValue )
```

Sets the possible values for the problem.

Parámetros

<i>possibleValue</i>	[int] The possible values to set.
----------------------	-----------------------------------

6.77.3.16. setState()

```
void es.ull.esit.app.problem.definition.Problem.setState (  
    State state )
```

Sets the current state of the problem.

Parámetros

<i>state</i>	[State] The state to set.
--------------	---------------------------

6.77.3.17. setTypeProblem()

```
void es.ull.esit.app.problem.definition.Problem.setTypeProblem (  
    ProblemType typeProblem )
```

Sets the type of the problem.

Parámetros

<i>typeProblem</i>	[ProblemType] The type of the problem to set.
--------------------	---

6.77.3.18. setTypeSolutionMethod()

```
void es.ull.esit.app.problem.definition.Problem.setTypeSolutionMethod (
    TypeSolutionMethod typeSolutionMethod )
```

Sets the type of solution method used in the problem.

Parámetros

<i>typeSolutionMethod</i>	[TypeSolutionMethod] The type of solution method to set.
---------------------------	--

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/definition/Problem.java](#)

6.78. Referencia de la enumeración

es.ull.esit.app.problem.definition.Problem.ProblemType

Atributos públicos

- [MAXIMIZAR](#)
- [MINIMIZAR](#)

6.78.1. Documentación de datos miembro**6.78.1.1. MAXIMIZAR**

```
es.ull.esit.app.problem.definition.Problem.ProblemType.MAXIMIZAR
```

6.78.1.2. MINIMIZAR

```
es.ull.esit.app.problem.definition.Problem.ProblemType.MINIMIZAR
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/definition/Problem.java](#)

6.79. Referencia de la clase

es.ull.esit.app.local_search.candidate_type.RandomCandidate

Diagrama de herencia de es.ull.esit.app.local_search.candidate_type.RandomCandidate

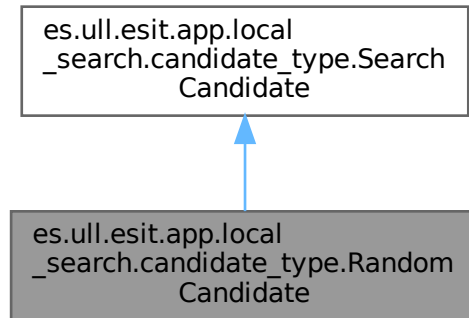
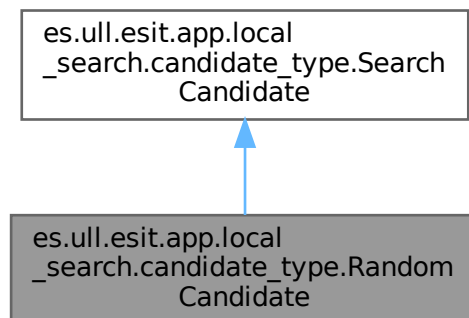


Diagrama de colaboración de es.ull.esit.app.local_search.candidate_type.RandomCandidate:



Métodos públicos

- State [stateSearch](#) (List< State > listNeighborhood)

6.79.1. Descripción detallada

Class that implements a random candidate selection strategy for local search.

6.79.2. Documentación de funciones miembro

6.79.2.1. stateSearch()

```
State es.ull.esit.app.local_search.candidate_type.RandomCandidate.stateSearch (
    List< State > listNeighborhood )
```

Selects a random state from the provided list of neighboring states.

Parámetros

<i>listNeighborhood</i>	List of neighboring states.
-------------------------	-----------------------------

Devuelve

A randomly selected state from the list.

Reimplementado de [es.ull.esit.app.local_search.candidate_type.SearchCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/candidate_type/RandomCandidate.java`

6.80. Referencia de la clase **es.ull.esit.app.metaheuristics.generators.RandomSearch**

Diagrama de herencia de `es.ull.esit.app.metaheuristics.generators.RandomSearch`

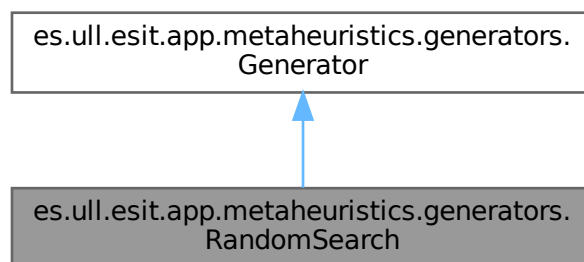
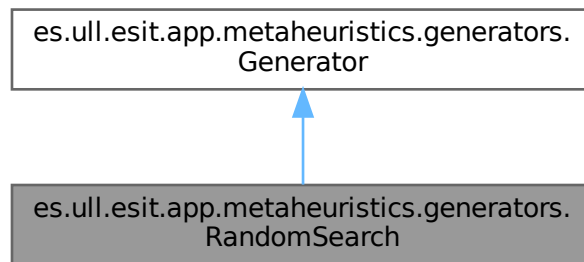


Diagrama de colaboración de `es.ull.esit.app.metaheuristics.generators.RandomSearch`:



Métodos públicos

- `RandomSearch` ()
- State `generate` (Integer operatornumber) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- State `getReference` ()
- void `setInitialReference` (State stateInitialRef)
- void `updateReference` (State stateCandidate, Integer countIterationsCurrent) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`
- `GeneratorType` `getType` ()
- `GeneratorType` `getTypeGenerator` ()
- void `setTypeGenerator` (`GeneratorType` typeGenerator)
- List< State > `getReferenceList` ()
- List< State > `getSonList` ()
- boolean `awardUpdateREF` (State stateCandidate)
- float `getWeight` ()
- void `setWeight` (float weight)
- int[] `getListCountBetterGender` ()
- int[] `getListCountGender` ()
- float[] `getTrace` ()

Métodos públicos estáticos

- static List< State > `getListStateReference` ()
- static void `setListStateReference` (List< State > listStateReference)

Otros miembros heredados

Atributos públicos heredados de `es.ull.esit.app.metaheuristics.generators.Generator`

- int `countGender`
- int `countBetterGender`
- int[] `listCountBetterGender`

6.80.1. Descripción detallada

Class that implements a Random Search generator.

6.80.2. Documentación de constructores y destructores

6.80.2.1. RandomSearch()

```
es.ull.esit.app.metaheuristics.generators.RandomSearch.RandomSearch ( )
```

Default constructor.

6.80.3. Documentación de funciones miembro

6.80.3.1. awardUpdateREF()

```
boolean es.ull.esit.app.metaheuristics.generators.RandomSearch.awardUpdateREF (
    State stateCandidate )
```

Awards an update to the reference state.

Parámetros

<i>stateCandidate</i>	[State] candidate state.
-----------------------	--------------------------

Devuelve

[boolean] whether the update is awarded.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.RandomSearch.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new random state using the underlying problem operator.

Parámetros

<i>operatornumber</i>	[Integer] Operator number to use for generation.
-----------------------	--

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.3. `getListCountBetterGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.RandomSearch.getListCountBetterGender ( )
```

Gets the list of better gender counts.

Devuelve

`[int[]]` List of better gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.4. `getListCountGender()`

```
int[] es.ull.esit.app.metaheuristics.generators.RandomSearch.getListCountGender ( )
```

Gets the list of gender counts.

Devuelve

`[int[]]` List of gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.5. `getListStateReference()`

```
static List< State > es.ull.esit.app.metaheuristics.generators.RandomSearch.getListState↔  
Reference ( ) [static]
```

Gets the global list of reference states.

Devuelve

`[State]` list of reference states.

6.80.3.6. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.RandomSearch.getReference ( )
```

Gets the current reference state for Random Search.

Devuelve

`[State]` Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.7. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.RandomSearch.getReferenceList ( )
```

Gets the list of reference states.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.8. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.RandomSearch.getSonList ( )
```

Gets the list of son states (empty for Random Search).

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.9. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.RandomSearch.getTrace ( )
```

Gets the trace of weights over iterations.

Devuelve

float[] trace of weights.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.10. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.RandomSearch.getType ( )
```

Gets the type of generator.

Devuelve

[GeneratorType] Type of generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.11. `getTypeGenerator()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.RandomSearch.getTypeGenerator ( )
```

Gets the type of generator.

Devuelve

[GeneratorType] Type of generator.

6.80.3.12. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.RandomSearch.getWeight ( )
```

Gets the weight of the generator.

Devuelve

[float] weight of the generator.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.13. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.RandomSearch.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state for Random Search.

Parámetros

<i>stateInitialRef</i>	[State] Initial reference state.
------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.14. `setListStateReference()`

```
static void es.ull.esit.app.metaheuristics.generators.RandomSearch.setListStateReference (
    List< State > listStateReference ) [static]
```

Sets the global list of reference states.

Parámetros

<i>listStateReference</i>	[State] list of reference states.
---------------------------	-----------------------------------

6.80.3.15. setTypeGenerator()

```
void es.ull.esit.app.metaheuristics.generators.RandomSearch.setTypeGenerator (
    GeneratorType typeGenerator )
```

Sets the type of generator.

Parámetros

<i>typeGenerator</i>	[GeneratorType] type of generator to set.
----------------------	---

6.80.3.16. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.RandomSearch.setWeight (
    float weight )
```

Sets the weight of the generator.

Parámetros

<i>weight</i>	[float] weight to set.
---------------	------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.80.3.17. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.RandomSearch.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
NoSuchMethodException
```

Updates the reference state based on the candidate state and acceptance criteria.

Parámetros

<i>stateCandidate</i>	[State] candidate state.
<i>countIterationsCurrent</i>	[Integer] current iteration count.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If illegal access occurs.
<i>InvocationTargetException</i>	If an invocation target error occurs.
<i>NoSuchMethodException</i>	If a method is not found.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/RandomSearch.java`

6.81. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.Range`

Métodos públicos

- [Probability](#) `getData ()`
- void `setData (Probability data)`
- float `getMax ()`
- void `setMax (float max)`
- float `getMin ()`
- void `setMin (float min)`

6.81.1. Descripción detallada

Class that represents a range with associated probability data.

6.81.2. Documentación de funciones miembro

6.81.2.1. `getData()`

```
Probability es.ull.esit.app.evolutionary_algorithms.complement.Range.getData ( )
```

Gets the probability data associated with this range.

Devuelve

[Probability] The probability data.

6.81.2.2. `getMax()`

```
float es.ull.esit.app.evolutionary_algorithms.complement.Range.getMax ( )
```

Gets the maximum value of the range.

Devuelve

[float] The maximum value.

6.81.2.3. getMin()

```
float es.ull.esit.app.evolutionary_algorithms.complement.Range.getMin ( )
```

Gets the minimum value of the range.

Devuelve

[float] The minimum value.

6.81.2.4. setData()

```
void es.ull.esit.app.evolutionary_algorithms.complement.Range.setData (
    Probability data )
```

Sets the probability data associated with this range.

Parámetros

<i>data</i>	[Probability] The probability data to set.
-------------	--

6.81.2.5. setMax()

```
void es.ull.esit.app.evolutionary_algorithms.complement.Range.setMax (
    float max )
```

Sets the maximum value of the range.

Parámetros

<i>max</i>	[float] The maximum value to set.
------------	-----------------------------------

6.81.2.6. setMin()

```
void es.ull.esit.app.evolutionary_algorithms.complement.Range.setMin (
    float min )
```

Sets the minimum value of the range.

Parámetros

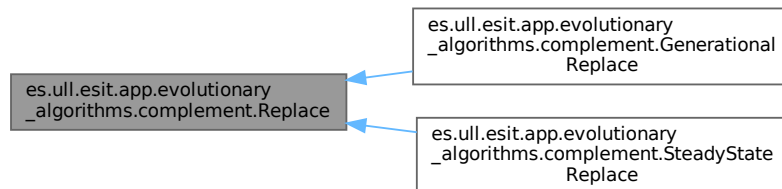
<i>min</i>	[float] The minimum value to set.
------------	-----------------------------------

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/[Range.java](#)

6.82. Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.Replace

Diagrama de herencia de es.ull.esit.app.evolutionary_algorithms.complement.Replace



Métodos públicos

- abstract List< State > **replace** (State stateCandidate, List< State > listState) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.82.1. Descripción detallada

Abstract class representing a replacement strategy in an evolutionary algorithm.

6.82.2. Documentación de funciones miembro

6.82.2.1. replace()

```

abstract List< State > es.ull.esit.app.evolutionary_algorithms.complement.Replace.replace (
    State stateCandidate,
    List< State > listState ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget←
    Exception, NoSuchMethodException [abstract]
  
```

Method to replace states in the population based on a candidate state.

Parámetros

<i>stateCandidate</i>	[State] The candidate state to be considered for replacement.
<i>listState</i>	[List<State>] The current list of states in the population.

Devuelve

[List<State>] The updated list of states after applying the replacement strategy.

Excepciones

<i>IllegalArgumentException</i>	
<i>SecurityException</i>	
<i>ClassNotFoundException</i>	
<i>InstantiationException</i>	
<i>IllegalAccessException</i>	
<i>InvocationTargetException</i>	
<i>NoSuchMethodException</i>	

Reimplementado en [es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace](#) y [es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace](#)

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replace.java](#)

6.83. Referencia de la enumeración

es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType

Atributos públicos

- [STEADY_STATE_REPLACE](#)
- [GENERATIONAL_REPLACE](#)

6.83.1. Descripción detallada

Enum representing the types of replacement strategies in an evolutionary algorithm.

6.83.2. Documentación de datos miembro

6.83.2.1. GENERATIONAL_REPLACE

```
es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType.GENERATIONAL_REPLACE
```

Replacement strategy where only a few individuals are replaced at a time

6.83.2.2. STEADY_STATE_REPLACE

```
es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType.STEADY_STATE_REPLACE
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/ReplaceType.java](#)

6.84. Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection

Diagrama de herencia de es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection

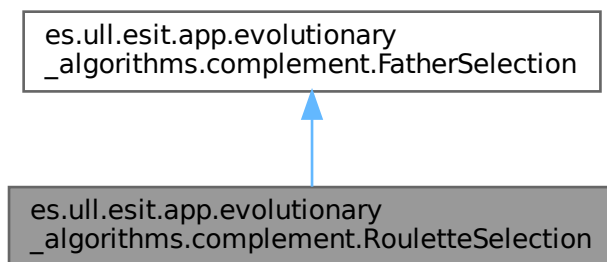
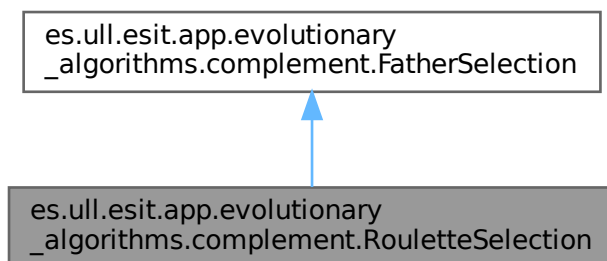


Diagrama de colaboración de es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection:



Métodos públicos

- List< State > [selection](#) (List< State > listState, int truncation)

6.84.1. Descripción detallada

Class implementing the roulette wheel selection method for selecting parents in an evolutionary algorithm.

6.84.2. Documentación de funciones miembro

6.84.2.1. selection()

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection.selection (
    List< State > listState,
    int truncation )
```

Selects parents from the given list of states using the roulette wheel selection method.

Parámetros

<i>listState</i>	[List<State>] The list of candidate states (individuals) to select from.
<i>truncation</i>	[int] The number of individuals to select (not used in this implementation).

Devuelve

[List<State>] The list of selected parent states.

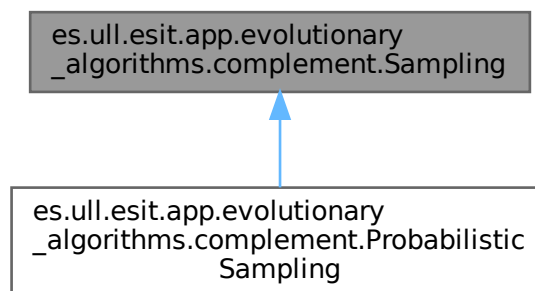
Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/RouletteSelection.java`

6.85. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.Sampling`

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.Sampling`



Métodos públicos

- `abstract List< State > sampling (List< State > fathers, int countInd)`

6.85.1. Descripción detallada

Abstract class defining the sampling method for selecting individuals in an evolutionary algorithm.

6.85.2. Documentación de funciones miembro

6.85.2.1. `sampling()`

```
abstract List< State > es.ull.esit.app.evolutionary_algorithms.complement.Sampling.sampling (
    List< State > fathers,
    int countInd ) [abstract]
```

Samples individuals from the given list of fathers.

Parámetros

<i>fathers</i>	[List<State>] The list of father states (individuals) to sample from.
<i>countInd</i>	[int] The number of individuals to sample.

Devuelve

[List<State>] The list of sampled individuals.

Reimplementado en [es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Sampling.java](#)

6.86. Referencia de la enumeración **es.ull.esit.app.evolutionary_algorithms.complement.SamplingType**

Atributos públicos

- [PROBABILISTIC_SAMPLING](#)

6.86.1. Descripción detallada

Enum defining the types of sampling methods available for selecting individuals in an evolutionary algorithm.

6.86.2. Documentación de datos miembro

6.86.2.1. PROBABILISTIC_SAMPLING

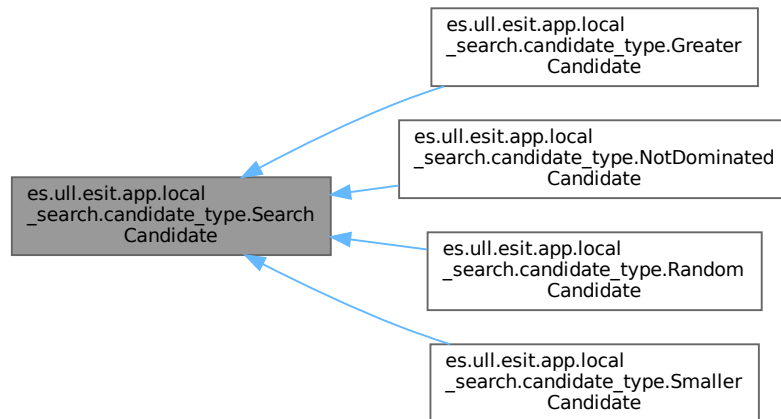
```
es.ull.esit.app.evolutionary_algorithms.complement.SamplingType.PROBABILISTIC_SAMPLING
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SamplingType.java](#)

6.87. Referencia de la clase es.ull.esit.app.local_search.candidate_type.SearchCandidate

Diagrama de herencia de es.ull.esit.app.local_search.candidate_type.SearchCandidate



Métodos públicos

- abstract State [stateSearch](#) (List< State > listNeighborhood) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.87.1. Descripción detallada

Abstract class defining the interface for candidate selection strategies in local search.

6.87.2. Documentación de funciones miembro

6.87.2.1. stateSearch()

```

abstract State es.ull.esit.app.local_search.candidate_type.SearchCandidate.stateSearch (
    List< State > listNeighborhood ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException [abstract]
  
```

Method to select a state from the list of neighboring states.

Parámetros

<i>listNeighborhood</i>	[List<State>] List of neighboring states.
-------------------------	---

Devuelve

[State] Selected state from the list.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class cannot be found.
<i>InstantiationException</i>	If an object cannot be instantiated.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method cannot be found.

Reimplementado en [es.ull.esit.app.local_search.candidate_type.GreaterCandidate](#), [es.ull.esit.app.local_search.candidate_type.NotD](#), [es.ull.esit.app.local_search.candidate_type.RandomCandidate](#) y [es.ull.esit.app.local_search.candidate_type.SmallerCandidate](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/local_search/candidate_type/SearchCandidate.java](#)

6.88. Referencia de la enumeración

es.ull.esit.app.evolutionary_algorithms.complement.SelectionType

Atributos públicos

- [ROULETTE_SELECTION](#)
- [TRUNCATION_SELECTION](#)

6.88.1. Descripción detallada

Enum defining the types of selection methods available for selecting parents in an evolutionary algorithm.

6.88.2. Documentación de datos miembro**6.88.2.1. ROULETTE_SELECTION**

```
es.ull.esit.app.evolutionary_algorithms.complement.SelectionType.ROULETTE_SELECTION
```

6.88.2.2. TRUNCATION_SELECTION

```
es.ull.esit.app.evolutionary_algorithms.complement.SelectionType.TRUNCATION_SELECTION
```

Roulette wheel selection method

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SelectionType.java](#)

6.89. Referencia de la clase es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing

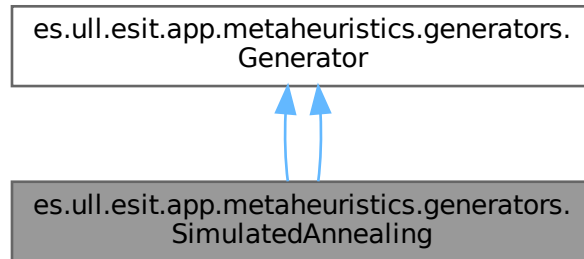
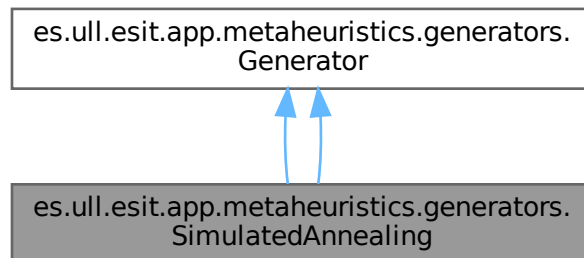


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing:



Métodos públicos

- [SimulatedAnnealing \(\)](#)
- State [generate](#) (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- State [getReference](#) ()
- void [setStateRef](#) (State stateRef)
- void [setInitialReference](#) (State stateInitialRef)
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- [GeneratorType getType](#) ()
- [GeneratorType getTypeGenerator](#) ()

- void [setTypeGenerator](#) ([GeneratorType](#) typeGenerator)
- List< State > [getReferenceList](#) ()
- List< State > [getSonList](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()
- [SimulatedAnnealing](#) ()
- State [generate](#) (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- State [getReference](#) ()
- void [setStateRef](#) (State stateRef)
- void [setInitialReference](#) (State stateInitialRef)
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- [GeneratorType](#) [getType](#) ()
- [GeneratorType](#) [getTypeGenerator](#) ()
- void [setTypeGenerator](#) ([GeneratorType](#) typeGenerator)
- List< State > [getReferenceList](#) ()
- List< State > [getSonList](#) ()
- boolean [awardUpdateREF](#) (State stateCandidate)
- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()

Métodos públicos estáticos

- static Double [getAlpha](#) ()
- static void [setAlpha](#) (Double aAlpha)
- static Double [getTinitial](#) ()
- static void [setTinitial](#) (Double aTinitial)
- static Double [getTfinal](#) ()
- static void [setTfinal](#) (Double aTfinal)
- static int [getCountIterationsT](#) ()
- static void [setCountIterationsT](#) (int aCountIterationsT)
- static Double [getAlpha](#) ()
- static void [setAlpha](#) (Double aAlpha)
- static Double [getTinitial](#) ()
- static void [setTinitial](#) (Double aTinitial)
- static Double [getTfinal](#) ()
- static void [setTfinal](#) (Double aTfinal)
- static int [getCountIterationsT](#) ()
- static void [setCountIterationsT](#) (int aCountIterationsT)

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- `int` `countGender`
- `int` `countBetterGender`
- `int[]` `listCountBetterGender`

6.89.1. Descripción detallada

Class that implements a Simulated Annealing generator.

6.89.2. Documentación de constructores y destructores

6.89.2.1. `SimulatedAnnealing()` [1/2]

```
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.SimulatedAnnealing ( )
```

Default constructor.

6.89.2.2. `SimulatedAnnealing()` [2/2]

```
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.SimulatedAnnealing ( )
```

Default constructor.

6.89.3. Documentación de funciones miembro

6.89.3.1. `awardUpdateREF()` [1/2]

```
boolean es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.awardUpdateREF (
    State stateCandidate )
```

For Simulated Annealing, we do not use an additional “award” criterion beyond the acceptance rule, so this always returns `false`.

Parámetros

<code>stateCandidate</code>	[State] Candidate state.
-----------------------------	--------------------------

Devuelve

[boolean] Always false.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.2. awardUpdateREF() [2/2]

```
boolean es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.awardUpdateREF (
    State stateCandidate )
```

For Simulated Annealing, we do not use an additional “award” criterion beyond the acceptance rule, so this always returns `false`.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
-----------------------	--------------------------

Devuelve

[boolean] Always false.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.3. generate() [1/2]

```
State es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new candidate state from the current reference using the neighbourhood defined by the operator.

Parámetros

<i>operatornumber</i>	[Integer] Operator index.
-----------------------	---------------------------

Devuelve

[State] Generated candidate state or `null` if strategy/problem is not available.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.4. generate() [2/2]

```
State es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new candidate state from the current reference using the neighbourhood defined by the operator.

Parámetros

<i>operatornumber</i>	[Integer] Operator index.
-----------------------	---------------------------

Devuelve

[State] Generated candidate state or `null` if strategy/problem is not available.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.5. `getAlpha()` [1/2]

```
static Double es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getAlpha ( ) [static]
```

Gets the cooling factor.

Devuelve

[Double] Cooling factor.

6.89.3.6. `getAlpha()` [2/2]

```
static Double es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getAlpha ( ) [static]
```

Gets the cooling factor.

Devuelve

[Double] Cooling factor.

6.89.3.7. `getCountIterationsT()` [1/2]

```
static int es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getCountIterationsT (
) [static]
```

Gets the iteration index for the next temperature update.

Devuelve

[int] Iteration index.

6.89.3.8. `getCountIterationsT()` [2/2]

```
static int es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getCountIterationsT (
) [static]
```

Gets the iteration index for the next temperature update.

Devuelve

[int] Iteration index.

6.89.3.9. getListCountBetterGender() [1/2]

```
int[] es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getListCountBetterGender ( )
```

Returns the internal statistics array of “better gender” counts.

Devuelve

[int[]] Better gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.10. getListCountBetterGender() [2/2]

```
int[] es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getListCountBetterGender ( )
```

Returns the internal statistics array of “better gender” counts.

Devuelve

[int[]] Better gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.11. getListCountGender() [1/2]

```
int[] es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getListCountGender ( )
```

Returns the internal statistics array of gender counts.

Devuelve

[int[]] Gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.12. getListCountGender() [2/2]

```
int[] es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getListCountGender ( )
```

Returns the internal statistics array of gender counts.

Devuelve

[int[]] Gender counts.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.13. `getReference()` [1/2]

```
State es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getReference ( )
```

Returns the current reference state.

Devuelve

[State] Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.14. `getReference()` [2/2]

```
State es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getReference ( )
```

Returns the current reference state.

Devuelve

[State] Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.15. `getReferenceList()` [1/2]

```
List< State > es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getReferenceList ( )
```

Returns the list of reference states visited so far. Each call appends the current reference state if not null.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.16. `getReferenceList()` [2/2]

```
List< State > es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getReferenceList ( )
```

Returns the list of reference states visited so far. Each call appends the current reference state if not null.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.17. getSonList() [1/2]

```
List< State > es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getSonList ( )
```

Simulated Annealing does not explicitly maintain a son list. We return an empty list to avoid null handling.

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.18. getSonList() [2/2]

```
List< State > es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getSonList ( )
```

Simulated Annealing does not explicitly maintain a son list. We return an empty list to avoid null handling.

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.19. getTfinal() [1/2]

```
static Double es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTfinal ( ) [static]
```

Gets the final temperature.

Devuelve

[Double] Final temperature.

6.89.3.20. getTfinal() [2/2]

```
static Double es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTfinal ( ) [static]
```

Gets the final temperature.

Devuelve

[Double] Final temperature.

6.89.3.21. `getTinitial()` [1/2]

```
static Double es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTinitial ( )  
[static]
```

Gets the initial temperature.

Devuelve

[Double] Initial temperature.

6.89.3.22. `getTinitial()` [2/2]

```
static Double es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTinitial ( )  
[static]
```

Gets the initial temperature.

Devuelve

[Double] Initial temperature.

6.89.3.23. `getTrace()` [1/2]

```
float[] es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTrace ( )
```

Returns the trace of weight values.

Devuelve

[float[]] Weight trace.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.24. `getTrace()` [2/2]

```
float[] es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTrace ( )
```

Returns the trace of weight values.

Devuelve

[float[]] Weight trace.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.25. getType() [1/2]

```
GeneratorType es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getType ( )
```

Returns the type of this generator.

Devuelve

[GeneratorType] Generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.26. getType() [2/2]

```
GeneratorType es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getType ( )
```

Returns the type of this generator.

Devuelve

[GeneratorType] Generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.27. getTypeGenerator() [1/2]

```
GeneratorType es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTypeGenerator ( )
```

Returns the generator type (explicit getter).

Devuelve

[GeneratorType] Generator type.

6.89.3.28. getTypeGenerator() [2/2]

```
GeneratorType es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getTypeGenerator ( )
```

Returns the generator type (explicit getter).

Devuelve

[GeneratorType] Generator type.

6.89.3.29. `getWeight()` [1/2]

```
float es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getWeight ( )
```

Returns the weight associated with this generator.

Devuelve

[float] Weight value.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.30. `getWeight()` [2/2]

```
float es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.getWeight ( )
```

Returns the weight associated with this generator.

Devuelve

[float] Weight value.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.31. `setAlpha()` [1/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setAlpha (
    Double aAlpha ) [static]
```

Sets the cooling factor.

Parámetros

<i>aAlpha</i>	[Double] New cooling factor.
---------------	------------------------------

6.89.3.32. `setAlpha()` [2/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setAlpha (
    Double aAlpha ) [static]
```

Sets the cooling factor.

Parámetros

<i>aAlpha</i>	[Double] New cooling factor.
---------------	------------------------------

6.89.3.33. setCountIterationsT() [1/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setCountIterationsT (
    int aCountIterationsT ) [static]
```

Sets the iteration index for the next temperature update.

Parámetros

<i>aCountIterationsT</i>	[int] New iteration index.
--------------------------	----------------------------

6.89.3.34. setCountIterationsT() [2/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setCountIterationsT (
    int aCountIterationsT ) [static]
```

Sets the iteration index for the next temperature update.

Parámetros

<i>aCountIterationsT</i>	[int] New iteration index.
--------------------------	----------------------------

6.89.3.35. setInitialReference() [1/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state for Simulated Annealing.

Parámetros

<i>stateInitialRef</i>	[State] Initial reference state.
------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.36. setInitialReference() [2/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state for Simulated Annealing.

Parámetros

<i>stateInitialRef</i>	[State] Initial reference state.
------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.37. `setStateRef()` [1/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setStateRef (
    State stateRef )
```

Sets the current reference state.

Parámetros

<i>stateRef</i>	[State] New reference state.
-----------------	------------------------------

6.89.3.38. `setStateRef()` [2/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setStateRef (
    State stateRef )
```

Sets the current reference state.

Parámetros

<i>stateRef</i>	[State] New reference state.
-----------------	------------------------------

6.89.3.39. `setTfinal()` [1/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setTfinal (
    Double aTfinal ) [static]
```

Sets the final temperature.

Parámetros

<i>aTfinal</i>	[Double] New final temperature.
----------------	---------------------------------

6.89.3.40. `setTfinal()` [2/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setTfinal (
    Double aTfinal ) [static]
```

Sets the final temperature.

Parámetros

<i>aTfinal</i>	[Double] New final temperature.
----------------	---------------------------------

6.89.3.41. setTinitial() [1/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setTinitial (
    Double aTinitial ) [static]
```

Sets the initial temperature.

Parámetros

<i>aTinitial</i>	[Double] New initial temperature.
------------------	-----------------------------------

6.89.3.42. setTinitial() [2/2]

```
static void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setTinitial (
    Double aTinitial ) [static]
```

Sets the initial temperature.

Parámetros

<i>aTinitial</i>	[Double] New initial temperature.
------------------	-----------------------------------

6.89.3.43. setTypeGenerator() [1/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setTypeGenerator (
    GeneratorType typeGenerator )
```

Sets the generator type (mainly for testing or configuration).

Parámetros

<i>typeGenerator</i>	[GeneratorType] New generator type.
----------------------	-------------------------------------

6.89.3.44. setTypeGenerator() [2/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setTypeGenerator (
    GeneratorType typeGenerator )
```

Sets the generator type (mainly for testing or configuration).

Parámetros

<i>typeGenerator</i>	[GeneratorType] New generator type.
----------------------	-------------------------------------

6.89.3.45. `setWeight()` [1/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setWeight (
    float weight )
```

Sets the weight associated with this generator.

Parámetros

<i>weight</i>	[float] New weight value.
---------------	---------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.46. `setWeight()` [2/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.setWeight (
    float weight )
```

Sets the weight associated with this generator.

