

ANEXO A PLATAFORMAS PARA DESARROLLAR AGENTES

Hasta ahora diversas empresas y universidades han desarrollado diferentes plataformas para implementar sistemas de agentes, algunas de propósito específico y otras de propósito general. Más de la mitad de proyectos utilizan el lenguaje de programación Java como base para implantar sistemas de agentes por su capacidad de funcionar sobre Internet y por su máquina virtual que le permite funcionar en diferentes plataformas de hardware.

Los desarrollos se han hecho desde el punto de vista comercial y como proyectos académicos y de investigación¹⁶

La tabla que sigue contiene algunos de los proyectos que proponen una plataformas para desarrollo de agentes, en cada una se indica su nombre, la empresa o universidad que la desarrolla, la página web donde hay información sobre la misma, un resumen de sus principales características, si considera el manejo de ontologías mixtas, eventos inesperados, planeación y definición separada de agentes e interacciones.

PLATAFORMAS COMERCIALES

OM: Soporta Ontologías Mixtas
EI: Soporta Eventos Inesperados

P: Planifica automáticamente
SAI: Implementa Agentes separados de las Interacciones (escenarios)

| PROYECTO | DESCRIPCIÓN | OM | EI | P | SAI |
|--|--|----|----|----|-----|
| AgentTalk <i>NTT, Ishida Lab y Univ. Kioto Japón</i> | http://www.kecl.ntt.co.jp/csl/msrg/topics/at/ Es un lenguaje para descripción de protocolos de sistemas multiagentes, por ejemplo, el protocolo Contract Net. Escrito en Common Lisp. Producto disponible sólo en Japón. | no | no | no | no |
| AgentBuilder <i>Reticular Systems, Inc. USD\$999</i> | http://www.agentbuilder.com/ Es un conjunto de herramientas para construir agentes inteligentes de software. Consiste de dos componentes principales el Toolkit y el sistema de Run Time. Permite definir agentes en Java que se comunican en KQML. Permite construir agentes inteligentes y programas basados en Agentes. Soporta además CORBA y TCP/IP. | no | no | no | no |
| Agentx <i>International Knowledge Systems</i> | http://www.mobileagenttech.com/ Es una plataforma que aprovecha la máquina virtual de Java para ejecutar Agentes móviles que eficientemente pueden ejecutarse en un máquina, concurrentemente o migrar de un lugar a otro. Compatible 100% con JDK de Sun, además cuenta con funcionalidad de ORB. Permite que programas complejos se escriban como procesos pequeños que pueden distribuirse. | no | no | no | no |
| Aglets <i>IBM Japón</i> | http://www.trl.ibm.com/aglets/ Un aglet es un objeto Java que puede moverse de una computadora a otra en Internet llevándose sus datos y su estado de ejecución. Cuenta con un mecanismo de seguridad para proteger a los servidores de aglets intrusos. | no | no | no | no |
| Concordia <i>Mitsubishi Electric, Information Technology Center America</i> | http://www.merl.com/HSL/Projects/Concordia/ Es una plataforma de desarrollo de aplicaciones basadas en agentes móviles para acceder información anywhere, anytime y en dispositivos que soportan Java. Los usuarios pueden estar desconectados de la red durante algún tiempo. Los agentes pueden operar en LAN, Intranet o Internet. Puede usar comunicación wired o wireless. Soporta diversos dispositivos como computadoras de escritorio, PDAs, portátiles y teléfonos inteligentes. | no | no | no | no |

¹⁶ <http://www.agentbuilder.com/AgentTools>

PLATAFORMAS COMERCIALES

OM: Soporta Ontologías Mixtas
EI: Soporta Eventos Inesperados

P: Planifica automáticamente
SAI: Implementa Agentes separados de las Interacciones (escenarios)

