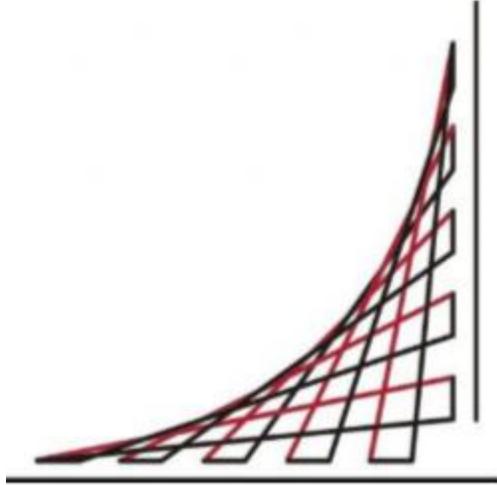


AMBIENTE JOYCE



PRESENTADO POR:

MARIA TATIANA TORRES TORRES

DIRECTOR DE PROYECTO DE GRADO:

GERARDO OSPINA HERNÁNDEZ

ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO

PROYECTO DE GRADO 2

BOGOTÁ D.C.

2016-i

Contenido

1	Contexto (Proyecto)	3
1.1	Planteamiento del problema.....	3
1.2	Marco teórico y estado del arte.....	3
1.3	Objetivos del proyecto: General y específicos	4
1.4	Justificación.....	4
1.5	Área de aplicación del producto resultado del proyecto.	5
2	Requerimientos	5
2.1	Descripción del sistema.....	5
2.2	Visión y alcance	5
2.3	Usuarios	6
3	Análisis Joyce	6
3.1	Descripción del subsistema	6
4	Análisis JADE	7
4.1	Descripción del subsistema	7
5	Diseño	7
5.1	Diagrama de clases	7
5.2	Vistas arquitectónicas	10
5.2.1	Vista lógica.....	10
5.2.2	Vista de desarrollo	13
6	Implementación	13
6.1	Especificación de estándares utilizados.....	13
7	Liberación	14
7.1	Configuración ambiente mínima/ideal.....	14
7.2	Manual de instalación.....	14
7.3	Manual de usuario	14

1 Contexto (Proyecto)

1.1 Planteamiento del problema.

Los sistemas concurrentes/paralelos se han convertido en un tema de actualidad dada la aparición masiva de sistemas multiprocesadores, reforzado por el desarrollo de hardware de alta velocidad que permite la conexión en red de dichos sistemas multiprocesadores. Una manera de aprovechar esta capacidad de computo, es la de escribir programas que aprovechen estos sistemas concurrentes/paralelos, lo cual, a su vez requiere, un lenguaje de programación que tenga primitivas para expresar concurrencia/paralelismo.

1.2 Marco teórico y estado del arte.

Existen lenguajes concurrentes los cuales permiten la ejecución de varios procesos simultáneamente. Algunas de las características de los lenguajes concurrentes son: el indeterminismo y la interacción entre procesos. El indeterminismo se refiere a los resultados diferentes que puede tener un mismo programa con los mismos datos de entrada puesto que el orden en el que se ejecutan las instrucciones en un programa concurrente es aleatorio. La interacción entre procesos se da en los lenguajes concurrentes dado que es indispensable pues los procesos que componen a un programa se tienen que comunicar. El lenguaje concurrente que se escogió para el desarrollo de este proyecto es Joyce, que trabaja con agentes los cuales se comunican a través de canales de transmisión.

También existen lenguajes que trabajan bajo un sistema multiagente que permite modelar aplicaciones con un conjunto de componentes llamados agentes que cuentan con características como lo son: autonomía, proactividad y capacidad de comunicación entre ellos. Estos sistemas se basan en una arquitectura AOP que permite el intercambio de mensajes punto a punto siendo todos los componentes agentes, es decir, no necesita

de un cliente y un servidor para llevar a cabo la comunicación. Entre algunos de estos sistemas se encuentran: JavaLog, JADE, IDK, ABLE, entre otros. Para este proyecto se eligió trabajar con JADE que es una plataforma implementada en su totalidad en Java y soporta servicios básicos definidos para la comunicación entre agentes tales como: movilidad, gestión y localización de agentes. Adicionalmente a estos servicios JADE les añade utilidades gráficas para facilitar la administración de las plataformas y la comunicación entre agentes en tiempo de ejecución.

Por otro lado encontramos también los generadores de compiladores que aportan para la realización de estos. Actualmente existen diferentes herramientas que cumplen con el objetivo de generar compiladores entre las que se encuentran: Bison, Coco/R, Flex, Lex, SDGLL1, TS, y YACC, para el desarrollo del compilador de Joyce se eligió la herramienta Coco/R la cual es una herramienta que mediante la descripción del lenguaje con una gramática LL1 genera analizadores léxicos y sintácticos para dicho lenguaje definido.