Parámetros

<i>weight</i>	[float] New weight value.
---------------	---------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.47. `updateReference()` [1/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, Security↵
Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocation↵
TargetException, NoSuchMethodException
```

Updates the reference state according to the Simulated Annealing acceptance criteria and cooling schedule.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
<i>countIterationsCurrent</i>	[Integer] Current global iteration count.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.89.3.48. `updateReference()` [2/2]

```
void es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing.updateReference (
    State stateCandidate,
```

`Integer countIterationsCurrent`) throws `IllegalArgumentException`, `SecurityException`, `ClassNotFoundException`, `InstantiationException`, `IllegalAccessException`, `InvocationTargetException`, `NoSuchMethodException`

Updates the reference state according to the Simulated Annealing acceptance criteria and cooling schedule.

Parámetros

<code>stateCandidate</code>	[State] Candidate state.
<code>countIterationsCurrent</code>	[Integer] Current global iteration count.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada de los siguientes archivos:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java`
- `src/test/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java`

6.90. Referencia de la clase

es.ull.esit.app.local_search.candidate_type.SmallerCandidate

Diagrama de herencia de `es.ull.esit.app.local_search.candidate_type.SmallerCandidate`

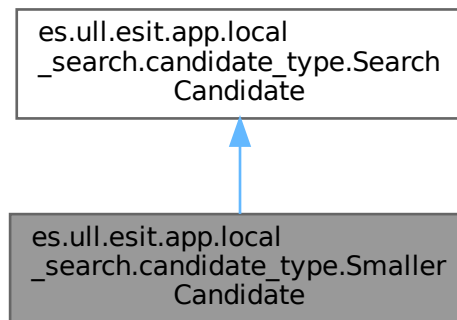
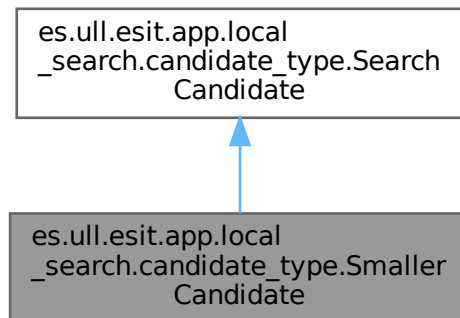


Diagrama de colaboración de es.ull.esit.app.local_search.candidate_type.SmallerCandidate:



Métodos públicos

- State [stateSearch](#) (List< State > listNeighborhood) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchElementException

6.90.1. Descripción detallada

Class that represents a candidate with the smallest evaluation in local search algorithms.

6.90.2. Documentación de funciones miembro

6.90.2.1. stateSearch()

```

State es.ull.esit.app.local_search.candidate_type.SmallerCandidate.stateSearch (
    List< State > listNeighborhood ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchElementException
  
```

Creates a new State from the list of neighborhood states that has the smallest evaluation.

Parámetros

<i>listNeighborhood</i>	[List<State>] List of neighborhood states.
-------------------------	--

Devuelve

[State] State with the smallest evaluation.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If there is an illegal access.
<i>InvocationTargetException</i>	If the method invocation fails.
<i>NoSuchMethodException</i>	If the method is not found.

Reimplementado de [es.ull.esit.app.local_search.candidate_type.SearchCandidate](#).

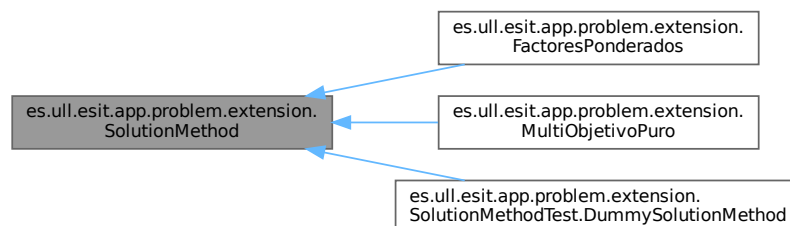
La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/candidate_type/SmallerCandidate.java`

6.91. Referencia de la clase

es.ull.esit.app.problem.extension.SolutionMethod

Diagrama de herencia de es.ull.esit.app.problem.extension.SolutionMethod



Métodos públicos

- abstract void [evaluationState](#) (State state)

6.91.1. Descripción detallada

Abstract class for solution methods.

6.91.2. Documentación de funciones miembro

6.91.2.1. evaluationState()

```

abstract void es.ull.esit.app.problem.extension.SolutionMethod.evaluationState (
    State state ) [abstract]
  
```

Method to evaluate a state.

Parámetros

<code>state</code>	[State] State to be evaluated.
--------------------	--------------------------------

Reimplementado en [es.ull.esit.app.problem.extension.FactoresPonderados](#) y [es.ull.esit.app.problem.extension.MultiObjetivoPuro](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/problem/extension/SolutionMethod.java`

6.92. Referencia de la clase `es.ull.esit.app.problem.definition.State`

Métodos públicos

- `State (State ps)`
- `State (List< Object > code)`
- `State ()`
- `List< Object > getCode ()`
- `void setCode (List< Object > listCode)`
- `GeneratorType getTypeGenerator ()`
- `void setTypeGenerator (GeneratorType typeGenerator)`
- `List< Double > getEvaluation ()`
- `void setEvaluation (List< Double > evaluation)`
- `int getNumber ()`
- `void setNumber (int number)`
- `State copy ()`
- `Object getCopy ()`
- `boolean comparator (State state)`
- `double distance (State state)`

Atributos protegidos

- `GeneratorType typeGenerator`
- `List< Double > evaluation`
- `int number`
- `List< Object > code`

6.92.1. Descripción detallada

Class that defines the state of the problem.

6.92.2. Documentación de constructores y destructores

6.92.2.1. `State()` [1/3]

```
es.ull.esit.app.problem.definition.State.State (
    State ps )
```

Copy constructor.

Parámetros

<i>ps</i>	[State] State to copy.
-----------	------------------------

6.92.2.2. State() [2/3]

```
es.ull.esit.app.problem.definition.State.State (
    List< Object > code )
```

Constructor with code initialization.

Parámetros

<i>code</i>	[List<Object>] Code to initialize the state.
-------------	--

6.92.2.3. State() [3/3]

```
es.ull.esit.app.problem.definition.State.State ( )
```

Default constructor.

6.92.3. Documentación de funciones miembro**6.92.3.1. comparator()**

```
boolean es.ull.esit.app.problem.definition.State.comparator (
    State state )
```

Compares the current state with another state.

Parámetros

<i>state</i>	[State] The state to compare with.
--------------	------------------------------------

Devuelve

[boolean] True if the states are equal, false otherwise.

6.92.3.2. copy()

```
State es.ull.esit.app.problem.definition.State.copy ( )
```

Creates a copy of the current state.

Devuelve

[State] A new State object that is a copy of the current state.

6.92.3.3. `distance()`

```
double es.ull.esit.app.problem.definition.State.distance (
    State state )
```

Calculates the distance between the current state and another state.

Parámetros

<code>state</code>	[State] The state to calculate the distance to.
--------------------	---

Devuelve

[double] The distance between the two states.

6.92.3.4. `getCode()`

```
List< Object > es.ull.esit.app.problem.definition.State.getCode ( )
```

Gets the code of the state.

Devuelve

[List<Object>] The code of the state.

6.92.3.5. `getCopy()`

```
Object es.ull.esit.app.problem.definition.State.getCopy ( )
```

Creates a copy of the current state.

Devuelve

[Object] A new Object that is a copy of the current state.

6.92.3.6. `getEvaluation()`

```
List< Double > es.ull.esit.app.problem.definition.State.getEvaluation ( )
```

Gets the evaluation values for the state.

Devuelve

[ArrayList<Double>] The evaluation values for the state.

6.92.3.7. `getNumber()`

```
int es.ull.esit.app.problem.definition.State.getNumber ( )
```

Gets the unique number identifying the state.

Devuelve

[int] The unique number identifying the state.

6.92.3.8. `getTypeGenerator()`

```
GeneratorType es.ull.esit.app.problem.definition.State.getTypeGenerator ( )
```

Gets the type of generator used to create the state.

Devuelve

[GeneratorType] The type of generator used to create the state.

6.92.3.9. `setCode()`

```
void es.ull.esit.app.problem.definition.State.setCode (
    List< Object > listCode )
```

Sets the code of the state.

Parámetros

<i>listCode</i>	[List<Object>] The code to set for the state.
-----------------	---

6.92.3.10. `setEvaluation()`

```
void es.ull.esit.app.problem.definition.State.setEvaluation (
    List< Double > evaluation )
```

Sets the evaluation values for the state.

Parámetros

<i>evaluation</i>	[ArrayList<Double>] The evaluation values to set for the state.
-------------------	---

6.92.3.11. `setNumber()`

```
void es.ull.esit.app.problem.definition.State.setNumber (
    int number )
```

Sets the unique number identifying the state.

Parámetros

<i>number</i>	[int] The unique number to set for the state.
---------------	---

6.92.3.12. setTypeGenerator()

```
void es.ull.esit.app.problem.definition.State.setTypeGenerator (
    GeneratorType typeGenerator )
```

Sets the type of generator used to create the state.

Parámetros

<i>typeGenerator</i>	[GeneratorType] The type of generator to set.
----------------------	---

6.92.4. Documentación de datos miembro

6.92.4.1. code

```
List<Object> es.ull.esit.app.problem.definition.State.code [protected]
```

6.92.4.2. evaluation

```
List<Double> es.ull.esit.app.problem.definition.State.evaluation [protected]
```

6.92.4.3. number

```
int es.ull.esit.app.problem.definition.State.number [protected]
```

6.92.4.4. typeGenerator

```
GeneratorType es.ull.esit.app.problem.definition.State.typeGenerator [protected]
```

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/problem/definition/State.java`

6.93. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace`.↔

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace`

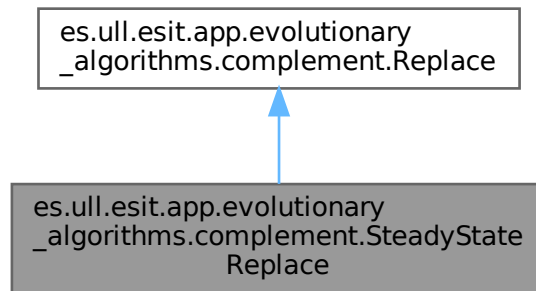
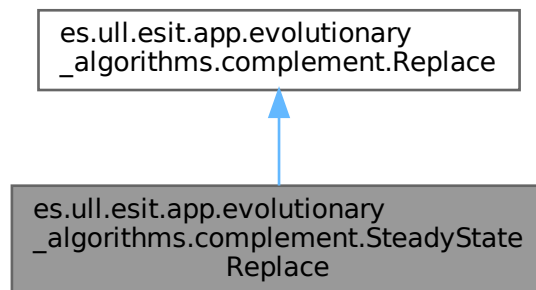


Diagrama de colaboración de `es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace`:



Métodos públicos

- `List< State > replace (State stateCandidate, List< State > listState)`
- `State minValue (List< State > listState)`
- `State maxValue (List< State > listState)`

6.93.1. Descripción detallada

Class that implements the steady state replacement method.

6.93.2. Documentación de funciones miembro

6.93.2.1. `maxValue()`

```
State es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace.maxValue (
    List< State > listState )
```

Finds the state with the maximum evaluation value in a list of states.

Parámetros

<i>listState</i>	[List<State>] The list of states to search.
------------------	---

Devuelve

[State] The state with the maximum evaluation value.

6.93.2.2. `minValue()`

```
State es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace.minValue (
    List< State > listState )
```

Finds the state with the minimum evaluation value in a list of states.

Parámetros

<i>listState</i>	[List<State>] The list of states to search.
------------------	---

Devuelve

[State] The state with the minimum evaluation value.

6.93.2.3. `replace()`

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace.replace (
    State stateCandidate,
    List< State > listState )
```

Replaces a state in the population using the steady state replacement method.

Parámetros

<i>stateCandidate</i>	[State] The candidate state to be considered for replacement.
<i>listState</i>	[List<State>] The current population of states.

Devuelve

[List<State>] The updated population of states after replacement.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Replace](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SteadyStateReplace.java](#)

6.94. Referencia de la clase

es.ull.esit.app.local_search.complement.StopExecute

Métodos públicos

- Boolean [stopIterations](#) (int countIterationsCurrent, int countmaxIterations)

6.94.1. Descripción detallada

Class that determines whether to stop the execution of local search based on iteration count.

6.94.2. Documentación de funciones miembro

6.94.2.1. stopIterations()

```
Boolean es.ull.esit.app.local_search.complement.StopExecute.stopIterations (
    int countIterationsCurrent,
    int countmaxIterations )
```

Determines if the local search should stop based on the current iteration count and maximum allowed iterations.

Parámetros

<i>countIterationsCurrent</i>	[int] Current iteration count.
<i>countmaxIterations</i>	[int] Maximum allowed iterations.

Devuelve

[Boolean] True if the search should stop, false otherwise.

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/local_search/complement/StopExecute.java](#)

6.95. Referencia de la clase es.ull.esit.app.metaheuristics.strategy.Strategy

Métodos públicos

- void [executeStrategy](#) (int countmaxIterations, int countIterationsChange, int operatornumber, Generator↵ Type generatorType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- void [updateCountGender](#) ()
- void [updateWeight](#) ()
- void [update](#) (Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, Class↵ NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuch↵ MethodException
- Generator [newGenerator](#) (GeneratorType generatortype) throws IllegalArgumentException, Security↵ Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵ Exception, NoSuchMethodException
- State [getBestState](#) ()
- void [setBestState](#) (State bestState)
- StopExecute [getStopexecute](#) ()
- int [getCountMax](#) ()
- void [setCountMax](#) (int countMax)
- void [setStopexecute](#) (StopExecute stopexecute)
- UpdateParameter [getUpdateparameter](#) ()
- void [setUpdateparameter](#) (UpdateParameter updateparameter)
- Problem [getProblem](#) ()
- void [setProblem](#) (Problem problem)
- List< String > [getListKey](#) ()
- void [initializeGenerators](#) () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- void [initialize](#) () throws IllegalArgumentException, SecurityException, ClassNotFoundException, Instantiation↵ Exception, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- int [getCountCurrent](#) ()
- void [setCountCurrent](#) (int countCurrent)
- double [getThreshold](#) ()
- void [setThreshold](#) (double threshold)
- void [calculateOffLinePerformance](#) (float sumMax, int countOff)
- void [updateRef](#) (GeneratorType generatorType)
- void [updateRefMultiG](#) ()
- void [updateRefGenerator](#) (Generator generator)
- SortedMap< GeneratorType, Generator > [getMapGenerators](#) ()
- void [setMapGenerators](#) (SortedMap< GeneratorType, Generator > mapGenerators)
- Generator [getGenerator](#) ()
- void [setGenerator](#) (Generator generator)
- List< State > [getListStates](#) ()
- void [setListStates](#) (List< State > listStates)
- List< State > [getListRefPoblacFinal](#) ()
- void [setListRefPoblacFinal](#) (List< State > listRefPoblacFinal)

Métodos públicos estáticos

- static synchronized [Strategy](#) [getStrategy](#) ()
- static void [destroyExecute](#) ()

6.95.1. Descripción detallada

Singleton class that manages the execution strategy of metaheuristic algorithms.

6.95.2. Documentación de funciones miembro

6.95.2.1. calculateOffLinePerformance()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.calculateOffLinePerformance (
    float sumMax,
    int countOff )
```

Calculates the offline performance.

Parámetros

<i>sumMax</i>	[float] The sum of maximum values.
<i>countOff</i>	[int] The count for offline performance.

6.95.2.2. destroyExecute()

```
static void es.ull.esit.app.metaheuristics.strategy.Strategy.destroyExecute ( ) [static]
```

Destroys the singleton instance and clears references.

6.95.2.3. executeStrategy()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.executeStrategy (
    int countmaxIterations,
    int countIterationsChange,
    int operatornumber,
    GeneratorType generatorType ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
Exception, NoSuchMethodException
```

Executes the strategy with the specified parameters.

Parámetros

<i>countmaxIterations</i>	[int] Maximum number of iterations.
<i>countIterationsChange</i>	[int] Number of iterations before a change occurs.
<i>operatornumber</i>	[int] Number of operators to be used.
<i>generatorType</i>	[GeneratorType] Type of generator to be used.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
---------------------------------	-------------------------------------

Excepciones

<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method is not found.

6.95.2.4. getBestState()

```
State es.ull.esit.app.metaheuristics.strategy.Strategy.getBestState ( )
```

Gets the best state found during the execution.

Devuelve

[State] The best state found.

6.95.2.5. getCountCurrent()

```
int es.ull.esit.app.metaheuristics.strategy.Strategy.getCountCurrent ( )
```

Gets the current count.

Devuelve

[int] The current count.

6.95.2.6. getCountMax()

```
int es.ull.esit.app.metaheuristics.strategy.Strategy.getCountMax ( )
```

Gets the maximum count.

Devuelve

[int] The maximum count.

6.95.2.7. getGenerator()

```
Generator es.ull.esit.app.metaheuristics.strategy.Strategy.getGenerator ( )
```

Gets the current generator.

Devuelve

[Generator] The current generator.

6.95.2.8. getListKey()

```
List< String > es.ull.esit.app.metaheuristics.strategy.Strategy.getListKey ( )
```

Gets a list of keys from the map of generators.

Devuelve

[List<String>] A list of generator type keys.

6.95.2.9. getListRefPoblacFinal()

```
List< State > es.ull.esit.app.metaheuristics.strategy.Strategy.getListRefPoblacFinal ( )
```

Gets the list of reference non-dominated population.

Devuelve

[List<State>] The list of reference non-dominated population.

6.95.2.10. getListStates()

```
List< State > es.ull.esit.app.metaheuristics.strategy.Strategy.getListStates ( )
```

Gets the list of states.

Devuelve

[List<State>] The list of states.

6.95.2.11. getMapGenerators()

```
SortedMap< GeneratorType, Generator > es.ull.esit.app.metaheuristics.strategy.Strategy.getMapGenerators ( )
```

Gets the map of generators.

Devuelve

[SortedMap<GeneratorType, Generator>] The map of generators.

6.95.2.12. getProblem()

```
Problem es.ull.esit.app.metaheuristics.strategy.Strategy.getProblem ( )
```

Gets the Problem instance.

Devuelve

[Problem] The Problem instance.

6.95.2.13. getStopexecute()

```
StopExecute es.ull.esit.app.metaheuristics.strategy.Strategy.getStopexecute ( )
```

Gets the StopExecute instance.

Devuelve

[StopExecute] The StopExecute instance.

6.95.2.14. getStrategy()

```
static synchronized Strategy es.ull.esit.app.metaheuristics.strategy.Strategy.getStrategy ( )  
[static]
```

Gets the singleton instance of the Strategy class.

Devuelve

[Strategy] The singleton instance.

6.95.2.15. getThreshold()

```
double es.ull.esit.app.metaheuristics.strategy.Strategy.getThreshold ( )
```

Gets the threshold value.

Devuelve

[double] The threshold value.

6.95.2.16. getUpdateparameter()

```
UpdateParameter es.ull.esit.app.metaheuristics.strategy.Strategy.getUpdateparameter ( )
```

Gets the UpdateParameter instance.

Devuelve

[UpdateParameter] The UpdateParameter instance.

6.95.2.17. initialize()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.initialize ( ) throws IllegalArgumentException↔  
Exception, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException↔  
Exception, InvocationTargetException, NoSuchMethodException
```

Initializes the strategy.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method cannot be found.

6.95.2.18. initializeGenerators()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.initializeGenerators ( ) throws Illegal←
ArgumentException, SecurityException, ClassNotFoundException, InstantiationException, Illegal←
AccessException, InvocationTargetException, NoSuchMethodException
```

Initializes the generators.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class cannot be found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method cannot be found.

6.95.2.19. newGenerator()

```
Generator es.ull.esit.app.metaheuristics.strategy.Strategy.newGenerator (
    GeneratorType generatortype ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget←
Exception, NoSuchMethodException
```

Creates a new generator based on the specified generator type.

Parámetros

<i>generatortype</i>	[GeneratorType] The type of generator to create.
----------------------	--

Devuelve

[Generator] A new generator instance.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class cannot be found.
<i>InstantiationException</i>	If an instance cannot be created.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method cannot be found.

6.95.2.20. `setBestState()`

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setBestState (
    State bestState )
```

Sets the best state found during the execution.

Parámetros

<i>bestState</i>	[State] The best state to set.
------------------	--------------------------------

6.95.2.21. `setCountCurrent()`

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setCountCurrent (
    int countCurrent )
```

Sets the current count.

Parámetros

<i>countCurrent</i>	[int] The current count to set.
---------------------	---------------------------------

6.95.2.22. `setCountMax()`

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setCountMax (
    int countMax )
```

Sets the maximum count.

Parámetros

<i>countMax</i>	[int] The maximum count to set.
-----------------	---------------------------------

6.95.2.23. setGenerator()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setGenerator (
    Generator generator )
```

Sets the current generator.

Parámetros

<i>generator</i>	[Generator] The current generator to set.
------------------	---

6.95.2.24. setListRefPoblacFinal()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setListRefPoblacFinal (
    List< State > listRefPoblacFinal )
```

Sets the list of reference non-dominated population.

Parámetros

<i>listRefPoblacFinal</i>	[List<State>] The list of reference non-dominated population to set.
---------------------------	--

6.95.2.25. setListStates()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setListStates (
    List< State > listStates )
```

Sets the list of states.

Parámetros

<i>listStates</i>	[List<State>] The list of states to set.
-------------------	--

6.95.2.26. setMapGenerators()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setMapGenerators (
    SortedMap< GeneratorType, Generator > mapGenerators )
```

Sets the map of generators.

Parámetros

<i>mapGenerators</i>	[SortedMap<GeneratorType, Generator>] The map of generators to set.
----------------------	---

6.95.2.27. `setProblem()`

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setProblem (
    Problem problem )
```

Sets the Problem instance.

Parámetros

<i>problem</i>	[Problem] The Problem instance to set.
----------------	--

6.95.2.28. `setStopexecute()`

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setStopexecute (
    StopExecute stopexecute )
```

Sets the StopExecute instance.

Parámetros

<i>stopexecute</i>	[StopExecute] The StopExecute instance to set.
--------------------	--

6.95.2.29. `setThreshold()`

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setThreshold (
    double threshold )
```

Sets the threshold value.

Parámetros

<i>threshold</i>	[double] The threshold value to set.
------------------	--------------------------------------

6.95.2.30. `setUpdateparameter()`

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.setUpdateparameter (
    UpdateParameter updateparameter )
```

Sets the UpdateParameter instance.

Parámetros

<i>updateparameter</i>	[UpdateParameter] The UpdateParameter instance to set.
------------------------	--

6.95.2.31. update()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.update (
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityEx←
ception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocatio←
TargetException, NoSuchMethodException
```

Updates the generator based on the current iteration count.

Parámetros

<i>countIterationsCurrent</i>	[Integer] Current iteration count.
-------------------------------	------------------------------------

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class cannot be found.
<i>InstantiationException</i>	If an instance cannot be created.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method cannot be found.

6.95.2.32. updateCountGender()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.updateCountGender ( )
```

Updates the count of improvements and usage for each generator over a given period.

6.95.2.33. updateRef()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.updateRef (
    GeneratorType generatorType )
```

Updates the reference state based on the generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] The type of generator.
----------------------	--

6.95.2.34. updateRefGenerator()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.updateRefGenerator (
    Generator generator )
```

Updates the reference state for a specific generator.

Parámetros

<i>generator</i>	[Generator] The generator to update.
------------------	--------------------------------------

6.95.2.35. updateRefMultiG()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.updateRefMultiG ( )
```

Updates the reference states for all generators in the multi-generator setup.

6.95.2.36. updateWeight()

```
void es.ull.esit.app.metaheuristics.strategy.Strategy.updateWeight ( )
```

Updates the weights of the generators in the multi-generator setup.

La documentación de esta clase está generada del siguiente archivo:

- src/main/java/es/ull/esit/app/metaheuristics/strategy/[Strategy.java](#)

6.96. Referencia de la clase es.ull.esit.app.strategy.Strategy**Métodos públicos**

- void [executeStrategy](#) (int countmaxIterations, int countIterationsChange, int operatornumber, Generator↵ Type generatorType) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- void [updateCountGender](#) ()
- void [updateWeight](#) ()
- void [update](#) (Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, Class↵ NotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuch↵ MethodException
- Generator [newGenerator](#) (GeneratorType generatorType) throws IllegalArgumentException, Security↵ Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTarget↵ Exception, NoSuchMethodException
- State [getBestState](#) ()
- void [setBestState](#) (State besState)
- StopExecute [getStopexecute](#) ()
- int [getCountMax](#) ()
- void [setCountMax](#) (int countMax)
- void [setStopexecute](#) (StopExecute stopexecute)
- UpdateParameter [getUpdateparameter](#) ()
- void [setUpdateparameter](#) (UpdateParameter updateparameter)
- Problem [getProblem](#) ()
- void [setProblem](#) (Problem problem)
- List< String > [getListKey](#) ()
- void [initializeGenerators](#) () throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- void [initialize](#) () throws IllegalArgumentException, SecurityException, ClassNotFoundException, Instantiation↵ Exception, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- int [getCountCurrent](#) ()
- void [setCountCurrent](#) (int countCurrent)
- double [getThreshold](#) ()
- void [setThreshold](#) (double threshold)
- void [calculateOffLinePerformance](#) (float sumMax, int countOff)
- void [updateRef](#) (GeneratorType generatorType)
- void [updateRefMultiG](#) ()
- void [updateRefGenerator](#) (Generator generator)

Métodos públicos estáticos

- static synchronized [Strategy getStrategy \(\)](#)
- static void [destroyExecute \(\)](#)

Atributos públicos

- Generator [generator](#)
- double [threshold](#)
- Dominance [notDominated](#)

6.96.1. Descripción detallada

Singleton class that manages the execution strategy of metaheuristic algorithms.

6.96.2. Documentación de funciones miembro

6.96.2.1. calculateOffLinePerformance()

```
void es.ull.esit.app.strategy.Strategy.calculateOffLinePerformance (
    float sumMax,
    int countOff )
```

Calculates the offline performance metric.

Parámetros

<i>sumMax</i>	[float] Sum of maximum evaluations.
<i>countOff</i>	[int] Count of offline evaluations.

6.96.2.2. destroyExecute()

```
static void es.ull.esit.app.strategy.Strategy.destroyExecute ( ) [static]
```

Destroys the singleton instance of the Strategy class and resets references.

6.96.2.3. executeStrategy()

```
void es.ull.esit.app.strategy.Strategy.executeStrategy (
    int countmaxIterations,
    int countIterationsChange,
    int operatornumber,
    GeneratorType generatorType ) throws IllegalArgumentException, SecurityException,
ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
Exception, NoSuchMethodException
```

Executes the strategy with the specified parameters.

Parámetros

<i>countmaxIterations</i>	[int] Maximum number of iterations
<i>countIterationsChange</i>	[int] Number of iterations before a change occurs
<i>operatornumber</i>	[int] Operator number
<i>generatorType</i>	[GeneratorType] Type of generator to use

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a required class is not found.
<i>InstantiationException</i>	If an error occurs during instantiation.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an error occurs during method invocation.
<i>NoSuchMethodException</i>	If a required method is not found.

6.96.2.4. `getBestState()`

```
State es.ull.esit.app.strategy.Strategy.getBestState ( )
```

Gets the best state found during the execution.

Devuelve

[State] Best state found.

6.96.2.5. `getCountCurrent()`

```
int es.ull.esit.app.strategy.Strategy.getCountCurrent ( )
```

Gets the current iteration count.

Devuelve

[int] Current iteration count.

6.96.2.6. `getCountMax()`

```
int es.ull.esit.app.strategy.Strategy.getCountMax ( )
```

Gets the maximum iteration count.

Devuelve

[int] Maximum iteration count.

6.96.2.7. getListKey()

```
List< String > es.ull.esit.app.strategy.Strategy.getListKey ( )
```

Gets the list of generator keys.

Devuelve

[ArrayList<String>] List of generator keys.

6.96.2.8. getProblem()

```
Problem es.ull.esit.app.strategy.Strategy.getProblem ( )
```

Gets the problem instance associated with the strategy.

Devuelve

[Problem] Problem instance.

6.96.2.9. getStopexecute()

```
StopExecute es.ull.esit.app.strategy.Strategy.getStopexecute ( )
```

Gets the stop execution criteria.

Devuelve

[StopExecute] Stop execution criteria.

6.96.2.10. getStrategy()

```
static synchronized Strategy es.ull.esit.app.strategy.Strategy.getStrategy ( ) [static]
```

Method to get the singleton instance of the Strategy class

6.96.2.11. getThreshold()

```
double es.ull.esit.app.strategy.Strategy.getThreshold ( )
```

Gets the threshold value for the strategy.

Devuelve

[double] Threshold value.

6.96.2.12. getUpdateparameter()

```
UpdateParameter es.ull.esit.app.strategy.Strategy.getUpdateparameter ( )
```

Gets the update parameter strategy.

Devuelve

[UpdateParameter] Update parameter strategy.

6.96.2.13. initialize()

```
void es.ull.esit.app.strategy.Strategy.initialize ( ) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Initializes the generators used in the strategy.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a class is not found.
<i>InstantiationException</i>	If an error occurs during instantiation.
<i>IllegalAccessException</i>	If an access to a class or method is denied.
<i>InvocationTargetException</i>	If an error occurs during method invocation.
<i>NoSuchMethodException</i>	If a required method is not found.

6.96.2.14. initializeGenerators()

```
void es.ull.esit.app.strategy.Strategy.initializeGenerators ( ) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
```

Initializes the generators used in the strategy.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a required class is not found.
<i>InstantiationException</i>	If an error occurs during instantiation.
<i>IllegalAccessException</i>	If an access to a class or method is denied.
<i>InvocationTargetException</i>	If an error occurs during method invocation.
<i>NoSuchMethodException</i>	If a required method is not found.

6.96.2.15. newGenerator()

```
Generator es.ull.esit.app.strategy.Strategy.newGenerator (
    GeneratorType generatorType ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Creates a new generator based on the specified type.

Parámetros

<i>GeneratorType</i>	[GeneratorType] Type of generator to create.
----------------------	--

Devuelve

[Generator] Newly created generator instance.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a required class is not found.
<i>InstantiationException</i>	If an error occurs during instantiation.
<i>IllegalAccessException</i>	If an access to a class or method is denied.
<i>InvocationTargetException</i>	If an error occurs during method invocation.
<i>NoSuchMethodException</i>	If a required method is not found.

6.96.2.16. setBestState()

```
void es.ull.esit.app.strategy.Strategy.setBestState (
    State bestState )
```

Sets the best state found during the execution.

Parámetros

<i>bestState</i>	[State] Best state to set.
------------------	----------------------------

6.96.2.17. setCountCurrent()

```
void es.ull.esit.app.strategy.Strategy.setCountCurrent (
    int countCurrent )
```

Sets the current iteration count.

Parámetros

<i>countCurrent</i>	[int] Current iteration count.
---------------------	--------------------------------

6.96.2.18. `setCountMax()`

```
void es.ull.esit.app.strategy.Strategy.setCountMax (
    int countMax )
```

Sets the maximum iteration count.

Parámetros

<i>countMax</i>	[int] Maximum iteration count.
-----------------	--------------------------------

6.96.2.19. `setProblem()`

```
void es.ull.esit.app.strategy.Strategy.setProblem (
    Problem problem )
```

Sets the problem instance associated with the strategy.

Parámetros

<i>problem</i>	[Problem] Problem instance.
----------------	-----------------------------

6.96.2.20. `setStopexecute()`

```
void es.ull.esit.app.strategy.Strategy.setStopexecute (
    StopExecute stopexecute )
```

Sets the stop execution criteria.

Parámetros

<i>stopexecute</i>	[StopExecute] Stop execution criteria.
--------------------	--

6.96.2.21. `setThreshold()`

```
void es.ull.esit.app.strategy.Strategy.setThreshold (
    double threshold )
```

Sets the threshold value for the strategy.

Parámetros

<i>threshold</i>	[double] Threshold value.
------------------	---------------------------

6.96.2.22. setUpdateparameter()

```
void es.ull.esit.app.strategy.Strategy.setUpdateparameter (
    UpdateParameter updateparameter )
```

Sets the update parameter strategy.

Parámetros

<i>updateparameter</i>	[UpdateParameter] Update parameter strategy.
------------------------	--

6.96.2.23. update()

```
void es.ull.esit.app.strategy.Strategy.update (
    Integer countIterationsCurrent ) throws IllegalArgumentException, Security↵
Exception, ClassNotFoundException, InstantiationException, IllegalAccessException, Invocation↵
TargetException, NoSuchMethodException
```

Updates the strategy based on the current iteration count.

Parámetros

<i>countIterationsCurrent</i>	[Integer] Current iteration count.
-------------------------------	------------------------------------

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If a required class is not found.
<i>InstantiationException</i>	If an error occurs during instantiation.
<i>IllegalAccessException</i>	If an access to a class or method is denied.
<i>InvocationTargetException</i>	If an error occurs during method invocation.
<i>NoSuchMethodException</i>	If a required method is not found.

6.96.2.24. updateCountGender()

```
void es.ull.esit.app.strategy.Strategy.updateCountGender ( )
```

Updates the count of genders for each generator in MultiGenerator.

6.96.2.25. updateRef()

```
void es.ull.esit.app.strategy.Strategy.updateRef (
    GeneratorType generatorType )
```

Updates the reference states based on the generator type.

Parámetros

<i>generatorType</i>	[GeneratorType] Type of generator.
----------------------	------------------------------------

6.96.2.26. updateRefGenerator()

```
void es.ull.esit.app.strategy.Strategy.updateRefGenerator (
    Generator generator )
```

Updates the reference state(s) for the specified generator.

Parámetros

<i>generator</i>	[Generator] The generator whose reference state(s) will be updated.
------------------	---

6.96.2.27. updateRefMultiG()

```
void es.ull.esit.app.strategy.Strategy.updateRefMultiG ( )
```

Updates the reference states for all generators in MultiGenerator.

6.96.2.28. updateWeight()

```
void es.ull.esit.app.strategy.Strategy.updateWeight ( )
```

Updates the weights of each generator in MultiGenerator.

6.96.3. Documentación de datos miembro**6.96.3.1. generator**

```
Generator es.ull.esit.app.strategy.Strategy.generator
```

Generator used in the strategy

6.96.3.2. notDominated

```
Dominance es.ull.esit.app.strategy.Strategy.notDominated
```

Dominance criteria

6.96.3.3. threshold

```
double es.ull.esit.app.strategy.Strategy.threshold
```

Threshold value for the strategy

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/strategy/Strategy.java](#)

6.97. Referencia de la enumeración es.ull.esit.app.local_search.complement.StrategyType

Atributos públicos

- [TABU](#)
- [NORMAL](#)

6.97.1. Descripción detallada

Enumeration representing different strategy types for local search algorithms.

6.97.2. Documentación de datos miembro

6.97.2.1. NORMAL

```
es.ull.esit.app.local_search.complement.StrategyType.NORMAL
```

Tabu Search strategy

6.97.2.2. TABU

```
es.ull.esit.app.local_search.complement.StrategyType.TABU
```

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/local_search/complement/StrategyType.java](#)

6.98. Referencia de la clase es.ull.esit.app.metaheuristics.generators.TabuSearch

Diagrama de herencia de es.ull.esit.app.metaheuristics.generators.TabuSearch

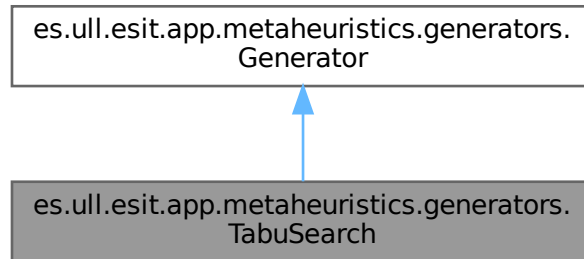
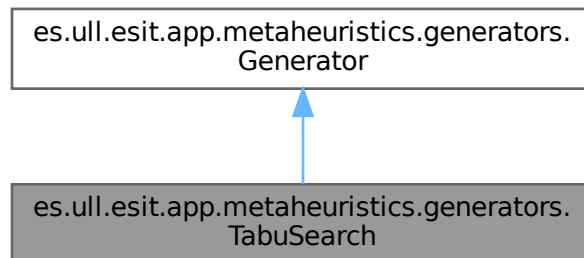


Diagrama de colaboración de es.ull.esit.app.metaheuristics.generators.TabuSearch:



Métodos públicos

- [TabuSearch](#) ()
- State [generate](#) (Integer operatornumber) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- State [getReference](#) ()
- void [setInitialReference](#) (State stateInitialRef)
- void [setStateRef](#) (State stateRef)
- void [updateReference](#) (State stateCandidate, Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException
- [GeneratorType](#) [getType](#) ()
- [GeneratorType](#) [getTypeGenerator](#) ()

- void [setTypeGenerator](#) ([GeneratorType](#) typeGenerator)
- List< State > [getReferenceList](#) ()
- List< State > [getSonList](#) ()
- void [setTypeCandidate](#) (CandidateType typeCandidate)
- boolean [awardUpdateREF](#) (State stateCandidate)
- float [getWeight](#) ()
- void [setWeight](#) (float weight)
- int[] [getListCountBetterGender](#) ()
- int[] [getListCountGender](#) ()
- float[] [getTrace](#) ()

Otros miembros heredados

Atributos públicos heredados de [es.ull.esit.app.metaheuristics.generators.Generator](#)

- int [countGender](#)
- int [countBetterGender](#)
- int[] [listCountBetterGender](#)

6.98.1. Descripción detallada

Class that implements a classic Tabu Search generator.

