





# Creación de sistemas multiagentes inteligentes con JADE



-  Sistemas Inteligentes
-  Grado en Ingeniería Informática
-  Diego M. Jiménez Bravo
-  [djimenez@fi.upm.es](mailto:djimenez@fi.upm.es)
-  13 de mayo de 2022

# Índice

1. ¿Qué es JADE?
2. ¿Cómo crear un agente con JADE?
3. ¿Cómo crear un comportamiento con JADE?
4. ¿Cómo se comunican los agentes en JADE?
5. Ejemplo simple de interacción entre agentes
6. Ejemplo de MAS inteligente con Weka
7. Recursos
8. Licencia

# Índice

1. **¿Qué es JADE?**
2. ¿Cómo crear un agente con JADE?
3. ¿Cómo crear un comportamiento con JADE?
4. ¿Cómo se comunican los agentes en JADE?
5. Ejemplo simple de interacción entre agentes
6. Ejemplo de MAS inteligente con Weka
7. Recursos
8. Licencia

# ¿Qué es JADE?



**JADE** (Java Agent DEvelopment Framework) es un marco de software totalmente implementado en el lenguaje de programación **Java**. Simplifica la implementación de sistemas multiagente a través de un *middleware* que cumple con las especificaciones **FIPA** y a través de un conjunto de herramientas gráficas que dan soporte a las fases de depuración y despliegue.

Fue creado por **Telecom Italia Lab**.



# Índice

1. ¿Qué es JADE?
2. **¿Cómo crear un agente con JADE?**
3. ¿Cómo crear un comportamiento con JADE?
4. ¿Cómo se comunican los agentes en JADE?
5. Ejemplo simple de interacción entre agentes
6. Ejemplo de MAS inteligente con Weka
7. Recursos
8. Licencia

# ¿Cómo crear un agente con JADE?

De forma general, siempre se va a crear un agente especializado en alguna tarea; por tanto, para crear con la librería JADE un agente de este tipo es necesario hacer lo siguiente:

1. Crear una clase Java que contendrá los atributos y métodos del agente especializado.
  - a. Esta clase será una **extensión de la clase “Agent”** de JADE (`jade.core.Agent`)
  - b. El **constructor** de esta clase además de incluir las inicializaciones de los atributos de la clase, debe incluir la llamada al **función `super()`** que inicializara los atributos de la clase madre “Agent” de JADE.
  - c. Será necesario **configurar un función `setup()`**; será el encargado de configurar el agente, su servicio/s, su comportamiento/s, ...

# Índice

1. ¿Qué es JADE?
2. ¿Cómo crear un agente con JADE?
3. **¿Cómo crear un comportamiento con JADE?**
4. ¿Cómo se comunican los agentes en JADE?
5. Ejemplo simple de interacción entre agentes
6. Ejemplo de MAS inteligente con Weka
7. Recursos
8. Licencia



# ¿Cómo crear un comportamiento con JADE?

Al igual que para resolver una determinada tarea se crean agentes especializados, también será necesario crear **comportamientos específicos** para cada uno de estos agentes. En JADE existen diferentes tipos de comportamientos:

- OneShotBehaviour.
- CyclicBehaviour.
- ParallelBehaviour.
- SequentialBehaviour.
- ...

Todos estos comportamientos se encuentran dentro del paquete `jade.core.behaviours` de JADE.



# ¿Cómo crear un comportamiento con JADE?

Para crear un comportamiento es necesario realizar lo siguiente:

1. Crear una clase Java con el comportamiento deseado.
  - a. Esta clase a de **extender de algún comportamiento** de los incluidos en el paquete `jade.core.behaviours`.
  - b. Se ha de configurar un **función `action()`** que será el que defina la funcionalidad del comportamiento.

