

Desarrollo de un agente de navegación de interiores basado en objetivos para la comprensión de técnicas de búsqueda

Alvarez Raynald Roberto David, Yunges Bárbara Carina

Estudiantes de la materia Inteligencia Artificial, carrera Ingeniería en Sistemas de Información de la Universidad Tecnológica Nacional, Facultad Regional Santa Fe

Profesora Dra. Gutierrez Milagros

Profesor Dr. Roa Jorge

Abstract. El presente trabajo fue desarrollado en el marco de la asignatura Inteligencia Artificial de la UTN-FRSF durante el primer cuatrimestre del ciclo lectivo 2016. Responde a la consigna de brindar una solución a un problema de búsqueda, implementando un Agente basado en objetivos que se desenvuelve dentro del plano de la facultad sobre un escenario simplificado en el que se descubrirán diferentes obstáculos, pretendiendo encontrar un camino desde la posición donde se halla el agente hasta el destino solicitado a través de diferentes estrategias de búsqueda y permitiendo comparar el comportamiento y resultados de cada una mediante sucesivas ejecuciones. Si durante el proceso, un camino seleccionado deja de ser válido, el agente percibirá la situación y ejecutará distintos operadores, de acuerdo a la estrategia de búsqueda, a los fines de encontrar otros posibles caminos para que el visitante pueda llegar al destino. Se consideran tanto estrategias de búsqueda no informadas (Costo uniforme, profundidad y amplitud) como heurísticas (Avara y A*) Se procedió a definir el problema de inteligencia artificial en cuestión, modelar el ambiente y el agente para su posterior implementación a través de un software que permite simular las operaciones en un entorno controlado, comparar estrategias y analizar resultados.

Keywords: Agente basado en objetivos, estrategias de búsqueda, búsqueda informada, búsqueda no informada, navegación de interiores.

1 Introducción

Uno de los principales problemas a los que se enfrenta una persona al visitar por primera vez a un organismo o institución pública es la falta de información sobre cómo encontrar un determinado lugar, oficina, departamento o persona.

Existen servicios de Navegación en Interiores basados en Internet que utilizan bases de datos de geolocalización, en conjunto con señales de wireless, de luces, y magnéticas, para ayudar a los visitantes.

Sin embargo, en un ambiente real, no detectan eventos tales como que un ascensor deje de funcionar, o se clausure un pasillo por motivos de limpieza, etc.

Por lo tanto, para dar soporte a la problemática planteada, el presente trabajo pretende desarrollar un agente de software que permita encontrar el camino entre dos puntos determinados dentro del interior de un edificio, teniendo en cuenta los distintos tipos de eventos que podrían ocurrir en el trayecto y ante los cuales el agente podría tener que buscar otros posibles caminos para que el visitante pueda llegar al destino, suponiendo que ya se dispone la información sobre geolocalización (mapas y lugares).

El agente está dotado de sensores y cámaras que le permitirían detectar en la posición en que se encuentra: Obstáculos en el camino (por ej., cinta que impide el paso debido a limpieza); si hay o no energía eléctrica; el "bip" del ascensor cuando está fuera de funcionamiento; personas bloqueando el camino o escalera.

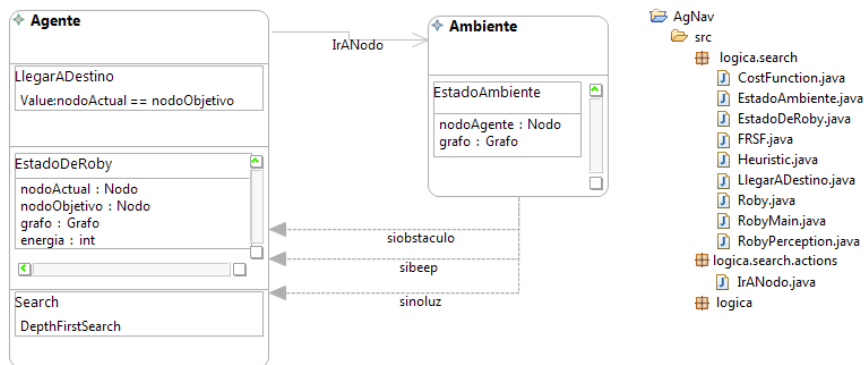
En lo referente a la energía del agente, se considera que se dispone de una carga inicial que se ve reducida de manera diferente entre desplazamientos normales de un punto a otro y aquellos que implican el uso de escaleras ascendentes o descendentes.