1.3 Objetivos del proyecto: General y específicos

Objetivo general:

Crear un sistema común que permita manejar la concurrencia/paralelismo apoyados en el lenguaje concurrente Joyce y el sistema multiagente JADE.

Objetivos específicos:

- Investigar y familiarizarse con Joyce y JADE.
- Construir un compilador que permita reconocer el lenguaje Joyce con todas sus características.
- Realizar propuestas de integración de los modelos de agente de Joyce y JADE
- Elaborar el modelo de integración entre los agentes de Joyce y los agentes de Jade e integrarlo al compilador elaborado para Joyce

1.4 Justificación

Es una necesidad porque este tipo de tecnologías no existen y pueden ser muy satisfactorias a la hora de buscar una solución para un problema con sistemas multiprocesadores concurrentes/paralelos, igualmente como ya se

ha mencionado, hoy en día está en auge la adopción de sistemas multiprocesadores que permitan manejar la concurrencia/paralelismo.

1.5 Área de aplicación del producto resultado del proyecto.

Grupo de investigación CTG-Informática

2 Requerimientos

2.1 Descripción del sistema

Una de las temáticas que se estudia en ingeniería de sistemas es la concurrencia, pero actualmente no existe un sistema que permita a los estudiantes aprender o profundizar sobre este tema; el presente trabajo se desarrolla para hacer frente a este problema y tiene el objetivo de mostrar el impacto del aprovechamiento de la aparición masiva de sistemas multiprocesadores y cómo esto se refuerza a través del desarrollo de hardware de alta velocidad, lo que en conjunto permite la conexión en red de dichos sistemas multiprocesadores y a su vez apoya en gran medida a la solución.

2.2 Visión y alcance

Es necesario conocer y entender para qué es y cómo funciona el lenguaje concurrente Joyce, después de conocer el lenguaje es necesario escribir un compilador para este. Un compilador es un programa el cual va a reconocer toda la gramática utilizada en Joyce y por supuesto verificar que este compilador funcione con alguna herramienta que tenga esta funcionalidad, la que se va a trabajar en este proyecto es Coco/r que es un generador de compiladores en el cual se escribe la gramática del lenguaje requerido para poder generar un escáner y un programa de análisis para este lenguaje.

Lo siguiente es entender JADE, qué es y cómo implementa los agentes para que luego de tener claro que es Joyce y JADE poder entrar a comparar los modelos de agentes y buscar la forma de integrarlos es decir que un agente

Joyce pueda traducirse a un agente JADE y poder unir esto en el compilador ya creado para Joyce.

2.3 Usuarios

- Estudiantes de la Escuela Colombiana de Ingeniería que necesiten trabajar con este lenguaje concurrente y su compilador
- Estudiantes de otras universidades que deseen trabajar con el compilador creado

3 Análisis Joyce

3.1 Descripción del subsistema

Joyce es un pequeño programa seguro de programación paralela, que fue principalmente diseñado para enseñar los principios y prácticas del procesamiento distribuido, el cual permite la activación de agentes que cuentan con sistemas de comunicación llamados canales, los cuales permiten transmitir mensajes entre dos o más agentes simultáneamente. Un canal puede ser utilizado por dos o más agentes si los agentes están listos para comunicarse por el mismo canal, es decir, para transmitir mensajes entre ellos. Es importante mencionar que el lenguaje Joyce está definido con notación BNF lo cual da facilidad en el momento realizar el analizador léxico y sintáctico de la gramática del lenguaje.

Finalmente, es importante resaltar que para el cumplimiento de los objetivos ya descritos se quiere aprovechar el uso de los sistemas multiprocesadores en el lenguaje de programación concurrente pero como estos lenguajes no hacen uso de los sistemas multiprocesadores, se vio la necesidad de utilizar una herramienta de desarrollo de sistemas multiagentes, la cual si los aproveche. A continuación, vamos a detallar estos sistemas y la plataforma que se escogió para trabajar.

Entregables:

- Analizador léxico y sintáctico

4 Análisis JADE**4.1 Descripción del subsistema**

Es una plataforma implementada en su totalidad en Java y soporta servicios básicos definidos para la comunicación entre agentes tales como: movilidad, gestión y localización de agentes. Adicionalmente a estos servicios JADE les añade utilidades gráficas para facilitar la administración de las plataformas y la comunicación entre agentes en tiempo de ejecución. Se reconoce que JADE es un sistema dinámico, es decir, que los agentes pueden crearse, migrar o terminar.