# Índice

1. ¿Qué es JADE?
2. ¿Cómo crear un agente con JADE?
3. ¿Cómo crear un comportamiento con JADE?
4. **¿Cómo se comunican los agentes en JADE?**
5. Ejemplo simple de interacción entre agentes
6. Ejemplo de MAS inteligente con Weka
7. Recursos
8. Licencia

# ¿Cómo se comunican los agentes en JADE?

Los agentes de un **Sistema MutiAgente** (MAS, *Multi-Agent System*) establecen **comunicaciones entre si** con el objetivo de resolver tareas y alcanzar el objetivo global del sistema. Estas comunicaciones se realizan a través de **mensajes** que se envían. De forma general, estos mensajes tienen una serie de campos que hay que configurar antes de enviar el mensaje para que la comunicación sea efectiva.

Para crear los mensajes en JADE es necesario, al menos, hacer lo siguiente:

1. Crear una instancia de la **clase “ACLMessage”** de JADE (`jade.lang.ac1.ACLMessage`).
  - a. Es necesario especificar el **tipo de mensaje** que se va a crear (INFORM, REQUEST, CONFIRM, ...)
  - b. Especificar el **agente/s receptor/es** del mensaje.
  - c. Especificar el **contenido del mensaje** (si fuese necesario).

Una vez hecho esto, el agente ya sería posible de **enviar el mensaje** mediante la función `send()`.

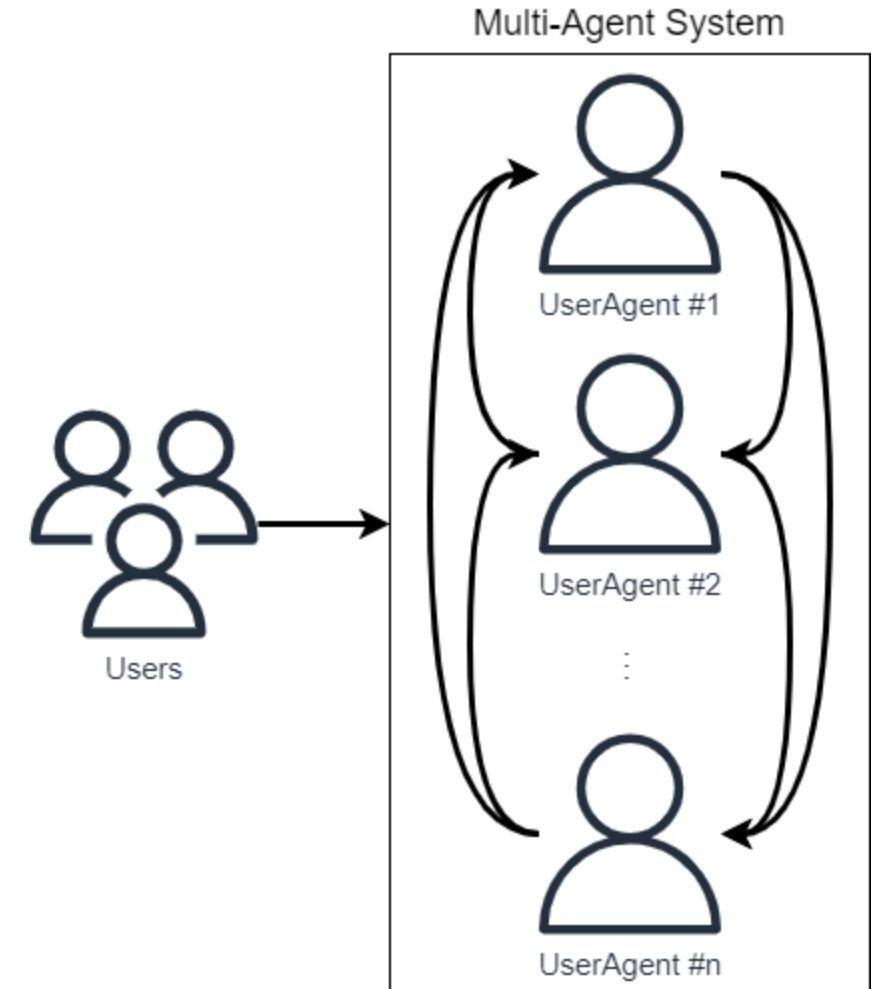
# Índice

1. ¿Qué es JADE?
2. ¿Cómo crear un agente con JADE?
3. ¿Cómo crear un comportamiento con JADE?
4. ¿Cómo se comunican los agentes en JADE?
- 5. Ejemplo simple de interacción entre agentes**
6. Ejemplo de MAS inteligente con Weka
7. Recursos
8. Licencia