It maintains:

- A current reference solution.
- A candidate selection rule.
- A Tabu list of forbidden solutions (via [TabuSolutions](#)).

6.98.2. Documentación de constructores y destructores

6.98.2.1. [TabuSearch\(\)](#)

```
es.ull.esit.app.metaheuristics.generators.TabuSearch.TabuSearch ( )
```

Default constructor.

It initialises:

- Acceptance: [AcceptType](#)#[AcceptAnyone](#)
- Strategy: [StrategyType](#)#[TABU](#)
- Candidate type: Greater/Smaller according to the problem type ([MAXIMIZAR](#) -> [GreaterCandidate](#), else [SmallerCandidate](#))

6.98.3. Documentación de funciones miembro

6.98.3.1. [awardUpdateREF\(\)](#)

```
boolean es.ull.esit.app.metaheuristics.generators.TabuSearch.awardUpdateREF (
    State stateCandidate )
```

For Tabu Search we do not use an additional “award” criterion, beyond the acceptance rule, so this always returns false.

Parámetros

<code>stateCandidate</code>	[State] Candidate state.
-----------------------------	--------------------------

Devuelve

[boolean] Always false.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.2. generate()

```
State es.ull.esit.app.metaheuristics.generators.TabuSearch.generate (
    Integer operatornumber ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Generates a new candidate solution from the current reference using the neighbourhood defined by the operator and the configured candidate selection rule.

Parámetros

<code>operatornumber</code>	[Integer] Operator index.
-----------------------------	---------------------------

Devuelve

[State] Generated candidate state, or `null` if context is missing.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.3. getListCountBetterGender()

```
int[] es.ull.esit.app.metaheuristics.generators.TabuSearch.getListCountBetterGender ( )
```

Returns the internal “better gender” statistics array.

Devuelve

[int[]] Better gender statistics.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.4. getListCountGender()

```
int[] es.ull.esit.app.metaheuristics.generators.TabuSearch.getListCountGender ( )
```

Returns the internal gender statistics array.

Devuelve

[int[]] Gender statistics.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.5. `getReference()`

```
State es.ull.esit.app.metaheuristics.generators.TabuSearch.getReference ( )
```

Returns the current reference state.

Devuelve

[State] Current reference state.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.6. `getReferenceList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.TabuSearch.getReferenceList ( )
```

Returns the list of reference states visited so far. Each call appends the current reference state if not null.

Devuelve

[List<State>] List of reference states.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.7. `getSonList()`

```
List< State > es.ull.esit.app.metaheuristics.generators.TabuSearch.getSonList ( )
```

Tabu Search does not explicitly maintain a son list. We return an empty list to avoid null checks.

Devuelve

[List<State>] Empty list.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.8. `getTrace()`

```
float[] es.ull.esit.app.metaheuristics.generators.TabuSearch.getTrace ( )
```

Returns the trace of weight values.

Devuelve

[float[]] Weight trace.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.9. `getType()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.TabuSearch.getType ( )
```

Returns the generator type.

Devuelve

[GeneratorType] Generator type.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.10. `getTypeGenerator()`

```
GeneratorType es.ull.esit.app.metaheuristics.generators.TabuSearch.getTypeGenerator ( )
```

Explicit generator type getter (for tests / configuration).

Devuelve

[GeneratorType] Generator type.

6.98.3.11. `getWeight()`

```
float es.ull.esit.app.metaheuristics.generators.TabuSearch.getWeight ( )
```

Returns the weight of this generator.

Devuelve

[float] Weight value.

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.12. `setInitialReference()`

```
void es.ull.esit.app.metaheuristics.generators.TabuSearch.setInitialReference (
    State stateInitialRef )
```

Sets the initial reference state for Tabu Search.

Parámetros

<code>stateInitialRef</code>	[State] Initial reference state.
------------------------------	----------------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.13. setStateRef()

```
void es.ull.esit.app.metaheuristics.generators.TabuSearch.setStateRef (
    State stateRef )
```

Sets the current reference state (explicit setter, useful in tests).

Parámetros

<i>stateRef</i>	[State] New reference state.
-----------------	------------------------------

6.98.3.14. setTypeCandidate()

```
void es.ull.esit.app.metaheuristics.generators.TabuSearch.setTypeCandidate (
    CandidateType typeCandidate )
```

Sets the candidate selection rule.

Parámetros

<i>typeCandidate</i>	[CandidateType] New candidate selection type.
----------------------	---

6.98.3.15. setTypeGenerator()

```
void es.ull.esit.app.metaheuristics.generators.TabuSearch.setTypeGenerator (
    GeneratorType typeGenerator )
```

Explicit generator type setter (for tests / configuration).

Parámetros

<i>typeGenerator</i>	[GeneratorType] New generator type.
----------------------	-------------------------------------

6.98.3.16. setWeight()

```
void es.ull.esit.app.metaheuristics.generators.TabuSearch.setWeight (
    float weight )
```

Sets the weight of this generator.

Parámetros

<i>weight</i>	[float] New weight value.
---------------	---------------------------

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

6.98.3.17. updateReference()

```
void es.ull.esit.app.metaheuristics.generators.TabuSearch.updateReference (
    State stateCandidate,
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException
```

Updates the reference state according to the acceptance rule and maintains the Tabu list:

- If the candidate is accepted, the reference state is updated.
- If strategy is TABU and the candidate is accepted, it is inserted into the Tabu list (unless already present), removing the oldest element when the list is full.

Parámetros

<i>stateCandidate</i>	[State] Candidate state.
<i>countIterationsCurrent</i>	[Integer] Current iteration (not used directly here).

Reimplementado de [es.ull.esit.app.metaheuristics.generators.Generator](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/metaheuristics/generators/TabuSearch.java`

6.99. Referencia de la clase `es.ull.esit.app.local_search.complement.TabuSolutions`

Métodos públicos

- `List< State > filterNeighborhood (List< State > listNeighborhood)`

Atributos públicos estáticos

- `static final List< State > listTabu = new ArrayList<>()`
- `static int maxelements = 100`

6.99.1. Descripción detallada

Class that manages the tabu list for Tabu Search in local search algorithms.

6.99.2. Documentación de funciones miembro**6.99.2.1. filterNeighborhood()**

```
List< State > es.ull.esit.app.local_search.complement.TabuSolutions.filterNeighborhood (
    List< State > listNeighborhood )
```

Filters the neighborhood states by removing those present in the tabu list.

Parámetros

<i>listNeighborhood</i>	[List<State>] List of neighborhood states.
-------------------------	--

Devuelve

[List<State>] Filtered list of neighborhood states.

6.99.3. Documentación de datos miembro**6.99.3.1. listTabu**

```
final List<State> es.ull.esit.app.local_search.complement.TabuSolutions.listTabu = new Array<↵
List<>() [static]
```

List of tabu states (accesible desde otros paquetes como MultiobjectiveTabuSearch y TabuSearch).

6.99.3.2. maxelements

```
int es.ull.esit.app.local_search.complement.TabuSolutions.maxelements = 100 [static]
```

Maximum number of elements allowed in the tabu list (nombre esperado: maxelements).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/local_search/complement/TabuSolutions.java](#)

6.100. Referencia de la clase es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation

Diagrama de herencia de es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation

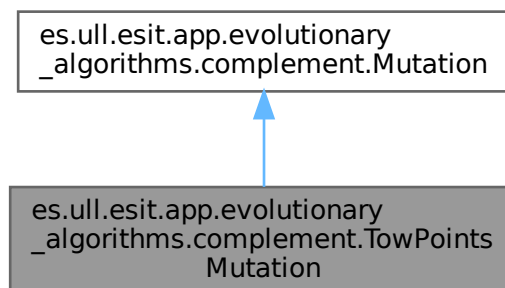
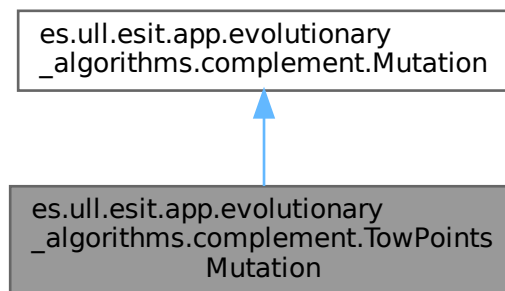


Diagrama de colaboración de `es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation`:



Métodos públicos

- State [mutation](#) (State *newind*, double *pm*)

6.100.1. Descripción detallada

Class that represents a two points mutation operator.

6.100.2. Documentación de funciones miembro

6.100.2.1. `mutation()`

```

State es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation.mutation (
    State newind,
    double pm )
  
```

Applies the two points mutation to the given individual with a certain probability.

Parámetros

<i>newind</i>	[State] The individual to be mutated.
<i>pm</i>	[double] The mutation probability.

Devuelve

[State] The mutated individual.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Mutation](#).

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/TowPointsMutation.java`

6.101. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection`.↩

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection`

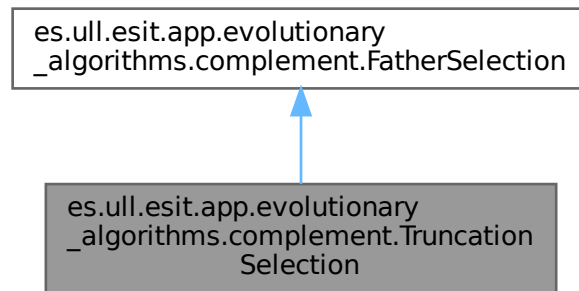
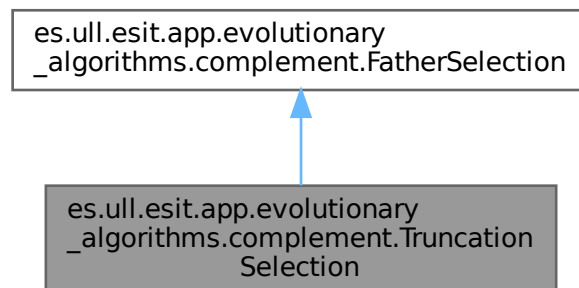


Diagrama de colaboración de `es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection`:



Métodos públicos

- `List< State > orderBetter (List< State > listState)`
- `List< State > ascOrderBetter (List< State > listState)`
- `List< State > selection (List< State > listState, int truncation)`

6.101.1. Descripción detallada

Class that represents the Truncation Selection method.

6.101.2. Documentación de funciones miembro

6.101.2.1. ascOrderBetter()

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection.asc↵
OrderBetter (
    List< State > listState )
```

Orders the list of states from worst to best.

Parámetros

<i>listState</i>	[List<State>] List of states to be ordered.
------------------	---

Devuelve

[List<State>] Ordered list of states.

6.101.2.2. orderBetter()

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection.order↵
Better (
    List< State > listState )
```

Orders the list of states from best to worst.

Parámetros

<i>listState</i>	[List<State>] List of states to be ordered.
------------------	---

Devuelve

[List<State>] Ordered list of states.

6.101.2.3. selection()

```
List< State > es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection.selection
(
    List< State > listState,
    int truncation )
```

Selects the best states from the list according to the truncation value.

Parámetros

<i>listState</i>	[List<State>] List of states to select from.
<i>truncation</i>	[int] Number of states to select.

Devuelve

[List<State>] Selected list of states.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection](#).

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/TruncationSelection.java](#)

6.102. Referencia de la clase [es.ull.esit.app.config.tspdynamic.TSPState](#)

Métodos públicos

- [int](#) [getValue](#) ()
- [void](#) [setValue](#) (int value)
- [int](#) [getIdCity](#) ()
- [void](#) [setIdCity](#) (int idCity)

6.102.1. Descripción detallada

Class that represents the state of the TSP problem.

6.102.2. Documentación de funciones miembro

6.102.2.1. [getIdCity\(\)](#)

```
int es.ull.esit.app.config.tspdynamic.TSPState.getIdCity ( )
```

Gets the identifier of the city associated with the state.

Devuelve

[int] the identifier of the city.

6.102.2.2. [getValue\(\)](#)

```
int es.ull.esit.app.config.tspdynamic.TSPState.getValue ( )
```

Gets the value associated with the state.

Devuelve

[int] the value.

6.102.2.3. [setIdCity\(\)](#)

```
void es.ull.esit.app.config.tspdynamic.TSPState.setIdCity (
    int idCity )
```

Sets the identifier of the city associated with the state.

Parámetros

<i>idCity</i>	[int] the identifier of the city to set.
---------------	--

6.102.2.4. setValue()

```
void es.ull.esit.app.config.tspdynamictspdynamic.TSPState.setValue (
    int value )
```

Sets the value associated with the state.

Parámetros

<i>value</i>	[int] the value to set.
--------------	-------------------------

La documentación de esta clase está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/config/tspdynamictspdynamic/TSPState.java](#)

6.103. Referencia de la enumeración `es.ull.esit.app.problem.extension.TypeSolutionMethod`

Atributos públicos

- [FactoresPonderados](#)
- [MultiObjetivoPuro](#)

6.103.1. Documentación de datos miembro**6.103.1.1. FactoresPonderados**

```
es.ull.esit.app.problem.extension.TypeSolutionMethod.FactoresPonderados
```

6.103.1.2. MultiObjetivoPuro

```
es.ull.esit.app.problem.extension.TypeSolutionMethod.MultiObjetivoPuro
```

Weighted Sum Method

La documentación de esta enumeración está generada del siguiente archivo:

- [src/main/java/es/ull/esit/app/problem/extension/TypeSolutionMethod.java](#)

6.104. Referencia de la clase `es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover`.↩

Diagrama de herencia de `es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover`

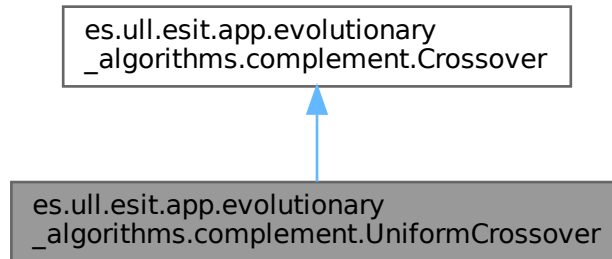
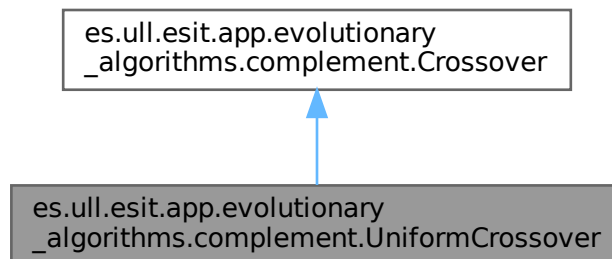


Diagrama de colaboración de `es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover`:



Métodos públicos

- `int[] mascara` (int length)
- `State crossover` (State father1, State father2, double pc)

6.104.1. Descripción detallada

Class that represents the Uniform Crossover method.

6.104.2. Documentación de funciones miembro

6.104.2.1. `crossover()`

```
State es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover.crossover (
    State father1,
    State father2,
    double pc )
```

Performs uniform crossover between two father states.

Parámetros

<i>father1</i>	[State] First father state.
<i>father2</i>	[State] Second father state.
<i>pc</i>	[double] Crossover probability.

Devuelve

[State] New state resulting from the crossover.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Crossover](#).

6.104.2.2. `maskara()`

```
int[] es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover.maskara (
    int length )
```

Generates a random binary mask of given length.

Parámetros

<i>length</i>	[int] Length of the mask.
---------------	---------------------------

Devuelve

[int[]] Binary mask array.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/UniformCrossover.java`

6.105. Referencia de la clase

es.ull.esit.app.evolutionary_algorithms.complement.Univariate

Diagrama de herencia de es.ull.esit.app.evolutionary_algorithms.complement.Univariate

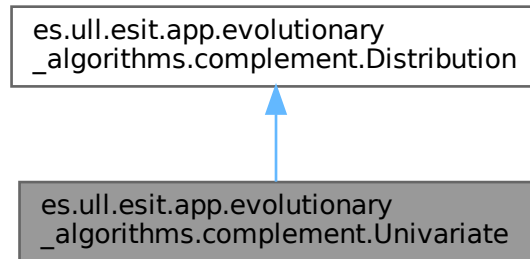
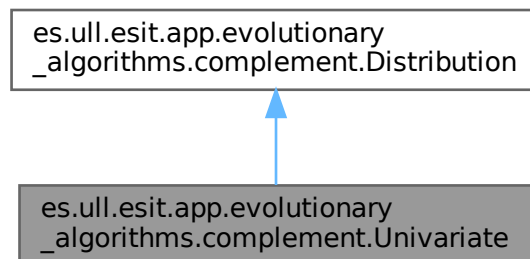


Diagrama de colaboración de es.ull.esit.app.evolutionary_algorithms.complement.Univariate:



Métodos públicos

- List< [Probability](#) > [distribution](#) (List< State > fathers)
- List< String > [getListKey](#) (SortedMap< String, Object > map)

6.105.1. Descripción detallada

Class that represents the Univariate Distribution method.

6.105.2. Documentación de funciones miembro

6.105.2.1. `distribution()`

```
List< Probability > es.ull.esit.app.evolutionary_algorithms.complement.Univariate.distribution
(
    List< State > fathers )
```

Calculates the univariate distribution of variable values from a list of father states.

Parámetros

<i>fathers</i>	[List<State>] List of father states.
----------------	--------------------------------------

Devuelve

[List<Probability>] List of Probability objects representing the distribution.

Reimplementado de [es.ull.esit.app.evolutionary_algorithms.complement.Distribution](#).

6.105.2.2. `getListKey()`

```
List< String > es.ull.esit.app.evolutionary_algorithms.complement.Univariate.getListKey (
    SortedMap< String, Object > map )
```

Generates a list of keys from a sorted map.

Parámetros

<i>map</i>	[SortedMap<String, Object>] Input sorted map.
------------	---

Devuelve

[List<String>] List of keys extracted from the map.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Univariate.java`

6.106. Referencia de la clase `es.ull.esit.app.local_search.complement.UpdateParameter`

Métodos públicos estáticos

- static Integer [updateParameter](#) (Integer countIterationsCurrent) throws IllegalArgumentException, SecurityException, ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException, NoSuchMethodException

6.106.1. Descripción detallada

Class responsible for updating parameters during the local search process.

6.106.2. Documentación de funciones miembro

6.106.2.1. updateParameter()

```
static Integer es.ull.esit.app.local_search.complement.UpdateParameter.updateParameter (
    Integer countIterationsCurrent ) throws IllegalArgumentException, SecurityException,
    ClassNotFoundException, InstantiationException, IllegalAccessException, InvocationTargetException,
    NoSuchMethodException [static]
```

Updates the current iteration count and changes the generator based on predefined reference counts.

Parámetros

<i>countIterationsCurrent</i>	[Integer] Current iteration count.
-------------------------------	------------------------------------

Devuelve

[Integer] Updated iteration count.

Excepciones

<i>IllegalArgumentException</i>	If an illegal argument is provided.
<i>SecurityException</i>	If a security violation occurs.
<i>ClassNotFoundException</i>	If the class is not found.
<i>InstantiationException</i>	If an instantiation error occurs.
<i>IllegalAccessException</i>	If access to a class or method is denied.
<i>InvocationTargetException</i>	If an exception occurs during method invocation.
<i>NoSuchMethodException</i>	If a method is not found.

La documentación de esta clase está generada del siguiente archivo:

- `src/main/java/es/ull/esit/app/local_search/complement/UpdateParameter.java`

Capítulo 7

Documentación de archivos

7.1. Referencia del archivo

src/main/java/es/ull/esit/app/config/tspdynamic/TSPState.java

Clases

- class [es.ull.esit.app.config.tspdynamic.TSPState](#)

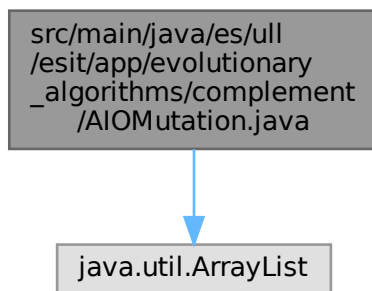
Paquetes

- package [es.ull.esit.app.config.tspdynamic](#)

7.2. Referencia del archivo **src/main/java/es/ull/esit/app/evolutionary_↵ algorithms/complement/AIOMutation.java**

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en AIOMutation.java:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation](#)

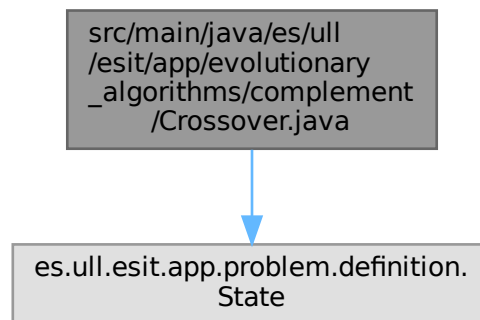
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.3. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Crossover.java`

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en `Crossover.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.Crossover](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.4. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/CrossoverType.java`

Clases

- enum [es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType](#)

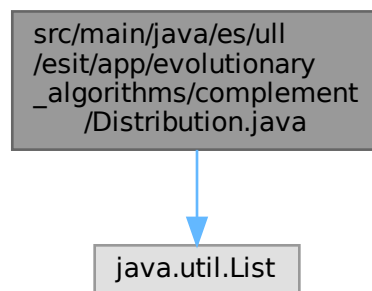
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.5. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Distribution.java ↩

```
import java.util.List;
```

Gráfico de dependencias incluidas en Distribution.java:

**Clases**

- class [es.ull.esit.app.evolutionary_algorithms.complement.Distribution](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.6. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/DistributionType.java ↩

Clases

- enum [es.ull.esit.app.evolutionary_algorithms.complement.DistributionType](#)

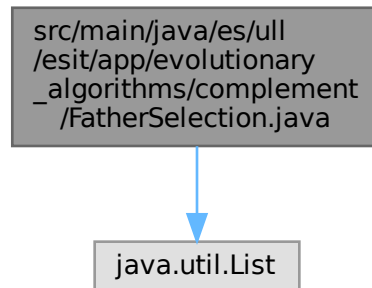
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.7. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/FatherSelection.java`

```
import java.util.List;
```

Gráfico de dependencias incluidas en `FatherSelection.java`:



Clases

- class `es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection`

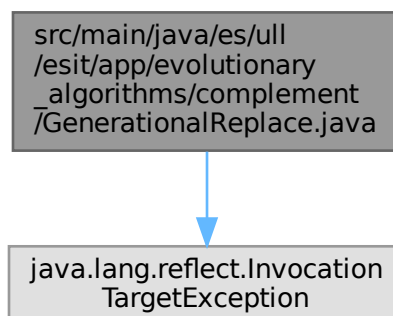
Paquetes

- package `es.ull.esit.app.evolutionary_algorithms.complement`

7.8. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/GenerationalReplace.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `GenerationalReplace.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace](#)

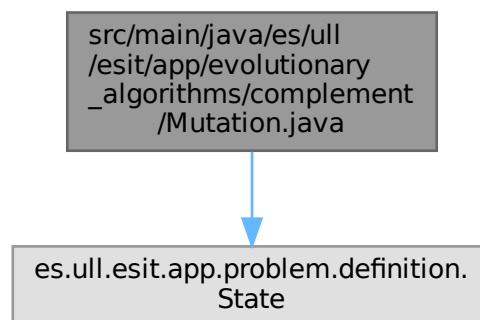
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.9. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation.java ↩

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en Mutation.java:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.Mutation](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.10. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/MutationType.java ↩

Clases

- enum [es.ull.esit.app.evolutionary_algorithms.complement.MutationType](#)

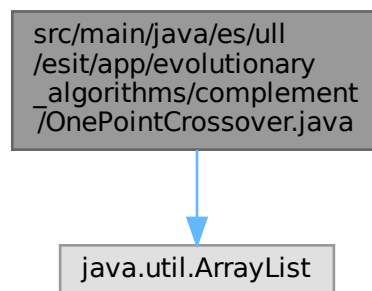
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.11. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointCrossover.java`

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en `OnePointCrossover.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover](#)

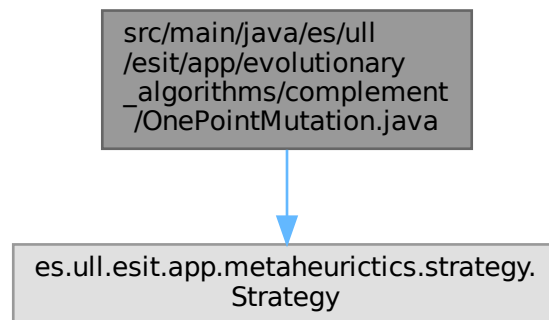
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.12. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointMutation.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en OnePointMutation.java:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation](#)

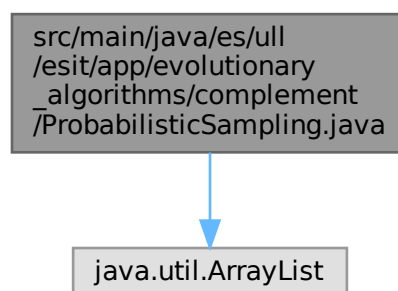
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.13. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/ProbabilisticSampling.java

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en ProbabilisticSampling.java:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.14. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Probability.java`

Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.Probability](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.15. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Range.java`

Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.Range](#)

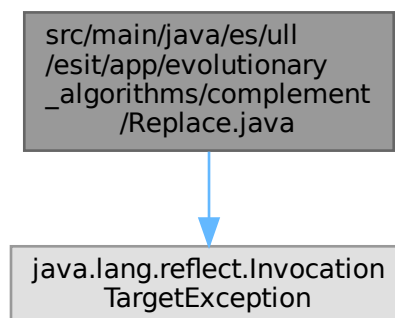
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.16. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replace.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `Replace.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.Replace](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.17. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/ReplaceType.java

Clases

- enum [es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType](#)

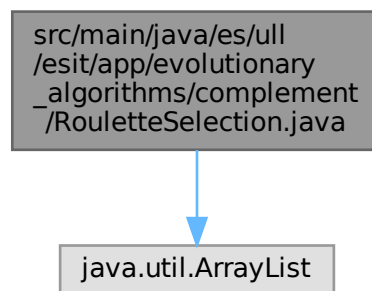
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.18. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/RouletteSelection.java

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en RouletteSelection.java:

**Clases**

- class [es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection](#)

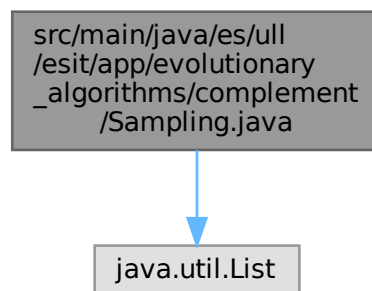
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.19. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Sampling.java`

```
import java.util.List;
```

Gráfico de dependencias incluidas en `Sampling.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.Sampling](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.20. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SamplingType.java`

Clases

- enum [es.ull.esit.app.evolutionary_algorithms.complement.SamplingType](#)

Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.21. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SelectionType.java

Clases

- enum [es.ull.esit.app.evolutionary_algorithms.complement.SelectionType](#)

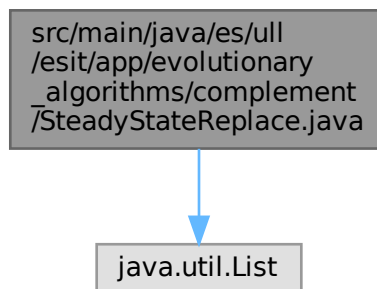
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.22. Referencia del archivo src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/SteadyStateReplace.java

