



Prácticas

Sistemas Inteligentes I

Sesión 1. AIMA

José A. Montenegro Montes

monte@lcc.uma.es

Resumen

- Definición
- Configurando entorno trabajo
- Primeros Pasos: Crear problemas Búsquedas
- Problemas. Puzzle-8
- Ejercicios

AIMA

DEFINICIÓN



¿Qué es AIMA?

Paquete de clases Java que permite definir y resolver problemas de Sistemas Inteligentes:

Inicialmente veremos algunos de los algoritmos de búsqueda vistos en teoría:

- **Búsqueda no Informada:** Profundidad y Amplitud
- **Búsqueda Heurística:** A*, IDA*
- **Búsqueda Local:** Basados en gradiente, Recorrido Simulado

El diseño permite separa la representación del problema de los algoritmos .

Contenido AIMA

- *aima-core* es la base del proyecto que incluye todas las implementaciones de los algoritmos descritos en la tercera edición del libro.
- *aima-gui* depende de *aima-core* y contiene los ejemplos visuales y demostraciones en línea de comando de los algoritmos definidos en *aima-core*.
- *aimax-osm* es una extensión del proyecto que demuestra como los algoritmos de *aima-core* pueden ser utilizados para crear Aplicaciones de Navegación utilizando los mapas del proyecto Open Street Maps.

Una vez configurado correctamente el entorno de trabajo, los programas de ejemplo estará en las siguientes rutas:

- *aima-gui/src/main/java/aima/gui/applications/*
- *aima-gui/src/main/java/aima/gui/demo* (demos línea comandos)
- *aimax-osm/src/main/java/aimax/osm/applications*

<http://code.google.com/p/aima-java/wiki/GettingStarted>

ANIMA

CONFIGURACIÓN

ENTORNO TRABAJO



Configurando entorno trabajo

- Las distribuciones de la librería pueden ser descargada de la página web del proyecto. Trabajaremos con la versión **aima-java-1.7.0-Chp4-Complete**

<http://code.google.com/p/aima-java/downloads/list>

- Es posible descargar la librería del repositorio de Subversion directamente. Esta opción requiere un cliente de Subversion. Instrucciones en:

1. <http://code.google.com/p/aima-java/source/checkout>

2. <http://subversion.apache.org/packages.html>

- Una vez que finalice la configuración el entorno de trabajo tendrá la siguiente organización:

<directorio-entorno-trabajo>/

aima-core/

aima-gui/

aimax-osm/

Configurando entorno trabajo. Pasos

Paso 0: Descargar Versión de Eclipse <http://www.eclipse.org/>
Descargar la versión de java 1.7 que es la versión que requiere AIMA

Paso 1: Dos posibles casos para instalar AIMA en eclipse:

1.1 Importando una distribución

1.2 Utilizando SVN

Paso 2: Instalar paquete Sistemas Inteligentes

Paso 3: Crear Proyecto para Clases de Ejercicios

Paso 1.1: Importando una distribución

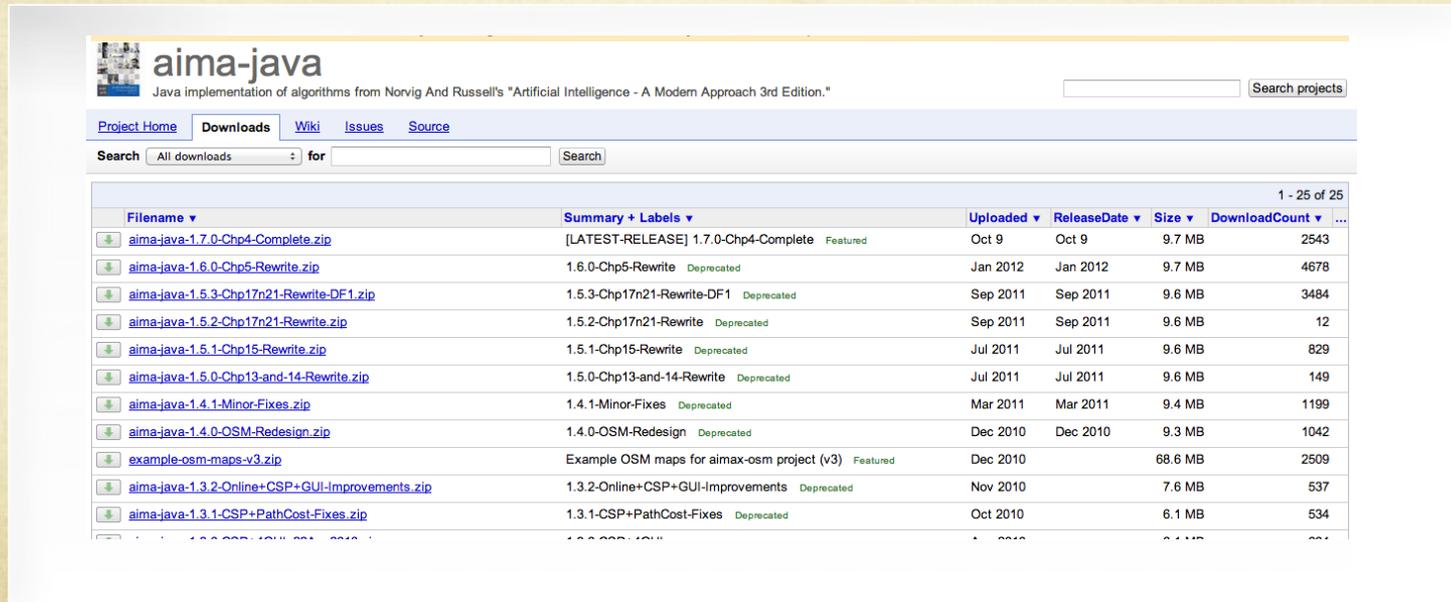
Importando una distribución

0. Descargamos la distribución de `aima-java-1.7.0-Chp4-Complete`

0.1 Ir <http://code.google.com/p/aima-java/downloads/list>

0.2 Seleccionar en la búsqueda “All Downloads” for “`aima-java`” y pulsar Search.

0.3 De la lista de versiones seleccionamos `aima-java-1.7.0-Chp4-Complete`

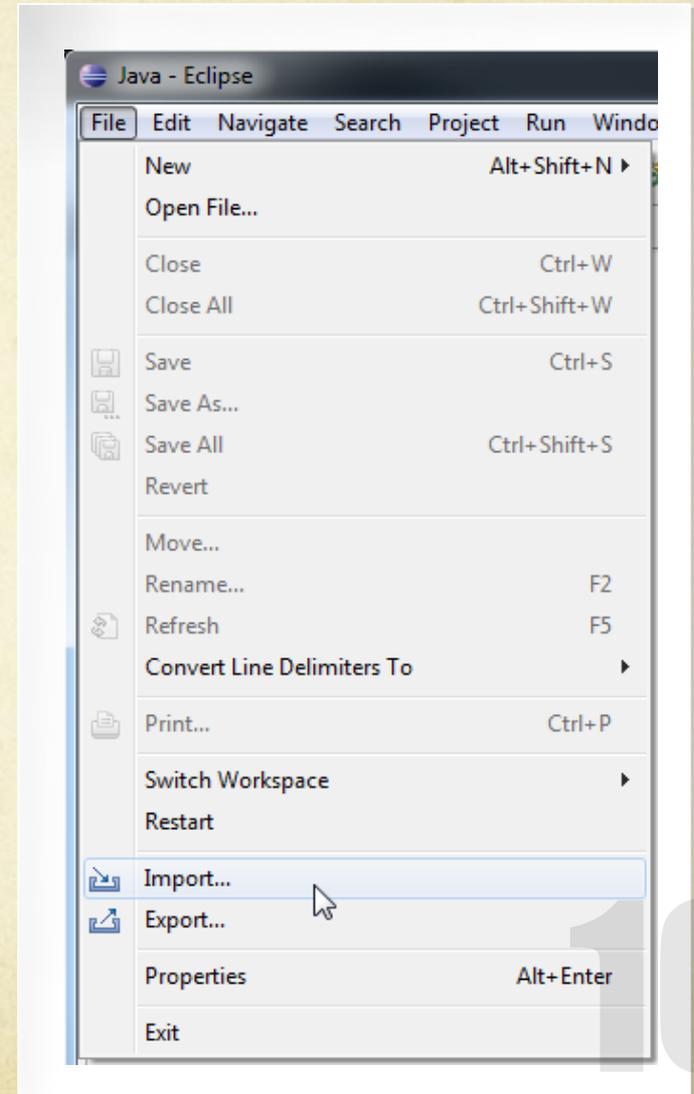


The screenshot displays the 'Downloads' section of the 'aima-java' project page. The page title is 'aima-java' with a subtitle 'Java implementation of algorithms from Norvig And Russell's "Artificial Intelligence - A Modern Approach 3rd Edition."'. The navigation menu includes 'Project Home', 'Downloads', 'Wiki', 'Issues', and 'Source'. A search bar is present with the text 'Search All downloads for'. Below the search bar is a table of downloads with the following columns: Filename, Summary + Labels, Uploaded, ReleaseDate, Size, and DownloadCount. The table shows 25 items, with the first item being 'aima-java-1.7.0-Chp4-Complete.zip'.

Filename	Summary + Labels	Uploaded	ReleaseDate	Size	DownloadCount
aima-java-1.7.0-Chp4-Complete.zip	[LATEST-RELEASE] 1.7.0-Chp4-Complete Featured	Oct 9	Oct 9	9.7 MB	2543
aima-java-1.6.0-Chp5-Rewrite.zip	1.6.0-Chp5-Rewrite Deprecated	Jan 2012	Jan 2012	9.7 MB	4678
aima-java-1.5.3-Chp17n21-Rewrite-DF1.zip	1.5.3-Chp17n21-Rewrite-DF1 Deprecated	Sep 2011	Sep 2011	9.6 MB	3484
aima-java-1.5.2-Chp17n21-Rewrite.zip	1.5.2-Chp17n21-Rewrite Deprecated	Sep 2011	Sep 2011	9.6 MB	12
aima-java-1.5.1-Chp15-Rewrite.zip	1.5.1-Chp15-Rewrite Deprecated	Jul 2011	Jul 2011	9.6 MB	829
aima-java-1.5.0-Chp13-and-14-Rewrite.zip	1.5.0-Chp13-and-14-Rewrite Deprecated	Jul 2011	Jul 2011	9.6 MB	149
aima-java-1.4.1-Minor-Fixes.zip	1.4.1-Minor-Fixes Deprecated	Mar 2011	Mar 2011	9.4 MB	1199
aima-java-1.4.0-OSM-Redesign.zip	1.4.0-OSM-Redesign Deprecated	Dec 2010	Dec 2010	9.3 MB	1042
example-osm-maps-v3.zip	Example OSM maps for aimax-osm project (v3) Featured	Dec 2010		68.6 MB	2509
aima-java-1.3.2-Online+CSP+GUI-Improvements.zip	1.3.2-Online+CSP+GUI-Improvements Deprecated	Nov 2010		7.6 MB	537
aima-java-1.3.1-CSP+PathCost-Fixes.zip	1.3.1-CSP+PathCost-Fixes Deprecated	Oct 2010		6.1 MB	534