# Ejemplo simple de interacción entre agentes

Este ejemplo tiene los siguientes requisitos:

- MAS que simule un chat.
- Los diferentes agentes podrán escribir y enviar mensajes.
- Los mensajes serán recibidos por el resto de agentes del sistema.
- Los usuarios se comunicarán con el sistema a través de una *User Interface* (UI) para cada uno de los agentes del sistema.
- A través de esta interfaz los usuarios podrán escribir y recibir los mensajes.

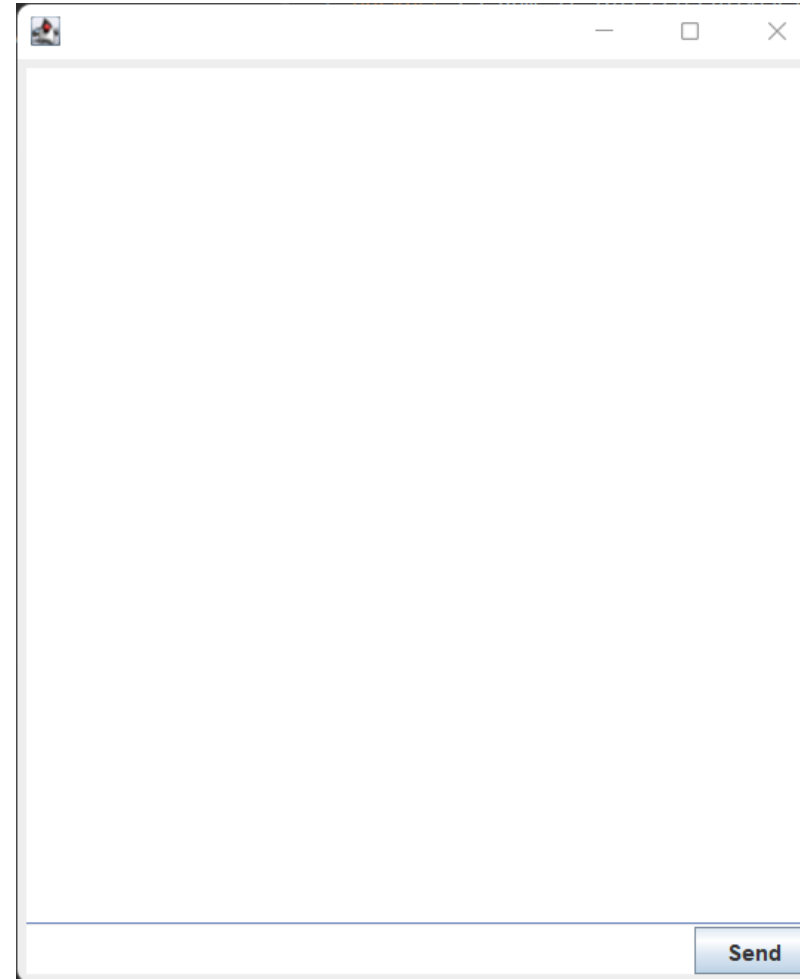


# Ejemplo simple de interacción entre agentes

Como punto de partida, diseñaremos  
la UI.

# Ejemplo simple de interacción entre agentes

Como punto de partida, diseñaremos la UI.