En consideración a las características de este problema, se encara su resolución a

través de la implementación de un agente basado en objetivos, puesto que el mismo, basa sus decisiones en la meta que persigue, teniendo en cuenta el estado del entorno. En las siguientes secciones, se definirá primero el problema de búsqueda identificado con las correspondientes operaciones que realizará el agente a los fines de alcanzar el cumplimiento de la prueba de meta. Luego se presentará la solución desarrollada, y se mostrará el uso del software para algunos ejemplos planteados dentro del plano de la facultad.

2 Definición conceptual del problema

Fig. 2.1. Diagrama conceptual del problema, con un único operador genérico, y la estructura de código fuente asociada.



En la Figura 2.1, a la izquierda se muestra el estado que tienen el ambiente y el agente, las percepciones que recibe el agente desde el ambiente, las operaciones que puede realizar el agente en el ambiente, y el estado del agente que indica el cumplimiento del objetivo. Para realizar el modelo, se empleó el software “Idemia”, que permite obtener código dependiente de la tecnología en forma semiautomática. Para ello, la herramienta utiliza un framework existente llamado FAIA.

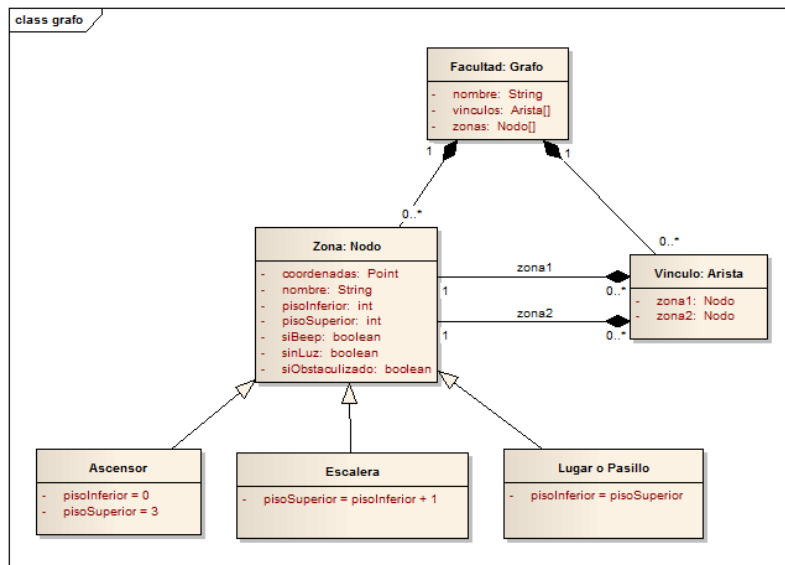
En la parte derecha de la Figura 2.1 podemos observar el árbol del proyecto resultante del diagrama. Se trata de una salida derivada inmediatamente de dicha representación. Se define un único operador “IrANodo”, que será luego instanciado una vez por cada nodo del grafo. Para la integridad del diagrama, se ha explicitado DepthFirstSearch como estrategia de búsqueda. Pero luego se modificará el código fuente generado para permitir la selección de otras.

2.1 Estado del ambiente

Dado que el problema versa esencialmente en la búsqueda de caminos y que el ambiente real consiste en sectores conectados entre sí, el modelado a través de un grafo resulta muy apropiado.

Cada nodo corresponde a algún tipo de estructura que puede aparecer en el recorrido real que se realizaría en ella: entradas principales, ascensores, escaleras, espacios de desplazamiento (pasillos, halls) o zonas (aulas, oficinas, baños), etc. Las aristas son bidireccionales e indicarían un tránsito directo entre los nodos que conectan. Estas zonas generalmente pertenecerán a un único piso.

Fig. 2.2. Representación del Estado del Ambiente mediante un diagrama de clase.



Los únicos tipos de nodos que presentarán valores diferentes para el piso Inferior y el piso Superior son el ascensor (que tiene acceso a todos los niveles) y la escalera (que pertenece a dos niveles sucesivos), brindando al agente navegabilidad entre pisos. La Figura 2.2 explica la estructura del grafo mediante un diagrama de clases.

Los nodos del grafo pueden tener algún “tipo de bloqueo” en relación al tipo de percepción que los origina (diferenciándose cuando se trata de alertas sonoras de beep, percepciones fotosensibles que indican falta de luz, o detección de obstáculos materiales), aunque a los fines prácticos, cualquiera de las mismas tendrá una única respuesta que indica la imposibilidad de transitabilidad a través del mismo.

También se incorporan las coordenadas de los nodos, que serán empleadas para estimar distancias en las estrategias heurísticas.