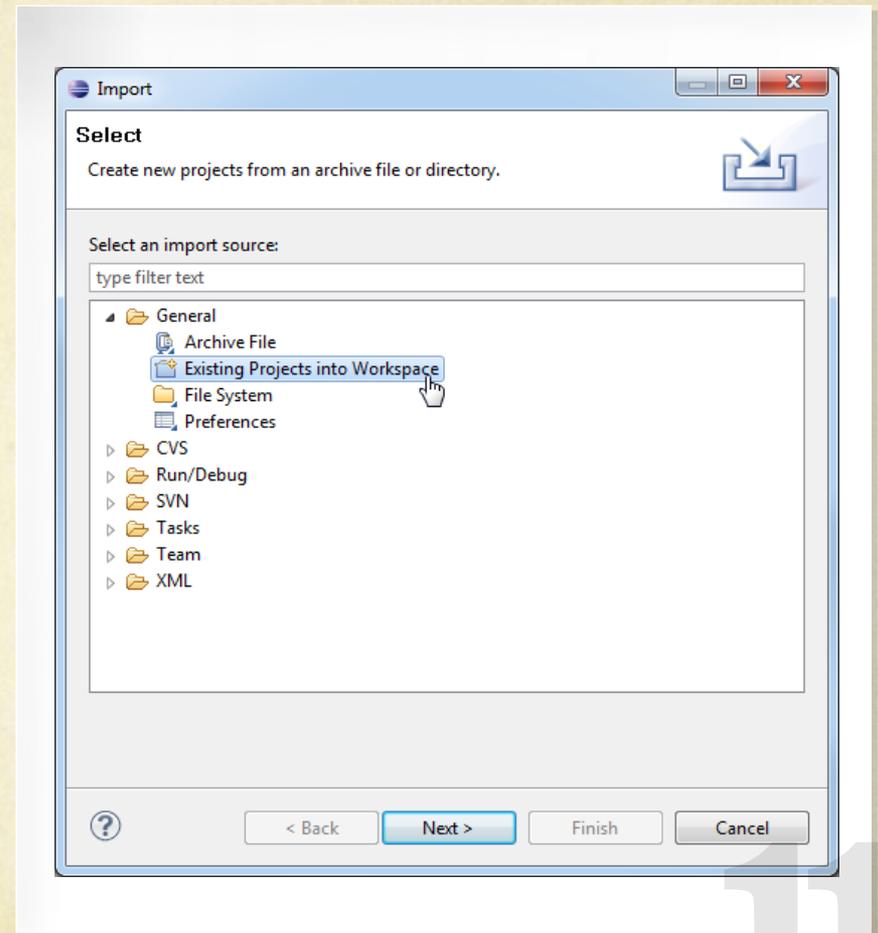
Paso 1.1: Importando una distribución

1. Selecciona la entrada del Menú 'File->Import...'



Paso 1.1: Importando una distribución

2. En el diálogo de Importar 'Select', seleccionamos 'General->Existing Projects into Workspace':



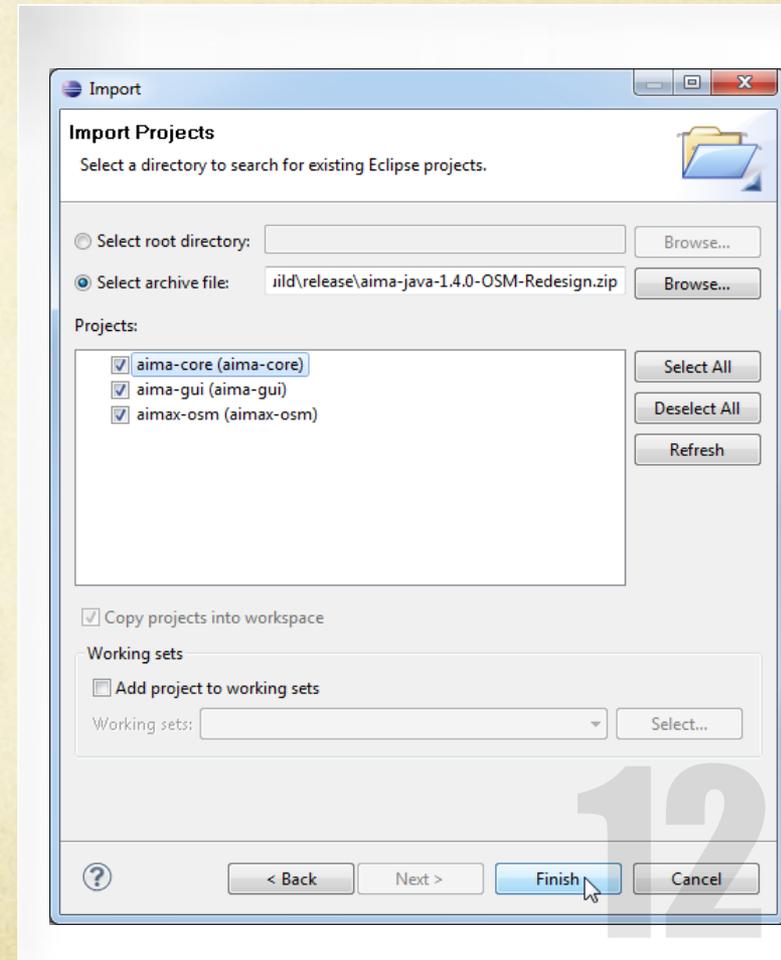
Paso 1.1: Importando una distribución

3. Pulsamos el botón 'Next >'.

4. En el 'Import Projects' seleccionamos 'Select archive file:' y 'Browse...' para seleccionar el archivo .zip bajado anteriormente de la web del proyecto.

Los tres subproyectos, *aima-core*, *aima-gui*, y *aimax-osm* deben aparecer y estar seleccionados automáticamente.

Finalizamos pulsando el botón 'Finish'.

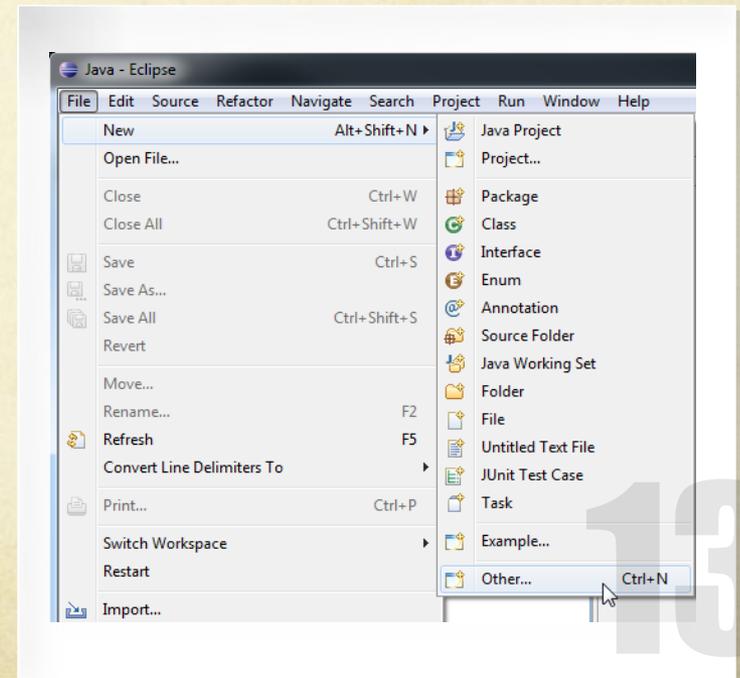


Paso 1.2: Utilizando un repositorio Subversion

Nota: El cliente Subversion Subclipse es un plugin para Eclipse IDE. Es necesario descargarlo y configurarlo, más información en <http://subclipse.tigris.org/>

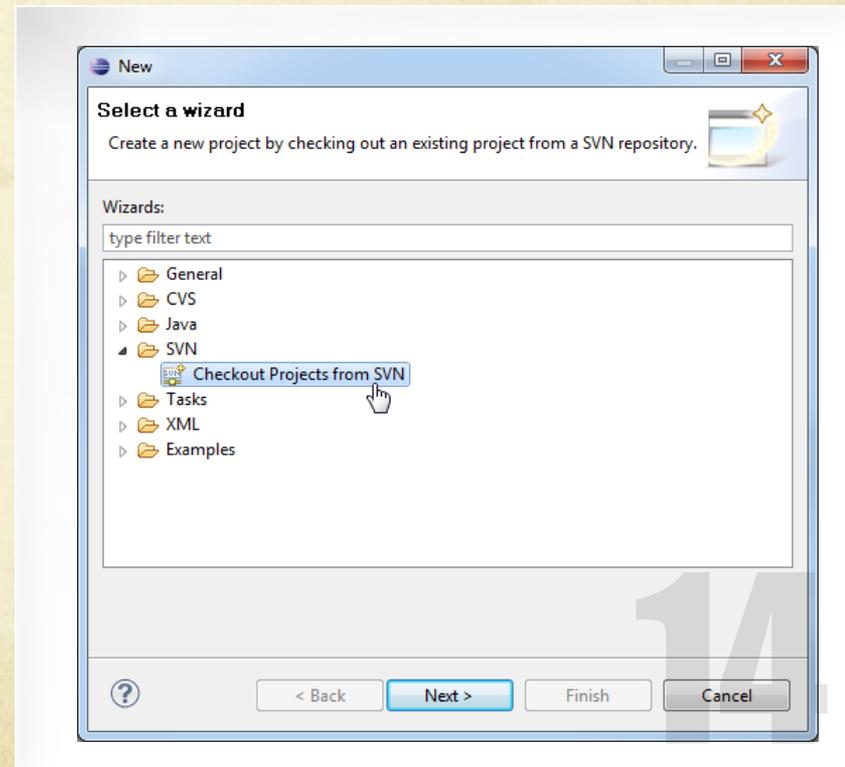
Las siguientes instrucciones detallan como descargar el proyecto AIMA-Java utilizando este plugin.

1. Seleccionar la entrada del menu 'File->New->Other':



Paso 1.2: Utilizando un repositorio Subversion

2. En el diálogo New 'Select a wizard', seleccionar 'SVN->Checkout Projects from SVN':



Paso 1.2: Utilizando un repositorio Subversion

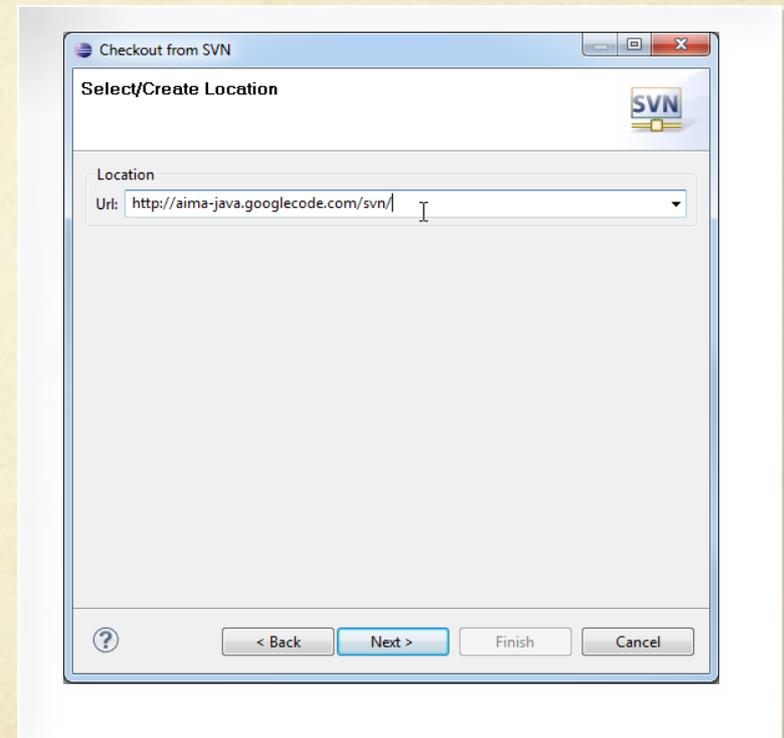
3. Pulsa el botón 'Next >'.

4. Si es la primera vez que accede al repositorio de AIMA-Java Subversion deberás seleccionar 'Create a new repository location' y pulsar el botón 'Next >'. Si no es la primera vez, introduce la ruta previa de la lista proporcionada.