| PROYECTO | DESCRIPCIÓN | OM | EI | P | SAI |
|---|--|----|----|----|-----|
| DirectIA SDK <i>MASA Group – Adaptive Objects</i> | http://www.animaths.com/ Es una plataforma que permite crear agentes que se adaptan a un ambiente dinámico para descubrir nuevos comportamientos y verificar su eficiencia en tiempo real. Se proporciona como bibliotecas en C++ para Windows 98 y NT. Comprende una aplicación de prueba, un tutorial, y herramientas para depurar y afinar los agentes. | no | no | no | no |
| Gossip <i>Tryllian</i> | http://www.tryllian.com/ Es una plataforma de agentes móviles que busca información en Internet para un usuario. Una vez que el agente conoce los intereses entra a un Marketplace en un servidor e intercambia información con otros agentes mediante una conversación. Los agentes pueden manejar hardware y software. | no | no | no | no |
| Grasshopper <i>IKV++</i> | http://www.grasshopper.de/index.html Es la primera plataforma de agentes móviles que cumple con el estándar MASIF construido sobre CORBA para proporcionar la integración de los sistemas cliente/servidor tradicionales con los agentes. Los agentes pueden aprovechar las líneas de alta velocidad de una red local y el acceso local a datos. | no | no | no | no |
| IGEN <i>CHI Systems</i> | http://www.cognitiveagent.com/ Es un toolkit y un ambiente de desarrollo para construir aplicaciones con agentes inteligentes. | no | no | no | no |
| Intelligent Agent <i>Factory Bits & Pixels</i> USD\$289 (incluyendo código fuente USD\$479) | http://www.bitpix.com/ Permite ahorrar tiempo en la construcción de soluciones basadas en agentes inteligentes. Los agentes se controlan mediante reglas escritas en CLIPS y se generan mediante especificaciones simples de workflows. | no | no | no | no |
| Intelligent Agent Library <i>Bits & Pixels</i> USD\$459 (incluyendo código fuente USD\$1495) | http://www.bitpix.com/ Es una biblioteca que proporciona un ambiente de trabajo que incluye facilidades para que los agentes se comuniquen y construir grandes conglomerados de agentes. Hay una plataforma basada en KQML que ilustra agentes que realizan actividades en la web. También soporta agentes móviles. | no | no | no | no |
| JACK <i>Intelligent Agents Agent Oriented Software Pty, Ltd</i> | http://www.agent-software.com.au/ Es un sistema de agentes que proporciona una arquitectura y capacidades para desarrollar agentes de software en aplicaciones distribuidas. Escrito en Java, usa una extensión de Java (JACK Agent Language) para proporcionar los beneficios de Java. | no | no | no | no |
| JAM <i>Intelligent Reasoning Systems</i> | http://members-http-2.rwc1.sfba.home.net/marcush/IRS/ Es una arquitectura basada en el modelo Belief-Desire-Intention sobre el Procedural Reasoning System (PRS) de Georgeoff, Rao, Lansky, et. al. JAM se puede aplicar a diferentes dominios. Soporta razonamiento top-down, goal-based y bottom-up data driven. | no | no | no | no |
| Jumping Beans <i>Ad Astra Engineering, Inc.</i> | http://www.jumpingbeans.com/ Es una plataforma que permite a los desarrolladores crear agentes móviles basados en Java. Aún en sitios donde la aplicación aún no estaba instalada. | no | no | no | no |
| LPA Agent Toolkit <i>Logic Programming Associates, Ltd.</i> | http://www.lpa.co.uk/atk.htm Es un toolkit que permite escribir programas orientados a agentes usando un enfoque basado en lógica. Este toolkit le proporciona al usuario un esqueleto de un agente el cual exhibe las principales características: autonomía, persistencia, cooperación y adaptabilidad. Los agentes se escriben en Prolog. | no | no | no | no |

PLATAFORMAS COMERCIALES

OM: Soporta Ontologías Mixtas
EI: Soporta Eventos Inesperados

P: Planifica automáticamente
SAI: Implementa Agentes separados de las Interacciones (escenarios)