Los agentes en JADE pueden tener varios tipos de comportamiento entre los cuales se encuentran: autónomo, social, reactivo y proactivo. Todo agente debe pertenecer a uno de los tipos mencionados anteriormente, sin embargo, un agente puede tener otras características si así lo necesita, pero estas no son obligatorias. Estas características son: Móvil, preciso, benevolente, racional o aprendiz. Toda la información de cada uno de los agentes, desde su creación hasta un poco después de su terminación, se encuentra en el cache del contenedor principal de agentes, el mantenimiento de esta información es de vital importancia en JADE en caso de que la plataforma llegase a fallar.

Entregables:

- Ejemplos generales de uso de la herramienta
- Ejemplos de integración

5 Diseño**5.1 Diagrama de clases****Convenciones:**

- Clases que dependen de la clase Agent
- Cases que dependen de la clase CyclicBehaviour
- Clase inicializadora

Diagrama de clases – ejemplo nim

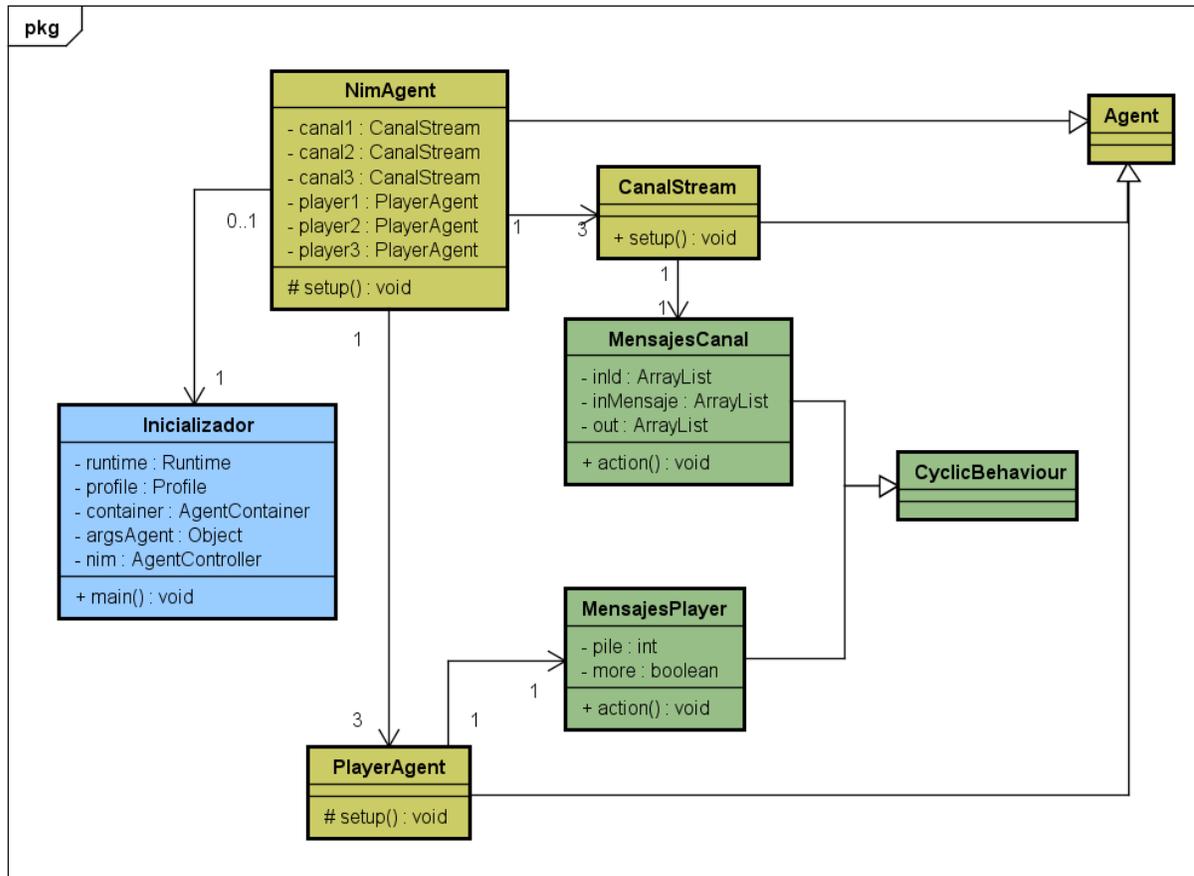
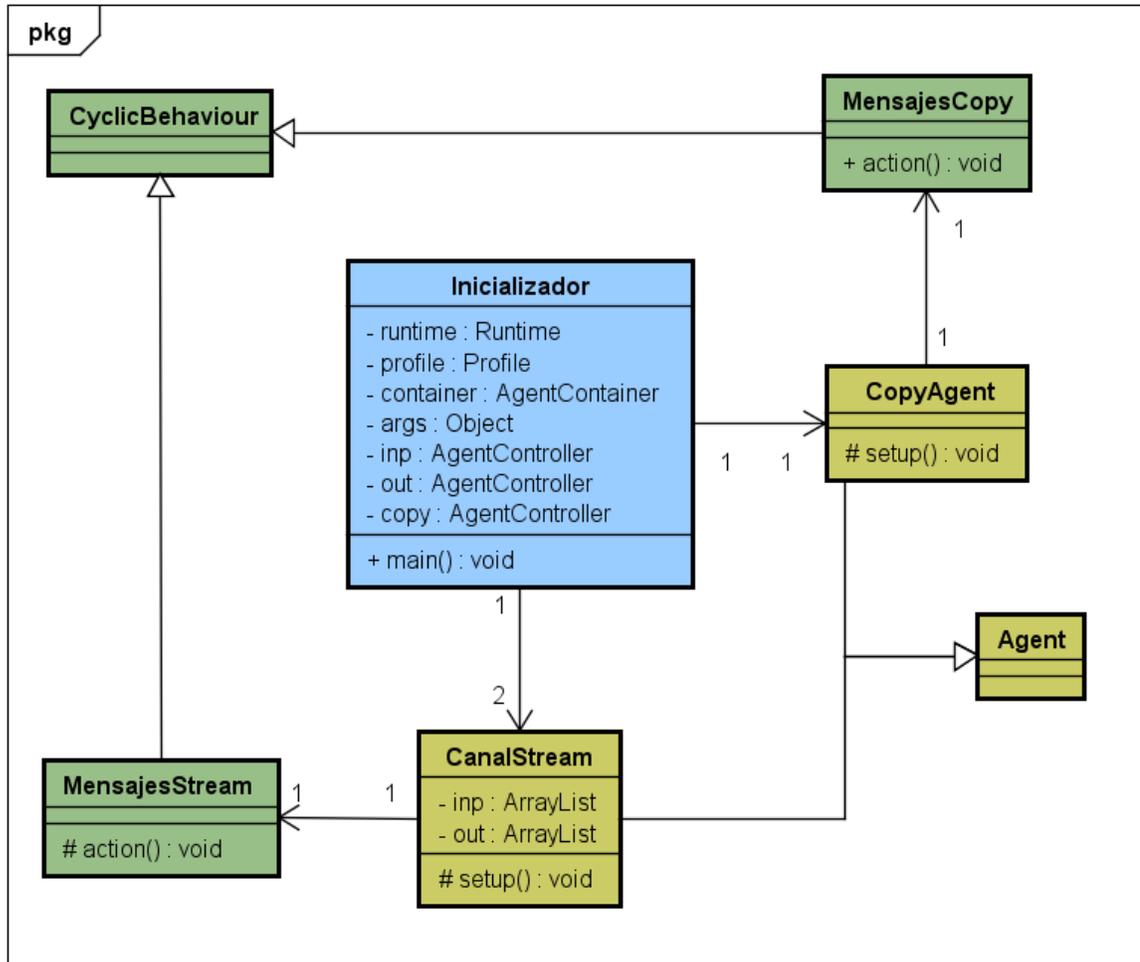


