

## Agente Controlador



Todos los Agentes que se han implementado se han lanzado desde el momento de abrir la plataforma o desde el menú propio de la plataforma.

Este taller tiene por objeto visualizar la forma en que un agente puede lanzar otro en cualquier momento que el defina.

Para lo cual se debe hacer lo siguiente:

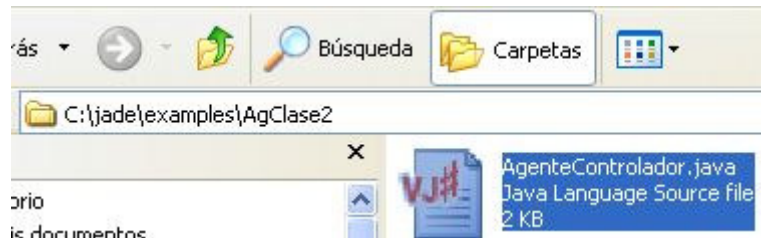
1. Se debe capturar el contenedor (Container de JADE) donde estará el agente.

```
PlatformController container = getContainerController();
```

2. Se crean los agentes con la instrucción CreateAgent. En esta instrucción se debe tener en cuenta la clase que almacena al nuevo agente, el nombre y los parámetros.

El código quedaría de la siguiente forma

1. Cree un archivo con nombre AgenteControlador.java (En el directorio jade/examples/AgClase2



2. Este Agente tiene la siguiente estructura:

```
package examples.AgClase2;

import jade.core.Agent;
import jade.core.AID;
import jade.domain.FIPAAgentManagement.*;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

import jade.core.Runtime;
import jade.core.Profile;
import jade.core.ProfileImpl;
import jade.wrapper.*;
|

public class AgenteControlador extends Agent {
```

3. Este Agente inicialmente no tiene definido ningún comportamiento (Behaviour), lo único que debe hacer es crear los otros Agentes (En este caso se crearán agentes Básicos , **AgBasico.java**, definidos en otro taller)

```
public class AgenteControlador extends Agent {

    protected void setup() {
        System.out.println(getLocalName()+" Empieza a Crear");

        try {
            // create the agent description of itself
            DFAgentDescription dfd = new DFAgentDescription();
            dfd.setName(getAID());
            // register the description with the DF
            DFService.register(this, dfd);
            System.out.println(getLocalName()+" Agente Registrado en el DF");
        } catch (FIPAException e) {
            e.printStackTrace();
        }

        // create another two ThanksAgent
        String t1AgentName = getLocalName()+"t1";
        String t2AgentName = getLocalName()+"t2";
```

4. A continuación se presenta el código que permite crear los agentes en el contenedor.

```
try {
PlatformController container = getContainerController(); // get a container controller
container.createNewAgent(t1AgentName, "examples.AgClase2.AgBasico", null).start();
System.out.println(getLocalName()+" Creacion de un Agente Basico :"+t1AgentName);
container.createNewAgent(t2AgentName, "examples.AgClase2.AgBasico", null).start();
System.out.println(getLocalName()+"Creacion de un Agente Basico :"+t2AgentName);
} catch (Exception any) {
any.printStackTrace();
}
```

5. El Agente espera 10 Segundos y a continuación envía un mensaje a los dos agentes creados.

```
//end comment was first here
doWait(10000);

// send them a GREETINGS message
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.setContent("SALUDOS");

msg.addReceiver(new AID(t1AgentName, AID.ISLOCALNAME));
msg.addReceiver(new AID(t2AgentName, AID.ISLOCALNAME));

send(msg);
System.out.println(getLocalName()+" ENVIANDO MENSAJE SALUDO: "+msg);
}
}
```

6. Compile los agentes y ejecutelos.

```
C:\jade>compileJade examples/AgClase2/AgenteControlador.java
```

7. Ejecute el Agente

Ejecute el agente de la siguiente forma:

```
C:\jade>runjade -gui ag2:examples.AgClase2.AgenteControlador
```

```
ag2 Empieza a Crear
ag2 Agente Registrado en el DF
ag2 Creacion de un Agente Basico :ag2t1
ag2 Creacion de un Agente Basico :ag2t2
Primer Agente con JADE
Primer Agente con JADE
ag2 ENVIANDO MENSAJE SALUDO: <INFORM
  :sender < agent-identifier :name ag2@castillo-vaio:1099/JADE :addresses <sequ
ence http://castillo-vaio:7778/acc > :X-JADE-agent-classname examples.AgClase2.A
genteControlador >
  :receiver <set < agent-identifier :name ag2t1@castillo-vaio:1099/JADE > < agen
t-identifier :name ag2t2@castillo-vaio:1099/JADE > >
  :content "SALUDOS"
>
```