| PROYECTO | DESCRIPCIÓN | OM | EI | P | SAI |
|---|--|----|----|----|-----|
| MadKit <i>Madkit Development Group</i> | http://www.madkit.org/ MadKit Es una plataforma multiagente basada en Java construido sobre un modelo organizacional. Provee las facilidades generales de agente (gestión de su ciclo de vida, paso de mensajes, distribución, entre otras), y permite alta heterogeneidad en arquitectura se agentes y lenguajes de comunicación además de otras personalizaciones. | no | no | no | no |
| Microsoft Agent <i>Microsoft Corp.</i> | http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msagent/agentstartpage_7gdh.asp El agente de Microsoft versión 2.0 proporciona una tecnología para crear una interfaz conversacionales para aplicaciones y páginas web. Proporciona capacidades de animación, interactividad y versatilidad con facilidades para su desarrollo. | no | no | no | no |
| Network Query Language <i>NQL Solutions</i> | http://www.nqli.com/ NQL Es un lenguaje de script para construir agentes y aplicaciones en la web. La comunicación es una parte importante de NQL. El acceso a Internet con protocolos comunes como http, ftp, nntp, y telnet estan interconstruidos en el lenguaje. Conexiones TCP/IP personalizadas pueden construirse usando sockets en NQL. Accesa e-mail, bases de datos y aplicaciones de escritorio. | no | no | no | no |
| Open Cybele <i>Intelligent Automation Inc.</i> | http://www.opencybele.org/view.asp?type=view&subID=16&topicID=1 Es un ambiente de ejecución construido sobre la plataforma de Java para el control de la ejecución de agentes. Los agentes y el manejo de eventos lo hace OpenCybele en vez de la MV de Java. | no | no | no | no |
| Pathwalker <i>Fujitsu</i> | http://www.labs.fujitsu.com/free/paw/ Pathwalker es una biblioteca de agentes orientada a procesos distribuidos escrita completamente en Java. Tiene algunas características que se usan para programación distribuida como mensajes asíncronos, identificador global único y procesos ligeros, manejo de código movil. | no | no | no | no |
| Swarm <i>Swarm Development Group</i> | http://www.swarm.org/ Es un paquete de software para simulación de sistemas multiagente. Se propone como una herramienta para investigadores de varias diciplinas, especialmente en vida artificial. La plataforma básica para este sistema es Solaris sobre Sparc Sun, GNU/Linux y Windows NT. | no | no | no | no |
| Topia Personal Agents <i>Topia Ventures</i> | http://www.topiaventures.com/index.asp Realizan tareas en nombre del usuario, como por ejemplo búsqueda, adquisición, filtrado y distribución de información, negociación para servicios y productos, colaborando con otros agentes para realizar las tareas y monitorear las actividades de los agentes. | no | no | no | no |
| UMPRS <i>Intelligent Reasoning Systems</i> | http://members-http-1.rwc1.sfba.home.net/marcush/IRS/ Es una arquitectura basada en el modelo Belief-Desire-Intention (BDI) sobre el Procedural Reasoning System (PRS). UMPRS se aplica a diferentes dominios, está desarrollado en C++. | no | no | no | no |
| Voyager <i>Object Space</i> | http://www.objectspace.com/ Es un ORB que combina las capacidades de los agentes autonomos móviles y la invocación de métodos remotos con soporte de CORBA y tiene servicios de directorio, persistencia y mensajes multicast. | no | no | no | no |

PLATAFORMAS ACADÉMICAS Y DE INVESTIGACIÓN

OM: Soporta Ontologías Mixtas
EI: Soporta Eventos Inesperados

P: Planifica automáticamente
SAI: Implementa Agentes separados de las Interacciones (escenarios)

| PROYECTO | DESCRIPCIÓN | OM | EI | P | SAI |
|--|---|----|----|----|-----|
| Agent Builder Shell (ABS) <i>Enterprise Integration Laboratory, University of Toronto</i> | http://www.kecl.ntt.co.jp/csl/msrg/topics/at/ Es un proyecto en donde se desarrolla un Kernel que proporciona capas reutilizables de lenguajes y servicios para construir sistemas de agentes: lenguajes de comunicación y coordinación, manejo de conocimiento basado en lógica, distribución de información cooperativa, administración de conflictos y modelos de organizaciones. El enfoque está siendo usado para desarrollar aplicaciones multiagente en el área de manufactura. | no | no | no | no |
| Agent TCL <i>Dartmouth University</i> | http://agent.cs.dartmouth.edu/general/agenttcl.html Agent TCL es una herramienta para desarrollar sistemas de agentes móviles. Los agentes se crean utilizando el lenguaje de comandos útiles (Tool Command Language, TCL). TCL es un lenguaje basado en scripts disponible libremente. Los agentes migran de un lugar a otro suspendiendo su ejecución y continuandola sin cambios en otro equipo. | no | no | no | no |
| Agent Tool <i>Kansas State University</i> | http://www.cis.ksu.edu/~sdeloach/ai/ Permite especificar formalmente la estructura y comportamiento de un sistema multiagente. Usa un mecanismo gráfico y una metodología de ingeniería de sistemas multiagente. | no | no | no | no |
| Architecture type-based Development Environment (ADE) <i>University of Postdam</i> | http://samuel.cs.uni-potsdam.de/soft/taxt/research/ade/ade.html Es una plataforma que modela explícitamente las interacciones entre agentes, a diferencia de otras plataformas, permitiendo mapear los sistemas y modelos en plataformas de agentes. Mediante ADE se pueden crear aplicaciones para varias plataformas. | no | no | no | no |
| Bee-gent <i>Toshiba Corp., Systems and Software Research Laboratories</i> | http://www2.toshiba.co.jp/beegent/index.htm Agentifica completamente la comunicación que se da entre aplicaciones de software. La plataforma consiste de dos tipos de agentes: envolventes, que se usan para agentificar las aplicaciones existentes y los mediadores, que soportan la coordinación interna de las aplicaciones y manejan todas las comunicaciones. | no | no | no | no |
| Bond Distributed Object System <i>Purdue University</i> | http://bond.cs.purdue.edu/ Proporciona un ambiente orientado al intercambio de mensajes desarrollado para realizar aplicaciones distribuidas. | no | no | no | no |
| Cable <i>Logica Corporation</i> | http://public.logica.com/~grace/Architecture/Cable/public/ Es una arquitectura genérica para desarrollar y ejecutar aplicaciones distribuidas que están basadas en la metáfora de múltiples agentes inteligentes cooperativos. Proporciona un lenguaje de definición de agentes (Agent Definition Language, ADL) y un traductor llamado Ascribe. | no | no | no | no |
| D ² Agents <i>Dartmouth University</i> | http://agent.cs.dartmouth.edu/ El objetivo de este proyecto es proporcionar soporte a las aplicaciones que requieren de la recuperación, organización y presentación de información distribuida en redes arbitrarias. | no | no | no | no |
| DECAF Agent Framework <i>University of Delaware</i> | http://www.eecis.udel.edu/~decaf/ Proporciona una plataforma para desarrollar agentes, la arquitectura básica está desarrollada en Java por lo que permite aprovechar sus características | no | no | no | no |