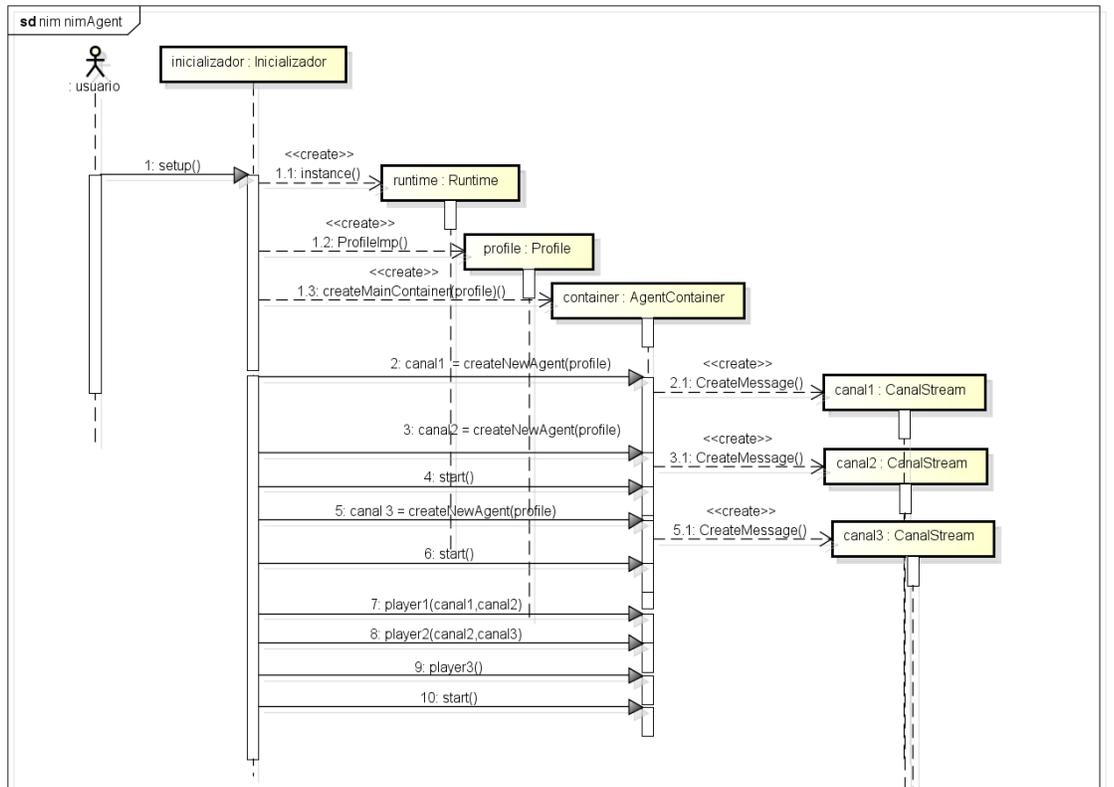
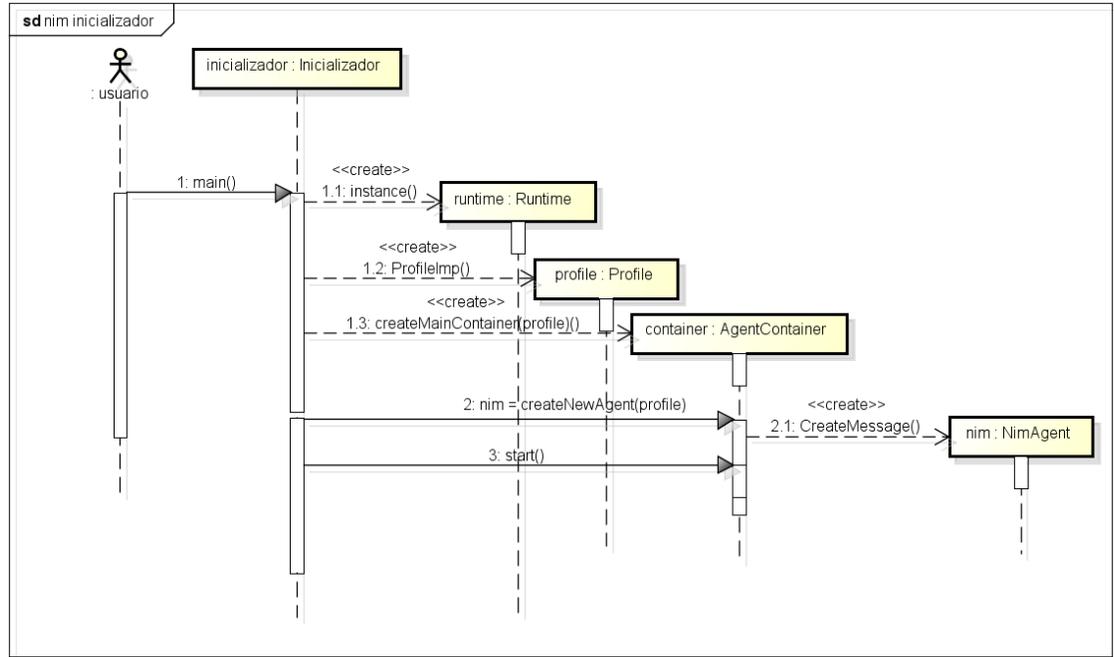
Diagrama de clases – ejemplo copy