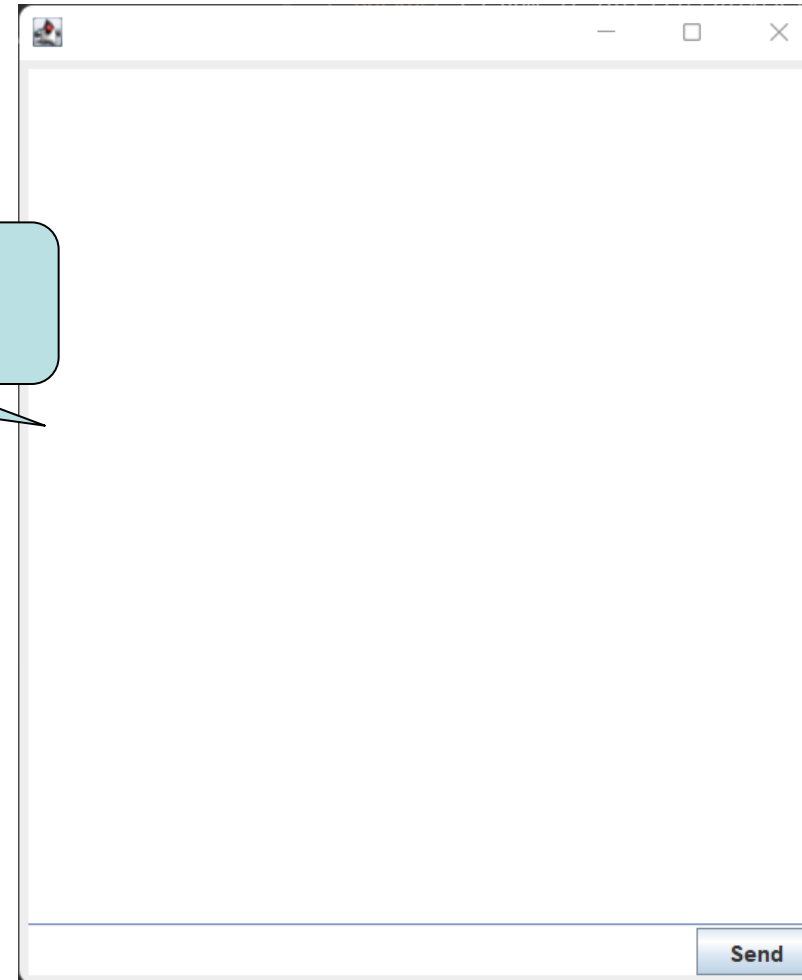




# Ejemplo simple de interacción entre agentes

Como punto de partida, diseñaremos la UI.

Area de mensajes enviados y recibidos.