PLATAFORMAS ACADÉMICAS Y DE INVESTIGACIÓN

OM: Soporta Ontologías Mixtas
EI: Soporta Eventos Inesperados

P: Planifica automáticamente
SAI: Implementa Agentes separados de las Interacciones (escenarios)

| PROYECTO | DESCRIPCIÓN | OM | EI | P | SAI |
|--|--|----|----|----|-----|
| EXCALIBUR <i>GMD First, Technical University of Berlin</i> | http://www.ai-center.com/projects/excalibur/publications.html Es una arquitectura genérica para agentes autónomos utilizada en ambientes de videojuegos. Los agentes son capaces de encontrar las acciones correctas en base a los objetivos que se tienen y adaptan su comportamiento para nuevos oponentes o ambientes. | no | no | SI | no |
| FIPA-OS <i>FIPA</i> | http://fipa-os.sourceforge.net/ Es una plataforma basada en componentes que permiten desarrollar agentes que cumplen con la definición de FIPA. | no | no | no | no |
| Gypsy <i>Technical University of Viena</i> | http://www.infosys.tuwien.ac.at/Gypsy/ Es un proyecto que utiliza Java para la implementación de un ambiente flexible para experimentar con agentes móviles. Se pretende que sea útil para aplicaciones en Internet como recuperación de información, comercio y computación móvil. | no | no | no | no |
| Hive <i>The Media Lab. MIT</i> | http://www.hivecell.net/ Es una plataforma de software para crear aplicaciones distribuidas. Los programadores pueden crear sistemas que se conectan y usan los datos de cualquier parte de Internet. Es un ambiente para que puedan vivir agentes distribuidos y que se comunican para lograr los objetivos. | no | no | no | no |
| Infospiders <i>University of California San Diego Computer Science Department</i> | http://dollar.biz.uiowa.edu/~fil/IS/ Es una plataforma basada en el modelo de vida artificial para la recuperación de información en grandes bases de datos dinámicas, distribuidas y heterogéneas. La energía para sobrevivir cada uno de los agentes la toma del ambiente y del usuario en los intercambios de información relevante. | no | no | no | no |
| JADE <i>CSELT S.p.A., University of Parma</i> | http://sharon.cselt.it/projects/jade/ Es una plataforma para desarrollar aplicaciones basadas en agentes que cumplen con los estándares de FIPA. JADE se basa en un agente genérico que el usuario puede extender para dotarlo de la funcionalidad esperada. | no | no | no | no |
| JAFMAS <i>University of Cincinnati</i> | http://www.ececs.uc.edu/~abaker/JAFMAS/ Es una plataforma para guiar el desarrollo de sistemas multiagente junto con un conjunto de clases para el desarrollo de agentes en Java. Soporta comunicación a uno o múltiples agentes usando KQML. | no | no | no | no |
| JATLite <i>Stanford University</i> | http://java.stanford.edu/java_agent/html/ Es un conjunto de paquetes de Java que permiten construir sistemas multiagente usando Java. | no | no | no | no |
| JATLiteBean <i>University of Otago</i> | http://kmi.open.ac.uk/people/emanuela/JATLiteBean/ Es una extensión a JavaBean y JATLite adicionando la funcionalidad de KQML. | no | no | no | no |
| JIAC <i>Technische Universität Berlin</i> | http://dai.cs.tu-berlin.de/english/forschung/projekte/JIAC/ Es una arquitectura de agentes abierta, escalable implementada en Java, ofrece agentes móviles basados en componentes, así como soporte para crear aplicaciones de comercio electrónico y servicios distribuidos de telecomunicaciones. | no | no | no | no |
| KLAIM <i>Universita'Di Firenze</i> | http://music.dsi.unifi.it/ Es una plataforma para soportar el paradigma de procesos junto con los datos pueden moverse de un equipo de computo a otro. | no | no | no | no |