2.2 Representación del estado del agente

Table 2.1. Gastos de energía asociados a operaciones entre nodos con diferentes combinaciones de piso base y límite entre los que existe posible conectividad.

Nodo Actual			Nodo Siguiete			Esfuerzo Asociado
Piso base	Piso límite	(corresponde a)	Piso base	Piso límite	(corresponde a)	
0	3	Ascensor	0	3	Ascensor	-
0	3	Ascensor	0	1	Escalera	-
0	3	Ascensor	1	1	Pasillo	-
0	3	Ascensor	1	2	Escalera	-
0	1	Escalera	0	3	Ascensor	-
0	1	Escalera	0	1	Escalera	-
0	1	Escalera	1	1	Pasillo	SUBIR
0	1	Escalera	1	2	Escalera	SUBIR
1	1	Pasillo	0	3	Ascensor	-
1	1	Pasillo	0	1	Escalera	-
1	1	Pasillo	1	1	Pasillo	-
1	1	Pasillo	1	2	Escalera	-
1	2	Escalera	0	3	Ascensor	-
1	2	Escalera	0	1	Escalera	BAJAR
1	2	Escalera	1	1	Pasillo	BAJAR
1	2	Escalera	1	2	Escalera	-

El estado del agente, se compondrá del nodo actual en el que se ubica, el nodo objetivo al que desea llegar, la energía disponible y el grafo que modela el ambiente.

El grafo que representa el escenario conocido será una réplica de aquel con el que se define el estado del ambiente, pero, dado que inicialmente se desconoce el estado de bloqueo de los nodos, al comenzar estarán todos seteados como transitables.

Como ya se indicó, cada nodo posee un nombre que se utilizará como identificador del mismo, un rango de pisos a los que pertenece simultáneamente y que debería responder a la conectividad permitida con otros nodos (ya que solo deberían conectarse con nodos que pertenezcan al rango). Dicha limitación en la conectividad, si bien ya deberá respetarse al establecer el escenario, adquiere importancia lógica al momento de definirse una ponderación de costo al desplazarse entre niveles, ya que existirá diferente gasto de energía en operaciones que impliquen moverse entre dos nodos que pertenezcan a diferentes pisos (de manera ascendente o descendente), según los criterios que se muestran en la tabla 2.1 (en la que se toman como ejemplo, las posibilidades que pueden ocurrir en el piso 1, pero la situación sería análoga en otros niveles).

2.3 Estado inicial y final

Table 2.2. Comparación entre el estado inicial y final del grafo

Estado Inicial del grafo	Estado Final del grafo
<ul style="list-style-type: none"> Nodos: Nodo(EntradaPrincipal, 0, 0, false, false, false, Point(50,100)) Nodo(EntradaTorreAdministrativa, 0, 0, false, false, false, Point(30,5)) Nodo(PasilloAulas1a7, 0, 0, false, false, false, Point(80,110)) Nodo(Escalera1PBaP1, 0, 1, false, false, false, Point(50,10)) ... 	<ul style="list-style-type: none"> Nodos: Nodo(EntradaPrincipal, 0, 0, false, false, false, Point(50,100)) Nodo(EntradaTorreAdministrativa, 0, 0, false, false, false, Point(30,5)) Nodo(PasilloAulas1a7, 0, 0, false, true, false, Point(80,110)) Nodo(Escalera1PBaP1, 0, 1, false, false, true, Point(50,10)) ...
<ul style="list-style-type: none"> Aristas: Arista(EntradaPrincipal, HallDeEntrada) Arista(HallDeEntrada, PasilloAulas1a7) 	<ul style="list-style-type: none"> Aristas: Arista(EntradaPrincipal, HallDeEntrada) Arista(HallDeEntrada, PasilloAulas1a7) ...
<ul style="list-style-type: none"> Posición actual: nodoEntradaPrincipal 	<ul style="list-style-type: none"> Posición actual: nodoDeptoDeSistemas
<ul style="list-style-type: none"> Posición Objetivo: nodoDptoSistemas. 	<ul style="list-style-type: none"> Posición Objetivo: nodoDeptoSistemas
<ul style="list-style-type: none"> Energía: 1000. 	<ul style="list-style-type: none"> Energía: 830

En la tabla 2.2 se presenta un esbozo a modo de ejemplo. Los atributos de los nodos del grafo corresponden a: Nodo(nombre, pisoBase, pisoLimite, siBeep, sinLuz, siObstaculizado, coordenadas).