En el diálogo 'Select/Create Location', introduce la url:

<http://aima-java.googlecode.com/svn/>

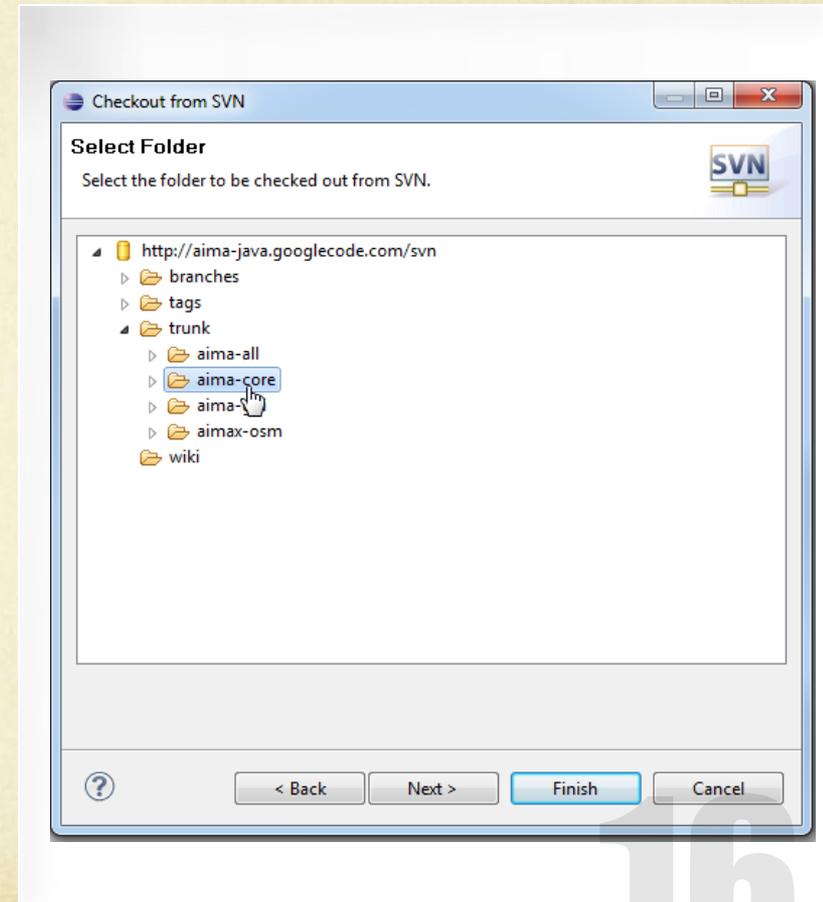
Y pulsamos el botón 'Next >'.



Paso 1.2: Utilizando un repositorio Subversion

5. En el diálogo 'Select Folder', selecciona el proyecto que quieres descargar en tu disco duro, comenzamos por ejemplo con *aima-core*:

6. Pulsaríamos el botón 'Finish' y el proyecto será descargado al repositorio local.



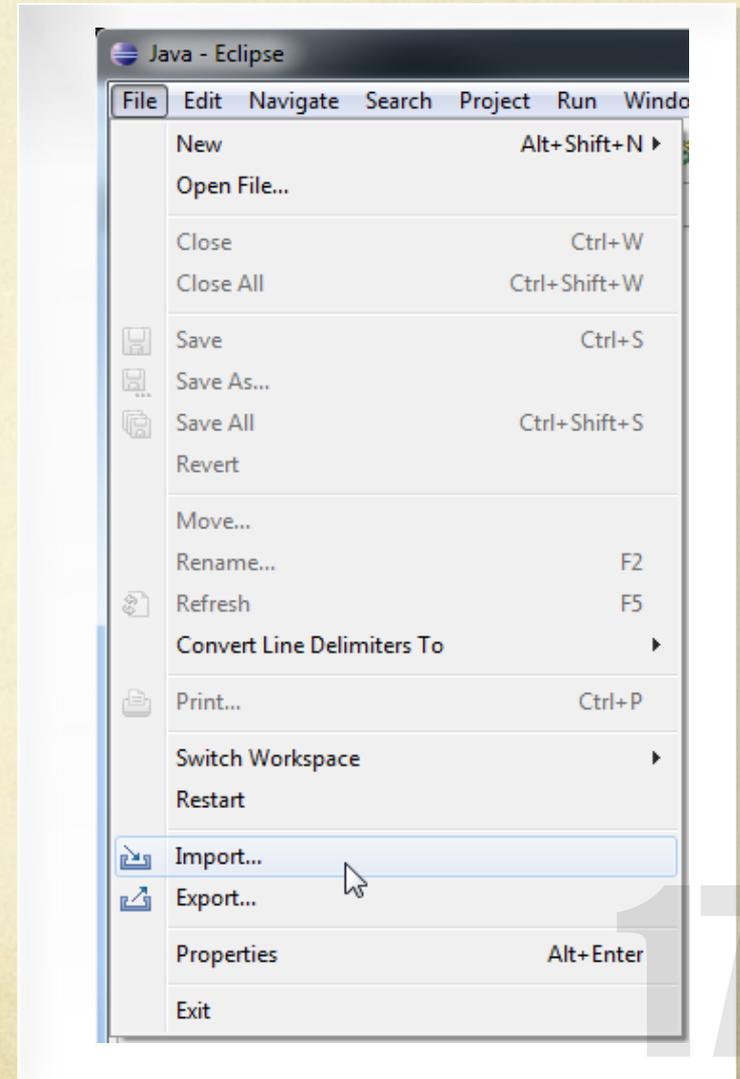
Paso 2: Instalar paquete Sist. Inteligentes

Importando paquete prácticas

0. Descargamos zip del campus virtual

intelligent-system.zip

1. Selecciona la entrada del Menú
'File->Import...'

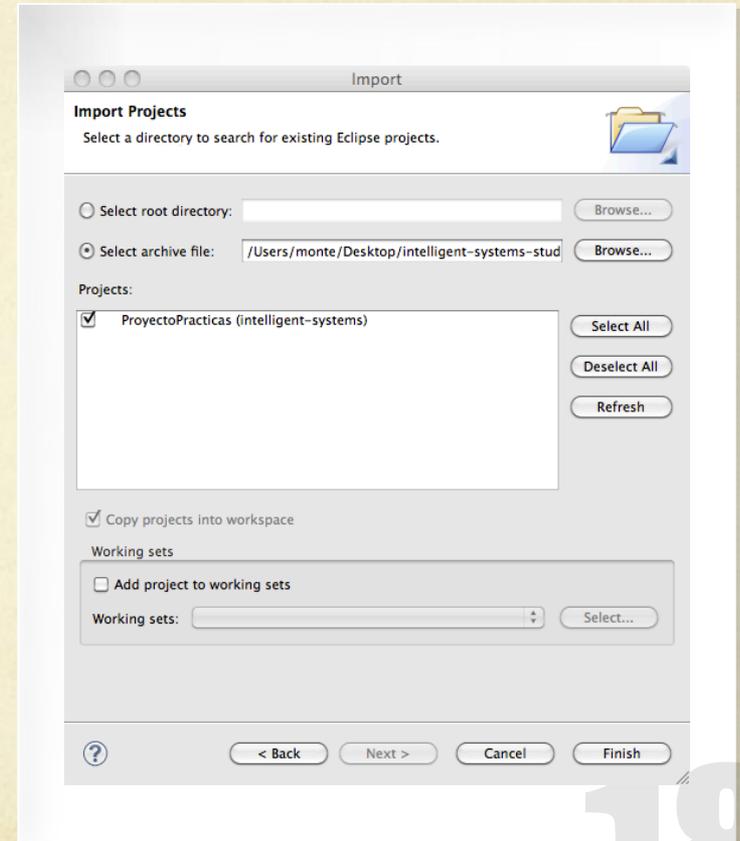


Paso 2: Instalar paquete Sist. Inteligentes

3. Pulsamos el botón 'Next >'.

4. En el 'Import Projects' seleccionamos 'Select archive file:' y 'Browse...' para seleccionar el archivo.zip bajado anteriormente del campus.

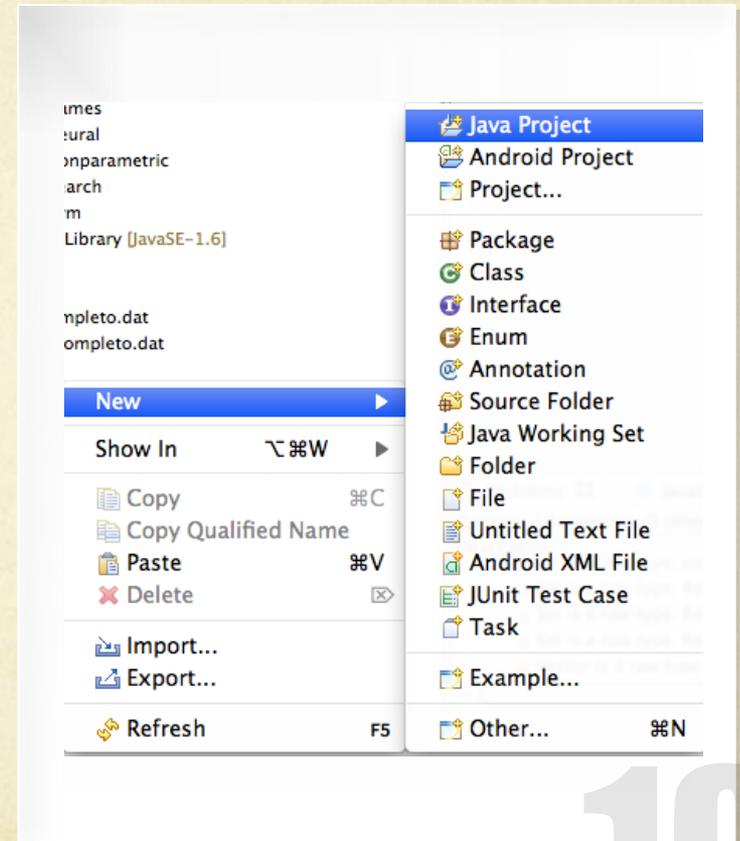
Finalizamos pulsando el botón 'Finish'.



Paso 3: Crear Proyecto Clases de Ejercicios

1. Pulsamos con el botón derecho ratón en el explorador de paquetes.

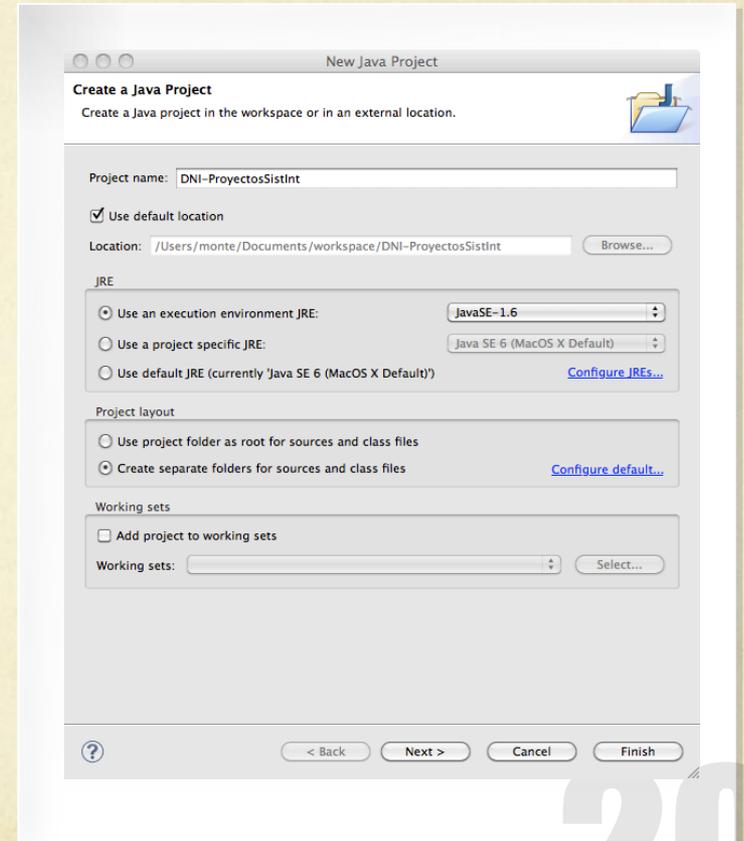
2. Seleccionamos “New->Java Project”