```
import java.util.List;
```

Gráfico de dependencias incluidas en SteadyStateReplace.java:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace](#)

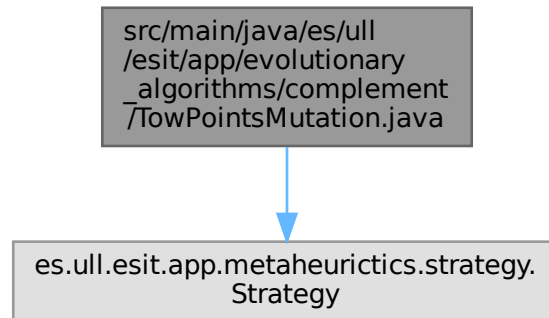
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.23. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/TowPointsMutation.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `TowPointsMutation.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation](#)

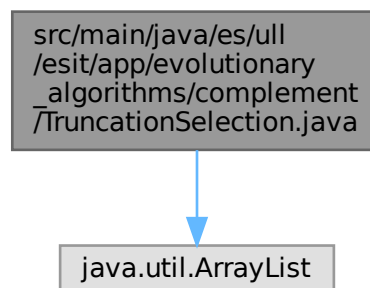
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.24. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/TruncationSelection.java`

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en `TruncationSelection.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection](#)

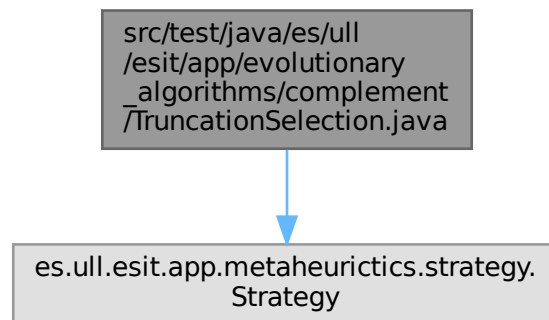
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.25. Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/TruncationSelection.java ↩

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en TruncationSelection.java:

**Clases**

- class [es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelectionTest](#)

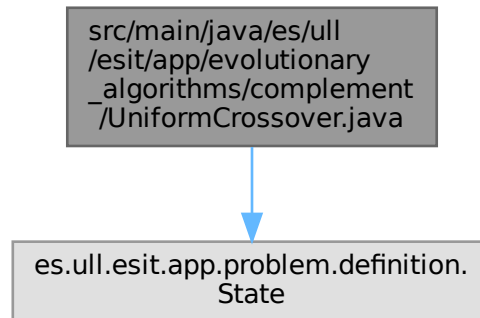
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.26. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/UniformCrossover.java`

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en `UniformCrossover.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover](#)

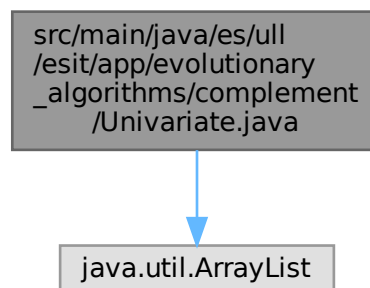
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.27. Referencia del archivo `src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Univariate.java`

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en `Univariate.java`:



Clases

- class [es.ull.esit.app.evolutionary_algorithms.complement.Univariate](#)

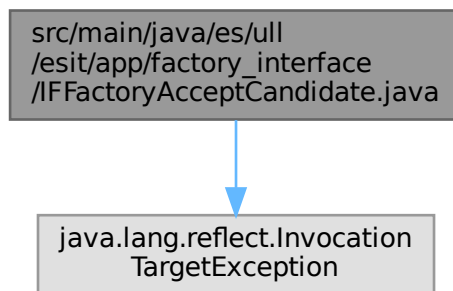
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.28. Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFactoryAcceptCandidate.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFactoryAcceptCandidate.java:



Clases

- interface [es.ull.esit.app.factory_interface.IFactoryAcceptCandidate](#)

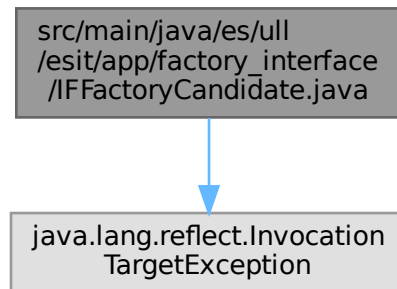
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.29. Referencia del archivo `src/main/java/es/ull/esit/app/factory_interface/IFactoryCandidate.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFactoryCandidate.java:



Clases

- interface [es.ull.esit.app.factory_interface.IFactoryCandidate](#)

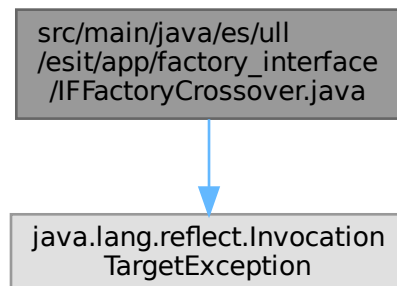
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.30. Referencia del archivo `src/main/java/es/ull/esit/app/factory_interface/IFactoryCrossover.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFactoryCrossover.java:



Clases

- interface [es.ull.esit.app.factory_interface.IFactoryCrossover](#)

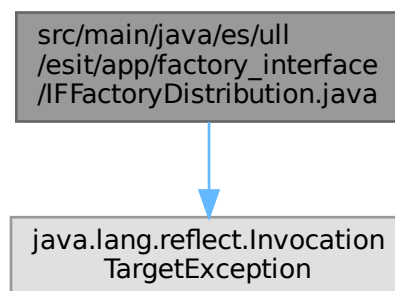
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.31. Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFactoryDistribution.java ↩

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFactoryDistribution.java:



Clases

- interface [es.ull.esit.app.factory_interface.IFactoryDistribution](#)

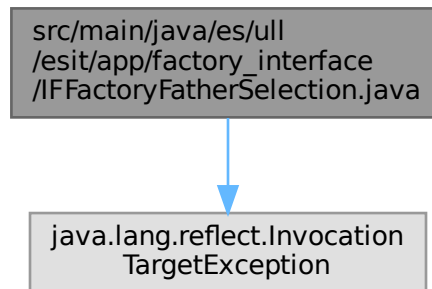
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.32. Referencia del archivo `src/main/java/es/ull/esit/app/factory_interface/IFactoryFatherSelection.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFactoryFatherSelection.java:



Clases

- interface [es.ull.esit.app.factory_interface.IFactoryFatherSelection](#)

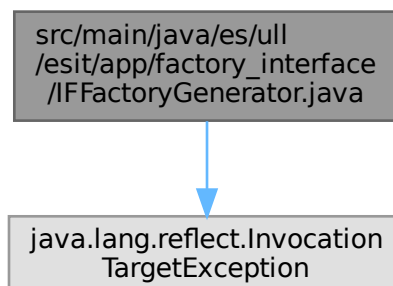
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.33. Referencia del archivo `src/main/java/es/ull/esit/app/factory_interface/IFactoryGenerator.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFactoryGenerator.java:



Clases

- interface [es.ull.esit.app.factory_interface.IFactoryGenerator](#)

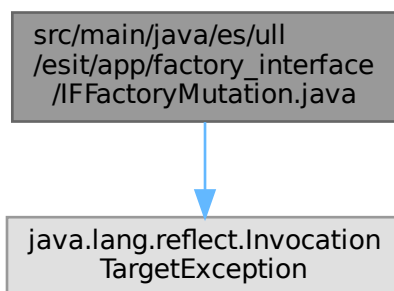
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.34. Referencia del archivo src/main/java/es/ull/esit/app/factory_↵ interface/IFactoryMutation.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFactoryMutation.java:



Clases

- interface [es.ull.esit.app.factory_interface.IFactoryMutation](#)

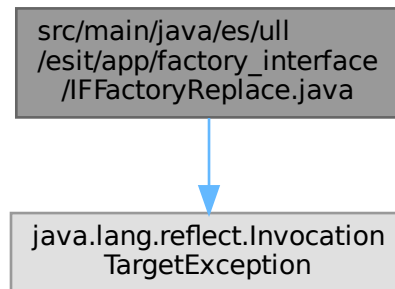
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.35. Referencia del archivo `src/main/java/es/ull/esit/app/factory_interface/IFactoryReplace.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `IFactoryReplace.java`:



Clases

- interface `es.ull.esit.app.factory_interface.IFactoryReplace`

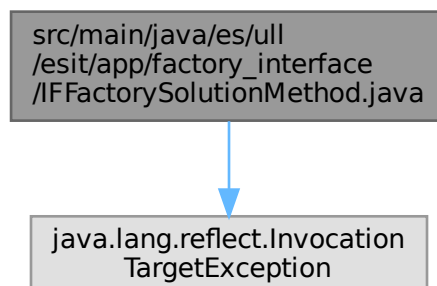
Paquetes

- package `es.ull.esit.app.factory_interface`

7.36. Referencia del archivo `src/main/java/es/ull/esit/app/factory_interface/IFactorySolutionMethod.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `IFactorySolutionMethod.java`:



Clases

- interface [es.ull.esit.app.factory_interface.IFFactorySolutionMethod](#)

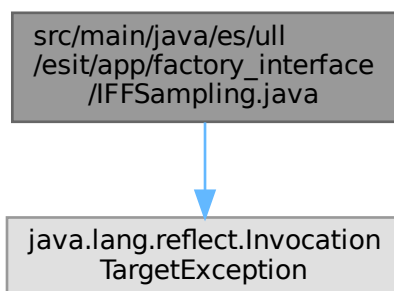
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.37. Referencia del archivo src/main/java/es/ull/esit/app/factory_interface/IFFSampling.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en IFFSampling.java:

**Clases**

- interface [es.ull.esit.app.factory_interface.IFFSampling](#)

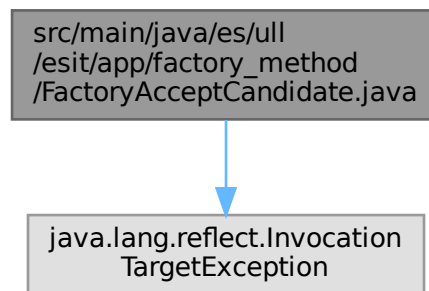
Paquetes

- package [es.ull.esit.app.factory_interface](#)

7.38. Referencia del archivo `src/main/java/es/ull/esit/app/factory_method/FactoryAcceptCandidate.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `FactoryAcceptCandidate.java`:



Clases

- class [es.ull.esit.app.factory_method.FactoryAcceptCandidate](#)

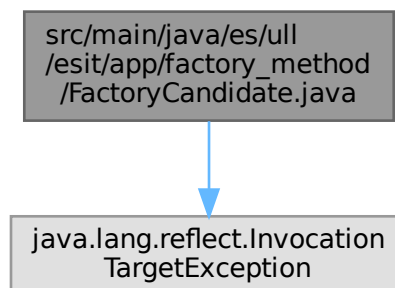
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.39. Referencia del archivo `src/main/java/es/ull/esit/app/factory_method/FactoryCandidate.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `FactoryCandidate.java`:



Clases

- class [es.ull.esit.app.factory_method.FactoryCandidate](#)

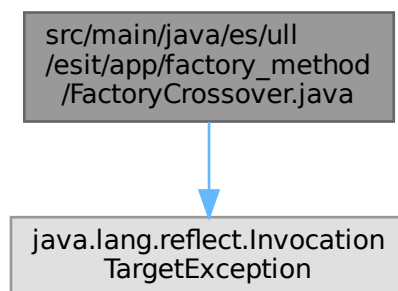
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.40. Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryCrossover.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactoryCrossover.java:



Clases

- class [es.ull.esit.app.factory_method.FactoryCrossover](#)

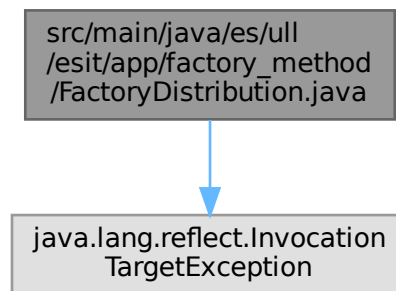
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.41. Referencia del archivo `src/main/java/es/ull/esit/app/factory_method/FactoryDistribution.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `FactoryDistribution.java`:



Clases

- class [es.ull.esit.app.factory_method.FactoryDistribution](#)

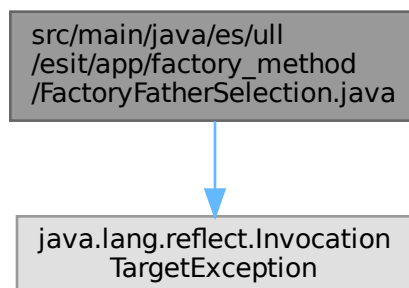
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.42. Referencia del archivo `src/main/java/es/ull/esit/app/factory_method/FactoryFatherSelection.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `FactoryFatherSelection.java`:



Clases

- class [es.ull.esit.app.factory_method.FactoryFatherSelection](#)

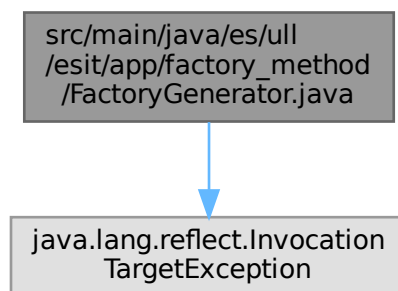
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.43. Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryGenerator.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactoryGenerator.java:



Clases

- class [es.ull.esit.app.factory_method.FactoryGenerator](#)

Paquetes

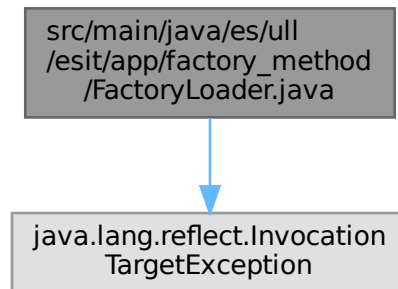
- package [es.ull.esit.app.factory_method](#)

7.44. Referencia del archivo

src/main/java/es/ull/esit/app/factory_method/FactoryLoader.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactoryLoader.java:



Clases

- class [es.ull.esit.app.factory_method.FactoryLoader](#)

Paquetes

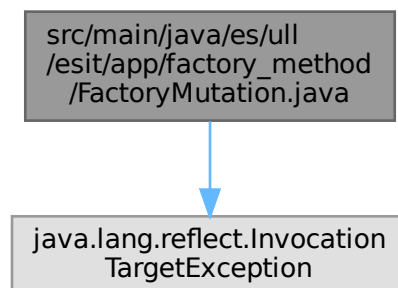
- package [es.ull.esit.app.factory_method](#)

7.45. Referencia del archivo

src/main/java/es/ull/esit/app/factory_method/FactoryMutation.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactoryMutation.java:



Clases

- class [es.ull.esit.app.factory_method.FactoryMutation](#)

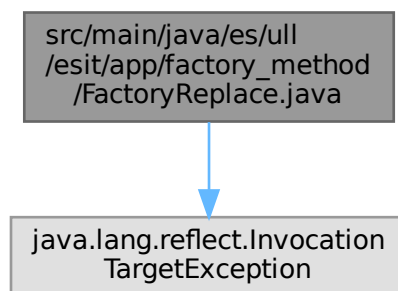
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.46. Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactoryReplace.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactoryReplace.java:



Clases

- class [es.ull.esit.app.factory_method.FactoryReplace](#)

Paquetes

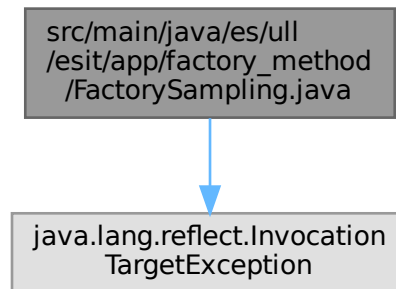
- package [es.ull.esit.app.factory_method](#)

7.47. Referencia del archivo

src/main/java/es/ull/esit/app/factory_method/FactorySampling.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactorySampling.java:



Clases

- class [es.ull.esit.app.factory_method.FactorySampling](#)

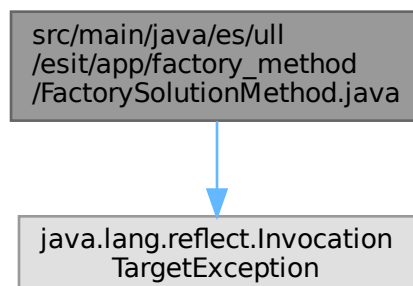
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.48. Referencia del archivo src/main/java/es/ull/esit/app/factory_method/FactorySolutionMethod.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en FactorySolutionMethod.java:



Clases

- class [es.ull.esit.app.factory_method.FactorySolutionMethod](#)

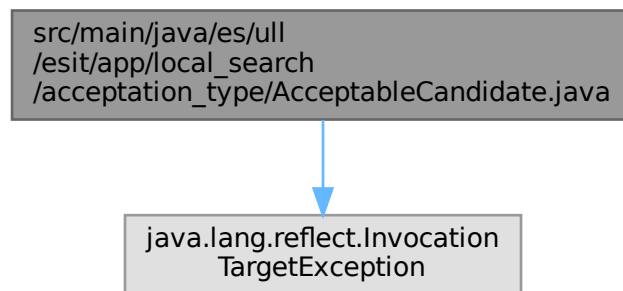
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.49. Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptableCandidate.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en AcceptableCandidate.java:

**Clases**

- class [es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate](#)

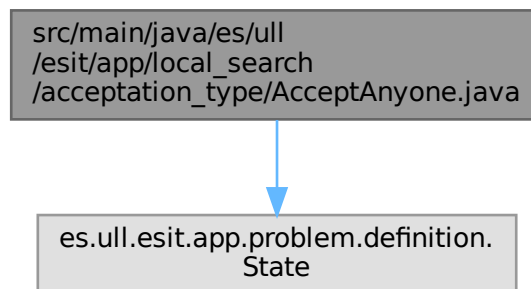
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.50. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptAnyone.java`

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en `AcceptAnyone.java`:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptAnyone`

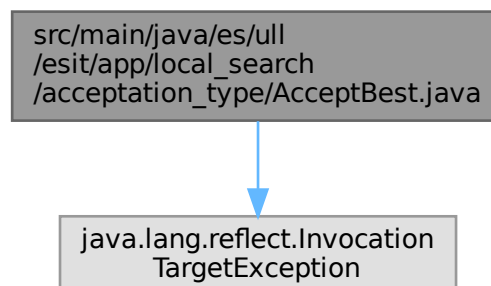
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.51. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptBest.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `AcceptBest.java`:



Clases

- class [es.ull.esit.app.local_search.acceptation_type.AcceptBest](#)

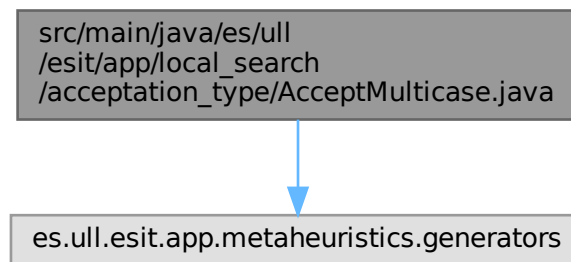
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.52. Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptMulticase.java

```
import es.ull.esit.app.metaheuristics.generators;
```

Gráfico de dependencias incluidas en AcceptMulticase.java:

**Clases**

- class [es.ull.esit.app.local_search.acceptation_type.AcceptMulticase](#)

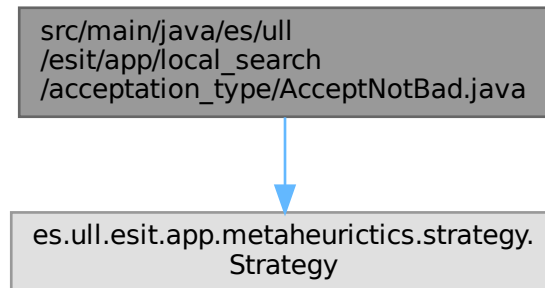
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.53. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBad.java` ↩

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `AcceptNotBad.java`:



Clases

- class [es.ull.esit.app.local_search.acceptation_type.AcceptNotBad](#)

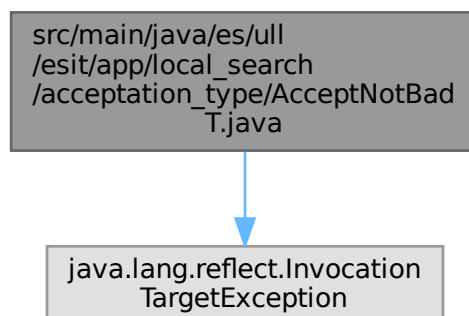
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.54. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadT.java` ↩

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `AcceptNotBadT.java`:



Clases

- class [es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT](#)

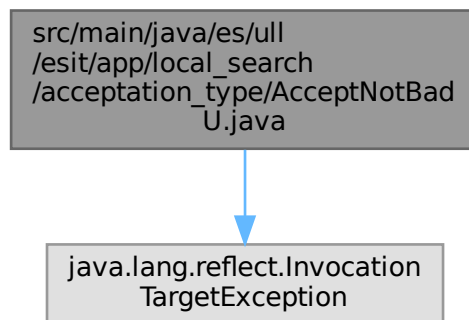
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.55. Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadU.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en AcceptNotBadU.java:

**Clases**

- class [es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU](#)

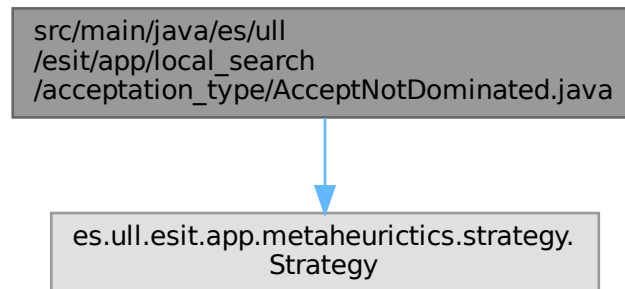
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.56. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominated.java` ↩

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `AcceptNotDominated.java`:



Clases

- class [es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated](#)

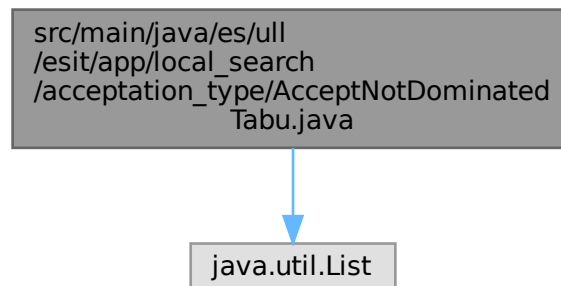
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.57. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominatedTabu.java` ↩

```
import java.util.List;
```

Gráfico de dependencias incluidas en `AcceptNotDominatedTabu.java`:



Clases

- class [es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu](#)

Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.58. Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/AcceptType.java ↩

Clases

- enum [es.ull.esit.app.local_search.acceptation_type.AcceptType](#)

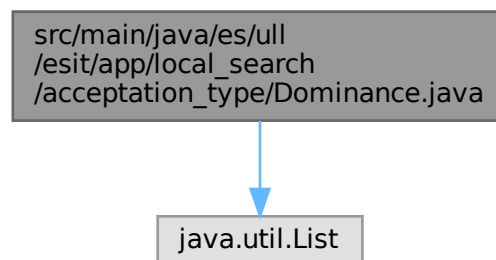
Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.59. Referencia del archivo src/main/java/es/ull/esit/app/local_search/acceptation_type/Dominance.java ↩

```
import java.util.List;
```

Gráfico de dependencias incluidas en Dominance.java:



Clases

- class [es.ull.esit.app.local_search.acceptation_type.Dominance](#)

Paquetes

- package [es.ull.esit.app.local_search.acceptation_type](#)

7.60. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/candidate_type/CandidateType.java`

Clases

- enum [es.ull.esit.app.local_search.candidate_type.CandidateType](#)

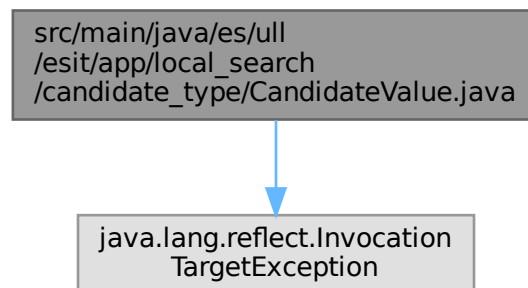
Paquetes

- package [es.ull.esit.app.local_search.candidate_type](#)

7.61. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/candidate_type/CandidateValue.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `CandidateValue.java`:



Clases

- class [es.ull.esit.app.local_search.candidate_type.CandidateValue](#)

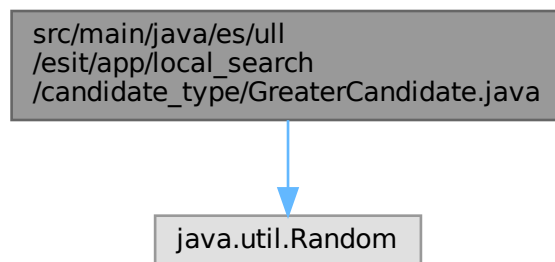
Paquetes

- package [es.ull.esit.app.local_search.candidate_type](#)

7.62. Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/GreaterCandidate.java ↩

```
import java.util.Random;
```

Gráfico de dependencias incluidas en GreaterCandidate.java:



Clases

- class [es.ull.esit.app.local_search.candidate_type.GreaterCandidate](#)

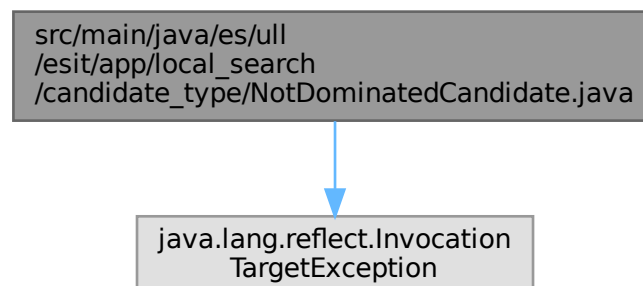
Paquetes

- package [es.ull.esit.app.local_search.candidate_type](#)

7.63. Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/NotDominatedCandidate.java ↩

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en NotDominatedCandidate.java:



Clases

- class [es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate](#)

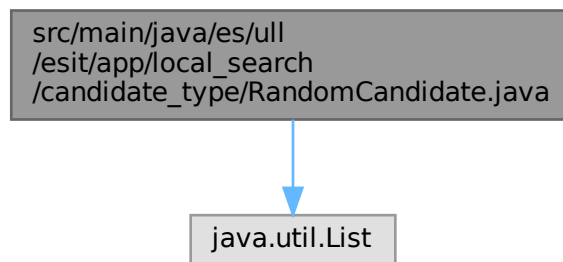
Paquetes

- package [es.ull.esit.app.local_search.candidate_type](#)

7.64. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/candidate_type/RandomCandidate.java`

```
import java.util.List;
```

Gráfico de dependencias incluidas en RandomCandidate.java:



Clases

- class [es.ull.esit.app.local_search.candidate_type.RandomCandidate](#)

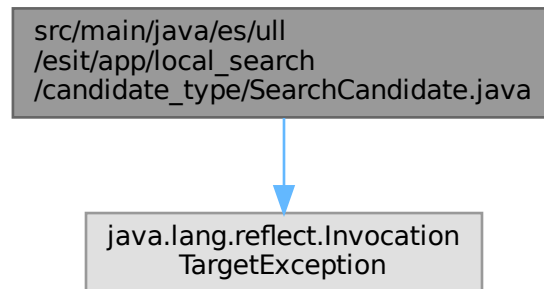
Paquetes

- package [es.ull.esit.app.local_search.candidate_type](#)

7.65. Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/SearchCandidate.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en SearchCandidate.java:



Clases

- class [es.ull.esit.app.local_search.candidate_type.SearchCandidate](#)

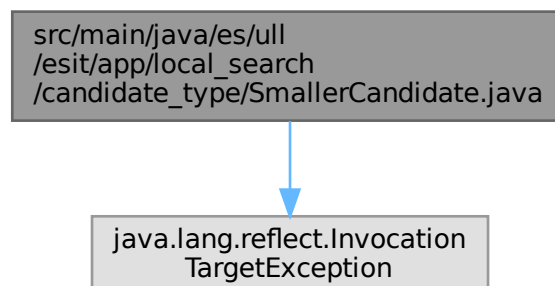
Paquetes

- package [es.ull.esit.app.local_search.candidate_type](#)

7.66. Referencia del archivo src/main/java/es/ull/esit/app/local_search/candidate_type/SmallerCandidate.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en SmallerCandidate.java:



Clases

- class [es.ull.esit.app.local_search.candidate_type.SmallerCandidate](#)

Paquetes

- package [es.ull.esit.app.local_search.candidate_type](#)

7.67. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/complement/StopExecute.java`

Clases

- class [es.ull.esit.app.local_search.complement.StopExecute](#)

Paquetes

- package [es.ull.esit.app.local_search.complement](#)

7.68. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/complement/StrategyType.java`

Clases

- enum [es.ull.esit.app.local_search.complement.StrategyType](#)

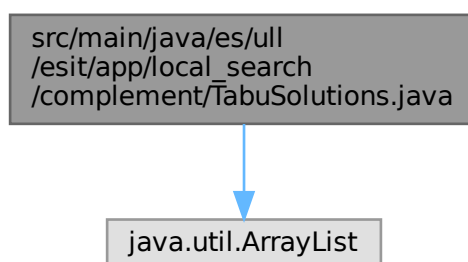
Paquetes

- package [es.ull.esit.app.local_search.complement](#)

7.69. Referencia del archivo `src/main/java/es/ull/esit/app/local_search/complement/TabuSolutions.java`

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en TabuSolutions.java:



Clases

- class [es.ull.esit.app.local_search.complement.TabuSolutions](#)

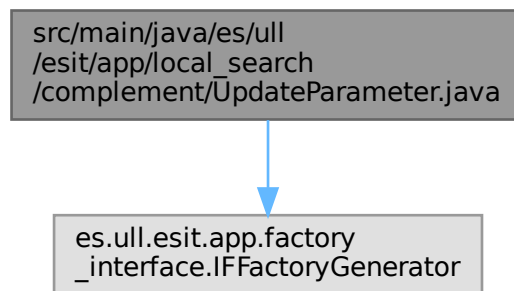
Paquetes

- package [es.ull.esit.app.local_search.complement](#)

7.70. Referencia del archivo src/main/java/es/ull/esit/app/local_search/complement/UpdateParameter.java

```
import es.ull.esit.app.factory_interface.IFFactoryGenerator;
```

Gráfico de dependencias incluidas en UpdateParameter.java:



Clases

- class [es.ull.esit.app.local_search.complement.UpdateParameter](#)

Paquetes

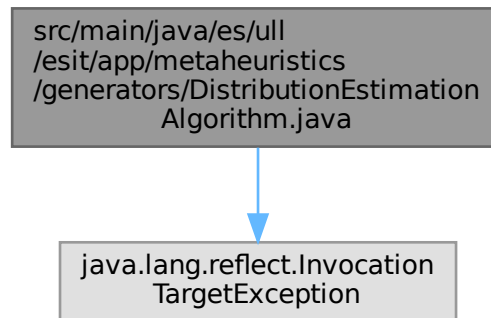
- package [es.ull.esit.app.local_search.complement](#)

7.71. Referencia del archivo

**src/main/java/es/ull/esit/app/metaheuristics/generators/↵
DistributionEstimationAlgorithm.java**

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en DistributionEstimationAlgorithm.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm](#)

Paquetes

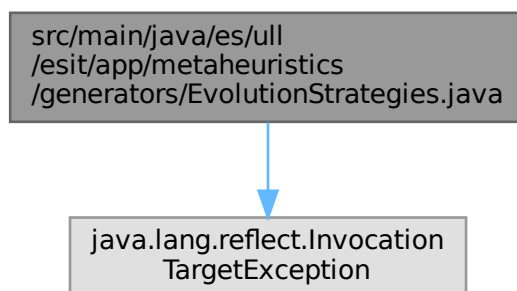
- package [es.ull.esit.app.metaheuristics.generators](#)

7.72. Referencia del archivo

**src/main/java/es/ull/esit/app/metaheuristics/generators/Evolution↵
Strategies.java**

```
import java.lang.reflect.InvocationTargetException;
```


Gráfico de dependencias incluidas en EvolutionStrategies.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.EvolutionStrategies](#)

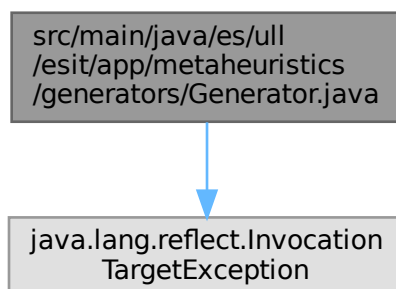
Paquetes

- package [es.ull.esit.app.metaheuristics.generators](#)

7.73. Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/Generator.java↵

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en Generator.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.Generator](#)

Paquetes

- package [es.ull.esit.app.metaheuristics.generators](#)

7.74. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/GeneratorType.java

Clases

- enum [es.ull.esit.app.metaheuristics.generators.GeneratorType](#)

Paquetes

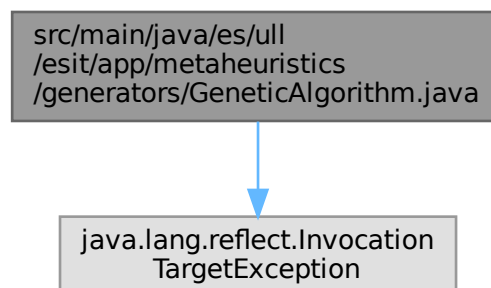
- package [es.ull.esit.app.metaheuristics.generators](#)

7.75. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/GeneticAlgorithm.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en GeneticAlgorithm.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm](#)

Paquetes

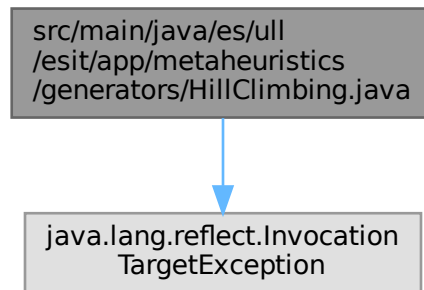
- package [es.ull.esit.app.metaheuristics.generators](#)

7.76. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbing.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en HillClimbing.java:

**Clases**

- class [es.ull.esit.app.metaheuristics.generators.HillClimbing](#)

Paquetes

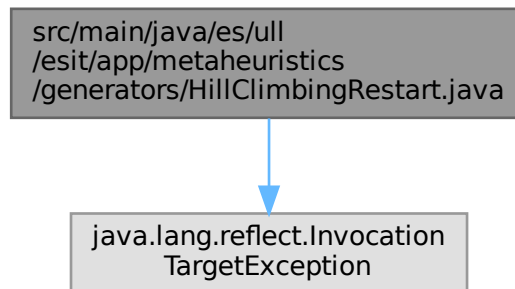
- package [es.ull.esit.app.metaheuristics.generators](#)

7.77. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbingRestart.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en HillClimbingRestart.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.HillClimbingRestart](#)

Paquetes

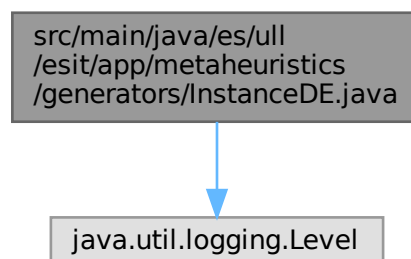
- package [es.ull.esit.app.metaheuristics.generators](#)

7.78. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceDE.java

```
import java.util.logging.Level;
```

Gráfico de dependencias incluidas en InstanceDE.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.InstanceDE](#)

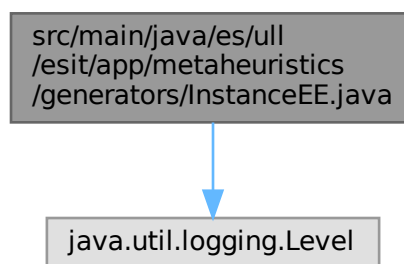
Paquetes

- package [es.ull.esit.app.metaheuristics.generators](#)

7.79. Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceEE.java

```
import java.util.logging.Level;
```

Gráfico de dependencias incluidas en InstanceEE.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.InstanceEE](#)

Paquetes

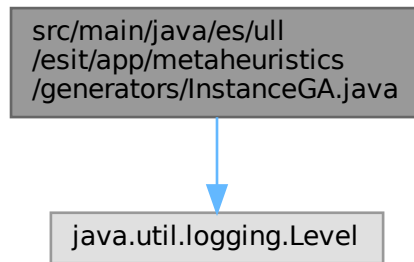
- package [es.ull.esit.app.metaheuristics.generators](#)

7.80. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceGA.java