2.4 Prueba de meta

Para alcanzar la meta, el agente deberá llegar al nodo objetivo. Lo cual se define por la condición: Posición Actual == Posición Objetivo

2.5 Operadores

Como ya se dijo, se define un único tipo de operador que indica que el agente se posicionará en un nodo específico: IrANodo

Se muestran 6 instancias de dicho tipo a modo de ejemplo, pero existiría uno por cada nodo del grafo, los cuales mantendrían un comportamiento semejante.

IrAHallDeEntrada; IrAPasilloAulas1-7; irEscalera1PBaP1; irEscalera1P1aP2; irAAscensor; irAAula2...

Las precondiciones para su ejecución son:

- En primer lugar, que posea energía suficiente. La insuficiencia definiría, en cambio, una condición de falla que finalizaría la ejecución total del agente.

- Que el nodo en el que está posicionado el agente no sea el mismo al que corresponde el operador.
- Que el nodo al que corresponde el operador no se encuentre bloqueado.
- Que el nodo en el que se encuentra posicionado el agente, posea una arista con el nodo al que corresponde el operador.
- Las postcondiciones del operador son:
 - Energía del agente decrementada según lo indicado en la Tabla 2.1.
 - La posición actual del agente se asigna al nodo que corresponde el operador.

A continuación se presenta el funcionamiento de un operador mediante pseudocódigo.

```

IrAHallDeEntrada
if(energia-costoOperacion(nodoActual, nodoObjetivo)>0
  &&nodoActual != nodoObjetivo&& (aristas.contains(nodoActual, hallDeEntrada) || aristas.contains(hallDeEntrada, nodoActual)) && !hallDeEntrada.isBloqueado())
  energia -= costoOperacion(nodoActual, nodoObjetivo)
  nodoActual = hallDeEntrada
endif
dondeisBloqueado() retorna: siBeep || sinLuz || siObstaculizado;
ycostoOperacion(nodoActual, nodoObjetivo)

```

Retorna el costo de desplazarse entre ambos nodos, según la Tabla 2.1.

2.6 Percepciones

Mediante éstas, el agente puede percibir del ambiente, la posibilidad de transitar a través de un nodo. El agente realiza tales percepciones de un nodo, al hallarse en alguno adyacente a aquel.

- Obstáculos (humanos o no): bloquean el acceso a un nodo
- Luz: permite al agente saber si no hay energía eléctrica
- Bip del ascensor: Indica que el ascensor está fuera de funcionamiento

2.7 Definición conceptual del problema de búsqueda

El objetivo del agente será encontrar un camino desde el nodo inicio hacia el nodo destino.

Cada nodo presentará una o varias aristas hacia otros nodos, algunos de los cuales podrían brindar conexiones que permitan acceder a otros pisos del edificio.

Por consiguiente, el comportamiento del agente consistirá en:

- Percibir cuáles de los nodos con los que se tienen aristas, se encuentran bloqueados.
- Calcular un camino posible hacia el destino, sin pasar por ninguno de los nodos que se saben bloqueados.
- Avanzar un nodo
- Si no se ha alcanzado el nodo objetivo, ni encontrado condición de falla (energía agotada), repetir el procedimiento

3 Diseño de la solución del problema

En la figura 3.3 se presenta el diagrama de clases de la solución desarrollada en java, a partir del código FAIA generado por IDEM-IA.

Se resaltan en color las clases que fueron implementadas extendiendo las provistas por el framework (que se ocupan de la ejecución de los algoritmos de búsqueda) a los fines de dar solución al problema vigente.

Al iniciarse el algoritmo, el constructor del agente creará las instancias de IrANodo correspondientes a cada nodo del grafo, y serán utilizados para realizar la expansión del árbol de búsqueda y para actuar sobre el ambiente al momento de tomarse la decisión.

El algoritmo ejecuta el ciclo de percepción-acción, hasta que se cumple la prueba de meta (establecida en la clase LlegarAObjetivo) o hasta que ocurre la condición de falla (energía agotada).