Paso 3: Crear Proyecto Clases de Ejercicios

3. Introducimos ProjectName:
“ID-Proyectos”

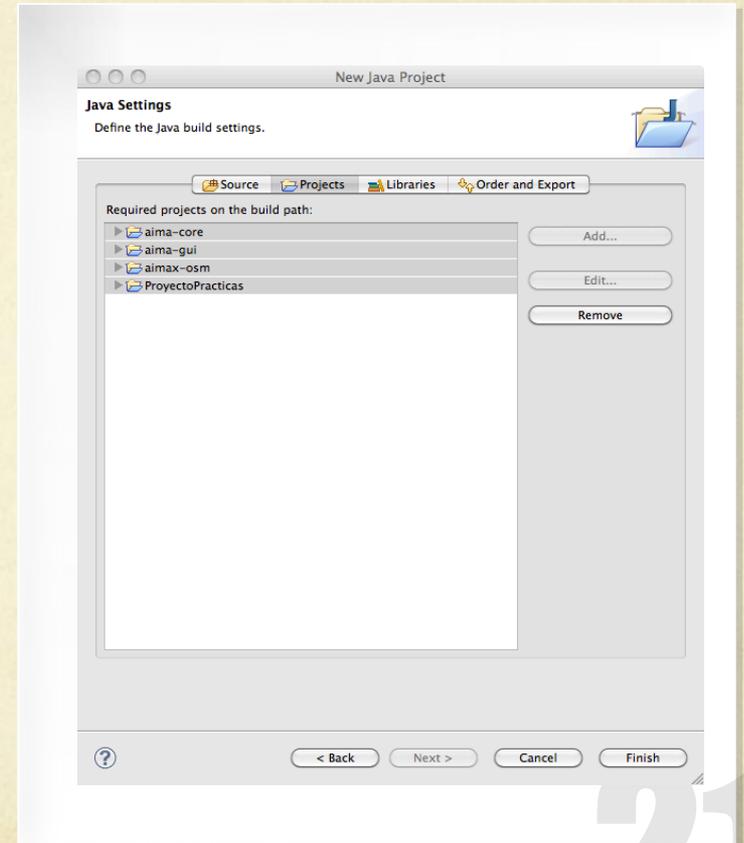
4. Seleccionamos “Next”



Paso 3: Crear Proyecto Clases de Ejercicios

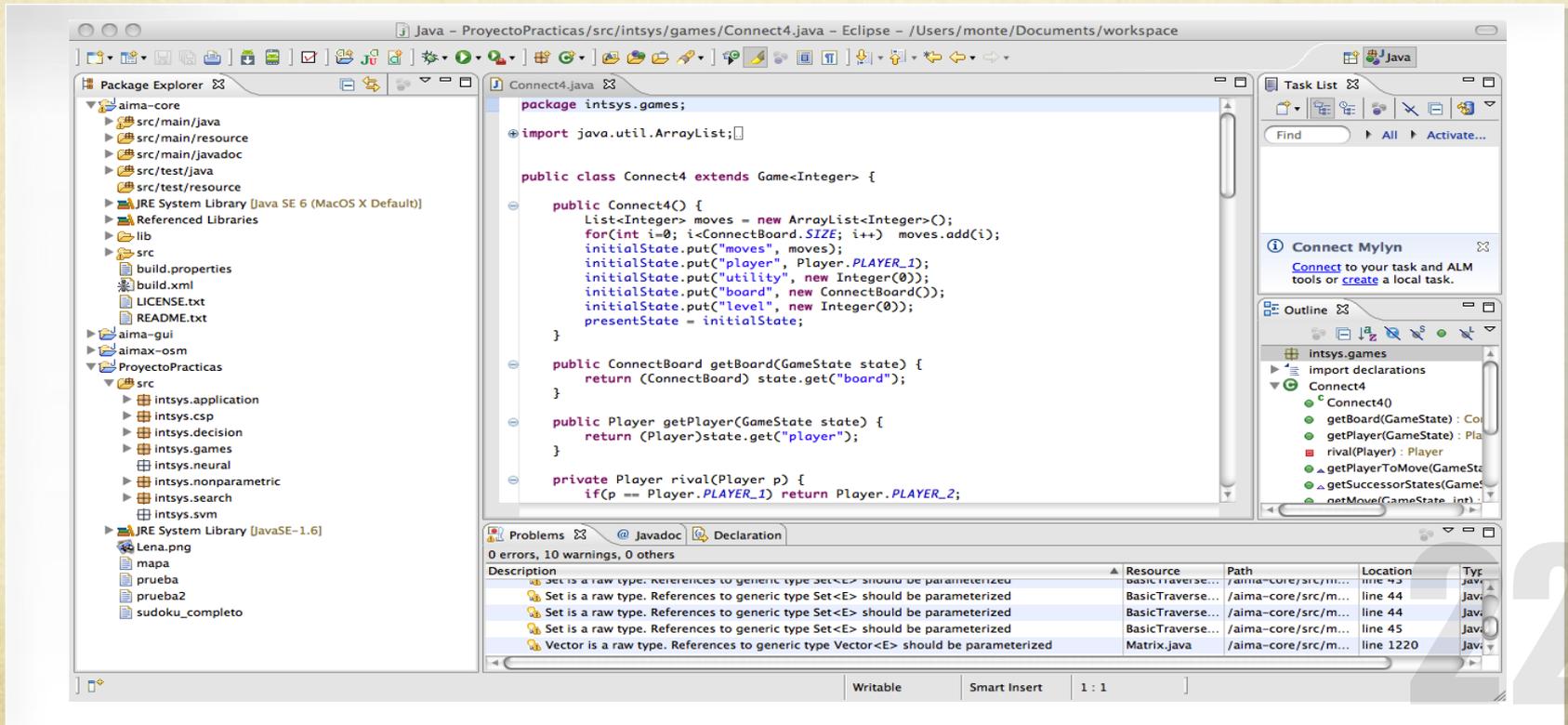
5. En la sección “Project” añadimos las referencias a AIMA y las clases de Sist. Int.

6. Seleccionamos “Finish”



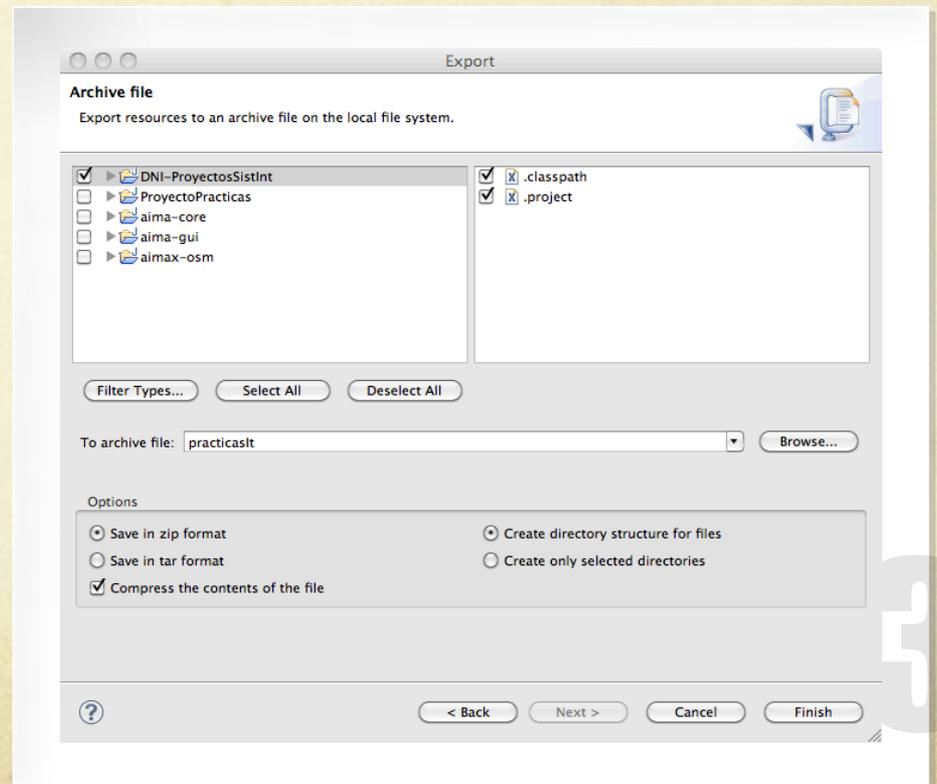
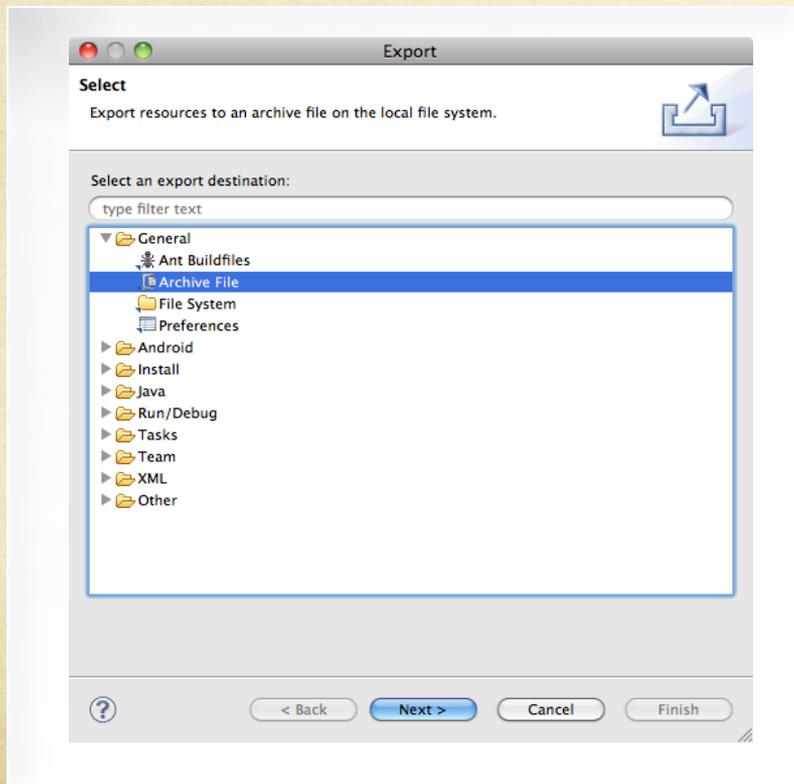
Finalización de la Instalación del Marco de Trabajo

Una vez instalado AIMA-Java en Eclipse y las clases de la asignatura, el entorno deberá parecerse a la siguiente figura:



Exportar Trabajo

Para Exportar el trabajo, seleccionaremos “Export” y realizamos los siguientes pasos:



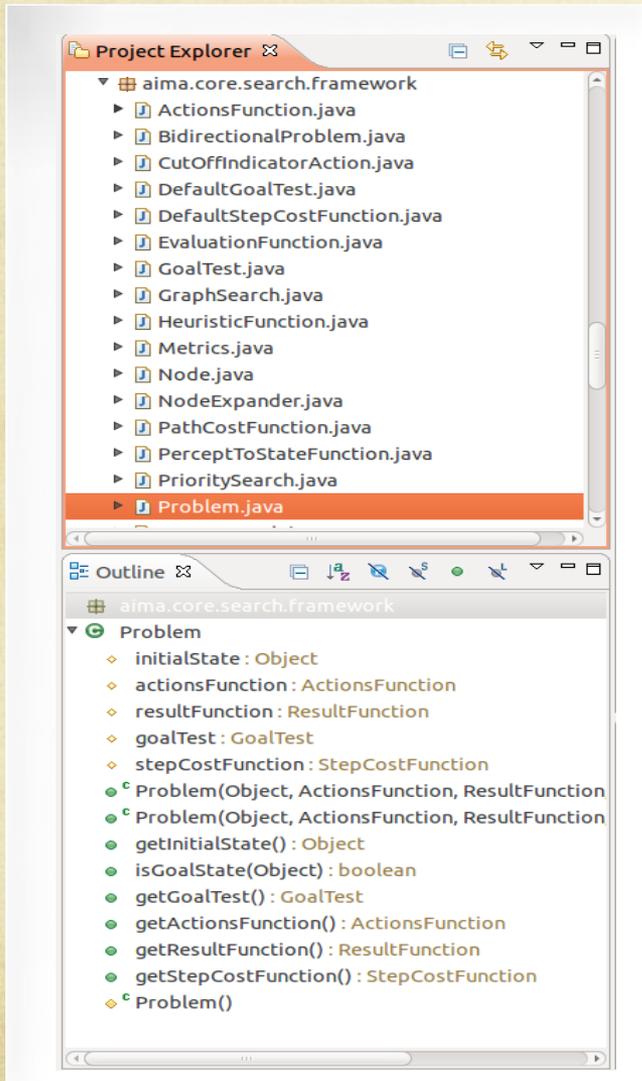
ANIMA

PRIMEROS PASOS

CREAR PROBLEMAS BÚSQUEDA



Definiendo el problema



Aima.core.search.framework

Clase Problem

Cinco componentes del problema:

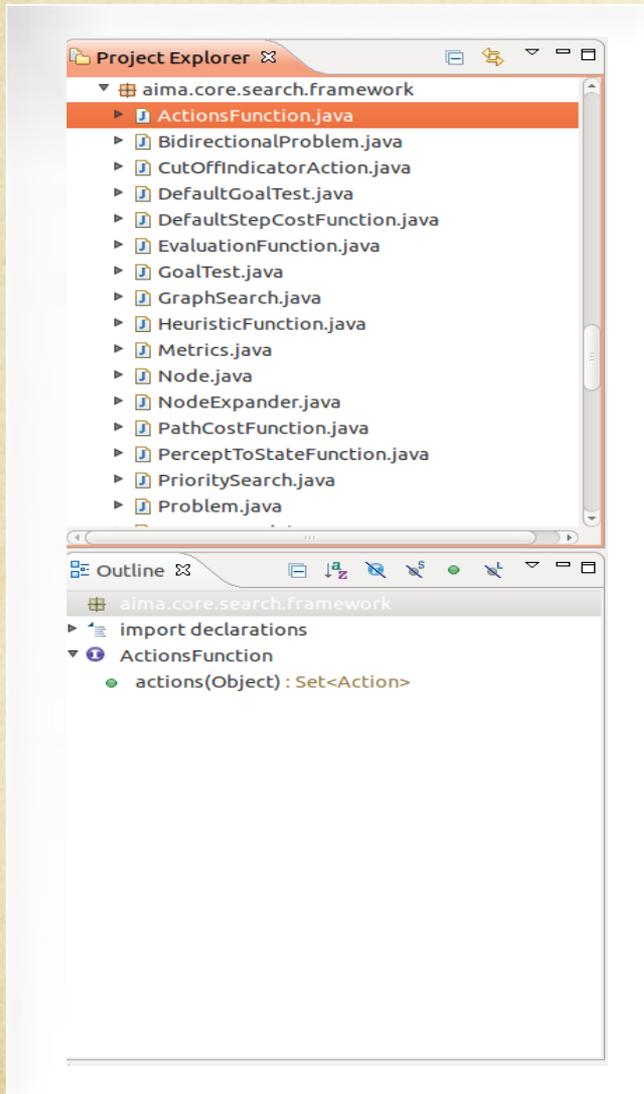
- InitialState: Estado Inicial
- ActionsFunction: Conjunto acciones ejecutadas estado dado
- ResultFunction: Modelo transición, el estado resultante de realizar una acción a un estado.
- GoalTest: Determina si un estado es el objetivo.
- StepCostFunction: Función que asigna coste numérico a cada ruta.

25

Definir el estado

- Es una clase independiente de la librería AIMA
- Debe tener uno o más constructores que genere el objeto que representa el estado inicial.
- Debe implementar los operadores de búsqueda del problema, para transitar de un estado a otro.
- Deseable implementar funciones para verificar que el operador de búsqueda puede ser aplicado a un estado.
- Algún otra función auxiliar que consideremos oportuna.

Clase Acciones



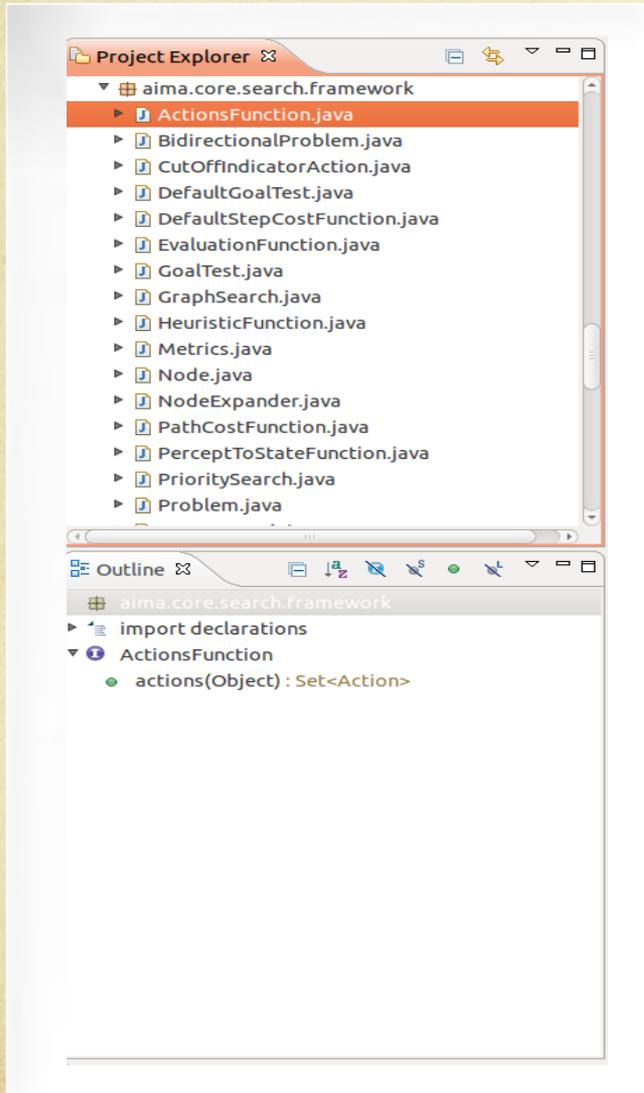
aima.core.search.framework.ActionsFunction

Set<Action> actions(Object s):

Devuelve un conjunto de acciones que pueden ser ejecutadas en el estado dado s.

Página 67: Artificial Intelligence A Modern Approach (3rd Edición)

Clase Resultado



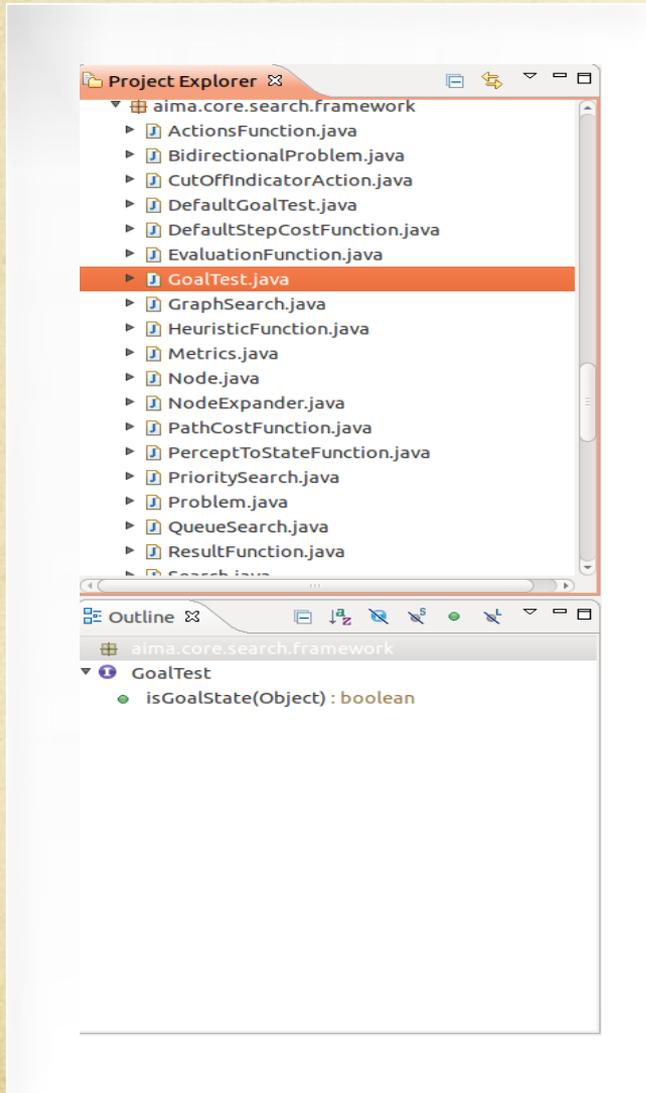
aima.core.search.framework.ResultFunction

Object result(Object s, Action a):

Devuelve el estado resultante de realizar una acción a en el estado s.

Página 67: Artificial Intelligence A Modern Approach (3rd Edición)

Clase Estado Objetivo



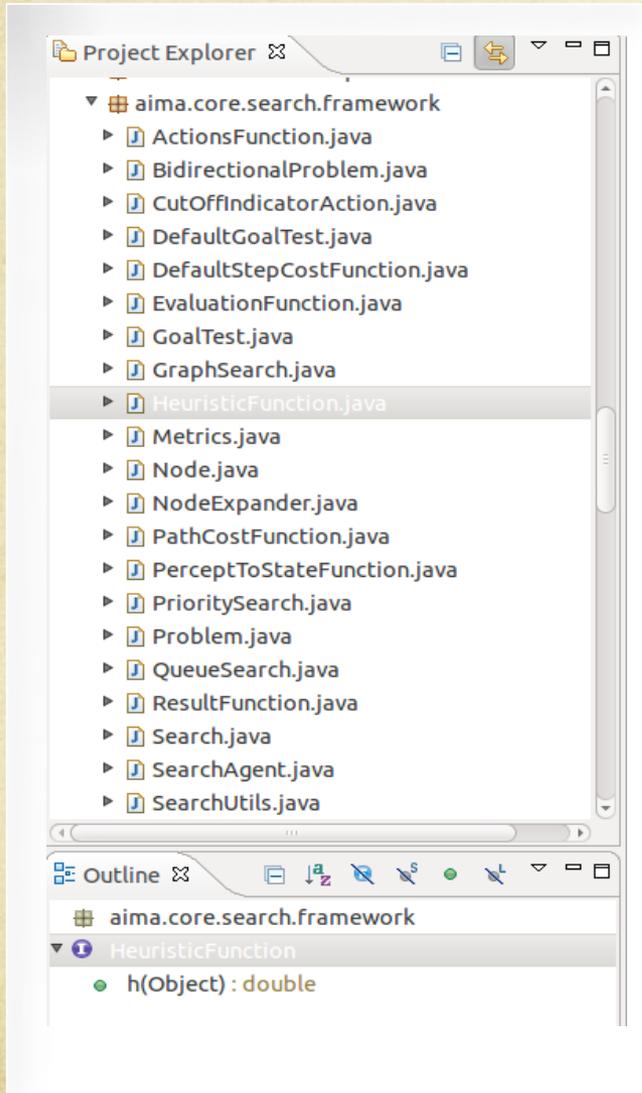
aima.core.search.framework.GoalTest

boolean isGoalState(Object state):

Determina si un estado dado es el objetivo.

Página 67: Artificial Intelligence A Modern Approach (3rd Edición)

Clase Heurística



aima.core.search.framework.HeuristicFunction

double h(Object state):

Define una función heurística $h(n)$.
Coste estimado del camino más “barato”
desde el estado del nodo n al estado
objetivo.

Página 92: Artificial Intelligence A Modern Approach (3rd Edición)

Clase Búsqueda

aima.core.search.framework.Search

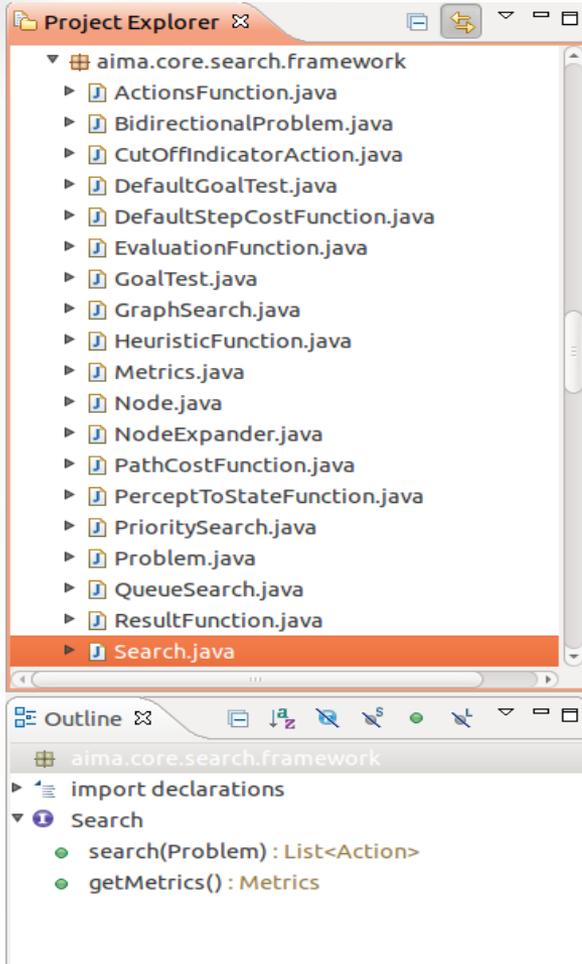
Define los algoritmos de búsqueda.