PLATAFORMAS ACADÉMICAS Y DE INVESTIGACIÓN

OM: Soporta Ontologías Mixtas
EI: Soporta Eventos Inesperados

P: Planifica automáticamente
SAI: Implementa Agentes separados de las Interacciones (escenarios)

| PROYECTO | DESCRIPCIÓN | OM | EI | P | SAI |
|---|--|----|----|----|-----|
| Knowbot System <i>Software CNRI</i> | http://www.cnri.reston.va.us/home/koe/ Es una infraestructura de investigación para agentes móviles, cuyo uso se pretende sea en sistemas altamente distribuidos tales como Internet. Está escrito en Python el cual es un lenguaje de programación orientado a objetos interpretado. | no | no | no | no |
| Mobiware Middleware Toolkit <i>Columbia University</i> | http://comet.columbia.edu/mobiware/ Es una tecnología de objetos distribuidos construida mediante Java y CORBA. Se ejecuta en dispositivos móviles, puntos de acceso inalámbricos y ruteadores con capacidad móvil proporcionando un conjunto de algoritmos e interfaces programables abierto para redes móviles adaptativas. | no | no | no | no |
| MOLE <i>University of Stuttgart</i> | http://mole.informatik.uni-stuttgart.de/ Es un prototipo que adiciona mecanismos de migración y comunicación usando el lenguaje de programación Java. | no | no | no | no |
| Multi-Agent Modeling Language (MAML) <i>Central European University</i> | http://www.maml.hu/ Es un lenguaje de programación para construir modelos basados en agentes. Es parte de investigaciones sobre modelación asistida por computadora. La disciplina de modelación soportada es basada en agentes. | no | no | no | no |
| MultiAgent System Tool (MAST) <i>Technical University of Madrid</i> | http://www.gsi.dit.upm.es/~mast/ Es una plataforma distribuida de propósito general para la cooperación de agentes heterogéneos. Consiste de dos entidades básicas: los agentes y la red a través de la cual interactúan ellos. | no | no | no | no |
| NOMADS <i>Institute for Human and Machine Cognition University of West Florida</i> | http://nomads.coginst.uwf.edu/ Es un sistema de agentes móviles que soporta la movilidad fuertemente y la ejecución segura en Java. Se compone de dos partes: un ambiente de ejecución llamado Oasis y una nueva Máquina Virtual compatible con Java llamada Aroma. | no | no | no | no |
| Open Agent Architecture <i>SRI International</i> | http://www.ai.sri.com/~oaa/ Es una plataforma para integrar una comunidad de agentes de software heterogéneos en un ambiente distribuido. En este contexto un agente se define como un proceso que registra sus servicios en una forma aceptable, utiliza el Lenguaje de Comunicación Interagente (ICL) y comparte funcionalidades comunes a todos los agentes. | no | no | no | no |
| ProcessLink <i>Stanford University</i> | http://www-cdr.stanford.edu/ProcessLink/ Es una plataforma que consiste de agentes genéricos y un protocolo de mensajes para integrar proyectos distribuidos y multidisciplinarios. Permite envolver software legado con código que le habilita para proporcionarle funciones de coordinación. | no | no | no | no |
| RETSINA <i>Carnegie Mellon University</i> | http://www-2.cs.cmu.edu/~softagents/ Tiene cuatro módulos para comunicación, planeación, calendarización y monitoreo de la ejecución de tareas y requerimientos de otros agentes. | no | no | no | no |
| Sensible Agents <i>The University of Texas at Austin, Laboratory for Intelligent Processes and Systems</i> | http://www.lips.utexas.edu/agents_one_pager.htm Representan una ayuda nueva para planeación, monitoreo y decisión. Los agentes en este ambiente son capaces de evaluar la propagación de las decisiones locales. Priorizan planes dinámicamente. Proporcionan sugerencias para las interacciones más eficientes y productivas con otros agentes. Pueden replanificar cuando ocurren cambios en el ambiente. | no | no | SI | no |