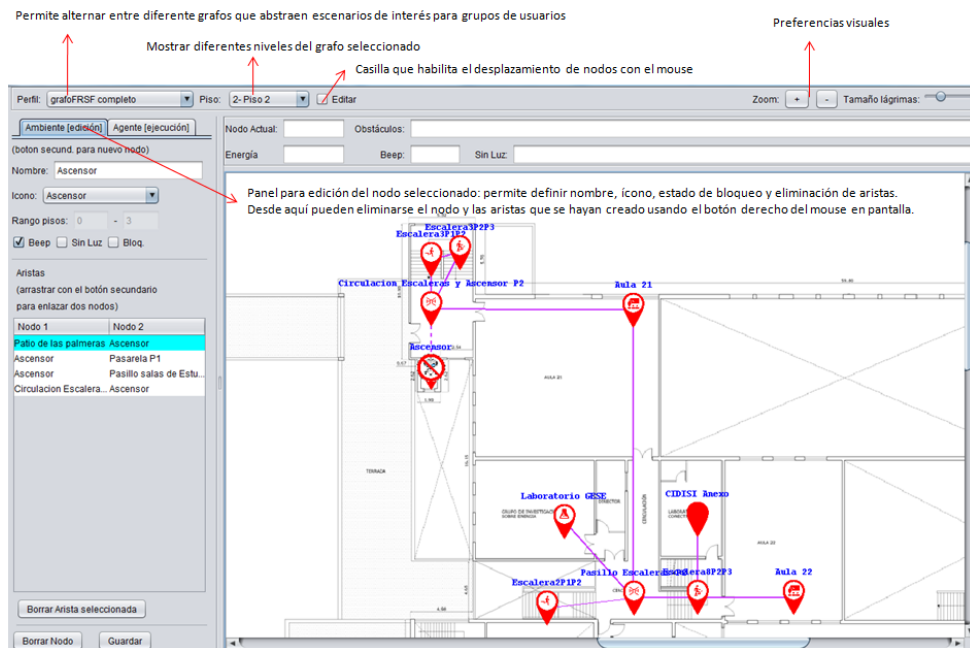
Las estrategias de búsqueda implementadas son:

- No informadas: Costo uniforme, profundidad y amplitud
- Informadas: Avara y A*

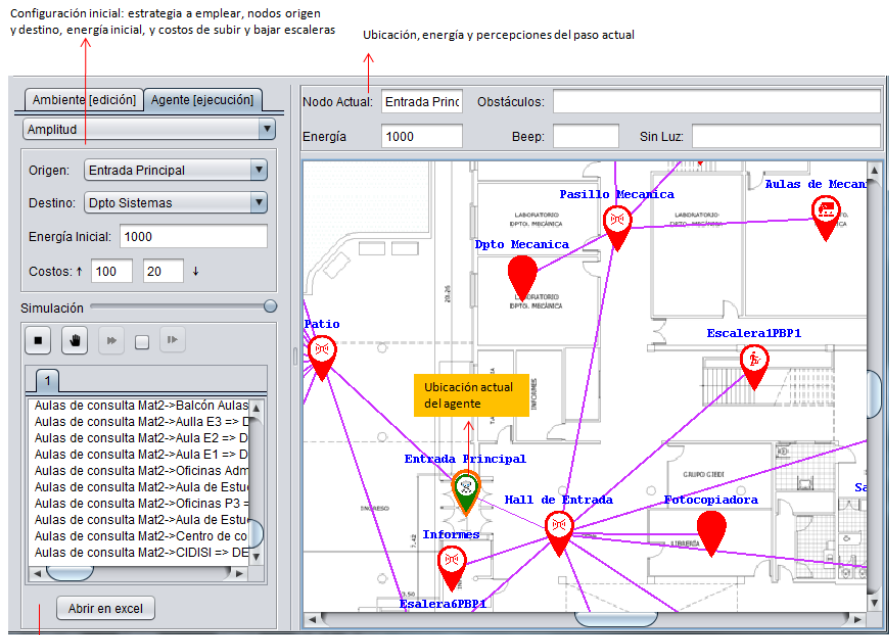
De la figura 3.3 puede notarse que para las funciones heurísticas se calcula el costo mediante las distancias (ya que permite definir una función monótona decreciente para la función heurística) mientras que en la de Costo Uniforme se utiliza la energía empleada (que depende de la cantidad de iteraciones realizadas), ya que es la variable que interesa medir a la hora de evaluar resultados.

El software resultante permite la definición del ambiente sobre el que operará el agente, y la ejecución de la simulación con diferentes valores para los atributos y la selección de la estrategia de búsqueda a emplear, presentando una salida en pantalla del “razonamiento” desarrollado, lo anterior se encuentra señalado en la fig. 3.1 y 3.2.