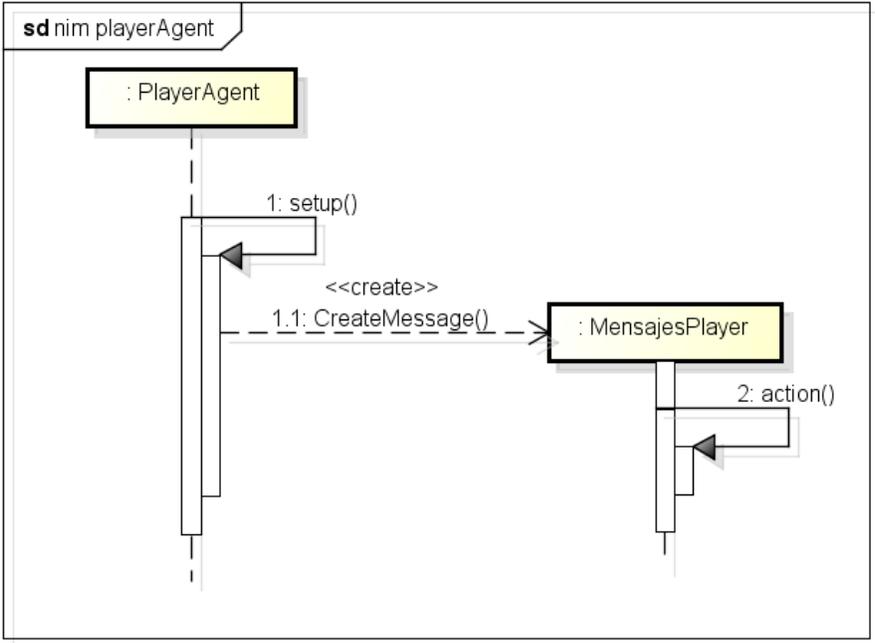
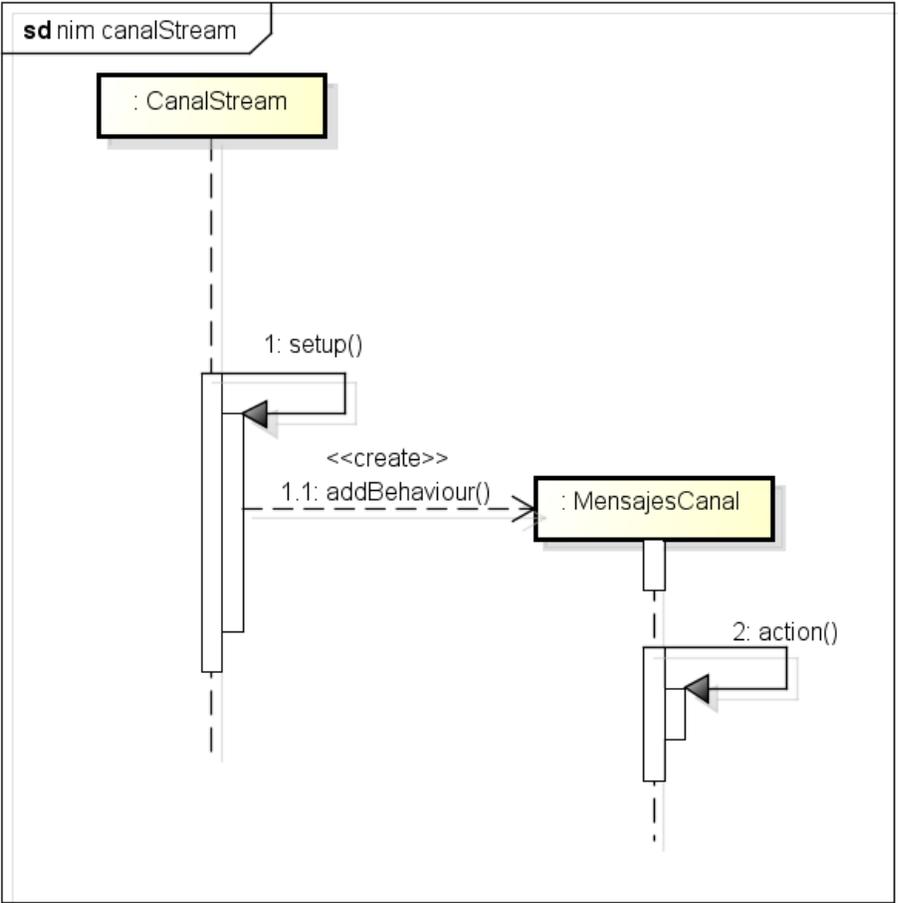