```
import java.util.logging.Level;
```

Gráfico de dependencias incluidas en InstanceGA.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.InstanceGA](#)

Paquetes

- package [es.ull.esit.app.metaheuristics.generators](#)

7.81. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/LimitRoulette.java

Clases

- class [es.ull.esit.app.metaheuristics.generators.LimitRoulette](#)

Paquetes

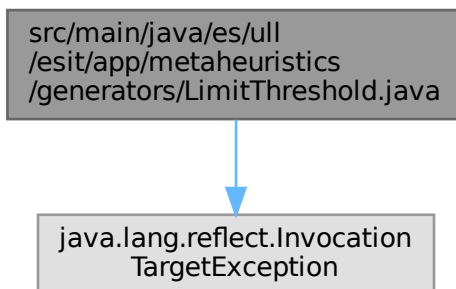
- package [es.ull.esit.app.metaheuristics.generators](#)

7.82. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/LimitThreshold.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en LimitThreshold.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.LimitThreshold](#)

Paquetes

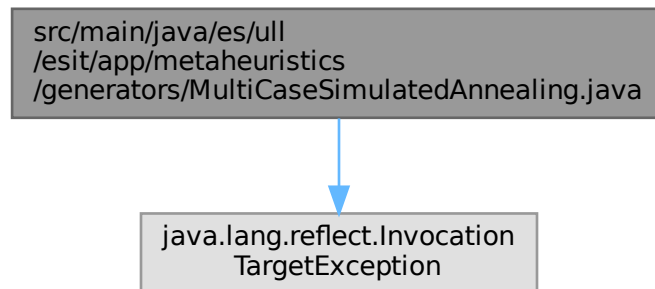
- package [es.ull.esit.app.metaheuristics.generators](#)

7.83. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/MultiCaseSimulatedAnnealing.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en MultiCaseSimulatedAnnealing.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing](#)

Paquetes

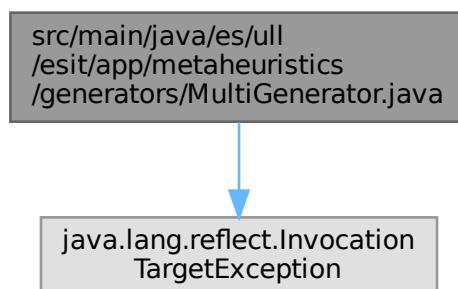
- package [es.ull.esit.app.metaheuristics.generators](#)

7.84. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/MultiGenerator.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en MultiGenerator.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.MultiGenerator](#)

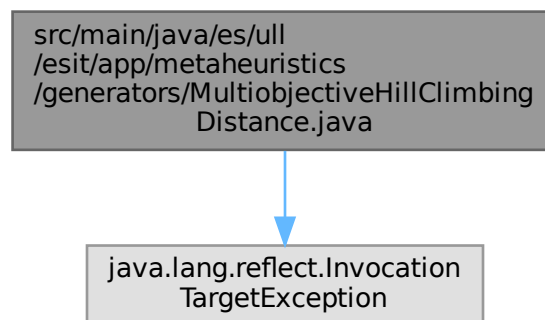
Paquetes

- package [es.ull.esit.app.metaheuristics.generators](#)

7.85. Referencia del archivo

**src/main/java/es/ull/esit/app/metaheuristics/generators/↵
MultiobjectiveHillClimbingDistance.java**

import java.lang.reflect.InvocationTargetException;
Gráfico de dependencias incluidas en MultiobjectiveHillClimbingDistance.java:

**Clases**

- class [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance](#)

Paquetes

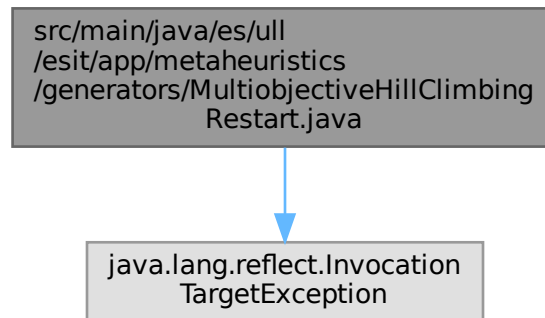
- package [es.ull.esit.app.metaheuristics.generators](#)

7.86. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/↵
MultiobjectiveHillClimbingRestart.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en MultiobjectiveHillClimbingRestart.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart](#)

Paquetes

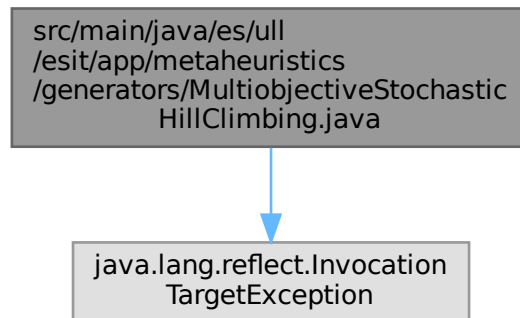
- package [es.ull.esit.app.metaheuristics.generators](#)

7.87. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/↵
MultiobjectiveStochasticHillClimbing.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en MultiobjectiveStochasticHillClimbing.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing](#)

Paquetes

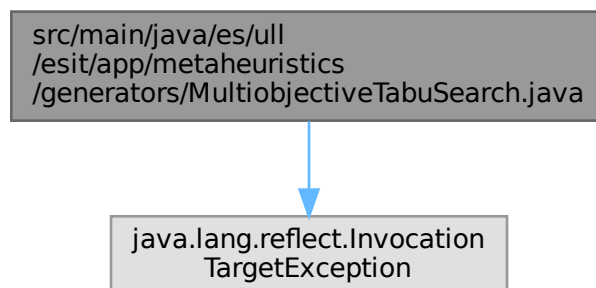
- package [es.ull.esit.app.metaheuristics.generators](#)

7.88. Referencia del archivo

**src/main/java/es/ull/esit/app/metaheuristics/generators/↵
MultiobjectiveTabuSearch.java**

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en MultiobjectiveTabuSearch.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch](#)

Paquetes

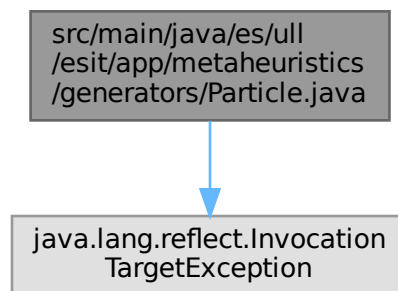
- package [es.ull.esit.app.metaheuristics.generators](#)

7.89. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/Particle.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en Particle.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.Particle](#)

Paquetes

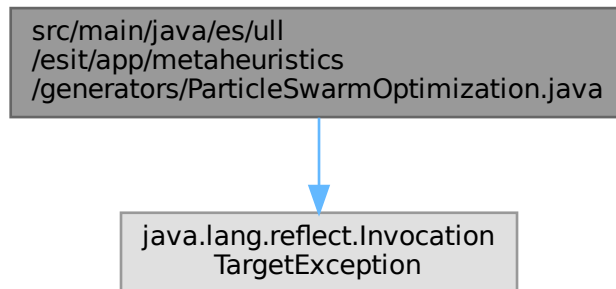
- package [es.ull.esit.app.metaheuristics.generators](#)

7.90. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/ParticleSwarmOptimization.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en ParticleSwarmOptimization.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization](#)

Paquetes

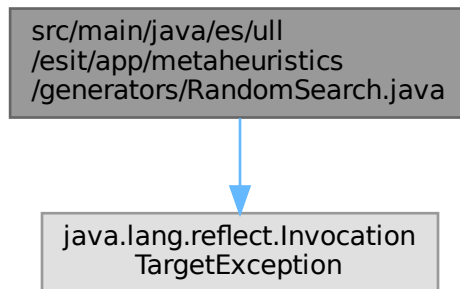
- package [es.ull.esit.app.metaheuristics.generators](#)

7.91. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/RandomSearch.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en RandomSearch.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.RandomSearch](#)

Paquetes

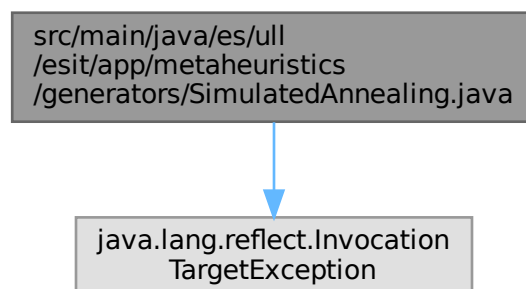
- package [es.ull.esit.app.metaheuristics.generators](#)

7.92. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java ↔

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en SimulatedAnnealing.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#)

Paquetes

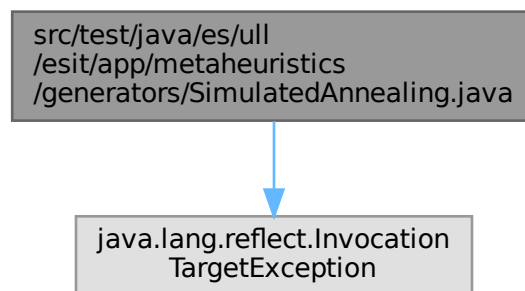
- package [es.ull.esit.app.metaheuristics.generators](#)

7.93. Referencia del archivo

`src/test/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en `SimulatedAnnealing.java`:



Clases

- class [es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing](#)

Paquetes

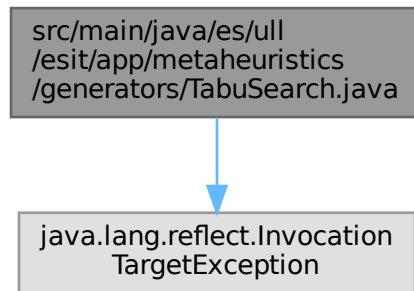
- package [es.ull.esit.app.metaheuristics.generators](#)

7.94. Referencia del archivo

src/main/java/es/ull/esit/app/metaheuristics/generators/TabuSearch.java↔

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en TabuSearch.java:



Clases

- class [es.ull.esit.app.metaheuristics.generators.TabuSearch](#)

Paquetes

- package [es.ull.esit.app.metaheuristics.generators](#)

7.95. Referencia del archivo

src/main/java/es/ull/esit/app/problem/definition/Codification.java

Clases

- class [es.ull.esit.app.problem.definition.Codification](#)

Paquetes

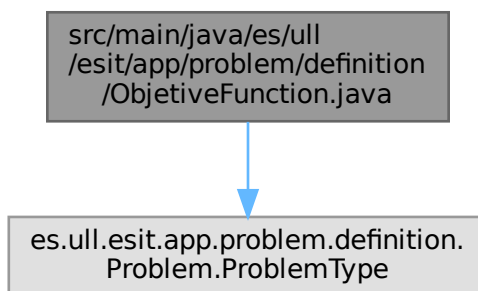
- package [es.ull.esit.app.problem.definition](#)

7.96. Referencia del archivo

src/main/java/es/ull/esit/app/problem/definition/ObjetiveFunction.java

```
import es.ull.esit.app.problem.definition.Problem.ProblemType;
```

Gráfico de dependencias incluidas en ObjetiveFunction.java:



Clases

- class [es.ull.esit.app.problem.definition.ObjetiveFunction](#)

Paquetes

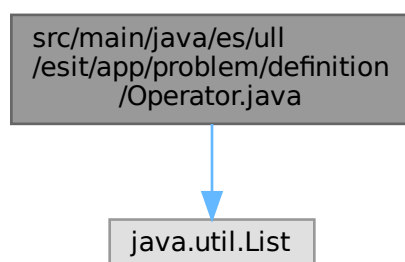
- package [es.ull.esit.app.problem.definition](#)

7.97. Referencia del archivo

src/main/java/es/ull/esit/app/problem/definition/Operator.java

```
import java.util.List;
```

Gráfico de dependencias incluidas en Operator.java:



Clases

- class [es.ull.esit.app.problem.definition.Operator](#)

Paquetes

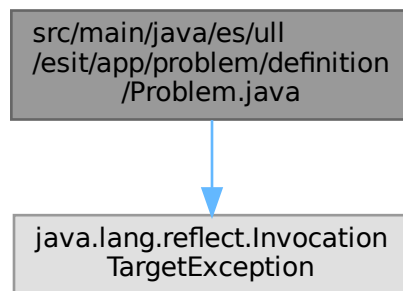
- package [es.ull.esit.app.problem.definition](#)

7.98. Referencia del archivo

src/main/java/es/ull/esit/app/problem/definition/Problem.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en Problem.java:



Clases

- class [es.ull.esit.app.problem.definition.Problem](#)
- enum [es.ull.esit.app.problem.definition.Problem.ProblemType](#)

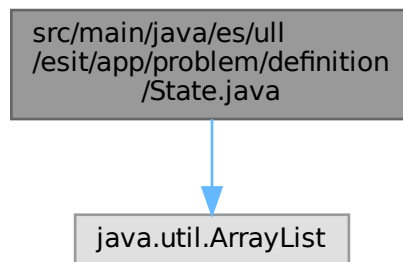
Paquetes

- package [es.ull.esit.app.problem.definition](#)

7.99. Referencia del archivo src/main/java/es/ull/esit/app/problem/definition/State.java

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en State.java:



Clases

- class [es.ull.esit.app.problem.definition.State](#)

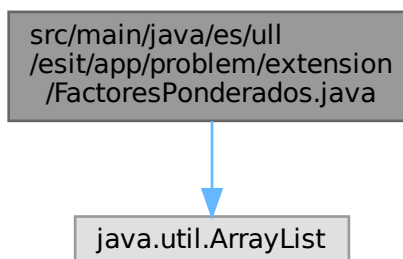
Paquetes

- package [es.ull.esit.app.problem.definition](#)

7.100. Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/FactoresPonderados.java↔

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en FactoresPonderados.java:



Clases

- class [es.ull.esit.app.problem.extension.FactoresPonderados](#)

Paquetes

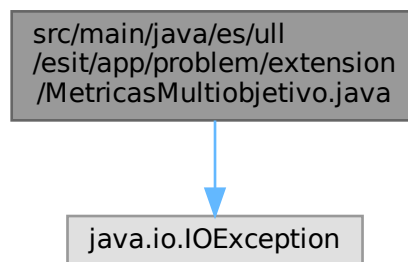
- package [es.ull.esit.app.problem.extension](#)

7.101. Referencia del archivo

src/main/java/es/ull/esit/app/problem/extension/MetricasMultiobjetivo.java

```
import java.io.IOException;
```

Gráfico de dependencias incluidas en MetricasMultiobjetivo.java:



Clases

- class [es.ull.esit.app.problem.extension.MetricasMultiobjetivo](#)

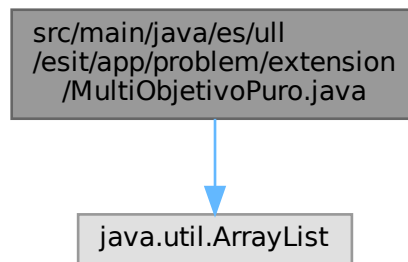
Paquetes

- package [es.ull.esit.app.problem.extension](#)

7.102. Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/MultiObjetivoPuro.java↔

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en MultiObjetivoPuro.java:



Clases

- class [es.ull.esit.app.problem.extension.MultiObjetivoPuro](#)

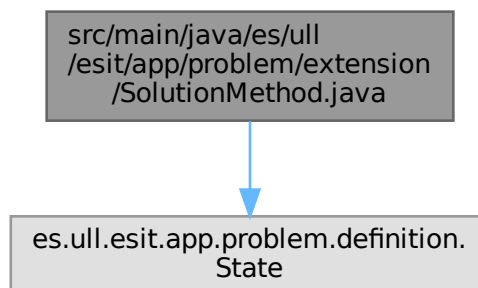
Paquetes

- package [es.ull.esit.app.problem.extension](#)

7.103. Referencia del archivo src/main/java/es/ull/esit/app/problem/extension/SolutionMethod.java↔

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en SolutionMethod.java:



Clases

- class [es.ull.esit.app.problem.extension.SolutionMethod](#)

Paquetes

- package [es.ull.esit.app.problem.extension](#)

7.104. Referencia del archivo

src/main/java/es/ull/esit/app/problem/extension/TypeSolutionMethod.java↔

Clases

- enum [es.ull.esit.app.problem.extension.TypeSolutionMethod](#)

Paquetes

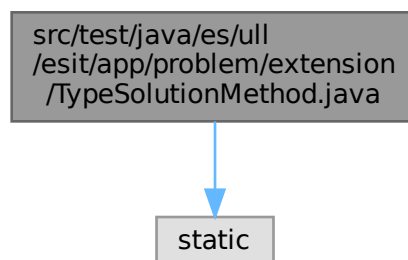
- package [es.ull.esit.app.problem.extension](#)

7.105. Referencia del archivo

src/test/java/es/ull/esit/app/problem/extension/TypeSolutionMethod.java↔

```
import static;
```

Gráfico de dependencias incluidas en TypeSolutionMethod.java:



Clases

- class [es.ull.esit.app.problem.extension.TypeSolutionMethodTest](#)

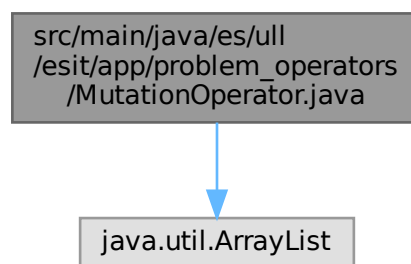
Paquetes

- package [es.ull.esit.app.problem.extension](#)

7.106. Referencia del archivo src/main/java/es/ull/esit/app/problem_operators/MutationOperator.java

```
import java.util.ArrayList;
```

Gráfico de dependencias incluidas en MutationOperator.java:



Clases

- class [es.ull.esit.app.problem_operators.MutationOperator](#)

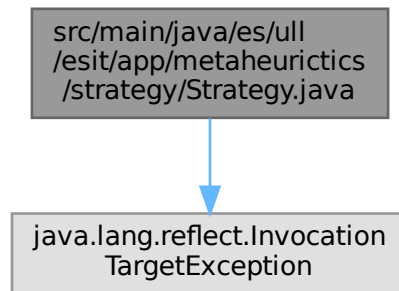
Paquetes

- package [es.ull.esit.app.problem_operators](#)

7.107. Referencia del archivo src/main/java/es/ull/esit/app/metaheuristics/strategy/Strategy.java

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en Strategy.java:



Clases

- class [es.ull.esit.app.metaheuristics.strategy.Strategy](#)

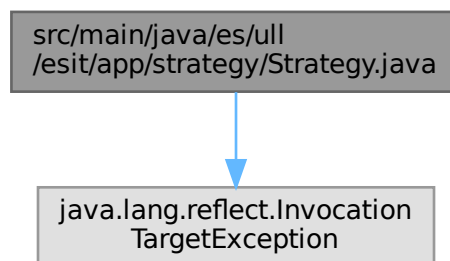
Paquetes

- package [es.ull.esit.app.metaheuristics.strategy](#)

7.108. Referencia del archivo `src/main/java/es/ull/esit/app/strategy/Strategy.java`

```
import java.lang.reflect.InvocationTargetException;
```

Gráfico de dependencias incluidas en Strategy.java:



Clases

- class [es.ull.esit.app.strategy.Strategy](#)

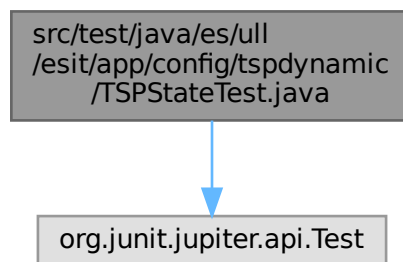
Paquetes

- package [es.ull.esit.app.strategy](#)

7.109. Referencia del archivo src/test/java/es/ull/esit/app/config/tspdynamic/TSPStateTest.java

```
import org.junit.jupiter.api.Test;
```

Gráfico de dependencias incluidas en TSPStateTest.java:

**Clases**

- class `es.ull.esit.app.config.tspdynamic.TSPStateTest`

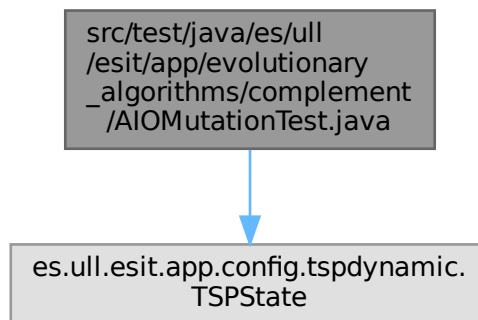
Paquetes

- package [es.ull.esit.app.config.tspdynamic](#)

7.110. Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_↵ algorithms/complement/AIOMutationTest.java

```
import es.ull.esit.app.config.tspdynamic.TSPState;
```

Gráfico de dependencias incluidas en AIOMutationTest.java:



Clases

- class `es.ull.esit.app.evolutionary_algorithms.complement.AIOMutationTest`

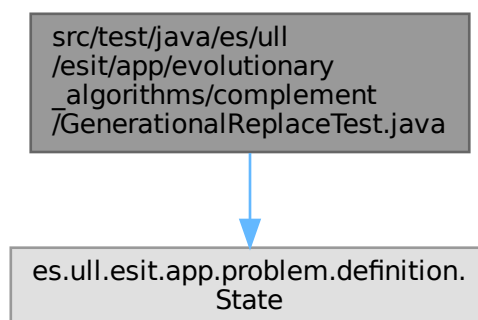
Paquetes

- package `es.ull.esit.app.evolutionary_algorithms.complement`

7.111. Referencia del archivo `src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/GenerationalReplaceTest.java`

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en GenerationalReplaceTest.java:



Clases

- class **es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplaceTest**

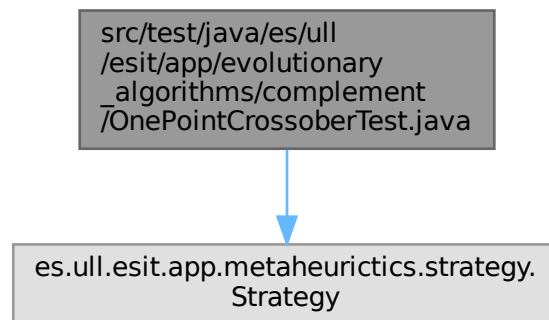
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.112. Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointCrossoverTest.java

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en OnePointCrossoverTest.java:

**Clases**

- class **es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossoverTest**

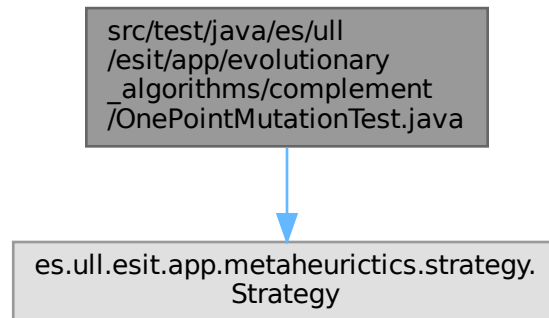
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.113. Referencia del archivo `src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePointMutationTest.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `OnePointMutationTest.java`:



Clases

- class `es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutationTest`

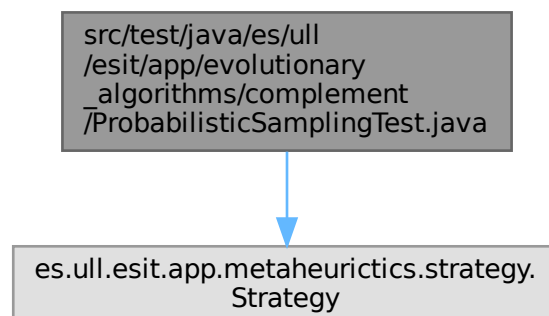
Paquetes

- package `es.ull.esit.app.evolutionary_algorithms.complement`

7.114. Referencia del archivo `src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/ProbabilisticSamplingTest.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `ProbabilisticSamplingTest.java`:



Clases

- class **es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSamplingTest**

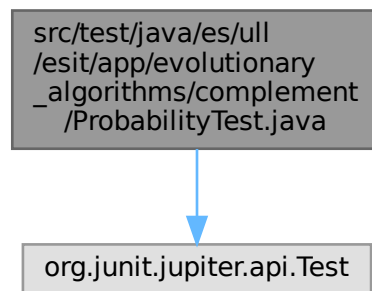
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.115. Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/ProbabilityTest.java

```
import org.junit.jupiter.api.Test;
```

Gráfico de dependencias incluidas en ProbabilityTest.java:

**Clases**

- class **es.ull.esit.app.evolutionary_algorithms.complement.ProbabilityTest**

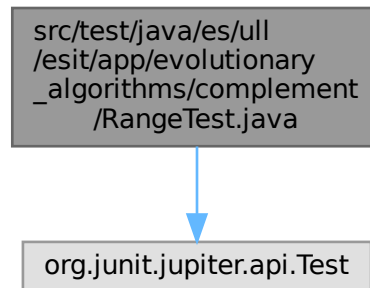
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.116. Referencia del archivo `src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/RangeTest.java`

```
import org.junit.jupiter.api.Test;
```

Gráfico de dependencias incluidas en `RangeTest.java`:



Clases

- class `es.ull.esit.app.evolutionary_algorithms.complement.RangeTest`

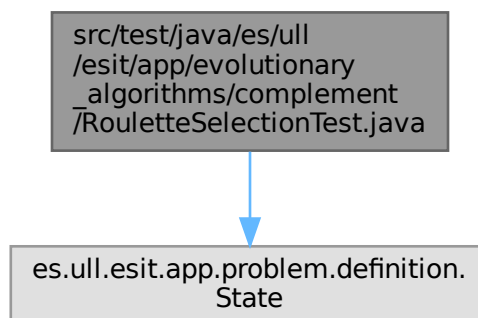
Paquetes

- package `es.ull.esit.app.evolutionary_algorithms.complement`

7.117. Referencia del archivo `src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/RouletteSelectionTest.java`

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en `RouletteSelectionTest.java`:



Clases

- class **es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelectionTest**

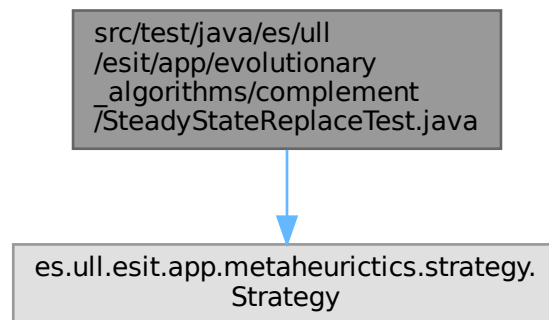
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.118. Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/SteadyStateReplaceTest.java

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en SteadyStateReplaceTest.java:

**Clases**

- class **es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplaceTest**

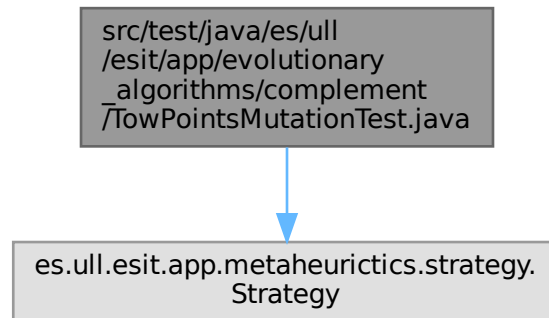
Paquetes

- package [es.ull.esit.app.evolutionary_algorithms.complement](#)

7.119. Referencia del archivo `src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/TowPointsMutationTest.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `TowPointsMutationTest.java`:



Clases

- class `es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutationTest`

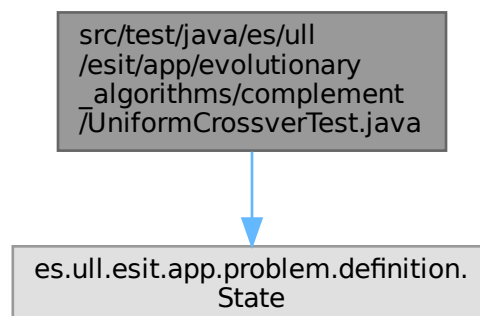
Paquetes

- package `es.ull.esit.app.evolutionary_algorithms.complement`

7.120. Referencia del archivo `src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/UniformCrossverTest.java`

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en `UniformCrossverTest.java`:



Clases

- class `es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossoverTest`

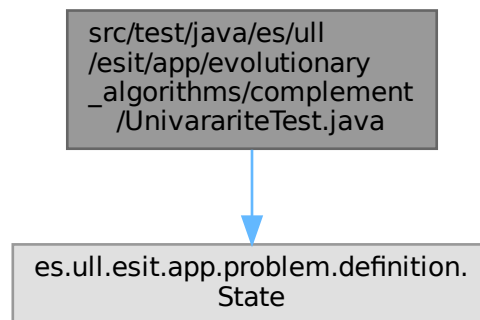
Paquetes

- package `es.ull.esit.app.evolutionary_algorithms.complement`

7.121. Referencia del archivo src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/UnivarariteTest.java

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en UnivarariteTest.java:

**Clases**

- class `es.ull.esit.app.evolutionary_algorithms.complement.UnivariateTest`

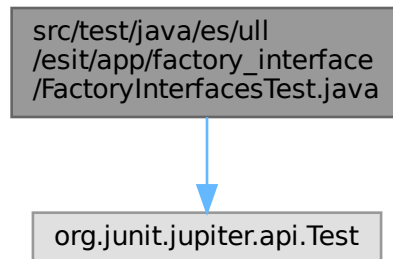
Paquetes

- package `es.ull.esit.app.evolutionary_algorithms.complement`

7.122. Referencia del archivo `src/test/java/es/ull/esit/app/factory_interface/FactoryInterfacesTest.java`

```
import org.junit.jupiter.api.Test;
```

Gráfico de dependencias incluidas en `FactoryInterfacesTest.java`:



Clases

- class `es.ull.esit.app.factory_interface.FactoryInterfacesTest`

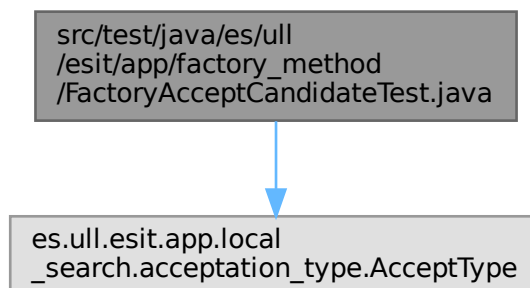
Paquetes

- package `es.ull.esit.app.factory_interface`

7.123. Referencia del archivo `src/test/java/es/ull/esit/app/factory_method/FactoryAcceptCandidateTest.java`

```
import es.ull.esit.app.local_search.acceptation_type.AcceptType;
```

Gráfico de dependencias incluidas en `FactoryAcceptCandidateTest.java`:



Clases

- class **es.ull.esit.app.factory_method.FactoryAcceptCandidateTest**

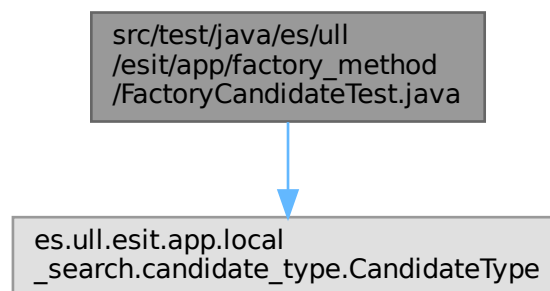
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.124. Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryCandidateTest.java

```
import es.ull.esit.app.local_search.candidate_type.CandidateType;
```

Gráfico de dependencias incluidas en FactoryCandidateTest.java:



Clases

- class **es.ull.esit.app.factory_method.FactoryCandidateTest**

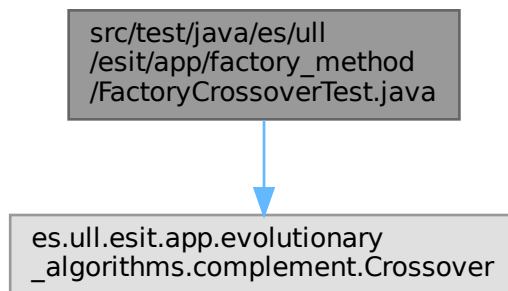
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.125. Referencia del archivo `src/test/java/es/ull/esit/app/factory_method/FactoryCrossoverTest.java`

```
import es.ull.esit.app.evolutionary_algorithms.complement.Crossover;
```

Gráfico de dependencias incluidas en `FactoryCrossoverTest.java`:



Clases

- class `es.ull.esit.app.factory_method.FactoryCrossoverTest`

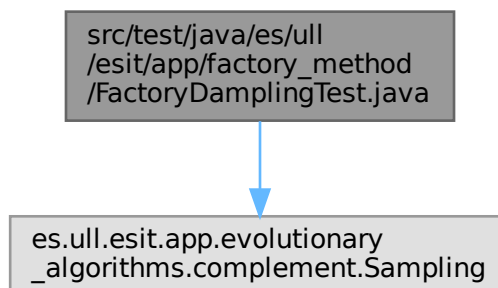
Paquetes

- package `es.ull.esit.app.factory_method`

7.126. Referencia del archivo `src/test/java/es/ull/esit/app/factory_method/FactoryDamplingTest.java`

```
import es.ull.esit.app.evolutionary_algorithms.complement.Sampling;
```

Gráfico de dependencias incluidas en `FactoryDamplingTest.java`:



Clases

- class **es.ull.esit.app.factory_method.FactorySamplingTest**

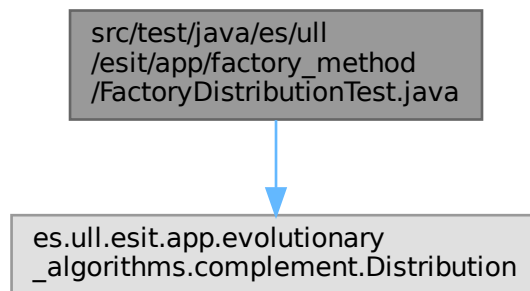
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.127. Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryDistributionTest.java

```
import es.ull.esit.app.evolutionary_algorithms.complement.Distribution;
```

Gráfico de dependencias incluidas en FactoryDistributionTest.java:

**Clases**

- class **es.ull.esit.app.factory_method.FactoryDistributionTest**

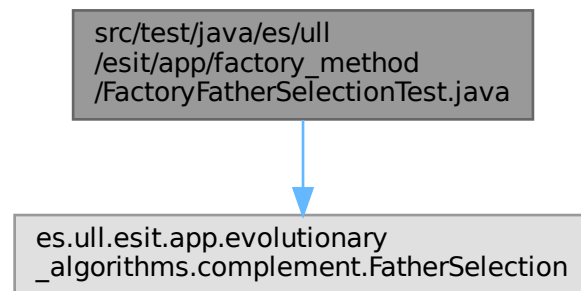
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.128. Referencia del archivo `src/test/java/es/ull/esit/app/factory_method/FactoryFatherSelectionTest.java`

```
import es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection;
```

Gráfico de dependencias incluidas en `FactoryFatherSelectionTest.java`:



Clases

- class `es.ull.esit.app.factory_method.FactoryFatherSelectionTest`

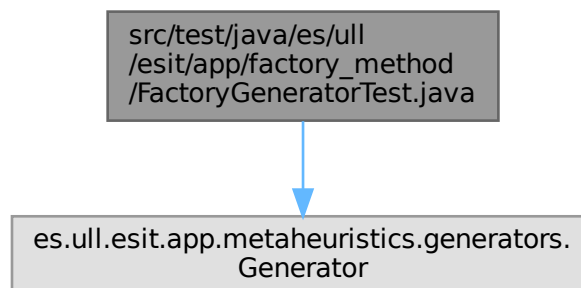
Paquetes

- package `es.ull.esit.app.factory_method`

7.129. Referencia del archivo `src/test/java/es/ull/esit/app/factory_method/FactoryGeneratorTest.java`

```
import es.ull.esit.app.metaheuristics.generators.Generator;
```

Gráfico de dependencias incluidas en `FactoryGeneratorTest.java`:



Clases

- class **es.ull.esit.app.factory_method.FactoryGeneratorTest**

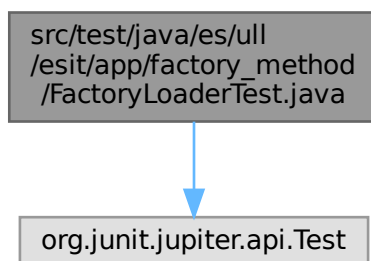
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.130. Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactoryLoaderTest.java ↩

```
import org.junit.jupiter.api.Test;
```

Gráfico de dependencias incluidas en FactoryLoaderTest.java:



Clases

- class **es.ull.esit.app.factory_method.FactoryLoaderTest**

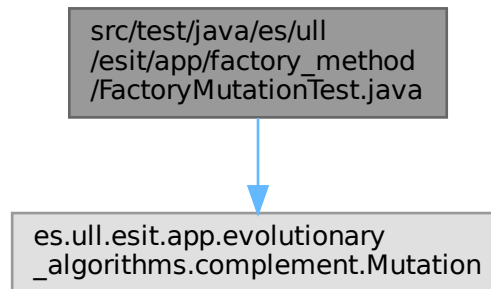
Paquetes

- package [es.ull.esit.app.factory_method](#)

7.131. Referencia del archivo `src/test/java/es/ull/esit/app/factory_method/FactoryMutationTest.java`

```
import es.ull.esit.app.evolutionary_algorithms.complement.Mutation;
```

Gráfico de dependencias incluidas en `FactoryMutationTest.java`:



Clases

- class `es.ull.esit.app.factory_method.FactoryMutationTest`

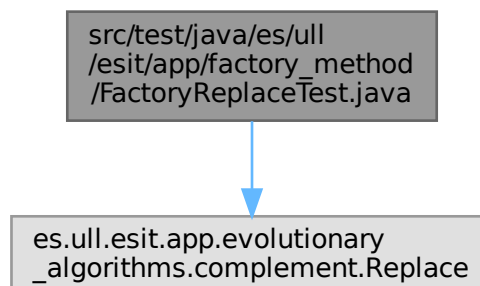
Paquetes

- package `es.ull.esit.app.factory_method`

7.132. Referencia del archivo `src/test/java/es/ull/esit/app/factory_method/FactoryReplaceTest.java`

```
import es.ull.esit.app.evolutionary_algorithms.complement.Replace;
```

Gráfico de dependencias incluidas en `FactoryReplaceTest.java`:



Clases

- class `es.ull.esit.app.factory_method.FactoryReplaceTest`

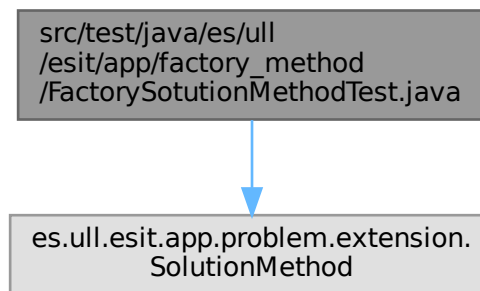
Paquetes

- package `es.ull.esit.app.factory_method`

7.133. Referencia del archivo src/test/java/es/ull/esit/app/factory_method/FactorySotutionMethodTest.java