PLATAFORMAS ACADÉMICAS Y DE INVESTIGACIÓN

OM: Soporta Ontologías Mixtas
EI: Soporta Eventos Inesperados

P: Planifica automáticamente
SAI: Implementa Agentes separados de las Interacciones (escenarios)

| PROYECTO | DESCRIPCIÓN | OM | EI | P | SAI |
|--|--|----|----|----|-----|
| Social Interaction Framework (SIF) <i>DFKI (German Research Institute for AI)</i> | http://www.lhbasif.com/agentworkshop/index.asp Es una herramienta para generar prototipos de simulaciones de en donde se involucran múltiples agentes. | no | no | no | no |
| Sodabot <i>MIT Artificial Intelligence Lab</i> | http://www.ai.mit.edu/people/sodabot/sodabot.html Es un ambiente de propósito general para desarrollo y ejecución de agentes de software. Este proyecto desarrolla un lenguaje de programación para programar a los agentes mediante primitivas que describen la actividad del agente. | no | no | no | no |
| SOMA (Secure and Open Mobile Agent) <i>University of Bologna</i> | http://lia.deis.unibo.it/Research/SOMA/ Es una plataforma basada en Java utilizada en Internet, diseñada para los objetivos de seguridad e interoperabilidad. Es compatible con CORBA y MASIF. | no | no | no | no |
| TeamBots <i>The Robotics Institute, Carnegie Mellon University</i> | http://www.teambots.org/ Es una colección de programas de aplicación basada en Java para investigar robots móviles multiagente. El ambiente de simulación está escrito en Java. | no | no | no | no |
| TuCSoN <i>Universita di Bologna</i> | http://www.lia.deis.unibo.it/Research/TuCSoN/ Es un modelo para la coordinación de agentes de Internet. | no | no | no | no |
| Zeus <i>British Telecommunications Labs</i> | http://193.113.209.147/projects/agents.htm Es un ambiente colaborativo y una biblioteca de componentes para la construcción de agentes escritos en Java. Consta de tres capas. La capa de definición representa las habilidades del agente, como razonamiento, aprendizaje, sus metas, recursos, habilidades, creencias, preferencias, entre otros. La capa de organización describe las relaciones de un agente con otros. La capa de coordinación describe la coordinación y técnicas de negociación del agente, en este se encuentra el módulo de comunicación. | no | SI | SI | no |

ANEXO B GRAMÁTICA DE LIA

En este anexo se indica el archivo con la gramática de LIA como se utiliza en el traductor. Las palabras en mayúsculas son símbolos no terminales, las escritas con minúsculas son los símbolos terminales, los números indican rutinas semánticas y los símbolos que inician con el símbolo # son los indicadores de error en la sintaxis.

```

; Instituto Politécnico Nacional
; Centro de Investigación en Computación
;
; GRAMATICA DEL LENGUAJE DE INTERACCIONES
ENTRE AGENTES
;
; 6 diciembre 2000
;
; Es una Gramática Top-Down cuyos componentes
son:
; . simbolos no terminales (MAYÚSCULAS)
; . terminales (minúsculas)
; . número (rutinas semánticas)
; . #número (errores)
;=====
; SECCION INICIAL
;=====
INICIO -> GLOBAL REGIONAL AGENTES MAIN
INTERACCIONES
;
;=====
; VARIABLES GLOBALES
;=====
GLOBAL -> global 1 s{ LISTA_VARSG X1
X1 -> s}
X1 -> #1
GLOBAL -> global s{ #2
GLOBAL -> global #3
GLOBAL -> #4
;
LISTA_VARSG -> VARSG LISTA_VARSG
LISTA_VARSG -> VARSG
;
VARSG -> TIPO LISTAVG
TIPO -> int 2
TIPO -> char 2
TIPO -> float 2
TIPO -> port 2
;
LISTAVG -> VARG s, LISTAVG
LISTAVG -> VARG s;
LISTAVG -> #5
;
VARG -> id s[ numero s] s[ numero s] 3
VARG -> id s[ numero s] 4
VARG -> id 5
VARG -> #6
;
;=====
; VARIABLES REGIONALES
;=====
REGIONAL -> regional 1 s{ LISTA_VARSR X2
X2 -> s}
X2 -> #8
REGIONAL -> regional 1 s{ #9
REGIONAL -> regional #10
REGIONAL -> #11

;
LISTA_VARSR -> VARSR LISTA_VARSR
LISTA_VARSR -> VARSR
;
VARSR -> TIPO LISTAVR
;
LISTAVR -> VARR s, LISTAVR
LISTAVR -> VARR s;
LISTAVR -> #13
;
VARR -> id s[ numero s] s[ numero s] 3
VARR -> id s[ numero s] 4
VARR -> id 5
VARR -> #12
;
;=====
; A G E N T E S
;=====
AGENTES -> AGENTE AGENTES
AGENTES -> AGENTE
;
;-----
; Agente
;-----
AGENTE -> agent 1 id 6 s{ LREGIONAL INTERNAS
PROPOSITOS PERCIBE ACCIONES INICIALES s}
AGENTE -> agent id s{ LREGIONAL INTERNAS
PROPOSITOS #13
AGENTE -> agent id s{ LREGIONAL #14
AGENTE -> agent id s{ #15
AGENTE -> agent id #16
AGENTE -> agent #17
;
;-----
; Variables regionales del agente
;-----
LREGIONAL -> regional 1 s{ LISTAREG s}
LISTAREG -> id 6 s, LISTAREG
LISTAREG -> id 6 s;
;
;-----
; Variables internas
;-----
INTERNAS -> internal 1 s{ LISTA_VARSI X4
X4 -> s}
X4 -> #19
INTERNAS -> internal s{ #20
INTERNAS -> internal #21
INTERNAS -> #22
;
LISTA_VARSI -> VARSI LISTA_VARSI
LISTA_VARSI -> VARSI
;
VARSI -> TIPOI LISTAVI
TIPOI -> int 2
TIPOI -> char 2
TIPOI -> float 2
;
LISTAVI -> VARI s, LISTAVI
LISTAVI -> VARI s;