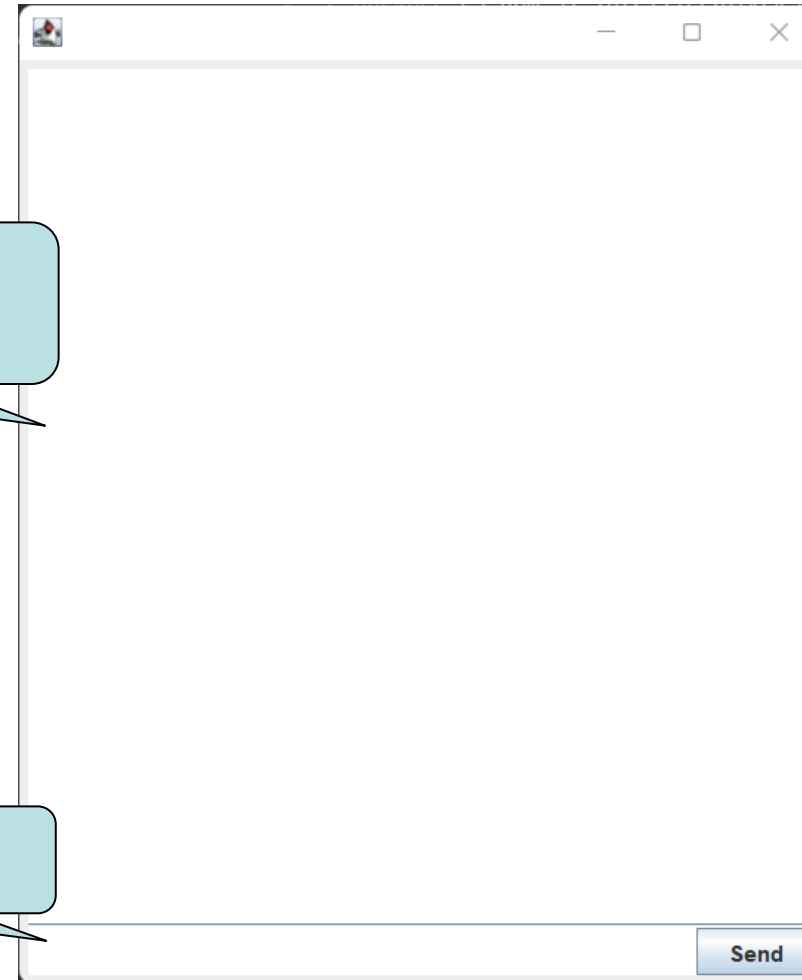


# Ejemplo simple de interacción entre agentes

Como punto de partida, diseñaremos la UI.

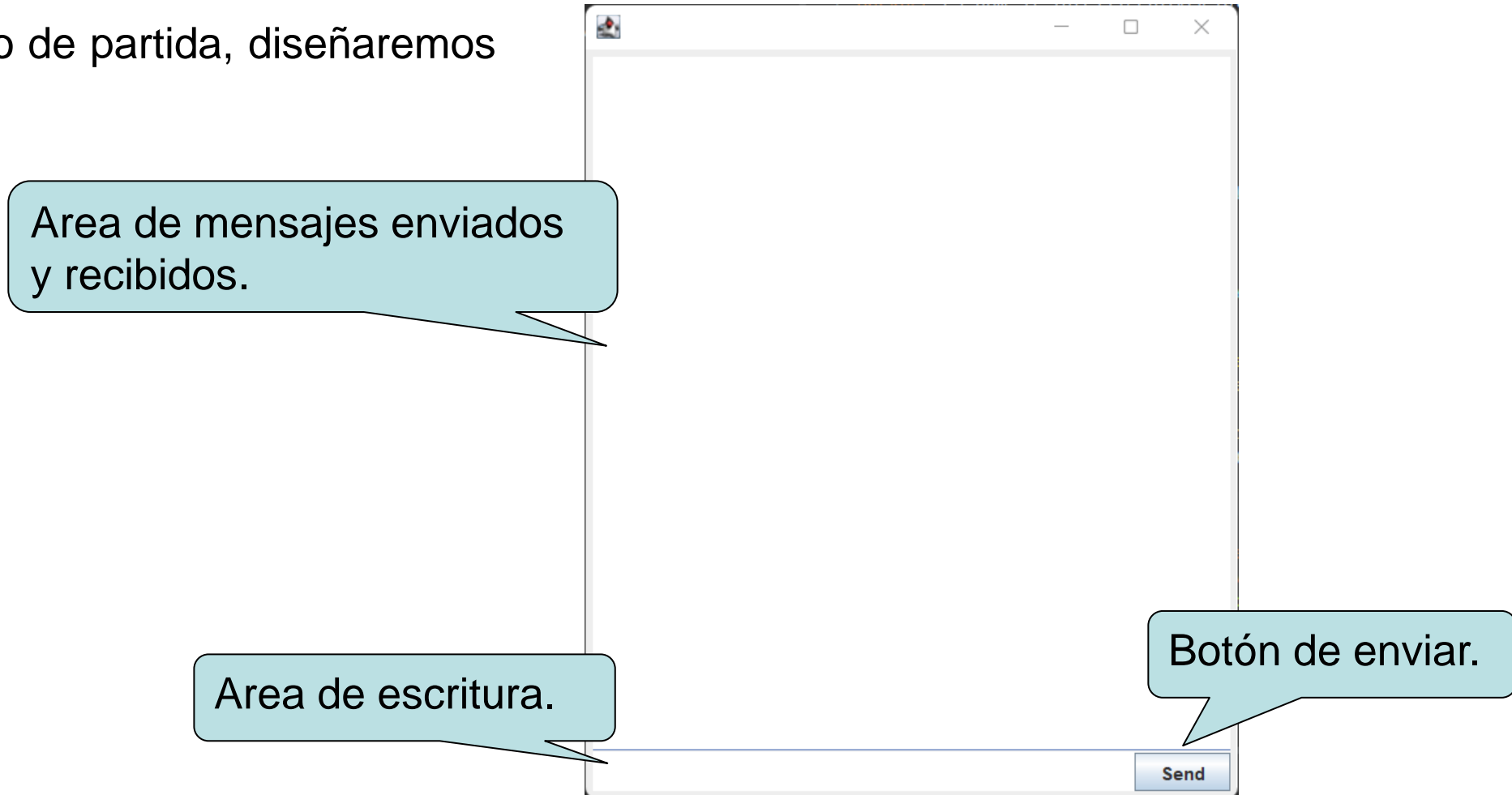
Area de mensajes enviados y recibidos.