```
import es.ull.esit.app.problem.extension.SolutionMethod;
```

Gráfico de dependencias incluidas en FactorySotutionMethodTest.java:



Clases

- class `es.ull.esit.app.factory_method.FactorySolutionMethodTest`

Paquetes

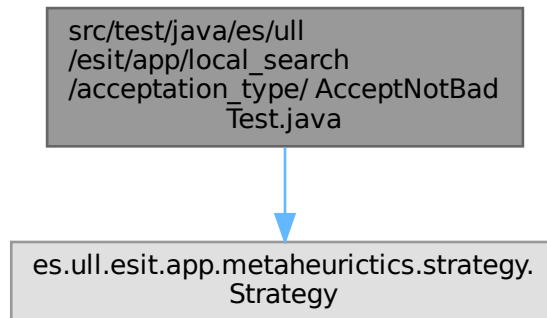
- package `es.ull.esit.app.factory_method`

7.134. Referencia del archivo

**src/test/java/es/ull/esit/app/local_search/acceptation_type/
AcceptNotBadTest.java**

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en AcceptNotBadTest.java:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptNotBadTest`

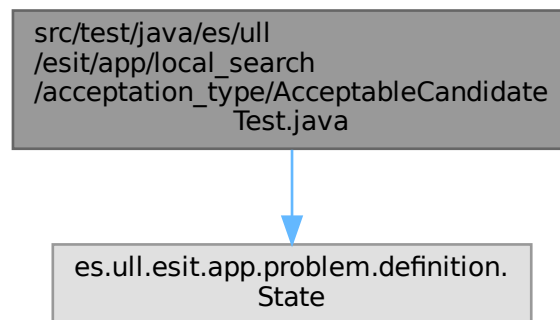
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.135. Referencia del archivo `src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptableCandidateTest.java`

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en AcceptableCandidateTest.java:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptableCandidateTest`

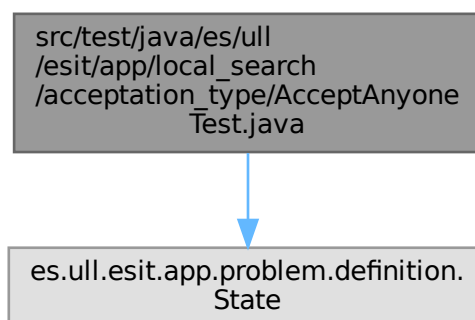
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.136. Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptAnyoneTest.java ↩

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en AcceptAnyoneTest.java:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptAnyoneTest`

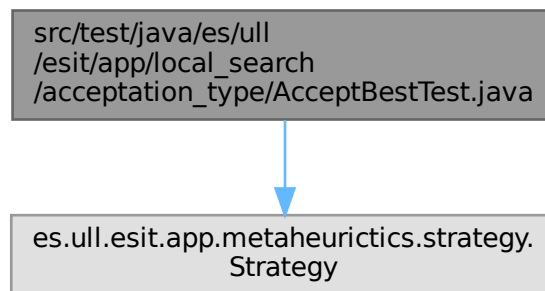
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.137. Referencia del archivo `src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptBestTest.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `AcceptBestTest.java`:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptBestTest`

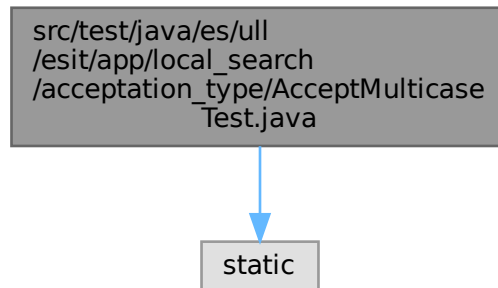
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.138. Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptMulticaseTest.java

```
import static;
```

Gráfico de dependencias incluidas en AcceptMulticaseTest.java:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptMulticaseTest`

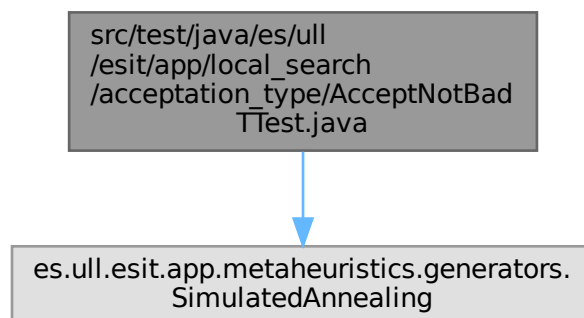
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.139. Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadTTest.java

```
import es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing;
```

Gráfico de dependencias incluidas en AcceptNotBadTTest.java:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptNotBadTTest`

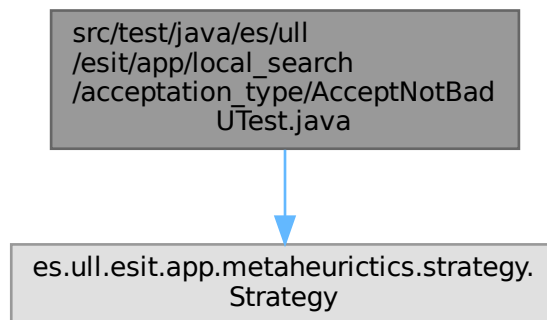
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.140. Referencia del archivo `src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadUTest.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `AcceptNotBadUTest.java`:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptNotBadUTest`

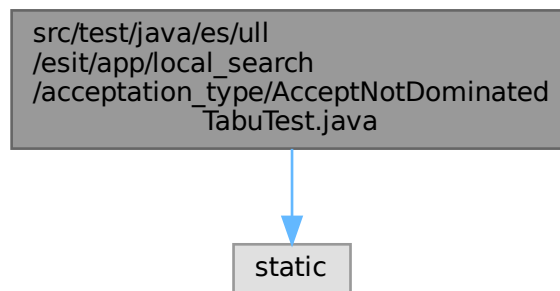
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.141. Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominatedTabuTest.java

```
import static;
```

Gráfico de dependencias incluidas en AcceptNotDominatedTabuTest.java:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabuTest`

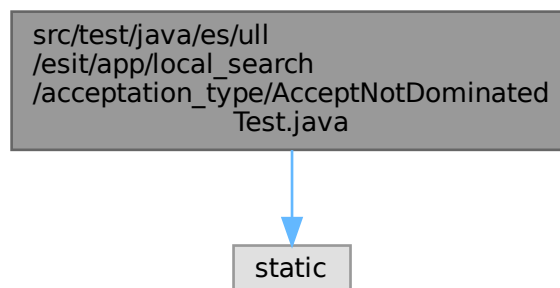
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.142. Referencia del archivo src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotDominatedTest.java

```
import static;
```

Gráfico de dependencias incluidas en AcceptNotDominatedTest.java:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTest`

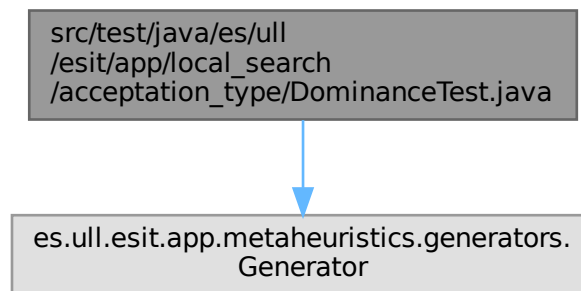
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.143. Referencia del archivo `src/test/java/es/ull/esit/app/local_search/acceptation_type/DominanceTest.java`

```
import es.ull.esit.app.metaheuristics.generators.Generator;
```

Gráfico de dependencias incluidas en `DominanceTest.java`:



Clases

- class `es.ull.esit.app.local_search.acceptation_type.DominanceTest`

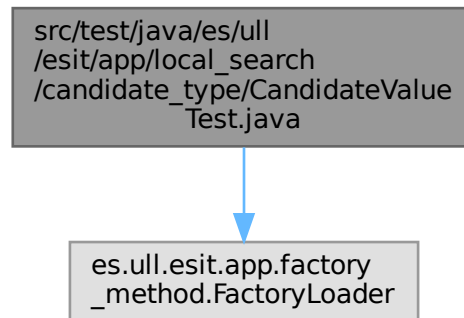
Paquetes

- package `es.ull.esit.app.local_search.acceptation_type`

7.144. Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/CandidateValueTest.java ↩

```
import es.ull.esit.app.factory_method.FactoryLoader;
```

Gráfico de dependencias incluidas en CandidateValueTest.java:



Clases

- class `es.ull.esit.app.local_search.candidate_type.CandidateValueTest`

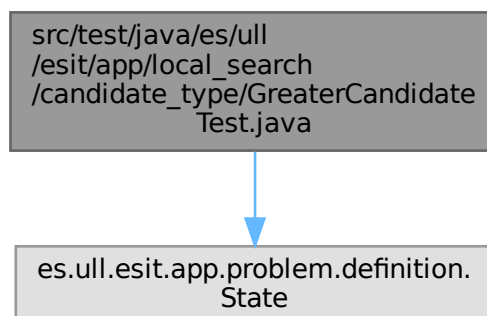
Paquetes

- package `es.ull.esit.app.local_search.candidate_type`

7.145. Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/GreaterCandidateTest.java ↩

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en GreaterCandidateTest.java:



Clases

- class `es.ull.esit.app.local_search.candidate_type.GreaterCandidateTest`

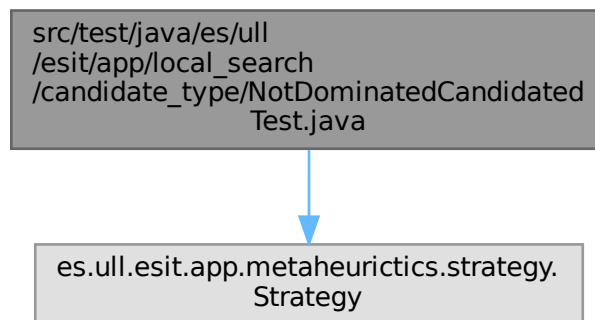
Paquetes

- package `es.ull.esit.app.local_search.candidate_type`

7.146. Referencia del archivo `src/test/java/es/ull/esit/app/local_search/candidate_type/NotDominatedCandidatedTest.java`

```
import es.ull.esit.app.metaheuristics.strategy.Strategy;
```

Gráfico de dependencias incluidas en `NotDominatedCandidatedTest.java`:



Clases

- class `es.ull.esit.app.local_search.candidate_type.NotDominatedCandidateTest`

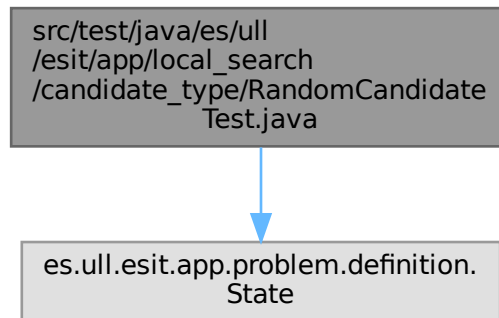
Paquetes

- package `es.ull.esit.app.local_search.candidate_type`

7.147. Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/RandomCandidateTest.java ↩

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en RandomCandidateTest.java:



Clases

- class `es.ull.esit.app.local_search.candidate_type.RandomCandidateTest`

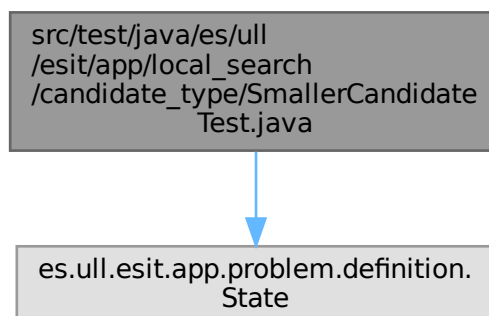
Paquetes

- package `es.ull.esit.app.local_search.candidate_type`

7.148. Referencia del archivo src/test/java/es/ull/esit/app/local_search/candidate_type/SmallerCandidateTest.java ↩

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en SmallerCandidateTest.java:



Clases

- class `es.ull.esit.app.local_search.candidate_type.SmallerCandidateTest`

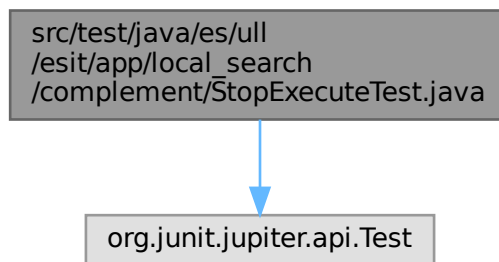
Paquetes

- package `es.ull.esit.app.local_search.candidate_type`

7.149. Referencia del archivo `src/test/java/es/ull/esit/app/local_search/complement/StopExecuteTest.java`

```
import org.junit.jupiter.api.Test;
```

Gráfico de dependencias incluidas en `StopExecuteTest.java`:



Clases

- class `es.ull.esit.app.local_search.complement.StopExecuteTest`

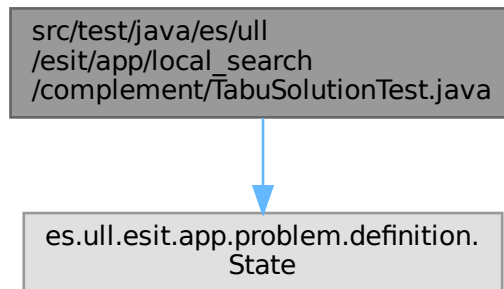
Paquetes

- package `es.ull.esit.app.local_search.complement`

7.150. Referencia del archivo src/test/java/es/ull/esit/app/local_search/complement/TabuSolutionTest.java

```
import es.ull.esit.app.problem.definition.State;
```

Gráfico de dependencias incluidas en TabuSolutionTest.java:



Clases

- class **es.ull.esit.app.local_search.complement.TabuSolutionsTest**

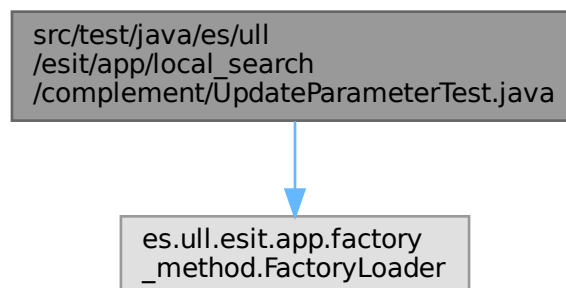
Paquetes

- package [es.ull.esit.app.local_search.complement](#)

7.151. Referencia del archivo src/test/java/es/ull/esit/app/local_search/complement/UpdateParameterTest.java

```
import es.ull.esit.app.factory_method.FactoryLoader;
```

Gráfico de dependencias incluidas en UpdateParameterTest.java:



Clases

- class **es.ull.esit.app.local_search.complement.UpdateParameterTest**

Paquetes

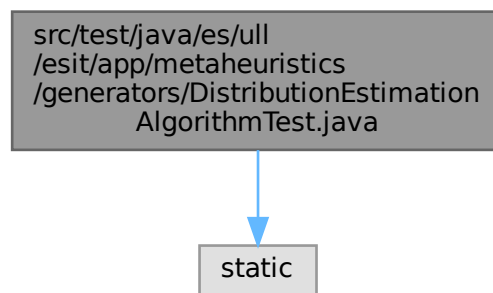
- package [es.ull.esit.app.local_search.complement](#)

7.152. Referencia del archivo

**src/test/java/es/ull/esit/app/metaheuristics/generators/↵
DistributionEstimationAlgorithmTest.java**

```
import static;
```

Gráfico de dependencias incluidas en DistributionEstimationAlgorithmTest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithmTest**

Paquetes

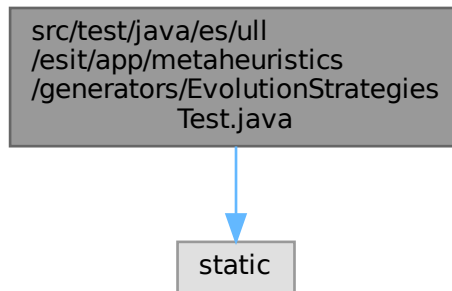
- package [es.ull.esit.app.metaheuristics.generators](#)

7.153. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/EvolutionStrategiesTest.java

```
import static;
```

Gráfico de dependencias incluidas en EvolutionStrategiesTest.java:



Clases

- class `es.ull.esit.app.metaheuristics.generators.EvolutionStrategiesTest`

Paquetes

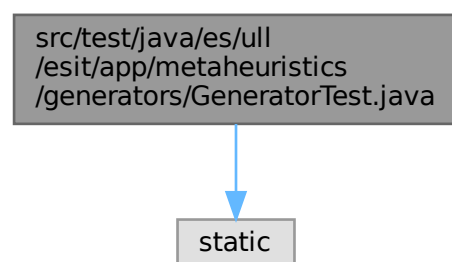
- package `es.ull.esit.app.metaheuristics.generators`

7.154. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/GeneratorTest.java

```
import static;
```

Gráfico de dependencias incluidas en GeneratorTest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.GeneratorTest**

Paquetes

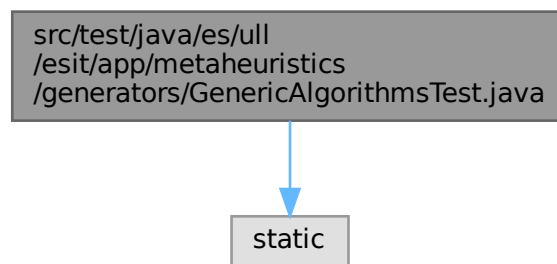
- package [es.ull.esit.app.metaheuristics.generators](#)

7.155. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/GenericAlgorithmsTest.java

```
import static;
```

Gráfico de dependencias incluidas en GenericAlgorithmsTest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.GeneticAlgorithmTest**

Paquetes

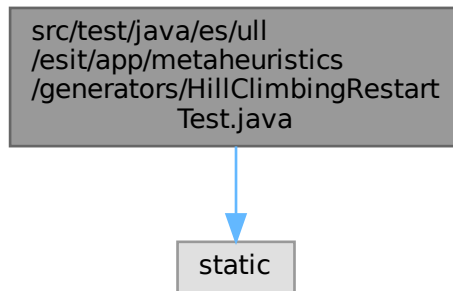
- package [es.ull.esit.app.metaheuristics.generators](#)

7.156. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/HillClimbingRestartTest.java↔

```
import static;
```

Gráfico de dependencias incluidas en HillClimbingRestartTest.java:



Clases

- class `es.ull.esit.app.metaheuristics.generators.HillClimbingRestartTest`

Paquetes

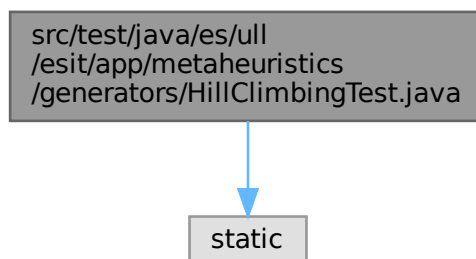
- package `es.ull.esit.app.metaheuristics.generators`

7.157. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/HillClimbingTest.java↔

```
import static;
```

Gráfico de dependencias incluidas en HillClimbingTest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.HillClimbingTest**

Paquetes

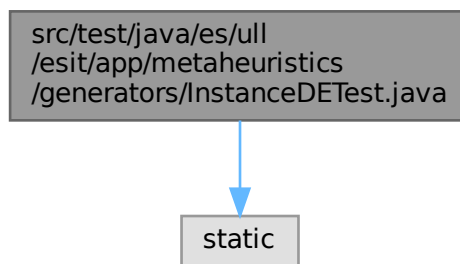
- package [es.ull.esit.app.metaheuristics.generators](#)

7.158. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/InstanceDETest.java

```
import static;
```

Gráfico de dependencias incluidas en InstanceDETest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.InstanceDETest**

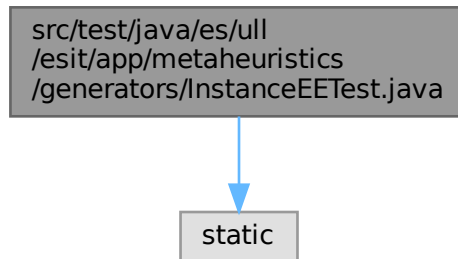
Paquetes

- package [es.ull.esit.app.metaheuristics.generators](#)

7.159. Referencia del archivo `src/test/java/es/ull/esit/app/metaheuristics/generators/InstanceEETest.java`

```
import static;
```

Gráfico de dependencias incluidas en `InstanceEETest.java`:



Clases

- class `es.ull.esit.app.metaheuristics.generators.InstanceEETest`

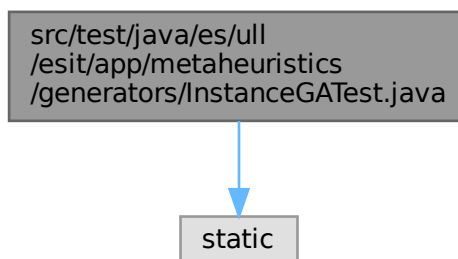
Paquetes

- package `es.ull.esit.app.metaheuristics.generators`

7.160. Referencia del archivo `src/test/java/es/ull/esit/app/metaheuristics/generators/InstanceGATest.java`

```
import static;
```

Gráfico de dependencias incluidas en `InstanceGATest.java`:



Clases

- class **es.ull.esit.app.metaheuristics.generators.InstanceGATest**

Paquetes

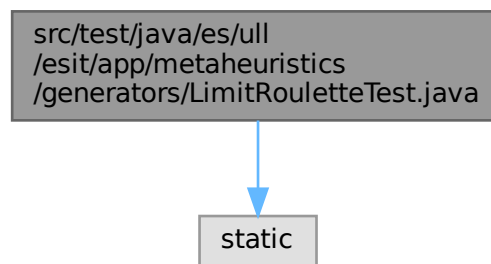
- package [es.ull.esit.app.metaheuristics.generators](#)

7.161. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/LimitRouletteTest.java↔

```
import static;
```

Gráfico de dependencias incluidas en LimitRouletteTest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.LimitRouletteTest**

Paquetes

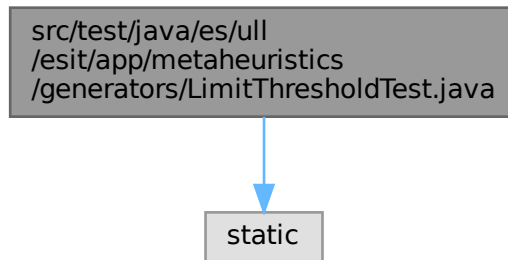
- package [es.ull.esit.app.metaheuristics.generators](#)

7.162. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/LimitThresholdTest.java

```
import static;
```

Gráfico de dependencias incluidas en LimitThresholdTest.java:



Clases

- class `es.ull.esit.app.metaheuristics.generators.LimitThresholdTest`

Paquetes

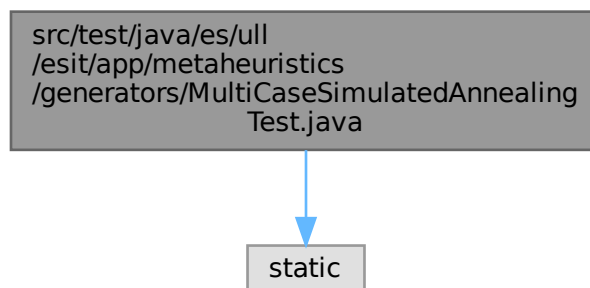
- package `es.ull.esit.app.metaheuristics.generators`

7.163. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/MultiCaseSimulatedAnnealingTest.java

```
import static;
```

Gráfico de dependencias incluidas en MultiCaseSimulatedAnnealingTest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealingTest**

Paquetes

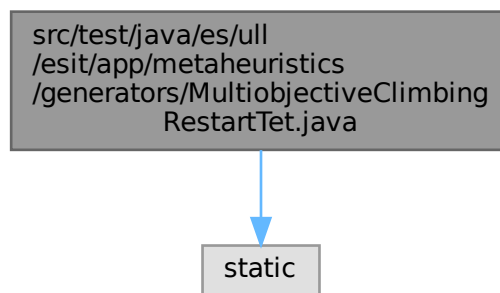
- package [es.ull.esit.app.metaheuristics.generators](#)

7.164. Referencia del archivo

**src/test/java/es/ull/esit/app/metaheuristics/generators/↵
MultiobjectiveClimbingRestartTet.java**

```
import static;
```

Gráfico de dependencias incluidas en MultiobjectiveClimbingRestartTet.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestartTest**

Paquetes

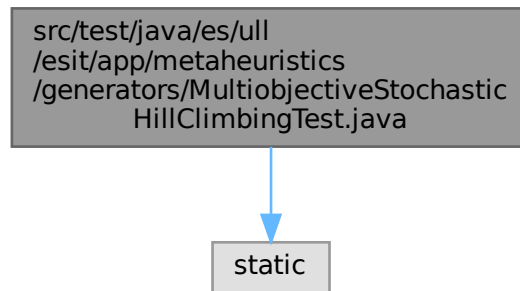
- package [es.ull.esit.app.metaheuristics.generators](#)

7.165. Referencia del archivo

**src/test/java/es/ull/esit/app/metaheuristics/generators/↵
MultiobjectiveStochasticHillClimbingTest.java**

```
import static;
```

Gráfico de dependencias incluidas en MultiobjectiveStochasticHillClimbingTest.java:



Clases

- class **es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbingTest**

Paquetes

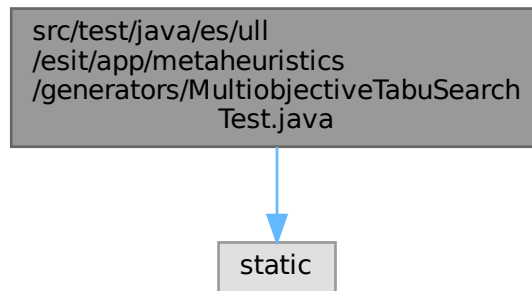
- package [es.ull.esit.app.metaheuristics.generators](#)

7.166. Referencia del archivo

**src/test/java/es/ull/esit/app/metaheuristics/generators/↵
MultiobjectiveTabuSearchTest.java**

```
import static;
```

Gráfico de dependencias incluidas en MultiobjectiveTabuSearchTest.java:



Clases

- class `es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearchTest`

Paquetes

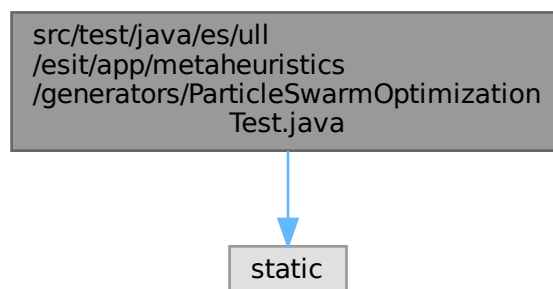
- package `es.ull.esit.app.metaheuristics.generators`

7.167. Referencia del archivo

`src/test/java/es/ull/esit/app/metaheuristics/generators/ParticleSwarmOptimizationTest.java`

```
import static;
```

Gráfico de dependencias incluidas en ParticleSwarmOptimizationTest.java:



Clases

- class `es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimizationTest`

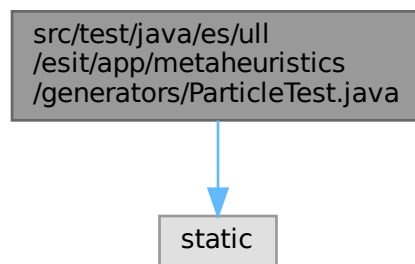
Paquetes

- package `es.ull.esit.app.metaheuristics.generators`

7.168. Referencia del archivo `src/test/java/es/ull/esit/app/metaheuristics/generators/ParticleTest.java`

```
import static;
```

Gráfico de dependencias incluidas en `ParticleTest.java`:



Clases

- class `es.ull.esit.app.metaheuristics.generators.ParticleTest`

Paquetes

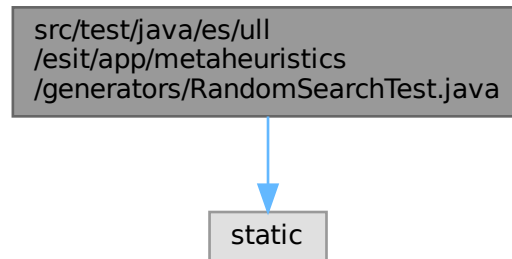
- package `es.ull.esit.app.metaheuristics.generators`

7.169. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/RandomSearchTest.java↔

```
import static;
```

Gráfico de dependencias incluidas en RandomSearchTest.java:



Clases

- class `es.ull.esit.app.metaheuristics.generators.RandomSearchTest`

Paquetes

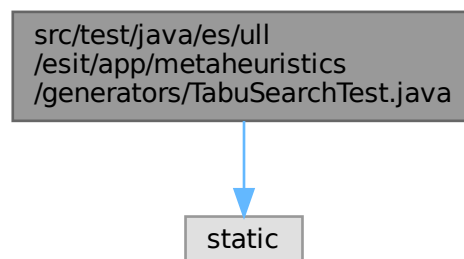
- package `es.ull.esit.app.metaheuristics.generators`

7.170. Referencia del archivo

src/test/java/es/ull/esit/app/metaheuristics/generators/TabuSearchTest.java↔

```
import static;
```

Gráfico de dependencias incluidas en TabuSearchTest.java:



Clases

- class `es.ull.esit.app.metaheuristics.generators.TabuSearchTest`

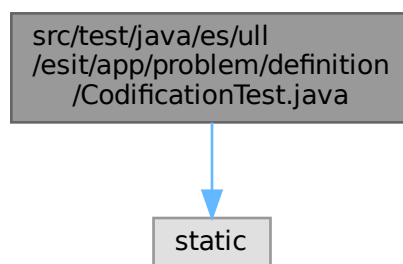
Paquetes

- package `es.ull.esit.app.metaheuristics.generators`

7.171. Referencia del archivo `src/test/java/es/ull/esit/app/problem/definition/CodificationTest.java`

```
import static;
```

Gráfico de dependencias incluidas en `CodificationTest.java`:



Clases

- class `es.ull.esit.app.problem.definition.CodificationTest`
- class `es.ull.esit.app.problem.definition.CodificationTest.DummyCodification`

Paquetes

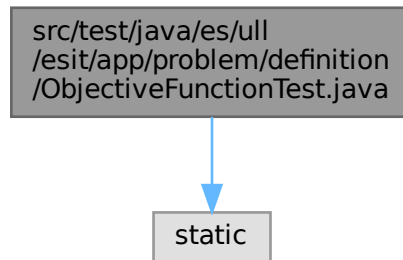
- package `es.ull.esit.app.problem.definition`

7.172. Referencia del archivo

src/test/java/es/ull/esit/app/problem/definition/ObjectiveFunctionTest.java

```
import static;
```

Gráfico de dependencias incluidas en ObjectiveFunctionTest.java:



Clases

- class `es.ull.esit.app.problem.definition.ObjectiveFunctionTest`
- class `es.ull.esit.app.problem.definition.ObjectiveFunctionTest.DummyObjectiveFunction`

Paquetes

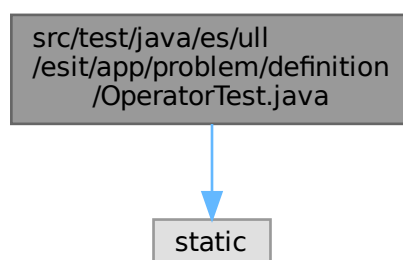
- package `es.ull.esit.app.problem.definition`

7.173. Referencia del archivo

src/test/java/es/ull/esit/app/problem/definition/OperatorTest.java

```
import static;
```

Gráfico de dependencias incluidas en OperatorTest.java:



Clases

- class `es.ull.esit.app.problem.definition.OperatorTest`
- class `es.ull.esit.app.problem.definition.OperatorTest.DummyOperator`

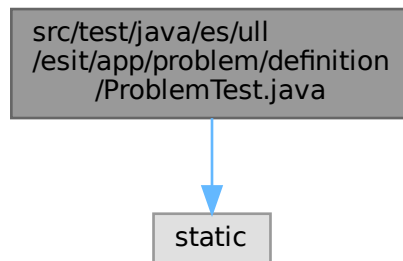
Paquetes

- package `es.ull.esit.app.problem.definition`

7.174. Referencia del archivo `src/test/java/es/ull/esit/app/problem/definition/ProblemTest.java`

```
import static;
```

Gráfico de dependencias incluidas en `ProblemTest.java`:



Clases

- class `es.ull.esit.app.problem.definition.ProblemTest`
- class `es.ull.esit.app.problem.definition.ProblemTest.LengthObjectiveFunction`
- class `es.ull.esit.app.problem.definition.ProblemTest.DummyCodification`
- class `es.ull.esit.app.problem.definition.ProblemTest.DummyOperator`

Paquetes

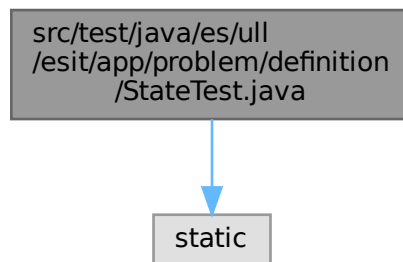
- package `es.ull.esit.app.problem.definition`

7.175. Referencia del archivo

src/test/java/es/ull/esit/app/problem/definition/StateTest.java

```
import static;
```

Gráfico de dependencias incluidas en StateTest.java:



Clases

- class `es.ull.esit.app.problem.definition.StateTest`

Paquetes

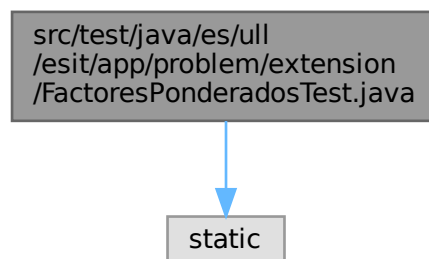
- package `es.ull.esit.app.problem.definition`

7.176. Referencia del archivo

src/test/java/es/ull/esit/app/problem/extension/FactoresPonderadosTest.java

```
import static;
```

Gráfico de dependencias incluidas en FactoresPonderadosTest.java:



Clases

- class `es.ull.esit.app.problem.extension.FactoresPonderadosTest`

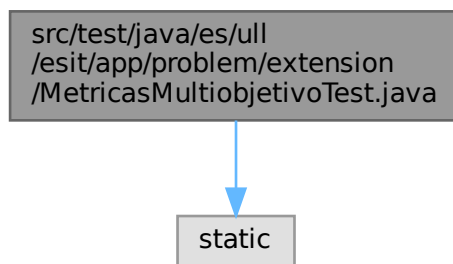
Paquetes

- package `es.ull.esit.app.problem.extension`

7.177. Referencia del archivo `src/test/java/es/ull/esit/app/problem/extension/MetricasMultiobjetivoTest.java`