5.2 Vistas arquitectónicas

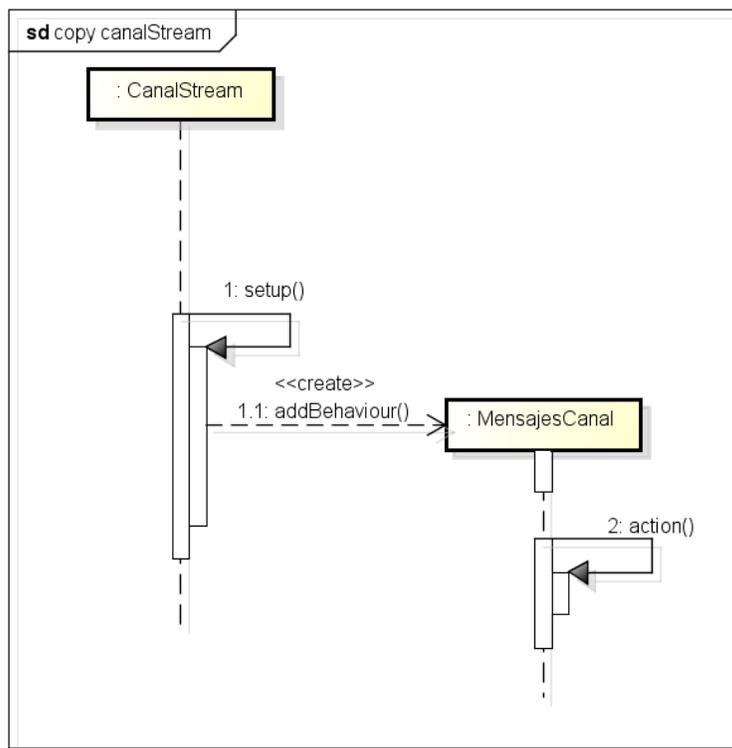
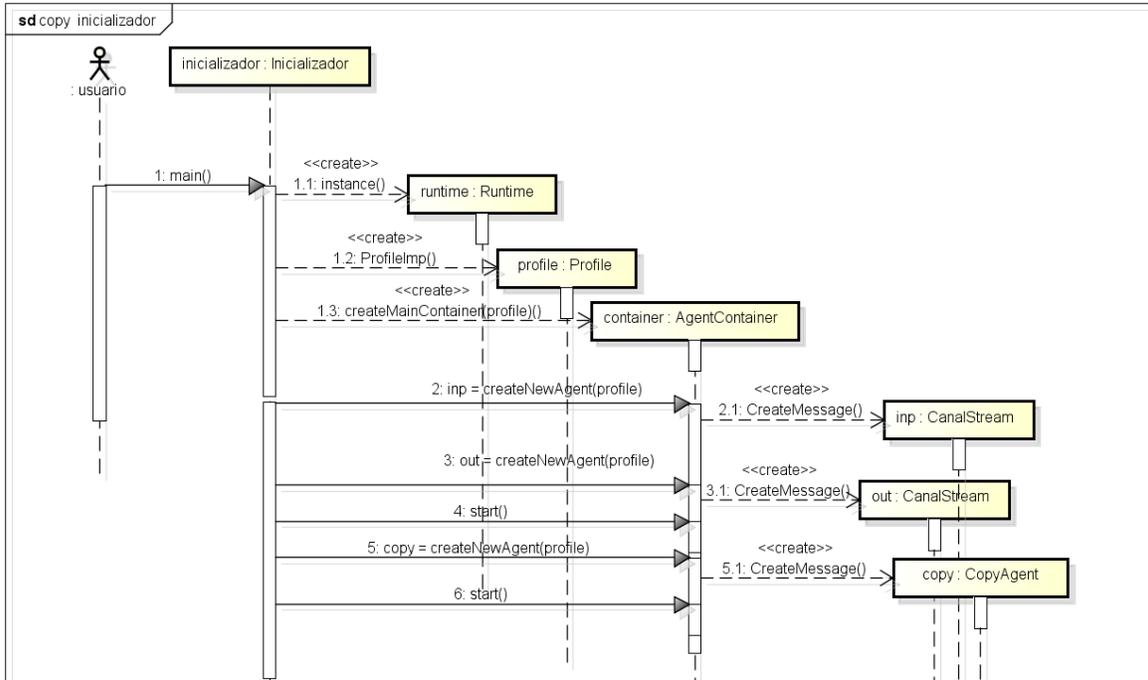
5.2.1 Vista lógica