```

```

LISTAVI -> #23
;
VARI -> id s[ numero s] s[ numero s] 3
VARI -> id s[ numero s] 4
VARI -> id 5
VARI -> #24
;
;-----
; Propósitos
;-----
PROPOSITOS -> purpose 1 s{ LISTAPROP s}
LISTAPROP -> id 6 s, LISTAPROP
LISTAPROP -> id 6 s;
;
;-----
; Eventos Inesperados percibidos
;-----
PERCIBE -> percieve 1 s{ LISTAPERCI s}
LISTAPERCI -> id 6 s, LISTAPERCI
LISTAPERCI -> id 6 s;
;
;-----
; Acciones para Eventos Inesperados
;-----
ACCIONES -> action 1 s{ LISTAACCIÓN s}
LISTAACCIÓN -> id 6 s, LISTAACCIÓN
LISTAACCIÓN -> id 6 s;
;
;-----
; Papeles iniciales
;-----
INICIALES -> initial 1 s{ LISTAINI s}
LISTAINI -> id 6 s, LISTAINI
LISTAINI -> id 6 s;
;
;=====
; INTERACCIONES
;=====
INTERACCIONES -> INTERACCION INTERACCIONES
INTERACCIONES -> INTERACCION
;
;-----
; Interacción
;-----
INTERACCION -> interaction 1 id 6 s{ PAPELES
s}
;
PAPELES -> PAPEL PAPELES
PAPELES -> PAPEL
;
;-----
; Papel
;-----
PAPEL -> role 1 id 6 s( PARAMS s[ instances
numero 10 s] s{ REQ POSITIVO NEGATIVO LOCALES
8 L_INSTR s} 9
;
;-----
; Parametros
;-----
PARAMS -> id 60 s, PARAMS
PARAMS -> id 60 s)
;
;-----
; Requisitos
;-----
REQ -> requisite 1 s{ LISTA_REQ s}
LISTA_REQ -> id 6 s, LISTA_REQ
LISTA_REQ -> id 6 s;