Fig. 3.1. Sistema funcionando, en modo edición del ambiente

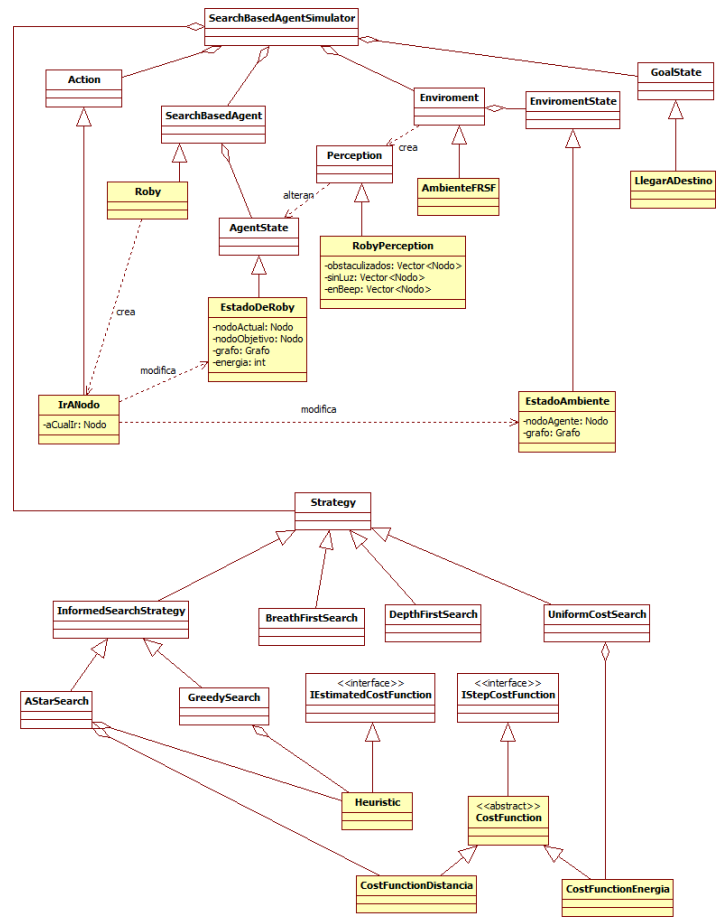


EST 2016, 19º Concurso de Trabajos Estudiantiles
Fig. 3.2. Sistema funcionando en modo simulación



Panel de ejecución: Los primeros botones permiten iniciar e interrumpir la simulación, pausar y avanzar de a un paso. La casilla de activación activa el "modo depuración", que permite visualizar el proceso de cada operador que se expande en el árbol de búsqueda. El área de texto inferior, muestra una pestaña por cada por cada sucesiva iteración indicando que nodos se pudieron expandir y el operador seleccionado.

Fig. 3.3. Diagrama de las clases implementadas que extienden FAIA



4 Resultados

En esta sección analizaremos cual o cuales son los métodos que nos permiten llegar a la solución óptima y el porqué.

A los fines de realizar la comparación entre las distintas estrategias, desarrollamos las siguientes pruebas con nodo inicio en la Entrada Principal. Y considerando que el costo de desplazamiento del agente entre 2 nodos es de 10 unidades de energía, salvo en el caso de las escaleras.

En las dos primeras el nodo destino es el Departamento Civil ubicado en el primer piso; en la tercera la finalidad es llegar al Departamento de Sistemas situado en el tercer piso, en la cuarta, el destino es el Aula 21 localizada en el segundo piso, y en la quinta se trata de la Secretaría académica, localizada en el primer piso. En las 2 últimas pruebas se considera una variación del esfuerzo para subir o bajar escaleras.

El ambiente seleccionado es un plano precargado, se puede visualizar en el sistema agNav para el perfil “grafoFRSF completo” que contiene casi 140 nodos.

Como estrategia de búsqueda informada utilizamos Búsqueda Avara y A*. En los ejemplos de la tabla 4.1 no se ve diferencia entre estos 2 métodos debido a las características del grafo. Ambos encuentran una solución en sus tres pruebas.

Avara utiliza la función heurística previamente definida, en cambio A* agrega la función de costo, midiendo así también la distancia entre un nodo determinado y el nodo al que se movió el agente.

En la prueba 2 de la tabla 4.1, vemos que el método de búsqueda costo uniforme obtuvo una mejor solución que las estrategias de búsquedas informadas, esto se debe a que el primero resuelve el problema utilizando una función que busca desarrollar primero la rama del árbol de búsqueda con menor costo acumulado, mientras que las heurísticas trabajan con la distancia (medida a la que no damos importancia en los resultados analizados).

La estrategia de profundidad, ver tabla 4.1, para este problema no brinda solución: ingresa en un bucle infinito, el programa finaliza cuando el agente se queda sin energía.