List<Action> search(Problem p):

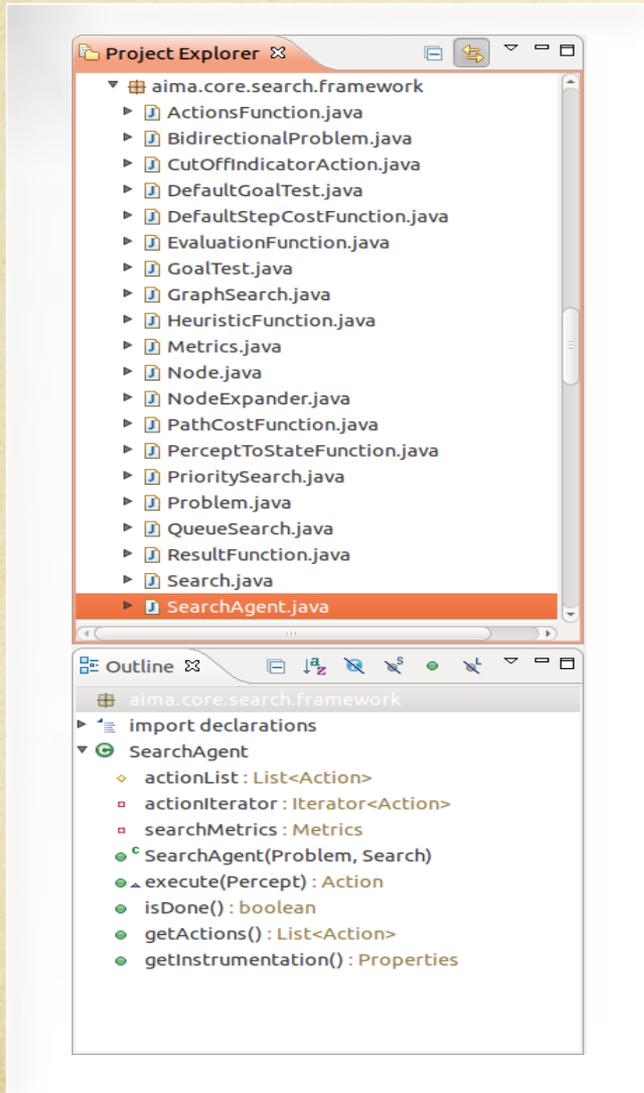
Devuelve una lista de acciones hasta el objetivo, si fue encontrado o una lista vacía en caso contrario.

Metrics getMetrics():

Devuelve las métricas de la búsqueda



Clase Agente Búsqueda

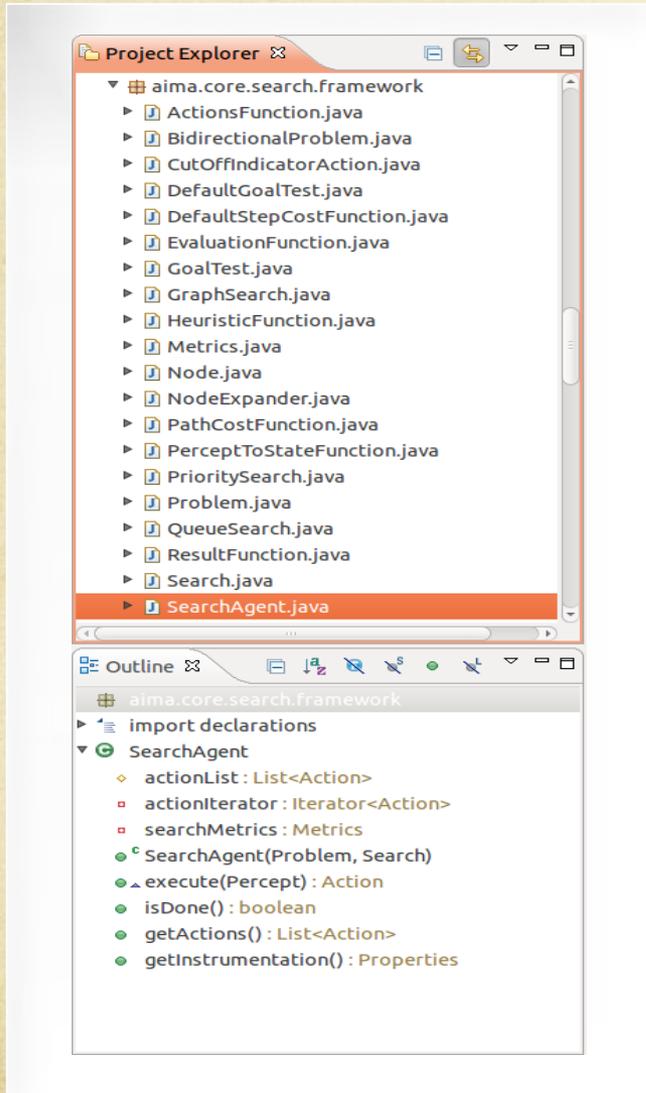


aima.core.search.framework.SearchAgent

SearchAgent(Problem p, Search search):

Realiza la búsqueda de un problema y un algoritmo de búsqueda dado.

Clase Función Coste Singular



aima.core.search.framework.StepCostFunction

public interface StepCostFunction

Calcula el coste de una acción del estado s al estado s'

ANIMA
PROBLEMAS

PUZZLE 8



8-Puzzle

- Un tablero cuadrado (3x3) en el que hay situados 8 bloques cuadrados numerados.
- Un bloque adyacente al hueco puede deslizarse hacia él.
- El juego consiste en transformar una posición inicial en la posición final mediante el deslizamiento de los bloques.

2	1	3
6	5	4
7	8	

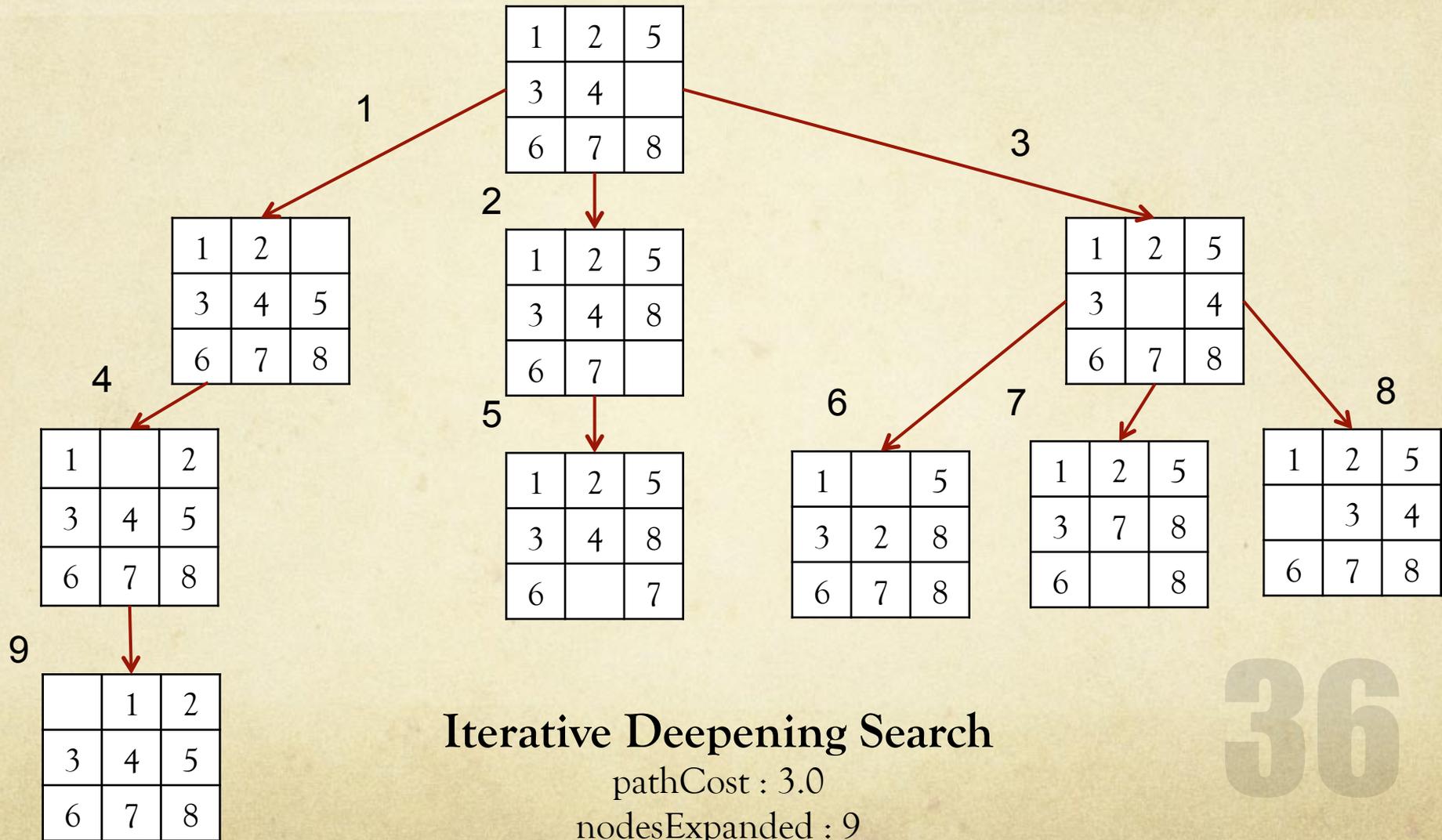
Posible estado inicial

	1	2
3	4	5
6	7	8

Estado final Libro

- Estados: cada una de las posibles configuraciones del tablero.
- Operadores: arriba, abajo, izquierda, derecha (movimientos del hueco)

Espacio Estados Puzzle Ocho



8-Puzzle heurística

- Distancia Manhattan: Distancia verticales+horizontales cada celda.

1	2	3	4	5	6	7	8
0	2	3	1	1	1	1	1

- Número Cuadros Desordenados: 7.

1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1

- Distancia blanco posición final: 4.

2	1	3
6	5	4
7	8	

Posible estado inicial

	1	2
3	4	5
6	7	8

Estado final Libro

Definir el estado

aima.core.environment.eightpuzzle.EightPuzzleBoard

Estado del puzzle como lista enteros.