```
import static;
```

Gráfico de dependencias incluidas en `MetricasMultiobjetivoTest.java`:



Clases

- class `es.ull.esit.app.problem.extension.MetricasMultiobjetivoTest`

Paquetes

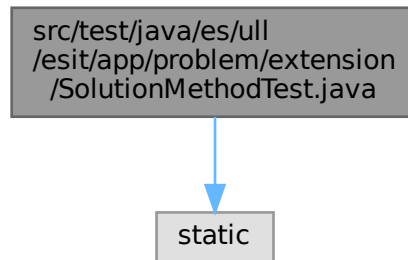
- package `es.ull.esit.app.problem.extension`

7.178. Referencia del archivo

src/test/java/es/ull/esit/app/problem/extension/SolutionMethodTest.java↔

```
import static;
```

Gráfico de dependencias incluidas en SolutionMethodTest.java:



Clases

- class `es.ull.esit.app.problem.extension.SolutionMethodTest`
- class `es.ull.esit.app.problem.extension.SolutionMethodTest.DummySolutionMethod`

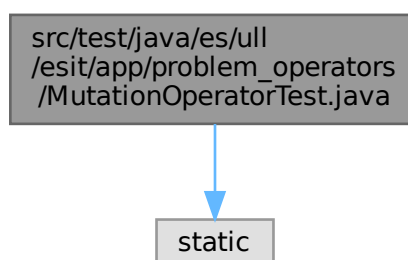
Paquetes

- package `es.ull.esit.app.problem.extension`

7.179. Referencia del archivo `src/test/java/es/ull/esit/app/problem_operators/MutationOperatorTest.java`↔

```
import static;
```

Gráfico de dependencias incluidas en MutationOperatorTest.java:



Clases

- class `es.ull.esit.app.problem_operators.MutationOperatorTest`

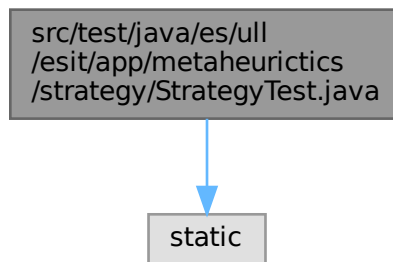
Paquetes

- package `es.ull.esit.app.problem_operators`

7.180. Referencia del archivo `src/test/java/es/ull/esit/app/metaheuristics/strategy/StrategyTest.java`

```
import static;
```

Gráfico de dependencias incluidas en `StrategyTest.java`:



Clases

- class `es.ull.esit.app.metaheuristics.strategy.StrategyTest`

Paquetes

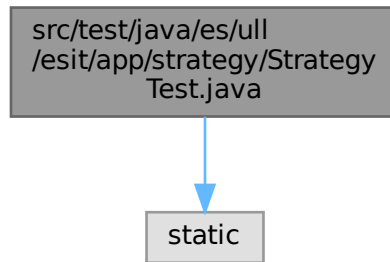
- package `es.ull.esit.app.metaheuristics.strategy`

7.181. Referencia del archivo

src/test/java/es/ull/esit/app/strategy/StrategyTest.java