El problema radica en que la estrategia en sí no sirve para este tipo de problemas: el algoritmo, siguiendo el orden definido de operadores, establece un camino y avanza un nodo, y repite el procedimiento expandiendo siempre el primer operador que satisface las precondiciones. Como resultado, no tarda mucho en ocurrir que en una iteración posterior se percibe como camino factible uno de sus pasos previos (ya que, además, existe conectividad bidireccional), y queda entonces iterando entre uno y otro hasta agotar la energía.

La estrategia en amplitud converge, ver tabla 4.1, pero muy difícilmente pueda encontrar un camino óptimo en un grafo de complejidad mínima, ya que la selección del operador a expandir solo depende del orden en que ha sido definido.

Para estas últimas estrategias es muy importante el orden en el que se definen los operadores, una definición alternativa cambiaría el resultado de la ejecución.

Los mejores resultados fueron encontrados con estrategias de búsqueda informada y costo uniforme, este último es óptimo cuando los bloqueos del ambiente no generan un redireccionamiento en el camino que debe seguir el agente.

Por otro lado, las estrategias heurísticas aplicadas se ven perjudicadas por el hecho de que están desarrolladas en función de distancias, mientras que el diseño del grafo, si bien respeta cierta precisión geográfica de los sectores representados, no es riguroso ni mucho menos, puesto que el foco está puesto en la conectividad entre componentes. A esto se añade que luego la eficacia del algoritmo no es medida por la distancia recorrida, sino por la energía consumida (y aunque existe cierta correspondencia, está muy afectada por la densidad del grafo).

En la prueba 4, ver Tabla 4.2, al ser más costoso subir las escaleras, la estrategia de costo uniforme dirige al agente por el ascensor disminuyendo así el consumo de energía al mínimo. Al ser más costoso bajar las escaleras, decide no hacerlo en el último tramo, por lo cual no puede utilizar el ascensor hasta el 3er piso, es por eso que sube por las escaleras.

Table 4.1. Primeras 3 pruebas realizadas

	Prueba 1	Prueba 2	Prueba 3
	Origen: Entrada Principal Destino = Depto. Civil Costo de subir escalera = 10 Costo de bajar escalera = 10	Origen: Entrada Principal Destino: Dpto Civil Costo de subir escalera = 30 Costo de bajar escalera = 10	Origen: Entrada Principal Destino = Depto. Sistemas Costo de subir escalera = 30 Costo de bajar escalera = 10
Amplitud	IrA Hall de Entrada IrA Esalera6BBP1 IrA Pasillo de Matemática IrA Pasillo Escaleras P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Esalera6BBP1 IrA Pasillo de Matemática IrA Pasillo Escaleras P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Pasillo Mecánica IrA Patio de las palmeras IrA Ascensor IrA Pasillo salas de Estudio IrA Pasillo Dptos Sistemas IrADpto Sistemas
U.Consum.	60u	80 u	70u
Profundidad	IrA Patio IrA Entrada Torre Adm. IrACirculacion Torre PB IrA EscaleraTorreBBP1 IrA EscaleraTorreP1P2 IrA EscaleraTorreP2P3 IrA EscaleraTorreP1P2... No se logra la meta	IrA Patio IrA Entrada Torre Adm. IrA Circulación Torre PB IrA EscaleraTorreBBP1 IrA EscaleraTorreP1P2 IrA EscaleraTorreP2P3 IrA EscaleraTorreP1P2... No se logra la meta	IrA Patio IrA Entrada Torre Adm. IrA Circulación Torre PB IrA EscaleraTorreBBP1 IrA EscaleraTorreP1P2 IrA EscaleraTorreP2P3 IrA EscaleraTorreP1P2.... No se logra la meta
U. Consum.	Todas las unidades disp.	Todas las unidades disp.	Todas las unidades disp.
Costo Uniforme	IrA Hall de Entrada IrA Esalera6BBP1 IrA Pasillo de Matemática IrA Pasillo Escaleras P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Pasillo Mecanica IrA Patio de las palmeras IrA Ascensor IrA Pasarela P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Pasillo Mecánica IrA Patio de las palmeras IrA Ascensor IrA Pasillo salas de Estudio IrA Pasillo Dptos Sistemas IrADpto Sistemas
U. Consum.	60u	70u	70u
Búsqueda Avara	IrA Hall de Entrada IrA Escalera1BBP1 IrA Pasillo Sanitarios P1 IrA Pasillo Escaleras P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Escalera1BBP1 IrA Pasillo Sanitarios P1 IrA Pasillo Escaleras P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Pasillo Mecánica IrA Patio de las palmeras IrA Ascensor IrA Pasillo salas de Estudio IrA Pasillo Dptos Sistemas IrADpto Sistemas
U. Consum.	60u	80u	70u
A*	IrA Hall de Entrada IrA Escalera1BBP1 IrA Pasillo Sanitarios P1 IrA Pasillo Escaleras P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Escalera1BBP1 IrA Pasillo Sanitarios P1 IrA Pasillo Escaleras P1 IrA Pasillo dpto Civil IrADpto Civil	IrA Hall de Entrada IrA Pasillo Mecánica IrA Patio de las palmeras IrA Ascensor IrA Pasillo salas de Estudio IrA Pasillo Dptos Sistemas IrADpto Sistemas
U. Consum.	60u	80u	70u