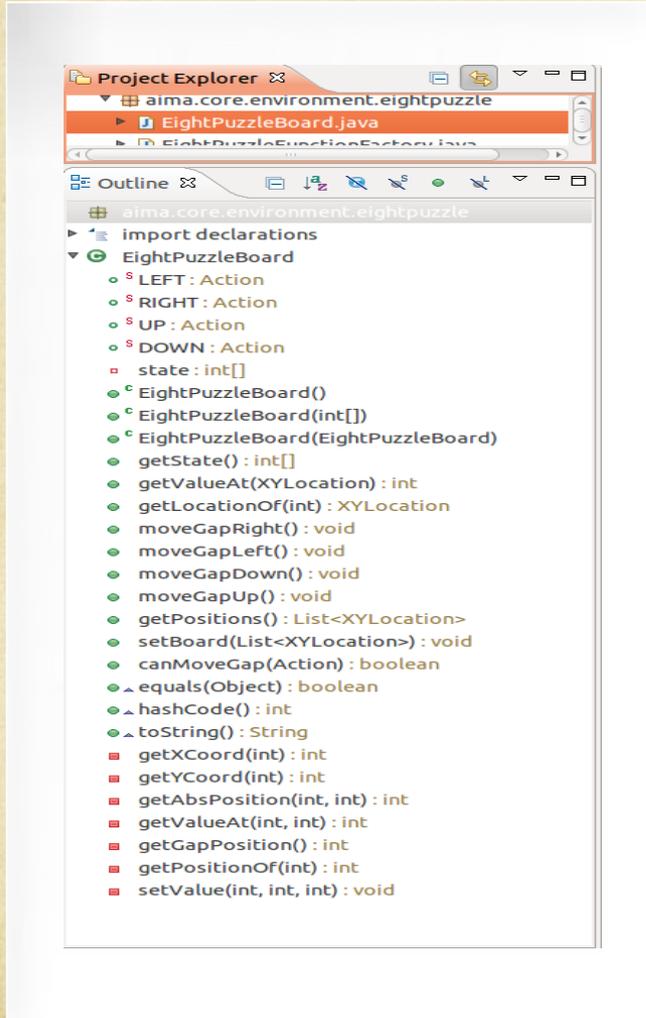
```
state = new int[] { 5, 4, 0, 6, 1, 8, 7, 3, 2 };
```

Las acciones posibles:

- MoveGapRight();
- MoveGapLeft();
- MoveGapUp();
- MoveGapDown();

Funciones auxiliares:

- toString: Imprimir puzzle
- getState: Obtener estado



Definir acciones y resultado

aima.core.environment.eightpuzzle.EightPuzzleFunctionFactory

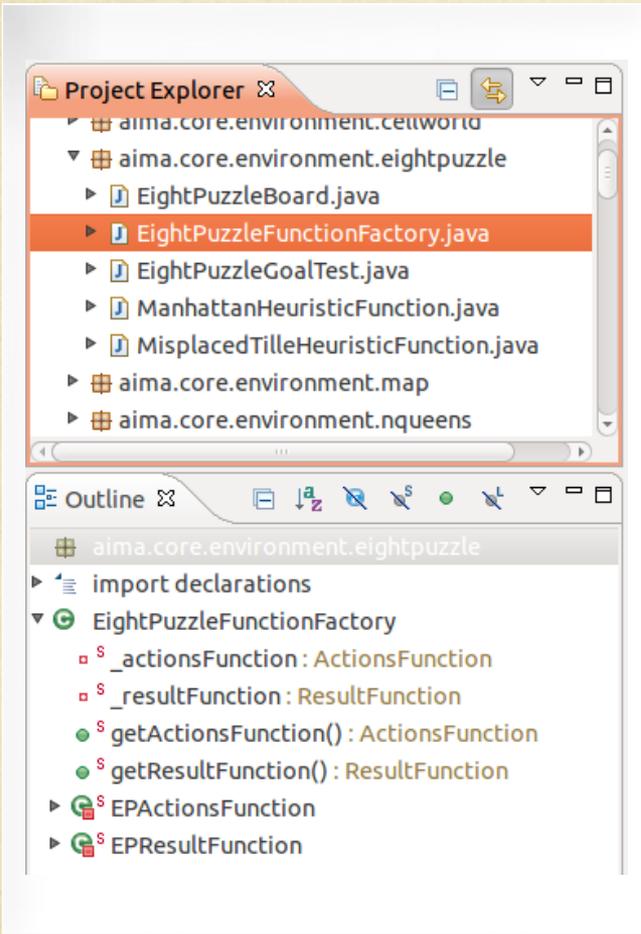
Implementa las acciones

```
if (board.canMoveGap(EightPuzzleBoard.UP)) {  
    actions.add(EightPuzzleBoard.UP);  
}
```

Implementa los resultados

```
if (EightPuzzleBoard.UP.equals(a) &  
    board.canMoveGap(EightPuzzleBoard.UP
```

```
EightPuzzleBoard newBoard = new EightPuzzleBoard(board);  
newBoard.moveGapUp();  
return newBoard;
```



Definir estado Objetivo

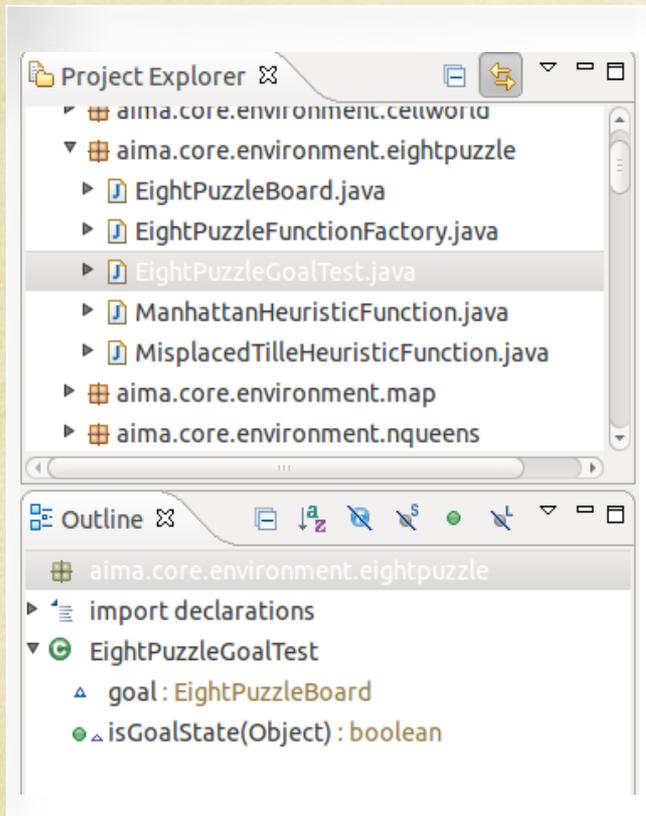
aima.core.environment.eightpuzzle.EightPuzzleGoalTest

Define si un estado es objetivo

```
EightPuzzleBoard board = (EightPuzzleBoard) state;  
return board.equals(goal);
```

EightPuzzleBoard goal =

```
new EightPuzzleBoard(new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8 });
```



Definir Heurística (Manhattan)

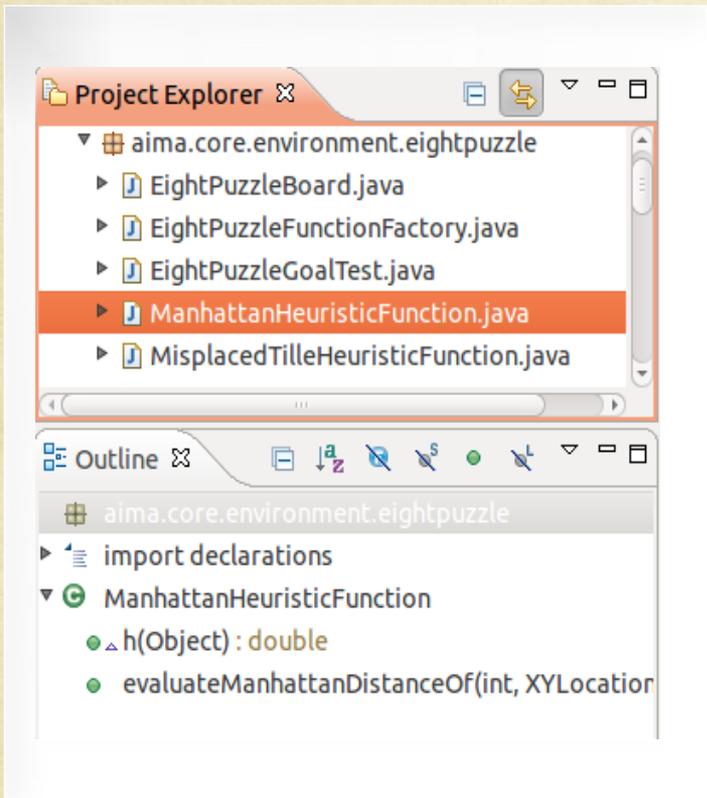
aima.core.environment.eightpuzzle.ManhattanHeuristicFunction

Define la heurística Manhattan

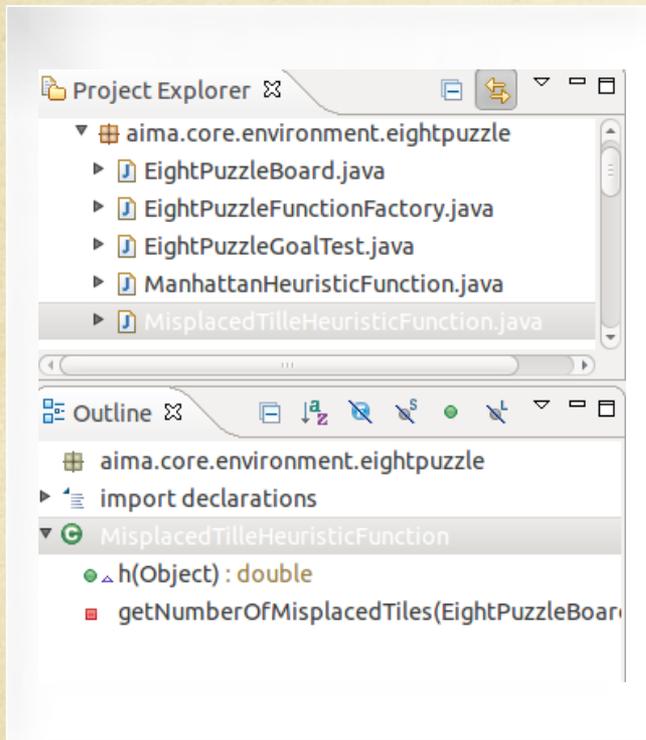
```
public double h(Object state) {  
    EightPuzzleBoard board = (EightPuzzleBoard) state;  
    int retVal = 0;  
    for (int i = 1; i < 9; i++) {  
        XYLocation loc = board.getLocationOf(i);  
        retVal += evaluateManhattanDistanceOf(i, loc);  
    }  
    return retVal;  
}
```

Ejemplo:

```
retVal = Math.abs(xpos - 0) + Math.abs(ypos - 1);
```



Definir Heurística (Cuadros Desordenados)



aima.core.environment.eightpuzzle.MisplacedTilleHeuristicFunction

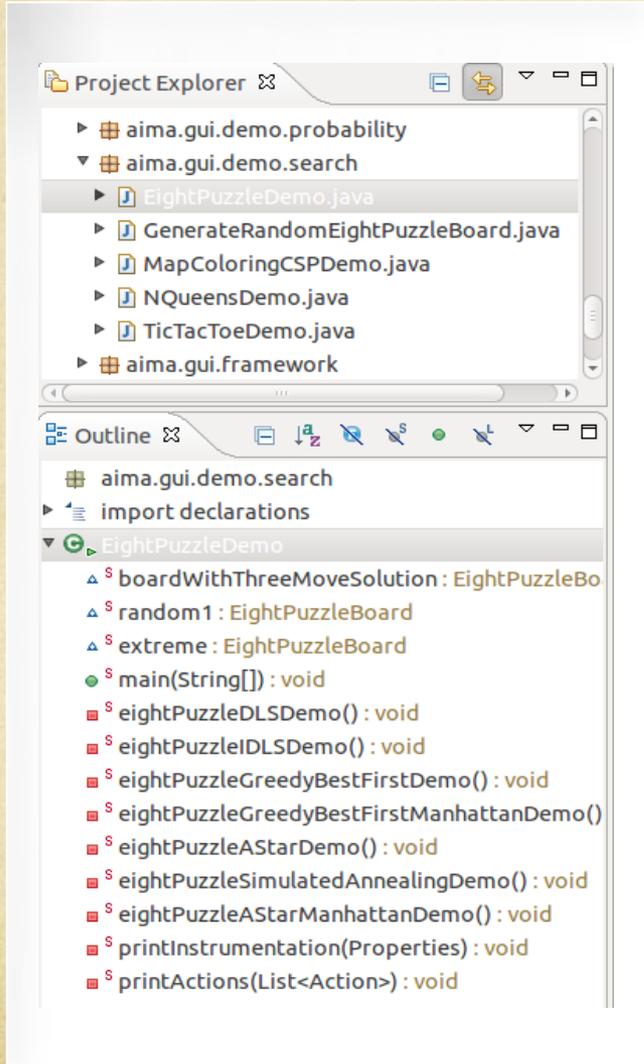
Define la heurística Cuadros Desordenados

```
public double h(Object state) {  
    EightPuzzleBoard board = (EightPuzzleBoard) state;  
    return getNumberOfMisplacedTiles(board);  
}
```

Ejemplo:

```
if (!(board.getLocationOf(0).equals(new XYLocation(0, 0)))) {  
    numberOfMisplacedTiles++;  
}
```

Demo Puzzle Ocho (texto)



aima.gui.demo.search.EightPuzzleDemo

Demo de Puzzle Ocho con los siguientes Algoritmos de Búsqueda:

`eightPuzzleDLSDemo()`:

- Profundidad Limitada, límite 9.