Area de escritura.



# Ejemplo simple de interacción entre agentes

Como punto de partida, diseñaremos la UI.



# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

```
public class UserAgent extends Agent{

    private static final long serialVersionUID = 1L;
    private ChatJFrame chatJFrame;
    protected CyclicBehaviourRecieveMessage recieveMessageBehaviour;

    public UserAgent(){
        super();
        this.chatJFrame = new ChatJFrame(this);
        this.recieveMessageBehaviour = new CyclicBehaviourRecieveMessage();
    }
    ...
}
```

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Extensión de la  
clase Agent.

```
public class UserAgent extends Agent{

    private static final long serialVersionUID = 1L;
    private ChatJFrame chatJFrame;
    protected CyclicBehaviourRecieveMessage recieveMessageBehaviour;

    public UserAgent(){
        super();
        this.chatJFrame = new ChatJFrame(this);
        this.recieveMessageBehaviour = new CyclicBehaviourRecieveMessage();
    }
    ...
}
```

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Extensión de la  
clase Agent.

```
public class UserAgent extends Agent{  
  
    private static final long serialVersionUID = 1L;  
    private ChatJFrame chatJFrame;  
    protected CyclicBehaviourRecieveMessage recieveMessageBehaviour;  
  
    public UserAgent(){  
        super();  
        this.chatJFrame = new ChatJFrame(this);  
        this.recieveMessageBehaviour = new CyclicBehaviourRecieveMessage();  
    }  
    ...  
}
```

Hay que configurar los *getters* y los *setters* por si fuesen necesarios.



# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Extensión de la  
clase Agent.

```
public class UserAgent extends Agent{  
  
    private static final long serialVersionUID = 1L;  
    private ChatJFrame chatJFrame;  
    protected CyclicBehaviourRecieveMessage recieveMessageBehaviour;  
  
    public UserAgent(  
        super();  
        this.chatJFrame = new ChatJFrame(this);  
        this.recieveMessageBehaviour = new CyclicBehaviourRecieveMessage();  
    }  
    ...  
}
```

Hay que configurar los *getters* y los *setters* por si fuesen necesarios.

Llamada a la  
función `super()`.

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Extensión de la  
clase Agent.

```
public class UserAgent extends Agent{  
  
    private static final long serialVersionUID = 1L;  
    private ChatJFrame chatJFrame;  
    protected CyclicBehaviourRecieveMessage recieveMessageBehaviour;  
  
    public UserAgent(  
        super();  
        this.chatJFrame = new ChatJFrame(this);  
        this.recieveMessageBehaviour = new CyclicBehaviourRecieveMessage();  
    }  
    ...  
}
```

Hay que configurar los *getters* y los *setters* por si fuesen necesarios.

Llamada a la  
función `super()`.

Inicializamos el atributo  
que contendrá el JFrame.

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Extensión de la  
clase Agent.

```
public class UserAgent extends Agent{  
  
    private static final long serialVersionUID = 1L;  
    private ChatJFrame chatJFrame;  
    protected CyclicBehaviourRecieveMessage recieveMessageBehaviour;  
  
    public UserAgent(  
        super();  
        this.chatJFrame = new ChatJFrame(this);  
        this.recieveMessageBehaviour = new CyclicBehaviourRecieveMessage();  
    }  
    ...  
}
```

Hay que configurar los *getters* y los *setters* por si fuesen necesarios.