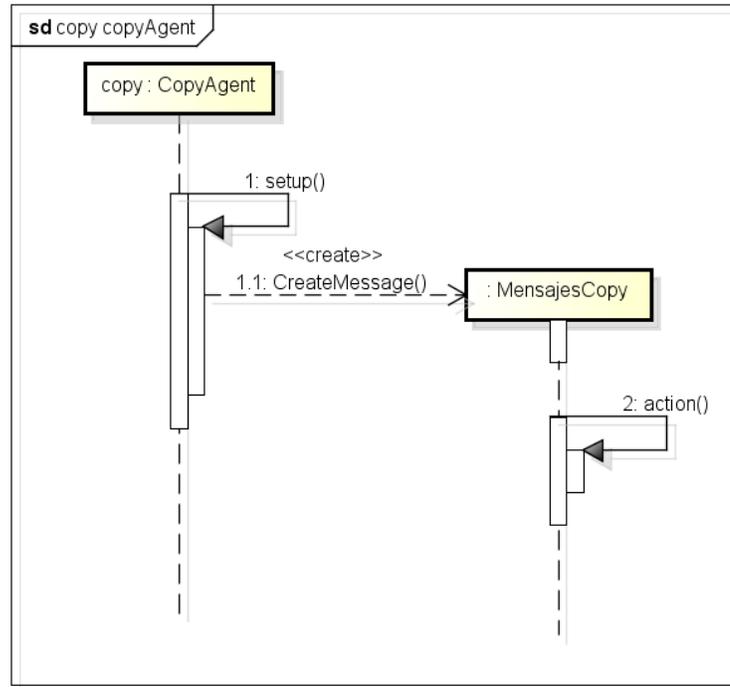
Los siguientes diagramas de secuencia corresponden al ejemplo NIM



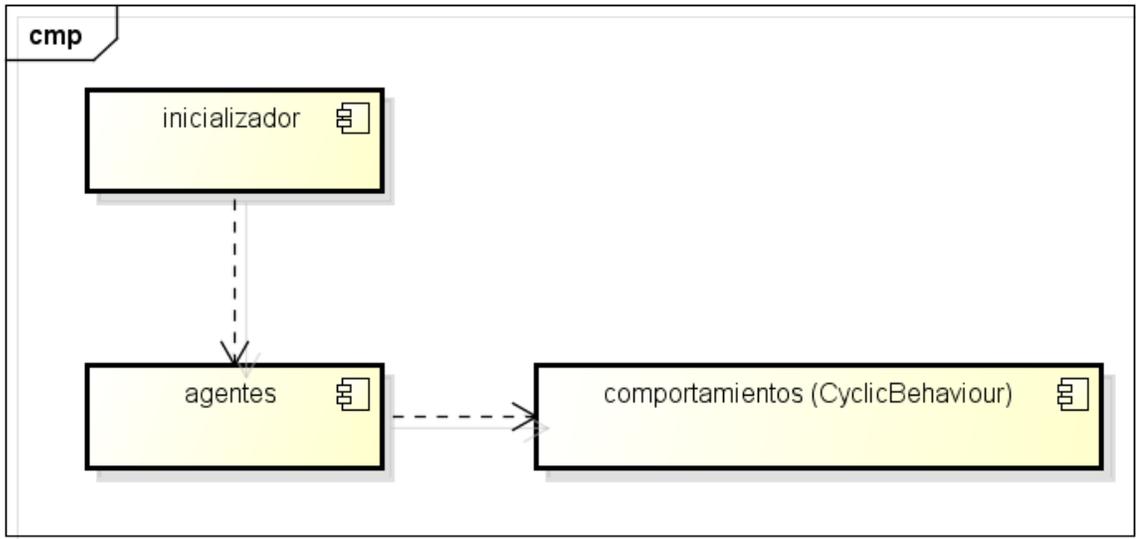


Los siguientes diagramas de secuencia corresponden al ejemplo COPY





5.2.2 Vista de desarrollo



6 Implementación

6.1 Especificación de estándares utilizados

JADE sigue los estándares de la FIPA (Foundation for Intelligent physical Agents), más específicamente sigue las especificaciones sobre comunicación, manejo y arquitectura de la FIPA2000. En general la FIPA

cuenta con una arquitectura básica, servicios entre los cuales se encuentra implementado el servicio de páginas blancas y las páginas amarillas, las páginas amarillas zona servicios a través de los cuales los agentes pueden publicar los servicios que proporcionan para que de esta manera otros agentes puedan acceder a estos el transporte de los mensajes y las páginas blancas se encarga de proveer servicios de ciclo de vida y de mantener el directorio de los identificadores de los agentes (AID) y finalmente ofrece un conjunto de herramientas gráficas que soportan la ejecución de agentes

7 Liberación

7.1 Configuración ambiente mínima/ideal

Los requerimiento mínimos que se deben tener en un equipo en que se desea ejecutar, ya sean los ejemplos o el analizador léxico y sintáctico es JAVA, por otro lado para poder ejecutar los ejemplo de JADE es necesario contar con netbeans.

7.2 Manual de instalación

En la carpeta docs encontrará el manual de instalación

7.3 Manual de usuario

En la carpeta docs encontrará el manual de usuario de los ejemplos de JADE y el del analizador léxico y sintáctico