`eightPuzzleIDLSDemo()`;

- Profundidad progresiva.

`eightPuzzleGreedyBestFirstDemo()`;

- Voraces. Heurística Cuadros Desordenados

`eightPuzzleGreedyBestFirstManhattanDemo()`;

- Voraces. Heurística Manhattan

`eightPuzzleAStarDemo()`;

- A*. Heurística Cuadros Desordenados

`eightPuzzleAStarManhattanDemo()`;

- A*. Heurística Manhattan

`eightPuzzleSimulatedAnnealingDemo()`

- Recocido simulado

Demo Puzzle Ocho (texto)

EightPuzzleDLSDemo

```
Problem problem = new Problem(boardWithThreeMoveSolution,  
                               EightPuzzleFunctionFactory.getActionsFunction(),  
                               EightPuzzleFunctionFactory.getResultFunction(),  
                               new EightPuzzleGoalTest());
```

```
Search search = new DepthLimitedSearch(9);  
SearchAgent agent = new SearchAgent(problem, search);
```

```
printActions(agent.getActions());  
printInstrumentation(agent.getInstrumentation());
```

Demo Puzzle Ocho (texto)

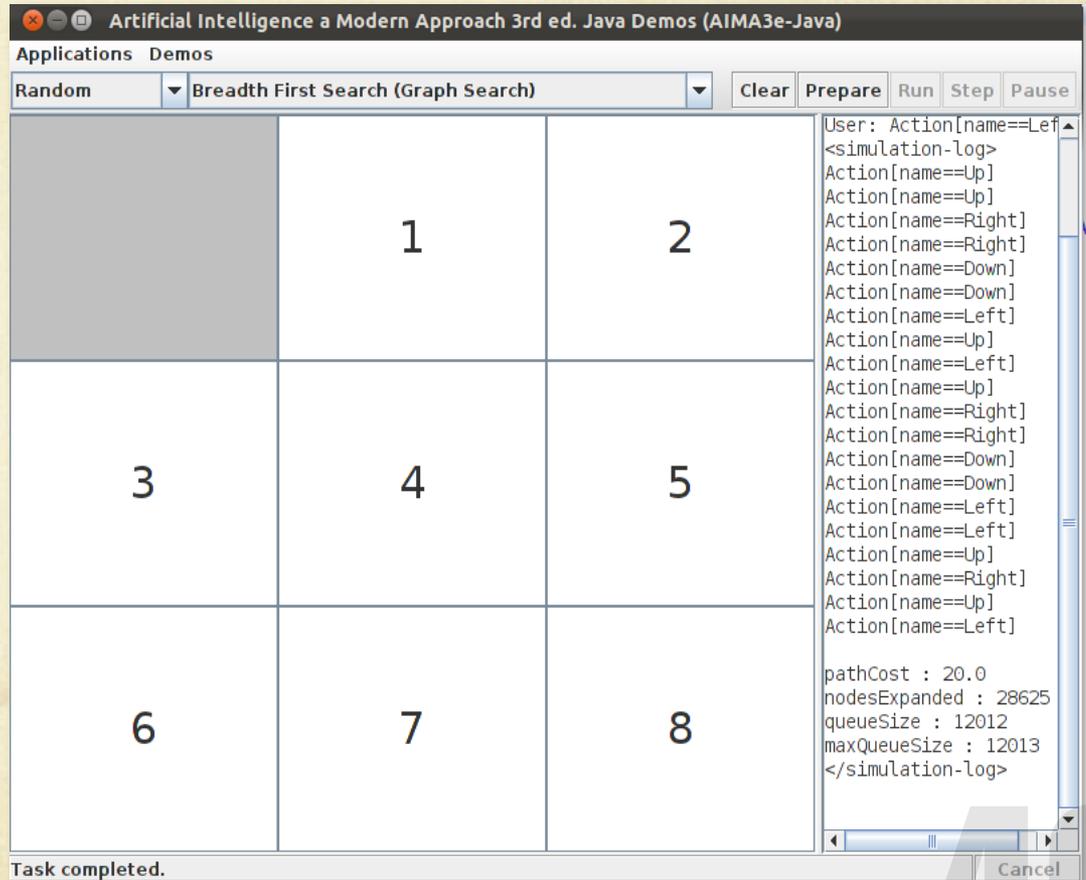
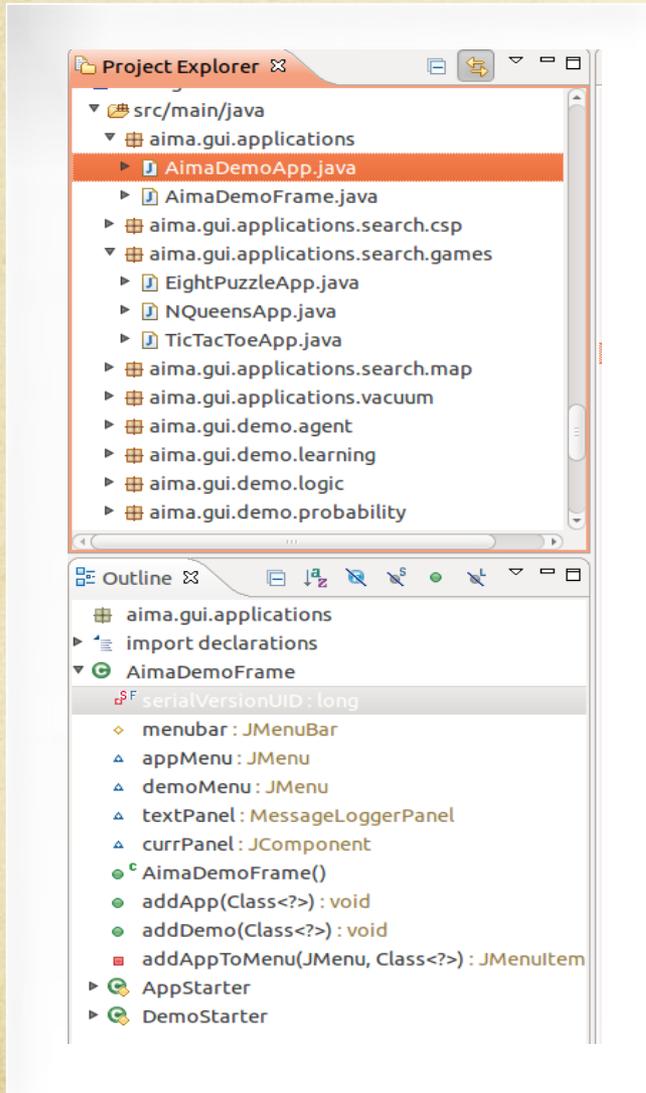
eightPuzzleAStarManhattanDemo

```
Problem problem = new Problem(random1,  
    EightPuzzleFunctionFactory.getActionsFunction(),  
    EightPuzzleFunctionFactory.getResultFunction(),  
    new EightPuzzleGoalTest());
```

```
Search search = new AStarSearch(new GraphSearch(), new ManhattanHeuristicFunction());  
SearchAgent agent = new SearchAgent(problem, search);
```

```
printActions(agent.getActions());  
printInstrumentation(agent.getInstrumentation());
```

Demo Puzzle Ocho (visual)



ALMA
EJERCICIOS
PUZZLE 8



Ejercicio 1 Espacio Estados Puzzle Ocho

Dibuje espacio búsqueda del puzzle 8 partiendo del estado inicial, utilizando el algoritmo “Depth Limited Search(9)”

Con solución:

pathCost : 9.0

nodesExpanded : 10

1	2	5
3	4	
6	7	8

Estado inicial

	1	2
3	4	5
6	7	8

Estado final

Ejercicio 2 Puzzle Ocho (visual)

Ejecute Puzzle Ocho Visual y complete la siguiente tabla comparativa de algoritmos (Estado Inicial Extreme).

Algoritmo	Coste Camino	Nodos Expandidos
Breadth First Search (Graph Search)		
Depth Limited Search (9)		
Iterative Deepening Search		
Greedy Best First Search(MisplacedTileHeuristic)		
Greedy Best First Search (ManhattanHeuristic)		
AStar Search (MisplacedTileHeuristic)		
AStar Search (ManhattanHeuristic)		
Simulated Annealing Search	XXXXXXXXXXXX	

Ejercicio 3 Puzzle Ocho (texto) cont...

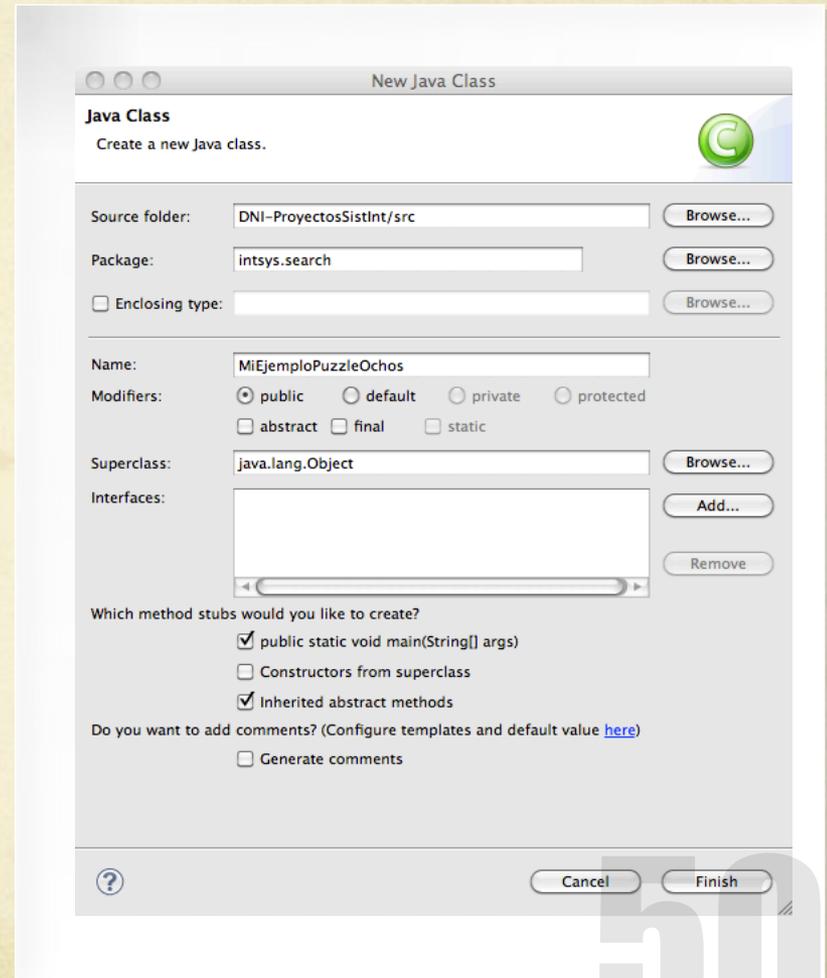
Cree una clase MiEjemploPuzzleOcho en **DNI-ProyectosSistInt** que implemente Profundidad en Grafo.

El ejemplo partirá desde siguiente estado inicial:

{ 1, 2, 5, 3, 4, 0, 6, 7, 8 }

Y obtendrá el estado objetivo con estos valores:

pathCost : 59123.0
nodesExpanded : 120491
queueSize : 39830
maxQueueSize : 42913



Ejercicio 3 Puzzle Ocho (texto)

ProfundidadGrafo

```
Problem problem = new
```

```
    Problem(boardWithThreeMoveSolution, EightPuzzleFunctionFactory.getActionsFunction(),  
            EightPuzzleFunctionFactory.getResultFunction(), new EightPuzzleGoalTest());
```

```
Search search = new DepthFirstSearch(new GraphSearch());
```

```
SearchAgent agent = new SearchAgent(problem, search);
```

```
printActions(agent.getActions());
```

```
printInstrumentation(agent.getInstrumentation())
```

Nota: Utilice de guía `aima.gui.demo.search.EightPuzzleDemo`, pero nunca
modifique el paquete AIMA ni proyecto Practicas



Sistemas Inteligentes

José A. Montenegro Montes

monte@lcc.uma.es