Llamada a la  
función `super()`.

Inicializamos el atributo  
que contendrá el JFrame.

Inicializamos el atributo que contendrá  
el comportamiento del agente.

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

```
...
protected void setup() {
    DFAgentDescription dfd = new
    DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new
    ServiceDescription();
    sd.setName("User communication");
    sd.setType("User communication");
    sd.addLanguages(new
    SLCodec().getName());
    dfd.addServices(sd);

    try {
        DFService.register(this, dfd);
    } ...
}
```

```
...
        catch(FIPAException e) {
            System.err.println("Agent " +
            this.getLocalName() + ": " + e.getMessage());
        }

        this.addBehaviour(this.getRecieveMessa
        geBehaviour());

        this.getChatJFrame().setVisible(true);
    }
}
```

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Método setup().

```
...
protected void setup() {
    DFAgentDescription dfd = new
    DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new
    ServiceDescription();
    sd.setName("User communication");
    sd.setType("User communication");
    sd.addLanguages(new
    SLCodec().getName());
    dfd.addServices(sd);

    try {
        DFService.register(this, dfd);
    } ...
}
```

```
...
        catch(FIPAException e) {
            System.err.println("Agent " +
            this.getLocalName() + ": " + e.getMessage());
        }

        this.addBehaviour(this.getRecieveMessa
        geBehaviour());

        this.getChatJFrame().setVisible(true);
    }
}
```

# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Método setup().

```
...  
protected void setup() {  
    DFAgentDescription dfd = new  
    DFAgentDescription();  
    dfd.setName(getAID());  
    ServiceDescription sd = new  
    ServiceDescription();  
    sd.setName("User communication");  
    sd.setType("User communication");  
    sd.addLanguages(new  
    SLCodec().getName());  
    dfd.addServices(sd);  
  
    try {  
        DFService.register(this, dfd);  
    } ...
```

Configuración y registro  
del servicio del agente.

```
...  
        catch(FIPAException e) {  
            System.err.println("Agent " +  
            this.getLocalName() + ": " + e.getMessage());  
        }  
  
        this.addBehaviour(this.getRecieveMessa  
geBehaviour());  
  
        this.getChatJFrame().setVisible(true);  
    }  
}
```



# Ejemplo simple de interacción entre agentes

Ahora hay que definir el agente.

Método setup().

```
...
protected void setup() {
    DFAgentDescription dfd = new
    DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new
    ServiceDescription();
    sd.setName("User communication");
    sd.setType("User communication");
    sd.addLanguages(new
    SLCodec().getName());
    dfd.addServices(sd);

    try {
        DFService.register(this, dfd);
    } ...
}
```

Configuración y registro  
del servicio del agente.

```
...
catch(FIPAException e) {
    System.err.println("Agent " +
    this.getLocalName() + ": " + e.getMessage());
}

this.addBehaviour(this.getRecieveMessa
geBehaviour());

this.getChatJFrame().setVisible(true);
}
}
```

Añadimos el comportamiento  
y hacemos visible el JFrame.

# Ejemplo simple de interacción entre agentes

Definimos el comportamiento.

# Ejemplo simple de interacción entre agentes

Definimos el comportamiento.

```
public class CyclicBehaviourRecieveMessage extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.INFORM));
            if(msg != null) {
                ((UserAgent) this.myAgent).getChatJFrame().addRecievedMessage(msg.getContent());
            }else {
                this.block();
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

# Ejemplo simple de interacción entre agentes

Definimos el comportamiento.

Extensión de la clase  
CyclicBehaviour.

```
public class CyclicBehaviourRecieveMessage extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.INFORM));
            if(msg != null) {
                ((UserAgent) this.myAgent).getChatJFrame().addRecievedMessage(msg.getContent());
            }else {
                this.block();
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

# Ejemplo simple de interacción entre agentes

Definimos el comportamiento.