```

```

;
;-----
; Efectos Positivos
;-----
POSITIVO -> positive 1 s{ LISTA_POS s}
LISTA_POS -> id 6 s, LISTA_POS
LISTA_POS -> id 6 s;
;
;-----
; Efectos Negativos
;-----
NEGATIVO -> negative 1 s{ LISTA_NEG s}
LISTA_NEG -> id 6 s, LISTA_NEG
LISTA_NEG -> id 6 s;
;
;-----
; Variables Locales
;-----
LOCALES -> local 1 s{ LISTALOC X5
X5 -> s}
X5 -> #30
LOCALES -> local s{ #31
LOCALES -> local #32
LOCALES -> #33
;
LISTALOC -> VARSLOC LISTALOC
LISTALOC -> VARSLOC
;
VARSLOC -> TIPOI LISTALO
;
LISTALO -> VARLO s, LISTALO
LISTALO -> VARLO s;
;
VARLO -> id s[ numero s] s[ numero s] 3
VARLO -> id s[ numero s] 4
VARLO -> id 5
VARLO -> #35
;
L_INSTR -> INSTR L_INSTR
L_INSTR -> INSTR
;
;-----
; INSTRUCCIONES DE LIA
;-----
INSTR -> print s( cadena 11 s) s;
INSTR -> print s( id 49 s) s;
INSTR -> if s( EXPR_L s) s{ L_INSTR s} else s{
L_INSTR s}
INSTR -> if s( EXPR_L s) 12 s{ L_INSTR s} 13
INSTR -> while 14 s( 15 16 EXPR_L s) 17 s{
L_INSTR s} 18
;INSTR -> optional s[ EXPR_ARITM s] s{ L_INSTR
s}
INSTR -> addagent s( id s, id s, 19 50
LISTA_PROP_AG LISTA_REC_AG s; 59
INSTR -> addinteract s( id s, id s) s; 20
INSTR -> addpurpose s( id s, id s) s; 21
INSTR -> delpurpose s( id s, id s) s; 22
INSTR -> out s( id s, id s, id 52 s, id s, id
53 s) s;
INSTR -> accept s( id s, id s, id 54 s, id 55
s) s;
INSTR -> append s( id s, cadena 46 s) s;
INSTR -> append s( id s, id 47 s) s;
INSTR -> extract s( id s, id s, id 56 s) s;
INSTR -> compare s( id s, id s, id 57 s) s;
INSTR -> delmessage s( id s, id 58 s) s;
INSTR -> wait s( numero s) s; 45
INSTR -> goal s( id 61 s) s;
;INSTR -> addevent agente evento
;INSTR -> delevent agente evento

```

```

;INSTR -> addaction agente action
;INSTR -> delaction agente action
;INSTR -> start papel tiempo prioridad
;INSTR -> kill papel tiempo prioridad
INSTR -> VAR_A_NUM s= 15 EXPR_ARITM s; 23
INSTR -> id s:= cadena s; 48
INSTR -> id s[ numero s] s:= cadena s;
INSTR -> id s:= id s[ id s] s;
;
;-----
; Propósitos Nuevos de un Agente
;-----
LISTA_PROP_AG -> id 6 s; LISTA_PROP_AG
LISTA_PROP_AG -> id 6 s,
;
;-----
; Recursos Iniciales de un Agente
;-----
LISTA_REC_AG -> ACCION_AG s; LISTA_REC_AG
LISTA_REC_AG -> ACCION_AG s)
;
ACCION_AG -> id s:= cadena 51
ACCION_AG -> VAR_A_NUM s= 15 EXPR_ARITM 23a
;
;-----
; Expresion Lógica
;-----
EXPR_L -> TERMINO_L s|| EXPR_L 32
EXPR_L -> TERMINO_L
TERMINO_L -> FACTOR_L s&& TERMINO_L 33
TERMINO_L -> FACTOR_L
FACTOR_L -> EXPR_ARITM OP_REL EXPR_ARITM 34
FACTOR_L -> s! FACTOR_L
FACTOR_L -> s( EXPR_L s)
;
OP_REL -> s> 35
OP_REL -> s< 36
OP_REL -> s== 37
OP_REL -> s>= 38
OP_REL -> s<= 39
OP_REL -> s!= 40
;
;-----
; Expresión Aritmética
;-----
EXPR_ARITM -> TERMINO s+ EXPR_ARITM 25

```

```

EXPR_ARITM -> TERMINO s- EXPR_ARITM 26
EXPR_ARITM -> TERMINO
TERMINO -> FACTOR s* TERMINO 27
TERMINO -> FACTOR s/ TERMINO 28
TERMINO -> FACTOR
FACTOR -> s( EXPR_ARITM s)
FACTOR -> numero 24
FACTOR -> VAR_NUM
;
VAR_NUM -> id s[ id s] s[ id s] 31
VAR_NUM -> id s[ id s] 30
VAR_NUM -> id 29
;
;-----
; Otra cosa
;-----
VAR_A_NUM -> id s[ id s] s[ id s] 41
VAR_A_NUM -> id s[ id s] 42
VAR_A_NUM -> id s[ numero s] 43
VAR_A_NUM -> id 44
;
VAR_PUERTO -> id s[ numero s] s[ numero s]
VAR_PUERTO -> id s[ id s] s[ id s]
VAR_PUERTO -> id s[ numero s]
VAR_PUERTO -> id s[ id s]
VAR_PUERTO -> id
;
EXPR_CADENA -> cadena
;
VALUE -> id
VALUE -> numero
;
TIEMPO -> id 65--
;
EMISOR -> self 60--
EMISOR -> id 59--
;
PUERTO -> VAR_PUERTO
;=====
; PAPEL PRINCIPAL
;=====
MAIN -> main s( s) 7 8 s{ L_INSTR s} 9
; fin gramatica

```