La definición de costos mayores entre subir y bajar escaleras, siempre afectarán la estrategia de costo uniforme priorizando los trayectos que eviten el uso de las escaleras, de modo que procurará acceder siempre a través de los ascensores cuando no impliquen un recorrido mayor a la cantidad de unidades en que lo supera dicho esfuerzo.

En realidad, siempre que movilizarse por escaleras implique mayor energía que los desplazamientos normales, el agente tenderá a moverse siempre en dirección vertical hacia su objetivo, no operando nunca en sentido opuesto, como si ocurriría en cambio, con amplitud o profundidad (que no presentarían problemas en subir una escalera y luego volver a bajar si el orden de los operadores así lo estableciese). A menos, claro, que descubra que el camino está inaccesible a posteriori.

Table 3.2. Prueba número 4

	Prueba #4 con costo de subir escaleras 5 veces mayor que bajar	Prueba #4 con costo de bajar escaleras 5 veces mayor que subir
	Destino = Aula 21 Unidades de energía del agente = 1000 Costo de subir escalera= 100; Costo de bajar escalera = 20 Origen: Pasillo Mecánica Destino: Secretaría Académica Ambiente: tiene bloqueado el ascensor del patio de las palmeras	Destino = Aula 21 Unidades de energía del agente = 1000 Costo de subir escalera= 20; Costo de bajar escalera = 100 Origen: Pasillo Mecánica Destino: Secretaría Académica Ambiente: tiene bloqueado el ascensor del patio de las palmeras
Costo Uniforme	IrA Patio de las palmeras IrA Pasillo Mecánica IrA Hall de Entrada IrA Entrada Principal IrA Patio IrA Entrada Torre Adm. IrA Circulación Torre PB IrA Ascensor 1 Torre Adm. IrA Circulación Torre P1 IrA Secretaría Académica	IrA Patio de las palmeras IrA Escalera3PBP1 IrA Pasarela P1 IrA Circulación Torre P1 IrA Secretaría Académica
Energía	Energía consumida = 100 Energía restante = 900	Energía consumida = 60 Energía restante = 940

5 Conclusión

Este trabajo práctico ha culminado con éxito el desarrollo de un agente inteligente permitiéndonos comprender como este se relaciona con el mundo en el cual se desenvuelve, visualizando, experimentando y comparando las técnicas vistas en clase que fueron utilizadas por el agente para tomar decisiones sobre las acciones a emprender. El uso del entorno de desarrollo IDEM-IA y el framework FAIA, provisto por la cátedra, ha sido de gran utilidad tanto para la comprensión de los conceptos como el desarrollo de este trabajo.

Los resultados obtenidos con las estrategias Avara, A* y de Costo Uniforme fueron muy satisfactorios, Amplitud siempre convergió en las pruebas realizadas, aunque de manera no eficiente, mientras que Profundidad siempre quedó en un bucle debido a la naturaleza del modelo.

Por otro lado, destacamos que el software obtenido puede brindar un aporte pedagógico al permitir la elección de la estrategia de búsqueda, edición completa del ambiente en el que se mueve el agente (no sólo desde el grafo, sino en la posibilidad de reemplazar los planos y los tipos de nodos, pudiéndose utilizar en otros edificios). Y por la capacidad de mostrar el paso a paso de la ejecución de los algoritmos, distinguiendo las operaciones correspondiente al despliegue del árbol de búsqueda y la acción seleccionada en cada iteración.

6 Agradecimientos

El presente trabajo fue realizado bajo la supervisión de los profesores de la cátedra de Inteligencia Artificial de la Universidad Tecnológica Nacional, FRSF, a quienes nos gustaría expresar nuestro agradecimiento por haber apoyado y contribuido a la presentación de este trabajo.

7 Referencias

1. Artificial Intelligence: A Modern Approach(Third edition) by Stuart Russell and Peter Norvig
2. <https://code.google.com/archive/p/faia/wikis>
3. <http://code.google.com/p/idemia>
4. <http://code.google.com/p/faia>