```
import static;
```

Gráfico de dependencias incluidas en StrategyTest.java:



Clases

- class **es.ull.esit.app.strategy.StrategyTest**

Paquetes

- package [es.ull.esit.app.strategy](#)

Índice alfabético

AcceptAnyone	ascOrderBetter
es.ull.esit.app.local_search.acceptation_type.AcceptType, es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelectio	
35	317
AcceptBest	awardUpdateREF
es.ull.esit.app.local_search.acceptation_type.AcceptType, es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgori	
35	49
acceptCandidate	es.ull.esit.app.metaheuristics.generators.EvolutionStrategies,
es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate,	
22	es.ull.esit.app.metaheuristics.generators.Generator,
es.ull.esit.app.local_search.acceptation_type.AcceptAnyone,	92
24	es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm,
es.ull.esit.app.local_search.acceptation_type.AcceptBest,	103
25	es.ull.esit.app.metaheuristics.generators.HillClimbing,
es.ull.esit.app.local_search.acceptation_type.AcceptMulticase,	113
27	es.ull.esit.app.metaheuristics.generators.HillClimbingRestart,
es.ull.esit.app.local_search.acceptation_type.AcceptNotBad,	119
29	es.ull.esit.app.metaheuristics.generators.LimitThreshold,
es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT,	145
30	es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing,
es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU,	154
32	es.ull.esit.app.metaheuristics.generators.MultiGenerator,
es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated,	161
33	es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDis,
es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu,	171
35	es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRe
AcceptMulticase	179
es.ull.esit.app.local_search.acceptation_type.AcceptType, es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillC	
36	185
AcceptNotBad	es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch,
es.ull.esit.app.local_search.acceptation_type.AcceptType,	192
36	es.ull.esit.app.metaheuristics.generators.Particle,
AcceptNotBadT	213
es.ull.esit.app.local_search.acceptation_type.AcceptType, es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization,	
36	221
AcceptNotBadU	es.ull.esit.app.metaheuristics.generators.RandomSearch,
es.ull.esit.app.local_search.acceptation_type.AcceptType,	246
36	es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing,
AcceptNotDominated	263
es.ull.esit.app.local_search.acceptation_type.AcceptType, es.ull.esit.app.metaheuristics.generators.TabuSearch,	
36	308
AcceptNotDominatedTabu	
es.ull.esit.app.local_search.acceptation_type.AcceptType,	317
36	es.ull.esit.app.problem.extension.MetricasMultiobjetivo,
activeGenerator	150
es.ull.esit.app.metaheuristics.generators.MultiGenerator,	
169	es.ull.esit.app.problem.extension.MetricasMultiobjetivo,
AIO_MUTATION	150
es.ull.esit.app.evolutionary_algorithms.complement.MutationType,	
202	es.ull.esit.app.problem.extension.MetricasMultiobjetivo,

- 151
- calcularMedia
 - es.ull.esit.app.problem.extension.MetricasMultiobjetivo, 151
- calcularMin
 - es.ull.esit.app.problem.extension.MetricasMultiobjetivo, 152
- calcularTasaError
 - es.ull.esit.app.problem.extension.MetricasMultiobjetivo, 152
- calculateOffLinePerformance
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 288
 - es.ull.esit.app.strategy.Strategy, 298
- CandidateValue
 - es.ull.esit.app.local_search.candidate_type.CandidateValue, 40
- candidatevalue
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance, 175
- code
 - es.ull.esit.app.problem.definition.State, 283
- comparator
 - es.ull.esit.app.problem.definition.State, 280
- copy
 - es.ull.esit.app.metaheuristics.generators.MultiGenerator, 163
 - es.ull.esit.app.problem.definition.State, 280
- countBetterGender
 - es.ull.esit.app.metaheuristics.generators.Generator, 98
- countGender
 - es.ull.esit.app.metaheuristics.generators.Generator, 98
- createAcceptCandidate
 - es.ull.esit.app.factory_interface.IFFactoryAcceptCandidate, 126
 - es.ull.esit.app.factory_method.FactoryAcceptCandidate, 70
- createCrossover
 - es.ull.esit.app.factory_interface.IFFactoryCrossover, 128
 - es.ull.esit.app.factory_method.FactoryCrossover, 73
- createDistribution
 - es.ull.esit.app.factory_interface.IFFactoryDistribution, 129
 - es.ull.esit.app.factory_method.FactoryDistribution, 75
- createdSolutionMethod
 - es.ull.esit.app.factory_interface.IFFactorySolutionMethod, 134
 - es.ull.esit.app.factory_method.FactorySolutionMethod, 87
- createGenerator
 - es.ull.esit.app.factory_interface.IFFactoryGenerator, 131
 - es.ull.esit.app.factory_method.FactoryGenerator, 79
- createInstanceGeneratorsBPP
 - es.ull.esit.app.metaheuristics.generators.MultiGenerator, 163
- createMutation
 - es.ull.esit.app.factory_interface.IFFactoryMutation, 132
- createReplace
 - es.ull.esit.app.factory_interface.IFFactoryReplace, 133
 - es.ull.esit.app.factory_method.FactoryReplace, 83
- createSampling
 - es.ull.esit.app.factory_interface.IFFSampling, 135
 - es.ull.esit.app.factory_method.FactorySampling, 85
- createSearchCandidate
 - es.ull.esit.app.factory_interface.IFFactoryCandidate, 171
 - es.ull.esit.app.factory_method.FactoryCandidate, 71
- createSelectFather
 - es.ull.esit.app.factory_interface.IFFactoryFatherSelection, 130
 - es.ull.esit.app.factory_method.FactoryFatherSelection, 77
- crossover
 - es.ull.esit.app.evolutionary_algorithms.complement.Crossover, 45
 - es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover, 207
 - es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover, 321
- destroyExecute
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 288
 - es.ull.esit.app.strategy.Strategy, 298
- destroyMultiGenerator
 - es.ull.esit.app.metaheuristics.generators.MultiGenerator, 163
- distance
 - es.ull.esit.app.problem.definition.State, 280
- distanceCalculateAdd
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance, 171
- distanceSolution
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance, 175
- distribution
 - es.ull.esit.app.evolutionary_algorithms.complement.Distribution, 47
 - es.ull.esit.app.evolutionary_algorithms.complement.Univariate, 323
- DistributionEstimationAlgorithm
 - es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 49
 - es.ull.esit.app.metaheuristics.generators.GeneratorType, 99

- dominance
 - es.ull.esit.app.local_search.acceptation_type.Dominance, 58
- es.ull.esit.app.config.tspdynamic, 15
- es.ull.esit.app.config.tspdynamic.TSPState, 318
 - getIdCity, 318
 - getValue, 318
 - setIdCity, 318
 - setValue, 319
- es.ull.esit.app.evolutionary_algorithms.complement, 15
- es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation, 37
 - fillPath, 38
 - mutation, 38
 - path, 39
 - sortedPathValue, 38
- es.ull.esit.app.evolutionary_algorithms.complement.Crossover, 45
 - crossover, 45
- es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType, 46
 - ONE_POINT_CROSSOVER, 46
 - UNIFORM_CROSSOVER, 46
- es.ull.esit.app.evolutionary_algorithms.complement.Distribution, 46
 - distribution, 47
- es.ull.esit.app.evolutionary_algorithms.complement.DistributionType, 57
 - UNIVARIATE, 57
- es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection, 87
 - selection, 88
- es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace, 89
 - replace, 90
- es.ull.esit.app.evolutionary_algorithms.complement.Mutation, 199
 - mutation, 199
- es.ull.esit.app.evolutionary_algorithms.complement.MutationType, 202
 - AIO_MUTATION, 202
 - ONE_POINT_MUTATION, 202
 - TOW_POINTS_MUTATION, 202
- es.ull.esit.app.evolutionary_algorithms.complement.OnePointCrossover, 206
 - crossover, 207
- es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation, 208
 - mutation, 209
- es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling, 233
 - listState, 234
 - sampling, 234
- es.ull.esit.app.evolutionary_algorithms.complement.Probability, 235
 - getKey, 235
 - getProbability, 235
 - getValue, 235
 - setKey, 235
 - setProbability, 236
 - setValue, 236
- es.ull.esit.app.evolutionary_algorithms.complement.Range, 251
 - getData, 251
 - getMax, 251
 - getMin, 251
 - setData, 252
 - setMax, 252
 - setMin, 252
- es.ull.esit.app.evolutionary_algorithms.complement.Replace, 253
 - replace, 253
- es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType, 254
 - GENERATIONAL_REPLACE, 254
 - STEADY_STATE_REPLACE, 254
- es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection, 255
 - selection, 256
- es.ull.esit.app.evolutionary_algorithms.complement.Sampling, 257
 - sampling, 257
- es.ull.esit.app.evolutionary_algorithms.complement.SamplingType, 258
 - PROBABILISTIC_SAMPLING, 258
- es.ull.esit.app.evolutionary_algorithms.complement.SelectionType, 260
 - ROULETTE_SELECTION, 260
 - TRUNCATION_SELECTION, 260
- es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace, 284
 - maxValue, 285
 - minValue, 285
 - replace, 285
- es.ull.esit.app.evolutionary_algorithms.complement.TowPointsMutation, 314
 - mutation, 315
- es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection, 316
 - ascOrderBetter, 317
 - orderBetter, 317
 - selection, 317
- es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover, 320
 - crossover, 321
- es.ull.esit.app.evolutionary_algorithms.complement.Univariate, 322
 - distribution, 323
 - getListKey, 323
- es.ull.esit.app.factory_interface, 16
- es.ull.esit.app.factory_interface.IFFactoryAcceptCandidate, 126
 - createAcceptCandidate, 126
- es.ull.esit.app.factory_interface.IFFactoryCandidate, 127

- createSearchCandidate, 127
- es.ull.esit.app.factory_interface.IFFactoryCrossover, 128
 - createCrossover, 128
- es.ull.esit.app.factory_interface.IFFactoryDistribution, 129
 - createDistribution, 129
- es.ull.esit.app.factory_interface.IFFactoryFatherSelection, 130
 - createSelectFather, 130
- es.ull.esit.app.factory_interface.IFFactoryGenerator, 131
 - createGenerator, 131
- es.ull.esit.app.factory_interface.IFFactoryMutation, 132
 - createMutation, 132
- es.ull.esit.app.factory_interface.IFFactoryReplace, 133
 - createReplace, 133
- es.ull.esit.app.factory_interface.IFFactorySolutionMethod, 134
 - createdSolutionMethod, 134
- es.ull.esit.app.factory_interface.IFFSampling, 135
 - createSampling, 135
- es.ull.esit.app.factory_method, 16
- es.ull.esit.app.factory_method.FactoryAcceptCandidate, 69
 - createAcceptCandidate, 70
- es.ull.esit.app.factory_method.FactoryCandidate, 70
 - createSearchCandidate, 71
- es.ull.esit.app.factory_method.FactoryCrossover, 72
 - createCrossover, 73
- es.ull.esit.app.factory_method.FactoryDistribution, 74
 - createDistribution, 75
- es.ull.esit.app.factory_method.FactoryFatherSelection, 76
 - createSelectFather, 77
- es.ull.esit.app.factory_method.FactoryGenerator, 78
 - createGenerator, 79
- es.ull.esit.app.factory_method.FactoryLoader, 79
 - getInstance, 80
- es.ull.esit.app.factory_method.FactoryMutation, 80
 - createMutation, 81
- es.ull.esit.app.factory_method.FactoryReplace, 82
 - createReplace, 83
- es.ull.esit.app.factory_method.FactorySampling, 84
 - createSampling, 85
- es.ull.esit.app.factory_method.FactorySolutionMethod, 86
 - createdSolutionMethod, 87
- es.ull.esit.app.local_search.acceptation_type, 17
- es.ull.esit.app.local_search.acceptation_type.AcceptableCandidate, 21
 - acceptCandidate, 22
- es.ull.esit.app.local_search.acceptation_type.AcceptAnyone, 23
 - acceptCandidate, 24
- es.ull.esit.app.local_search.acceptation_type.AcceptBest, 24
 - acceptCandidate, 25
- es.ull.esit.app.local_search.acceptation_type.AcceptMulticase, 26
 - acceptCandidate, 27
- es.ull.esit.app.local_search.acceptation_type.AcceptNotBad, 28
 - acceptCandidate, 29
- es.ull.esit.app.local_search.acceptation_type.AcceptNotBadT, 29
 - acceptCandidate, 30
- es.ull.esit.app.local_search.acceptation_type.AcceptNotBadU, 31
 - acceptCandidate, 32
- es.ull.esit.app.local_search.acceptation_type.AcceptNotDominated, 32
 - acceptCandidate, 33
- es.ull.esit.app.local_search.acceptation_type.AcceptNotDominatedTabu, 34
 - acceptCandidate, 35
- es.ull.esit.app.local_search.acceptation_type.AcceptType, 35
 - AcceptAnyone, 35
 - AcceptBest, 35
 - AcceptMulticase, 36
 - AcceptNotBad, 36
 - AcceptNotBadT, 36
 - AcceptNotBadU, 36
 - AcceptNotDominated, 36
 - AcceptNotDominatedTabu, 36
- es.ull.esit.app.local_search.acceptation_type.Dominance, 58
 - dominance, 58
 - listDominance, 58
- es.ull.esit.app.local_search.candidate_type, 17
- es.ull.esit.app.local_search.candidate_type.CandidateType, 39
 - GreaterCandidate, 39
 - NotDominatedCandidate, 39
 - RandomCandidate, 39
 - SmallerCandidate, 39
- es.ull.esit.app.local_search.candidate_type.CandidateValue, 40
 - CandidateValue, 40
 - getTabusolution, 41
 - newSearchCandidate, 41
 - setTabusolution, 42
 - stateCandidate, 42
- es.ull.esit.app.local_search.candidate_type.GreaterCandidate, 110
 - stateSearch, 111
- es.ull.esit.app.local_search.candidate_type.NotDominatedCandidate, 203
 - stateSearch, 204
- es.ull.esit.app.local_search.candidate_type.RandomCandidate, 243
 - stateSearch, 244
- es.ull.esit.app.local_search.candidate_type.SearchCandidate, 259
 - stateSearch, 259

- es.ull.esit.app.local_search.candidate_type.SmallerCandidates, [276](#)
 - stateSearch, [277](#)
- es.ull.esit.app.local_search.complement, [18](#)
- es.ull.esit.app.local_search.complement.StopExecute, [286](#)
 - stopIterations, [286](#)
- es.ull.esit.app.local_search.complement.StrategyType, [306](#)
 - NORMAL, [306](#)
 - TABU, [306](#)
- es.ull.esit.app.local_search.complement.TabuSolutions, [313](#)
 - filterNeighborhood, [313](#)
 - listTabu, [314](#)
 - maxelements, [314](#)
- es.ull.esit.app.local_search.complement.UpdateParameter, [323](#)
 - updateParameter, [324](#)
- es.ull.esit.app.metaheuristics.strategy, [18](#)
- es.ull.esit.app.metaheuristics.strategy.Strategy, [287](#)
 - calculateOffLinePerformance, [288](#)
 - destroyExecute, [288](#)
 - executeStrategy, [288](#)
 - getBestState, [289](#)
 - getCountCurrent, [289](#)
 - getCountMax, [289](#)
 - getGenerator, [289](#)
 - getListKey, [289](#)
 - getListRefPoblacFinal, [290](#)
 - getListStates, [290](#)
 - getMapGenerators, [290](#)
 - getProblem, [290](#)
 - getStopexecute, [290](#)
 - getStrategy, [291](#)
 - getThreshold, [291](#)
 - getUpdateparameter, [291](#)
 - initialize, [291](#)
 - initializeGenerators, [292](#)
 - newGenerator, [292](#)
 - setBestState, [293](#)
 - setCountCurrent, [293](#)
 - setCountMax, [293](#)
 - setGenerator, [293](#)
 - setListRefPoblacFinal, [294](#)
 - setListStates, [294](#)
 - setMapGenerators, [294](#)
 - setProblem, [294](#)
 - setStopexecute, [295](#)
 - setThreshold, [295](#)
 - setUpdateparameter, [295](#)
 - update, [295](#)
 - updateCountGender, [296](#)
 - updateRef, [296](#)
 - updateRefGenerator, [296](#)
 - updateRefMultiG, [297](#)
 - updateWeight, [297](#)
- es.ull.esit.app.metaheuristics.generators, [18](#)
- es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, [47](#)
 - awardUpdateREF, [49](#)
 - DistributionEstimationAlgorithm, [49](#)
 - generate, [49](#)
 - getCountRef, [50](#)
 - getDistributionType, [50](#)
 - getfathersList, [50](#)
 - getGeneratorType, [51](#)
 - getListCountBetterGender, [51](#)
 - getListCountGender, [51](#)
 - getListReference, [52](#)
 - getListStateRef, [52](#)
 - getReference, [52](#)
 - getReferenceList, [52](#)
 - getReplaceType, [53](#)
 - getSelectionType, [53](#)
 - getSonList, [53](#)
 - getTrace, [53](#)
 - getType, [54](#)
 - getWeight, [54](#)
 - maxValue, [54](#)
 - setCountRef, [55](#)
 - setDistributionType, [55](#)
 - setGeneratorType, [55](#)
 - setInitialReference, [55](#)
 - setListReference, [56](#)
 - setReplaceType, [56](#)
 - setSelectionType, [56](#)
 - setWeight, [56](#)
 - updateReference, [57](#)
- es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, [59](#)
 - awardUpdateREF, [61](#)
 - EvolutionStrategies, [60](#)
 - generate, [62](#)
 - getCountRef, [62](#)
 - getListCountBetterGender, [62](#)
 - getListCountGender, [63](#)
 - getListStateRef, [63](#)
 - getListStateReference, [63](#)
 - getReference, [63](#)
 - getReferenceList, [64](#)
 - getSonList, [64](#)
 - getTrace, [64](#)
 - getType, [64](#)
 - getTypeGenerator, [65](#)
 - getWeight, [65](#)
 - setCountRef, [65](#)
 - setInitialReference, [65](#)
 - setListStateReference, [66](#)
 - setStateRef, [66](#)
 - setTypeGenerator, [66](#)
 - setWeight, [66](#)
 - updateReference, [67](#)
- es.ull.esit.app.metaheuristics.generators.Generator, [91](#)
 - awardUpdateREF, [92](#)
 - countBetterGender, [98](#)

- countGender, 98
- generate, 92
- getListCountBetterGender, 93
- getListCountGender, 93
- getReference, 94
- getReferenceList, 94
- getSonList, 94
- getTrace, 95
- getType, 95
- getWeight, 95
- listCountBetterGender, 98
- setInitialReference, 96
- setWeight, 96
- updateReference, 97
- es.ull.esit.app.metaheuristics.generators.GeneratorType, 98
 - DistributionEstimationAlgorithm, 99
 - EvolutionStrategies, 99
 - GeneticAlgorithm, 99
 - HillClimbing, 99
 - HillClimbingDistance, 99
 - HillClimbingRestart, 99
 - LimitThreshold, 99
 - MultiCaseSimulatedAnnealing, 100
 - MultiGenerator, 100
 - MultiobjectiveHillClimbingDistance, 100
 - MultiobjectiveHillClimbingRestart, 100
 - MultiobjectiveStochasticHillClimbing, 100
 - MultiobjectiveTabuSearch, 100
 - ParticleSwarmOptimization, 100
 - RandomSearch, 100
 - SimulatedAnnealing, 101
 - StochasticHillClimbing, 101
 - TabuSearch, 101
- es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 101
 - awardUpdateREF, 103
 - generate, 103
 - GeneticAlgorithm, 103
 - getCountRef, 104
 - getGeneratorType, 104
 - getListCountBetterGender, 104
 - getListCountGender, 104
 - getListState, 105
 - getListStateRef, 105
 - getReference, 105
 - getReferenceList, 105
 - getSonList, 106
 - getTrace, 106
 - getType, 106
 - getWeight, 106
 - setCountRef, 107
 - setGeneratorType, 108
 - setInitialReference, 108
 - setListState, 108
 - setStateRef, 108
 - setWeight, 109
 - updateReference, 109
- es.ull.esit.app.metaheuristics.generators.HillClimbing, 112
 - awardUpdateREF, 113
 - generate, 114
 - getGeneratorType, 114
 - getListCountBetterGender, 114
 - getListCountGender, 114
 - getReference, 114
 - getReferenceList, 115
 - getSonList, 115
 - getTrace, 115
 - getType, 115
 - getWeight, 116
 - HillClimbing, 113
 - setGeneratorType, 116
 - setInitialReference, 116
 - setStateRef, 116
 - setTypeCandidate, 117
 - setWeight, 117
 - updateReference, 117
- es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 118
 - awardUpdateREF, 119
 - generate, 120
 - getCount, 120
 - getCountBetterGender, 120
 - getCountCurrent, 120
 - getCountGender, 121
 - getGeneratorType, 121
 - getListCountBetterGender, 121
 - getListCountGender, 121
 - getReference, 122
 - getReferenceList, 122
 - getSonList, 122
 - getTrace, 122
 - getType, 123
 - getWeight, 123
 - HillClimbingRestart, 119
 - setCount, 123
 - setCountBetterGender, 123
 - setCountCurrent, 124
 - setCountGender, 124
 - setGeneratorType, 124
 - setInitialReference, 124
 - setStateRef, 125
 - setTypeCandidate, 125
 - setWeight, 125
 - updateReference, 125
- es.ull.esit.app.metaheuristics.generators.InstanceDE, 136
 - isTerminate, 137
 - run, 137
 - setTerminate, 137
- es.ull.esit.app.metaheuristics.generators.InstanceEE, 138
 - isTerminate, 139
 - run, 139
 - setTerminate, 139

- es.ull.esit.app.metaheuristics.generators.InstanceGA,
 - 140
 - isTerminate, 141
 - run, 141
 - setTerminate, 141
- es.ull.esit.app.metaheuristics.generators.LimitRoulette,
 - 141
 - getGenerator, 142
 - getLimitHigh, 142
 - getLimitLow, 142
 - setGenerator, 142
 - setLimitHigh, 143
 - setLimitLow, 143
- es.ull.esit.app.metaheuristics.generators.LimitThreshold,
 - 143
 - awardUpdateREF, 145
 - generate, 145
 - getGeneratorType, 146
 - getListCountBetterGender, 146
 - getListCountGender, 146
 - getReference, 146
 - getReferenceList, 147
 - getSonList, 147
 - getTrace, 147
 - getType, 147
 - getWeight, 148
 - LimitThreshold, 145
 - setGeneratorType, 148
 - setInitialReference, 148
 - setStateRef, 148
 - setTypeCandidate, 149
 - setWeight, 149
 - updateReference, 149
- es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing,
 - 153
 - awardUpdateREF, 154
 - generate, 155
 - getListCountBetterGender, 155
 - getListCountGender, 156
 - getReference, 156
 - getReferenceList, 156
 - getSonList, 156
 - getTinitial, 157
 - getTrace, 157
 - getType, 157
 - getTypeGenerator, 157
 - getWeight, 158
 - MultiCaseSimulatedAnnealing, 154
 - setInitialReference, 158
 - setStateRef, 158
 - setTfinal, 158
 - setTinitial, 159
 - setTypeGenerator, 159
 - setWeight, 159
 - tfinal, 159
 - updateReference, 160
- es.ull.esit.app.metaheuristics.generators.MultiGenerator,
 - 161
 - activeGenerator, 169
 - awardUpdateREF, 163
 - copy, 163
 - createInstanceGeneratorsBPP, 163
 - destroyMultiGenerator, 163
 - generate, 163
 - getActiveGenerator, 164
 - getListCountBetterGender, 164
 - getListCountGender, 164
 - getListGenerators, 164
 - getReference, 165
 - getReferenceList, 165
 - getSonList, 165
 - getTrace, 165
 - getType, 165
 - getWeight, 166
 - initializeGenerators, 166
 - initializeListGenerator, 166
 - listGeneratedPP, 169
 - listStateReference, 169
 - MultiGenerator, 163
 - roulette, 166
 - searchState, 166
 - setActiveGenerator, 166
 - setGeneratorType, 167
 - setInitialReference, 167
 - setListGeneratedPP, 167
 - setListGenerators, 167
 - setWeight, 167
 - tournament, 168
 - updateAwardImp, 168
 - updateAwardSC, 168
 - updateReference, 168
 - updateWeight, 168
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,
 - 169
 - awardUpdateREF, 171
 - candidatevalue, 175
 - distanceCalculateAdd, 171
 - distanceSolution, 175
 - generate, 171
 - generatorType, 175
 - getGeneratorType, 172
 - getListCountBetterGender, 172
 - getListCountGender, 172
 - getReference, 172
 - getReferenceList, 172
 - getSonList, 173
 - getTrace, 173
 - getType, 173
 - getWeight, 173
 - ifacceptCandidate, 176
 - listStateReference, 176
 - listTrace, 176
 - MultiobjectiveHillClimbingDistance, 171
 - setGeneratorType, 174
 - setInitialReference, 174
 - setStateRef, 174

- setWeight, 174
 - sizeNeighbors, 176
 - stateReferenceHC, 176
 - strategy, 176
 - typeAcceptation, 176
 - typeCandidate, 176
 - updateReference, 175
 - weight, 176
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, 177
 - awardUpdateREF, 179
 - generate, 179
 - getGeneratorType, 179
 - getListCountBetterGender, 179
 - getListCountGender, 180
 - getReference, 180
 - getReferenceList, 180
 - getSizeNeighbors, 180
 - getSonList, 181
 - getTrace, 181
 - getType, 181
 - getWeight, 181
 - MultiobjectiveHillClimbingRestart, 178
 - setGeneratorType, 182
 - setInitialReference, 182
 - setSizeNeighbors, 182
 - setStateRef, 183
 - setWeight, 183
 - updateReference, 183
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing, 184
 - awardUpdateREF, 185
 - generate, 186
 - getGeneratorType, 186
 - getListCountBetterGender, 186
 - getListCountGender, 186
 - getReference, 187
 - getReferenceList, 187
 - getSonList, 187
 - getTrace, 187
 - getType, 188
 - getWeight, 188
 - MultiobjectiveStochasticHillClimbing, 185
 - setGeneratorType, 188
 - setInitialReference, 189
 - setStateRef, 189
 - setWeight, 189
 - updateReference, 189
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 190
 - awardUpdateREF, 192
 - generate, 192
 - getListCountBetterGender, 193
 - getListCountGender, 193
 - getReference, 193
 - getReferenceList, 193
 - getSonList, 194
 - getStateReferenceTS, 194
 - getTrace, 194
 - getType, 194
 - getTypeGenerator, 195
 - getWeight, 195
 - MultiobjectiveTabuSearch, 192
 - setInitialReference, 195
 - setStateRef, 195
 - setStateReferenceTS, 196
 - setTypeCandidate, 196
 - setTypeGenerator, 196
 - setWeight, 196
 - updateReference, 197
- es.ull.esit.app.metaheuristics.generators.Particle, 211
 - awardUpdateREF, 213
 - generate, 214
 - getListCountBetterGender, 214
 - getListCountGender, 214
 - getReference, 214
 - getReferenceList, 215
 - getSonList, 215
 - getStateActual, 215
 - getStatePBest, 215
 - getTrace, 216
 - getType, 216
 - getVelocity, 216
 - getWeight, 216
 - Particle, 213
 - setInitialReference, 216
 - setStateActual, 217
 - setStatePBest, 217
 - setVelocity, 217
 - setWeight, 217
 - updateReference, 218
- es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 219
 - awardUpdateREF, 221
 - gBestInical, 221
 - generate, 222
 - getConstriction, 222
 - getCountCurrentIterPSO, 222
 - getCountParticle, 222
 - getCountParticleBySwarm, 223
 - getCountRef, 223
 - getGBest, 223
 - getGeneratorType, 223
 - getLBest, 224
 - getLearning1, 224
 - getLearning2, 224
 - getListCountBetterGender, 224
 - getListCountGender, 225
 - getListParticle, 225
 - getListStateReference, 225
 - getReference, 225
 - getReferenceList, 226
 - getSonList, 226
 - getStateReferencePSO, 226
 - getTrace, 226
 - getType, 227

- getWeight, [227](#)
- getWmax, [227](#)
- getWmin, [227](#)
- inicialiceLBest, [228](#)
- isBinary, [228](#)
- ParticleSwarmOptimization, [221](#)
- setBinary, [228](#)
- setConstriction, [228](#)
- setCountCurrentIterPSO, [229](#)
- setCountRef, [229](#)
- setGBest, [229](#)
- setGeneratorType, [229](#)
- setInitialReference, [230](#)
- setLearning1, [230](#)
- setLearning2, [230](#)
- setListParticle, [230](#)
- setListStateReference, [231](#)
- setStateReferencePSO, [231](#)
- setWeight, [231](#)
- setWmax, [231](#)
- setWmin, [232](#)
- updateReference, [232](#)
- es.ull.esit.app.metaheuristics.generators.RandomSearch, [244](#)
 - awardUpdateREF, [246](#)
 - generate, [246](#)
 - getListCountBetterGender, [246](#)
 - getListCountGender, [247](#)
 - getListStateReference, [247](#)
 - getReference, [247](#)
 - getReferenceList, [247](#)
 - getSonList, [248](#)
 - getTrace, [248](#)
 - getType, [248](#)
 - getTypeGenerator, [248](#)
 - getWeight, [249](#)
 - RandomSearch, [246](#)
 - setInitialReference, [249](#)
 - setListStateReference, [249](#)
 - setTypeGenerator, [249](#)
 - setWeight, [250](#)
 - updateReference, [250](#)
- es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, [261](#)
 - awardUpdateREF, [263](#)
 - generate, [264](#)
 - getAlpha, [265](#)
 - getCountIterationsT, [265](#)
 - getListCountBetterGender, [265](#), [266](#)
 - getListCountGender, [266](#)
 - getReference, [266](#), [267](#)
 - getReferenceList, [267](#)
 - getSonList, [267](#), [268](#)
 - getTfinal, [268](#)
 - getTinitial, [268](#), [269](#)
 - getTrace, [269](#)
 - getType, [269](#), [270](#)
 - getTypeGenerator, [270](#)
 - getWeight, [270](#), [271](#)
 - setAlpha, [271](#)
 - setCountIterationsT, [271](#), [272](#)
 - setInitialReference, [272](#)
 - setStateRef, [272](#), [273](#)
 - setTfinal, [273](#)
 - setTinitial, [273](#), [274](#)
 - setTypeGenerator, [274](#)
 - setWeight, [274](#), [275](#)
 - SimulatedAnnealing, [263](#)
 - updateReference, [275](#)
- es.ull.esit.app.metaheuristics.generators.TabuSearch, [307](#)
 - awardUpdateREF, [308](#)
 - generate, [309](#)
 - getListCountBetterGender, [309](#)
 - getListCountGender, [309](#)
 - getReference, [309](#)
 - getReferenceList, [310](#)
 - getSonList, [310](#)
 - getTrace, [310](#)
 - getType, [310](#)
 - getTypeGenerator, [311](#)
 - getWeight, [311](#)
 - setInitialReference, [311](#)
 - setStateRef, [311](#)
 - setTypeCandidate, [312](#)
 - setTypeGenerator, [312](#)
 - setWeight, [312](#)
 - TabuSearch, [308](#)
 - updateReference, [312](#)
- es.ull.esit.app.problem.definition, [19](#)
- es.ull.esit.app.problem.definition.Codification, [43](#)
 - getAleatoryKey, [43](#)
 - getVariableAleatoryValue, [43](#)
 - getVariableCount, [44](#)
 - validState, [44](#)
- es.ull.esit.app.problem.definition.ObjectiveFunction, [204](#)
 - evaluation, [205](#)
 - getTypeProblem, [205](#)
 - getWeight, [205](#)
 - setTypeProblem, [205](#)
 - setWeight, [206](#)
- es.ull.esit.app.problem.definition.Operator, [210](#)
 - generatedNewState, [210](#)
 - generateRandomState, [211](#)
- es.ull.esit.app.problem.definition.Problem, [236](#)
 - evaluate, [237](#)
 - getCodification, [238](#)
 - getFactorySolutionMethod, [238](#)
 - getFunction, [238](#)
 - getOperator, [238](#)
 - getPossibleValue, [238](#)
 - getState, [239](#)
 - getTypeProblem, [239](#)
 - getTypeSolutionMethod, [239](#)
 - newSolutionMethod, [239](#)
 - Problem, [237](#)

- setCodification, 240
 - setFactorySolutionMethod, 240
 - setFunction, 240
 - setOperator, 241
 - setPossibleValue, 241
 - setState, 241
 - setTypeProblem, 241
 - setTypeSolutionMethod, 242
- es.ull.esit.app.problem.definition.Problem.ProblemType, 242
 - MAXIMIZAR, 242
 - MINIMIZAR, 242
- es.ull.esit.app.problem.definition.State, 279
 - code, 283
 - comparator, 280
 - copy, 280
 - distance, 280
 - evaluation, 283
 - getCode, 281
 - getCopy, 281
 - getEvaluation, 281
 - getNumber, 281
 - getTypeGenerator, 282
 - number, 283
 - setCode, 282
 - setEvaluation, 282
 - setNumber, 282
 - setTypeGenerator, 283
 - State, 279, 280
 - typeGenerator, 283
- es.ull.esit.app.problem.extension, 19
- es.ull.esit.app.problem.extension.FactoresPonderados, 68
 - evaluationState, 68
- es.ull.esit.app.problem.extension.MetricasMultiobjetivo, 150
 - calcularDispersion, 150
 - calcularDistanciaGeneracional, 150
 - calcularMax, 151
 - calcularMedia, 151
 - calcularMin, 152
 - calcularTasaError, 152
- es.ull.esit.app.problem.extension.MultiObjetivoPuro, 198
 - evaluationState, 198
- es.ull.esit.app.problem.extension.SolutionMethod, 278
 - evaluationState, 278
- es.ull.esit.app.problem.extension.TypeSolutionMethod, 319
 - FactoresPonderados, 319
 - MultiObjetivoPuro, 319
- es.ull.esit.app.problem_operators, 20
- es.ull.esit.app.problem_operators.MutationOperator, 200
 - generatedNewState, 201
 - generateRandomState, 201
- es.ull.esit.app.strategy, 20
- es.ull.esit.app.strategy.Strategy, 297
 - calculateOffLinePerformance, 298
 - destroyExecute, 298
 - executeStrategy, 298
 - generator, 305
 - getBestState, 299
 - getCountCurrent, 299
 - getCountMax, 299
 - getListKey, 299
 - getProblem, 300
 - getStopexecute, 300
 - getStrategy, 300
 - getThreshold, 300
 - getUpdateparameter, 300
 - initialize, 301
 - initializeGenerators, 301
 - newGenerator, 301
 - notDominated, 305
 - setBestState, 302
 - setCountCurrent, 302
 - setCountMax, 303
 - setProblem, 303
 - setStopexecute, 303
 - setThreshold, 303
 - setUpdateparameter, 303
 - threshold, 305
 - update, 304
 - updateCountGender, 304
 - updateRef, 304
 - updateRefGenerator, 305
 - updateRefMultiG, 305
 - updateWeight, 305
- evaluate
 - es.ull.esit.app.problem.definition.Problem, 237
- evaluation
 - es.ull.esit.app.problem.definition.ObjetivoFunction, 205
 - es.ull.esit.app.problem.definition.State, 283
- evaluationState
 - es.ull.esit.app.problem.extension.FactoresPonderados, 68
 - es.ull.esit.app.problem.extension.MultiObjetivoPuro, 198
 - es.ull.esit.app.problem.extension.SolutionMethod, 278
- EvolutionStrategies
 - es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 60
 - es.ull.esit.app.metaheuristics.generators.GeneratorType, 99
- executeStrategy
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 288
 - es.ull.esit.app.strategy.Strategy, 298
- FactoresPonderados
 - es.ull.esit.app.problem.extension.TypeSolutionMethod, 319
- fillPath

es.ull.esit.app.evolutionary_algorithms.complement.AlgorithmType
38

filterNeighborhood
es.ull.esit.app.local_search.complement.TabuSolution
313

gBestInicial
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization,
221

generate
es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm,
49
es.ull.esit.app.metaheuristics.generators.EvolutionStrategies
62
es.ull.esit.app.metaheuristics.generators.Generator,
92
es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm
103
es.ull.esit.app.metaheuristics.generators.HillClimbing,
114
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart
120
es.ull.esit.app.metaheuristics.generators.LimitThreshold
145
es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing,
155
es.ull.esit.app.metaheuristics.generators.MultiGenerator,
163
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance
171
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart
179
es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing,
186
es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch
192
es.ull.esit.app.metaheuristics.generators.Particle,
214
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization
222
es.ull.esit.app.metaheuristics.generators.RandomSearch,
246
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing
264
es.ull.esit.app.metaheuristics.generators.TabuSearch
309

generatedNewState
es.ull.esit.app.problem_definition.Operator, 210
es.ull.esit.app.problem_operators.MutationOperator,
201

generateRandomState
es.ull.esit.app.problem_definition.Operator, 211
es.ull.esit.app.problem_operators.MutationOperator,
201

GENERATIONAL_REPLACE
es.ull.esit.app.evolutionary_algorithms.complement.ReplaceType
254

generator
es.ull.esit.app.strategy.Strategy, 305

es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDi
175
es.ull.esit.app.metaheuristics.generators.GeneratorType,
99
es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm,
103
getActiveGenerator
es.ull.esit.app.metaheuristics.generators.MultiGenerator,
164
getAleatoryKey
es.ull.esit.app.problem_definition.Codification, 43
getAlpha
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing,
265
getBestState
es.ull.esit.app.metaheuristics.strategy.Strategy,
289
es.ull.esit.app.strategy.Strategy, 299
getCode
es.ull.esit.app.problem_definition.State, 281
getCodification
es.ull.esit.app.problem_definition.Problem, 238
getConstruction
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization,
222
getCopy
es.ull.esit.app.problem_definition.State, 281
getCount
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart,
120
getCountBetterGender
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart,
120
getCountCurrent
es.ull.esit.app.metaheuristics.strategy.Strategy,
289
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart,
120
es.ull.esit.app.strategy.Strategy, 299
getCountCurrentIterPSO
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization,
222
getCountGender
es.ull.esit.app.metaheuristics.generators.HillClimbingRestart,
121
getCountIterationsT
es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing,
265
getCountMax
es.ull.esit.app.metaheuristics.strategy.Strategy,
289
es.ull.esit.app.strategy.Strategy, 299
getCountParticle
es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization,
222
getCountParticleBySwarm

- 63
- es.ull.esit.app.metaheuristics.generators.Generator, 93
- es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 104
- es.ull.esit.app.metaheuristics.generators.HillClimbing, 114
- es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 121
- es.ull.esit.app.metaheuristics.generators.LimitThreshold, 146
- es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 156
- es.ull.esit.app.metaheuristics.generators.MultiGenerator, 164
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbing, 172
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, 180
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing, 186
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 193
- es.ull.esit.app.metaheuristics.generators.Particle, 214
- es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 225
- es.ull.esit.app.metaheuristics.generators.RandomSearch, 247
- es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 266
- es.ull.esit.app.metaheuristics.generators.TabuSearch, 309
- getListGenerators
 - es.ull.esit.app.metaheuristics.generators.MultiGenerator, 164
- getListKey
 - es.ull.esit.app.evolutionary_algorithms.complement.Univariate, 323
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 289
 - es.ull.esit.app.strategy.Strategy, 299
- getListParticle
 - es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 225
- getListReference
 - es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 52
- getListRefPoblacFinal
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 290
- getListState
 - es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 105
- getListStateRef
 - es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 52
 - es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 180
- 63
- es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 105
- es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 63
- es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 225
- es.ull.esit.app.metaheuristics.generators.RandomSearch, 247
- getListStates
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 290
- getMapGenerators
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 290
- getDistance, 280
- getMax
 - es.ull.esit.app.evolutionary_algorithms.complement.Range, 251
- getMin
 - es.ull.esit.app.evolutionary_algorithms.complement.Range, 251
- getNumber
 - es.ull.esit.app.problem.definition.State, 281
- getOperator
 - es.ull.esit.app.problem.definition.Problem, 238
- getPossibleValue
 - es.ull.esit.app.problem.definition.Problem, 238
- getProbability
 - es.ull.esit.app.evolutionary_algorithms.complement.Probability, 235
- getProblem
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 290
 - es.ull.esit.app.strategy.Strategy, 300
- getReference
 - es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 52
 - es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 63
 - es.ull.esit.app.metaheuristics.generators.Generator, 94
 - es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 105
 - es.ull.esit.app.metaheuristics.generators.HillClimbing, 114
 - es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 122
 - es.ull.esit.app.metaheuristics.generators.LimitThreshold, 146
 - es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 156
 - es.ull.esit.app.metaheuristics.generators.MultiGenerator, 165
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance, 178
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, 180

es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing, [187](#)
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, [193](#)
 es.ull.esit.app.metaheuristics.generators.Particle, [214](#)
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, [225](#)
 es.ull.esit.app.metaheuristics.generators.RandomSearch, [247](#)
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, [266](#), [267](#)
 es.ull.esit.app.metaheuristics.generators.TabuSearch, [309](#)
 getReferenceList
 es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, [52](#)
 es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, [64](#)
 es.ull.esit.app.metaheuristics.generators.Generator, [94](#)
 es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, [105](#)
 es.ull.esit.app.metaheuristics.generators.HillClimbing, [115](#)
 es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, [122](#)
 es.ull.esit.app.metaheuristics.generators.LimitThreshold, [147](#)
 es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, [156](#)
 es.ull.esit.app.metaheuristics.generators.MultiGenerator, [165](#)
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistribution, [173](#)
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, [181](#)
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing, [187](#)
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, [194](#)
 es.ull.esit.app.metaheuristics.generators.Particle, [215](#)
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, [226](#)
 es.ull.esit.app.metaheuristics.generators.RandomSearch, [248](#)
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, [267](#), [268](#)
 es.ull.esit.app.metaheuristics.generators.TabuSearch, [310](#)
 es.ull.esit.app.metaheuristics.generators.TabuSearch, [193](#)
 es.ull.esit.app.metaheuristics.generators.Particle, [215](#)
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, [226](#)
 es.ull.esit.app.metaheuristics.generators.RandomSearch, [247](#)
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, [267](#)
 es.ull.esit.app.metaheuristics.generators.TabuSearch, [310](#)
 getStateActual
 es.ull.esit.app.metaheuristics.generators.Particle, [215](#)
 getStatePBest
 es.ull.esit.app.metaheuristics.generators.Particle, [215](#)
 getStateReferencePSO
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, [226](#)
 getStateReferenceTS
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, [194](#)
 getStopexecute
 es.ull.esit.app.metaheuristics.strategy.Strategy, [300](#)
 getStrategy
 es.ull.esit.app.problem.definition.Problem, [239](#)
 getSelectionType
 es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, [53](#)
 getSizeNeighbors

es.ull.esit.app.metaheuristics.strategy.Strategy, 291
 es.ull.esit.app.strategy.Strategy, 300
 getTabusolution
 es.ull.esit.app.local_search.candidate_type.CandidateValue, 41
 getTfinal
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 268
 getThreshold
 es.ull.esit.app.metaheuristics.strategy.Strategy, 291
 es.ull.esit.app.strategy.Strategy, 300
 getTinitial
 es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 157
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 268, 269
 getTrace
 es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 53
 es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 64
 es.ull.esit.app.metaheuristics.generators.Generator, 95
 es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 106
 es.ull.esit.app.metaheuristics.generators.HillClimbing, 115
 es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 122
 es.ull.esit.app.metaheuristics.generators.LimitThreshold, 147
 es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 157
 es.ull.esit.app.metaheuristics.generators.MultiGenerator, 165
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDi, 173
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRe, 173
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillC, 173
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 194
 es.ull.esit.app.metaheuristics.generators.Particle, 216
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 226
 es.ull.esit.app.metaheuristics.generators.RandomSearch, 248
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 269, 270
 es.ull.esit.app.metaheuristics.generators.TabuSearch, 311
 getTypeGenerator
 es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 65
 es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 157
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDi, 173
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRe, 173
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillC, 173
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 194
 es.ull.esit.app.metaheuristics.generators.Particle, 216
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 226
 es.ull.esit.app.metaheuristics.generators.RandomSearch, 248
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 269, 270
 es.ull.esit.app.metaheuristics.generators.TabuSearch, 311
 getTypeProblem
 es.ull.esit.app.problem.definition.State, 282
 es.ull.esit.app.problem.definition.ObjectiveFunction, 205
 es.ull.esit.app.problem.definition.Problem, 239
 getTypeSolutionMethod
 es.ull.esit.app.problem.definition.Problem, 239
 getUpdateparameter
 es.ull.esit.app.metaheuristics.strategy.Strategy, 291
 es.ull.esit.app.strategy.Strategy, 300

getValue 39
 es.ull.esit.app.config.tspdynamic.TSPState, 318
 es.ull.esit.app.evolutionary_algorithms.complement.Probability, 235
 getVariableAleatoryValue 99
 es.ull.esit.app.problem.definition.Codification, 43
 getVariableCount 113
 es.ull.esit.app.problem.definition.Codification, 44
 getVelocity 99
 es.ull.esit.app.metaheuristics.generators.Particle, 216
 getWeight 99
 es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 54
 es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 65
 es.ull.esit.app.metaheuristics.generators.Generator, 95
 es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 106
 es.ull.esit.app.metaheuristics.generators.HillClimbing, 116
 es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 123
 es.ull.esit.app.metaheuristics.generators.LimitThreshold, 148
 es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 158
 es.ull.esit.app.metaheuristics.generators.MultiGenerator, 166
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance, 173
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, 181
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing, 188
 es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 195
 es.ull.esit.app.metaheuristics.generators.Particle, 216
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 227
 es.ull.esit.app.metaheuristics.generators.RandomSearch, 249
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 270, 271
 es.ull.esit.app.metaheuristics.generators.TabuSearch, 311
 es.ull.esit.app.problem.definition.ObjectiveFunction, 205
 getWmax 227
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 227
 getWmin 227
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 227
 GreaterCandidate 58
 es.ull.esit.app.local_search.candidate_type.CandidateType, 58

es.ull.esit.app.metaheuristics.generators.MultiGenerator,	es.ull.esit.app.metaheuristics.generators.GeneratorType,
169	100
listState	es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing,
es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling,	35
234	MultiobjectiveTabuSearch
listStateReference	es.ull.esit.app.metaheuristics.generators.GeneratorType,
es.ull.esit.app.metaheuristics.generators.MultiGenerator,	100
169	es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch,
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,	100
176	MultiObjetivoPuro
listTabu	es.ull.esit.app.problem.extension.TypeSolutionMethod,
es.ull.esit.app.local_search.complement.TabuSolutions,	319
314	mutation
listTrace	es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation,
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,	58
176	es.ull.esit.app.evolutionary_algorithms.complement.Mutation,
	199
maskara	es.ull.esit.app.evolutionary_algorithms.complement.OnePointMutation,
es.ull.esit.app.evolutionary_algorithms.complement.UniformCrossover,	259
321	es.ull.esit.app.evolutionary_algorithms.complement.TwoPointsMutation,
maxelements	315
es.ull.esit.app.local_search.complement.TabuSolutions,	
314	newGenerator
MAXIMIZAR	es.ull.esit.app.metaheuristics.strategy.Strategy,
es.ull.esit.app.problem.definition.Problem.ProblemType,	292
242	es.ull.esit.app.strategy.Strategy, 301
maxValue	newSearchCandidate
es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplapp,	41
285	es.ull.esit.app.local_search.candidate_type.CandidateValue,
es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm,	54
54	es.ull.esit.app.problem.definition.Problem, 239
MINIMIZAR	NORMAL
es.ull.esit.app.problem.definition.Problem.ProblemType,	es.ull.esit.app.local_search.complement.StrategyType,
242	306
minValue	notDominated
es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplapp,	305
285	NotDominatedCandidate
MultiCaseSimulatedAnnealing	es.ull.esit.app.local_search.candidate_type.CandidateType,
es.ull.esit.app.metaheuristics.generators.GeneratorType,	39
100	number
es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing,	es.ull.esit.app.problem.definition.State, 283
154	
MultiGenerator	ONE_POINT_CROSSOVER
es.ull.esit.app.metaheuristics.generators.GeneratorType,	es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType,
100	46
es.ull.esit.app.metaheuristics.generators.MultiGenerator,	ONE_POINT_MUTATION
163	es.ull.esit.app.evolutionary_algorithms.complement.MutationType,
MultiobjectiveHillClimbingDistance	202
es.ull.esit.app.metaheuristics.generators.GeneratorType,	orderBetter
100	es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection,
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,	317
171	
MultiobjectiveHillClimbingRestart	Particle
es.ull.esit.app.metaheuristics.generators.GeneratorType,	es.ull.esit.app.metaheuristics.generators.Particle,
100	213
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart,	ParticleSwarmOptimization
178	es.ull.esit.app.metaheuristics.generators.GeneratorType,
MultiobjectiveStochasticHillClimbing	100

es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 221
 path
 es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation, 39
 PROBABILISTIC_SAMPLING
 es.ull.esit.app.evolutionary_algorithms.complement.SamplingType, 258
 Problem
 es.ull.esit.app.problem.definition.Problem, 237
 RandomCandidate
 es.ull.esit.app.local_search.candidate_type.CandidateType, 39
 RandomSearch
 es.ull.esit.app.metaheuristics.generators.GeneratorType, 100
 es.ull.esit.app.metaheuristics.generators.RandomSearch, 246
 replace
 es.ull.esit.app.evolutionary_algorithms.complement.GenerationalReplace, 90
 es.ull.esit.app.evolutionary_algorithms.complement.Replace, 253
 es.ull.esit.app.evolutionary_algorithms.complement.SteadyStateReplace, 285
 roulette
 es.ull.esit.app.metaheuristics.generators.MultiGenerator, 166
 ROULETTE_SELECTION
 es.ull.esit.app.evolutionary_algorithms.complement.SelectionType, 260
 run
 es.ull.esit.app.metaheuristics.generators.InstanceDE, 137
 es.ull.esit.app.metaheuristics.generators.InstanceEE, 139
 es.ull.esit.app.metaheuristics.generators.InstanceGA, 141
 sampling
 es.ull.esit.app.evolutionary_algorithms.complement.ProbabilisticSampling, 234
 es.ull.esit.app.evolutionary_algorithms.complement.Sampling, 257
 searchState
 es.ull.esit.app.metaheuristics.generators.MultiGenerator, 166
 selection
 es.ull.esit.app.evolutionary_algorithms.complement.FatherSelection, 88
 es.ull.esit.app.evolutionary_algorithms.complement.RouletteSelection, 256
 es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection, 317
 setActiveGenerator
 es.ull.esit.app.metaheuristics.generators.MultiGenerator, 166
 setAlpha
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 271
 setBestState
 es.ull.esit.app.metaheuristics.strategy.Strategy, 293
 es.ull.esit.app.strategy.Strategy, 302
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 228
 setCode
 es.ull.esit.app.problem.definition.State, 282
 setCodification
 es.ull.esit.app.problem.definition.Problem, 240
 setConstriction
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 228
 setCount
 es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 123
 setCountBetterGender
 es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 123
 setCountCurrent
 es.ull.esit.app.metaheuristics.strategy.Strategy, 293
 es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 124
 es.ull.esit.app.strategy.Strategy, 302
 setCountCurrentIterPSO
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 229
 setCountGender
 es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 124
 setCountIterationsT
 es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 271, 272
 setCountMax
 es.ull.esit.app.metaheuristics.strategy.Strategy, 293
 es.ull.esit.app.strategy.Strategy, 303
 es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 55
 es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 65
 es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 107
 es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 229
 setData
 es.ull.esit.app.evolutionary_algorithms.complement.Range, 252
 es.ull.esit.app.evolutionary_algorithms.complement.TruncationSelection, 317
 es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 55
 setEvaluation
 es.ull.esit.app.problem.definition.State, 282

- setMapGenerators
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 294
- setMax
 - es.ull.esit.app.evolutionary_algorithms.complement.Rangees.ull.esit.app.metaheuristics.generators.TabuSearch, 252
- setMin
 - es.ull.esit.app.evolutionary_algorithms.complement.Rangees.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 252
- setNumber
 - es.ull.esit.app.problem.definition.State, 282
- setOperator
 - es.ull.esit.app.problem.definition.Problem, 241
- setPossibleValue
 - es.ull.esit.app.problem.definition.Problem, 241
- setProbability
 - es.ull.esit.app.evolutionary_algorithms.complement.PossibleValue, 236
- setProblem
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 294
 - es.ull.esit.app.strategy.Strategy, 303
- setReplaceType
 - es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 56
- setSelectionType
 - es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 56
- setSizeNeighbors
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, 182
- setState
 - es.ull.esit.app.problem.definition.Problem, 241
- setStateActual
 - es.ull.esit.app.metaheuristics.generators.Particle, 217
- setStatePBest
 - es.ull.esit.app.metaheuristics.generators.Particle, 217
- setStateRef
 - es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 66
 - es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 108
 - es.ull.esit.app.metaheuristics.generators.HillClimbing, 116
 - es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 125
 - es.ull.esit.app.metaheuristics.generators.LimitThreshold, 148
 - es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 158
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, 174
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart, 183
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing, 189
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 195
- es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 272, 273
- es.ull.esit.app.metaheuristics.generators.TabuSearch, 311
- setStateReferencePSO
 - es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization, 231
- setStateReferenceTS
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 196
- setStopexecute
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 295
 - es.ull.esit.app.strategy.Strategy, 303
- setTabu
 - es.ull.esit.app.local_search.candidate_type.CandidateValue, 42
- setTerminate
 - es.ull.esit.app.metaheuristics.generators.InstanceDE, 137
 - es.ull.esit.app.metaheuristics.generators.InstanceEE, 137
 - es.ull.esit.app.metaheuristics.generators.InstanceGA, 141
 - es.ull.esit.app.metaheuristics.generators.InstanceHillClimbingAlgorithm, 158
 - es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 173
 - es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 273
- setThreshold
 - es.ull.esit.app.metaheuristics.strategy.Strategy, 295
 - es.ull.esit.app.strategy.Strategy, 303
- setTinitial
 - es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 159
 - es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 273, 274
- setTypeCandidate
 - es.ull.esit.app.metaheuristics.generators.HillClimbing, 117
 - es.ull.esit.app.metaheuristics.generators.HillClimbingRestart, 125
 - es.ull.esit.app.metaheuristics.generators.LimitThreshold, 149
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 196
 - es.ull.esit.app.metaheuristics.generators.TabuSearch, 312
- setTypeGenerator
 - es.ull.esit.app.metaheuristics.generators.EvolutionStrategies, 66
 - es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing, 159
 - es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch, 196

- es.ull.esit.app.metaheuristics.generators.RandomSearch, 274, 275
- 249
- es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing, 312
- 274
- es.ull.esit.app.metaheuristics.generators.TabuSearch, 206
- 312
- es.ull.esit.app.problem.definition.State, 283
- setTypeProblem
- es.ull.esit.app.problem.definition.ObjetivoFunction, 205
- es.ull.esit.app.problem.definition.Problem, 241
- setTypeSolutionMethod
- es.ull.esit.app.problem.definition.Problem, 242
- setUpdateparameter
- es.ull.esit.app.metaheuristics.strategy.Strategy, 295
- es.ull.esit.app.strategy.Strategy, 303
- setValue
- es.ull.esit.app.config.tspdynamic.TSPState, 319
- es.ull.esit.app.evolutionary_algorithms.complement.PSOCandidate, 236
- setVelocity
- es.ull.esit.app.metaheuristics.generators.Particle, 217
- setWeight
- es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm, 56
- es.ull.esit.app.metaheuristics.generators.EvolutionStrategy, 66
- es.ull.esit.app.metaheuristics.generators.Generator, 96
- es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm, 109
- es.ull.esit.app.metaheuristics.generators.HillClimbing,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Distribut, 117
- es.ull.esit.app.metaheuristics.generators.HillClimbingRestart,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Distribut, 125
- es.ull.esit.app.metaheuristics.generators.LimitThreshold,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/FatherS, 149
- es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Generat, 159
- es.ull.esit.app.metaheuristics.generators.MultiGenerators,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation, 167
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistances,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation, 174
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoi, 183
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbing,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoi, 189
- es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Probabil, 196
- es.ull.esit.app.metaheuristics.generators.Particle,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Probabil, 217
- es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Range.j, 231
- es.ull.esit.app.metaheuristics.generators.RandomSearch,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replac, 250
- es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing,src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replac, 274, 275
- es.ull.esit.app.metaheuristics.generators.TabuSearch, 274
- es.ull.esit.app.problem.definition.ObjetivoFunction, 205
- es.ull.esit.app.problem.definition.Problem, 241
- es.ull.esit.app.problem.definition.Problem, 242
- es.ull.esit.app.metaheuristics.strategy.Strategy, 295
- es.ull.esit.app.strategy.Strategy, 303
- es.ull.esit.app.config.tspdynamic.TSPState, 319
- es.ull.esit.app.evolutionary_algorithms.complement.PSOCandidate, 236
- es.ull.esit.app.local_search.candidate_type.CandidateType, 39
- sortedPathValue
- es.ull.esit.app.evolutionary_algorithms.complement.AIOMutation, 38
- es.ull.esit.app.config.tspdynamic.TSPState.java, 325
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/AIOMuta, 325
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Crossov, 326
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Crossov, 326
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Distribut, 327
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Distribut, 327
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/FatherS, 328
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Generat, 328
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation, 329
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Mutation, 329
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoi, 330
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoi, 330
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Probabil, 331
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Probabil, 332
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Range.j, 332
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replac, 332
- src/main/java/es/ull/esit/app/evolutionary_algorithms/complement/Replac, 332

368
src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbing.java, 369
src/main/java/es/ull/esit/app/metaheuristics/generators/HillClimbingRestart.java, 369
src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceDE.java, 370
src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceEE.java, 371
src/main/java/es/ull/esit/app/metaheuristics/generators/InstanceGA.java, 372
src/main/java/es/ull/esit/app/metaheuristics/generators/LimitRoulette.java, 372
src/main/java/es/ull/esit/app/metaheuristics/generators/LimitThreshold.java, 373
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiCaseSimulatedAnnealing.java, 373
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiGenerator.java, 374
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingDistance.java, 375
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveHillClimbingRestart.java, 376
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveStochasticHillClimbing.java, 376
src/main/java/es/ull/esit/app/metaheuristics/generators/MultiobjectiveTabuSearch.java, 377
src/main/java/es/ull/esit/app/metaheuristics/generators/Particle.java, 378
src/main/java/es/ull/esit/app/metaheuristics/generators/ParticleSwarmOptimization.java, 379
src/main/java/es/ull/esit/app/metaheuristics/generators/RandomSearch.java, 379
src/main/java/es/ull/esit/app/metaheuristics/generators/SimulatedAnnealing.java, 380
src/main/java/es/ull/esit/app/metaheuristics/generators/TabuSearch.java, 382
src/main/java/es/ull/esit/app/problem/definition/Codification.java, 402
src/main/java/es/ull/esit/app/problem/definition/ObjetiveFunction.java, 402
src/main/java/es/ull/esit/app/problem/definition/Operator.java, 403
src/main/java/es/ull/esit/app/problem/definition/Problem.java, 404
src/main/java/es/ull/esit/app/problem/definition/State.java, 404
src/main/java/es/ull/esit/app/problem/extension/FactoresPonderados.java, 405
src/main/java/es/ull/esit/app/problem/extension/MetricasMultiobjetivo.java, 406
src/main/java/es/ull/esit/app/problem/extension/MultiObjetivoPuro.java, 406
src/main/java/es/ull/esit/app/problem/extension/SolutionMethod.java, 407
src/main/java/es/ull/esit/app/problem/extension/TypeSolutionMethod.java, 408
src/main/java/es/ull/esit/app/problem_operators/MutationOperator.java, 408

389
src/main/java/es/ull/esit/app/strategy/Strategy.java, 390
src/test/java/es/ull/esit/app/config/tspdynamic/TSPStateTest.java, 391
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/AIOMutation.java, 391
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Generational.java, 392
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoint.java, 393
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/OnePoint.java, 394
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Probabilistic.java, 394
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Probabilistic.java, 395
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/RangeTest.java, 396
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/RouletteSelection.java, 396
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/SteadyState.java, 397
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Tournament.java, 397
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/TruncationSelection.java, 398
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/UniformCrossover.java, 399
src/test/java/es/ull/esit/app/evolutionary_algorithms/complement/Univariate.java, 399
src/test/java/es/ull/esit/app/factory_interface/FactoryInterfacesTest.java, 400
src/test/java/es/ull/esit/app/factory_method/FactoryAcceptCandidateTest.java, 401
src/test/java/es/ull/esit/app/factory_method/FactoryCandidateTest.java, 401
src/test/java/es/ull/esit/app/factory_method/FactoryCrossoverTest.java, 401
src/test/java/es/ull/esit/app/factory_method/FactoryDampingTest.java, 402
src/test/java/es/ull/esit/app/factory_method/FactoryDistributionTest.java, 402
src/test/java/es/ull/esit/app/factory_method/FactoryFatherSelectionTest.java, 403
src/test/java/es/ull/esit/app/factory_method/FactoryGeneratorTest.java, 403
src/test/java/es/ull/esit/app/factory_method/FactoryLoaderTest.java, 404
src/test/java/es/ull/esit/app/factory_method/FactoryMutationTest.java, 405
src/test/java/es/ull/esit/app/factory_method/FactoryReplaceTest.java, 405
src/test/java/es/ull/esit/app/factory_method/FactorySolutionMethodTest.java, 406
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptNotBadTest.java, 408
src/test/java/es/ull/esit/app/local_search/acceptation_type/AcceptableCandidateTest.java, 408

es.ull.esit.app.evolutionary_algorithms.complement.ReplacementGender	254	es.ull.esit.app.metaheuristics.strategy.Strategy,	296
StochasticHillClimbing		es.ull.esit.app.strategy.Strategy, 304	
es.ull.esit.app.metaheuristics.generators.GeneratorType,	101	updateParameter	
stopIterations		es.ull.esit.app.local_search.complement.UpdateParameter,	324
es.ull.esit.app.local_search.complement.StopExecute,	286	updateRef	
strategy		es.ull.esit.app.metaheuristics.strategy.Strategy,	296
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,	176	es.ull.esit.app.strategy.Strategy, 304	
		updateReference	
TABU		es.ull.esit.app.metaheuristics.generators.DistributionEstimationAlgorithm,	57
es.ull.esit.app.local_search.complement.StrategyType,	306	es.ull.esit.app.metaheuristics.generators.EvolutionStrategies,	67
TabuSearch		es.ull.esit.app.metaheuristics.generators.Generator,	97
es.ull.esit.app.metaheuristics.generators.GeneratorType,	101	es.ull.esit.app.metaheuristics.generators.GeneticAlgorithm,	109
es.ull.esit.app.metaheuristics.generators.TabuSearch,	308	es.ull.esit.app.metaheuristics.generators.HillClimbing,	177
tfinal		es.ull.esit.app.metaheuristics.generators.HillClimbingRestart,	125
es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing,	159	es.ull.esit.app.metaheuristics.generators.LimitThreshold,	149
threshold		es.ull.esit.app.metaheuristics.generators.MultiCaseSimulatedAnnealing,	160
es.ull.esit.app.strategy.Strategy, 305		es.ull.esit.app.metaheuristics.generators.MultiGenerator,	168
tournament		es.ull.esit.app.evolutionary_algorithms.complement.MutationType,	202
es.ull.esit.app.metaheuristics.generators.MultiGenerator,	168	es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,	175
TOW_POINTS_MUTATION		es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingRestart,	183
es.ull.esit.app.evolutionary_algorithms.complement.MutationType,	202	es.ull.esit.app.metaheuristics.generators.MultiobjectiveStochasticHillClimbingDistance,	197
TRUNCATION_SELECTION		es.ull.esit.app.metaheuristics.generators.MultiobjectiveTabuSearch,	218
es.ull.esit.app.evolutionary_algorithms.complement.SelectionTypes,	260	es.ull.esit.app.metaheuristics.generators.Particle,	232
typeAcceptation		es.ull.esit.app.metaheuristics.generators.ParticleSwarmOptimization,	250
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,	176	es.ull.esit.app.metaheuristics.generators.RandomSearch,	275
typeCandidate		es.ull.esit.app.metaheuristics.generators.SimulatedAnnealing,	312
es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,	176	es.ull.esit.app.metaheuristics.generators.TabuSearch,	
typeGenerator		es.ull.esit.app.problem.definition.State, 283	
es.ull.esit.app.problem.definition.State, 283		updateRefGenerator	
UNIFORM_CROSSOVER		es.ull.esit.app.metaheuristics.strategy.Strategy,	296
es.ull.esit.app.evolutionary_algorithms.complement.CrossoverType,	46	es.ull.esit.app.strategy.Strategy, 305	
UNIVARIATE		updateRefMultiG	
es.ull.esit.app.evolutionary_algorithms.complement.DistributionType,	57	es.ull.esit.app.metaheuristics.strategy.Strategy,	297
update		es.ull.esit.app.strategy.Strategy, 305	
es.ull.esit.app.metaheuristics.strategy.Strategy,	295	updateWeight	
es.ull.esit.app.strategy.Strategy, 304		es.ull.esit.app.metaheuristics.strategy.Strategy,	
updateAwardImp		es.ull.esit.app.metaheuristics.strategy.Strategy,	
es.ull.esit.app.metaheuristics.generators.MultiGenerator,	168	es.ull.esit.app.strategy.Strategy, 305	
updateAwardSC			
es.ull.esit.app.metaheuristics.generators.MultiGenerator,	168		

[297](#)

es.ull.esit.app.metaheuristics.generators.MultiGenerator,

[168](#)es.ull.esit.app.strategy.Strategy, [305](#)

validState

es.ull.esit.app.problem.definition.Codification, [44](#)

weight

es.ull.esit.app.metaheuristics.generators.MultiobjectiveHillClimbingDistance,

[176](#)