Extensión de la clase  
CyclicBehaviour.

```
public class CyclicBehaviourRecieveMessage extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.INFORM));
            if(msg != null) {
                ((UserAgent) this.myAgent).getChatJFrame().addRecievedMessage(msg.getContent());
            }else {
                this.block();
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Recibir el mensaje.

# Ejemplo simple de interacción entre agentes

Para ejecutar el sistema es necesario configurar la ejecución de nuestro IDE:

1. Establecer como clase principal del programa la **clase Boot** de JADE (jade.Boot).
2. Tendremos que añadir una serie de **argumentos** a la ejecución:



```
-gui  
UserAgent1:es.upm.si.masChat.UserAgent;UserAg  
ent2:es.upm.si.masChat.UserAgent
```

# Ejemplo simple de interacción entre agentes

Para ejecutar el sistema es necesario configurar la ejecución de nuestro IDE:

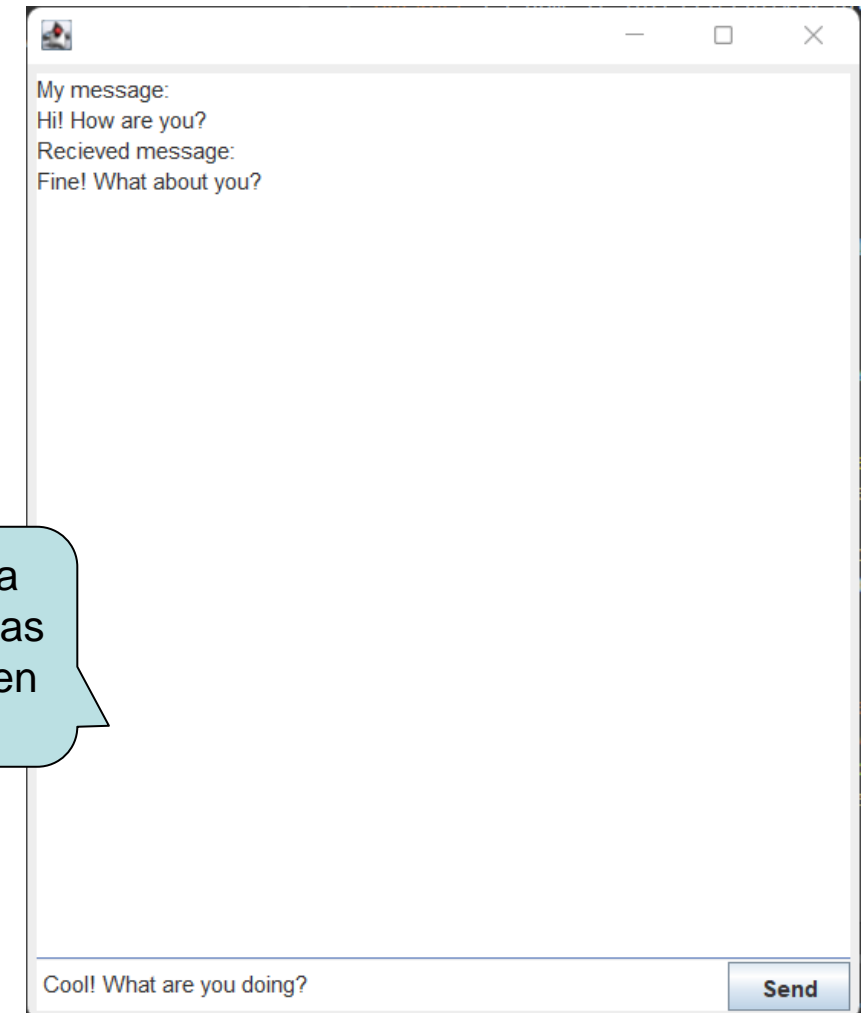
1. Establecer como clase principal del programa la **clase Boot** de JADE (jade.Boot).
2. Tendremos que añadir una serie de **argumentos** a la ejecución:

Interfaz final del sistema  
(existirán tantas ventanas  
como AgentUser haya en  
el sistema).



-gui

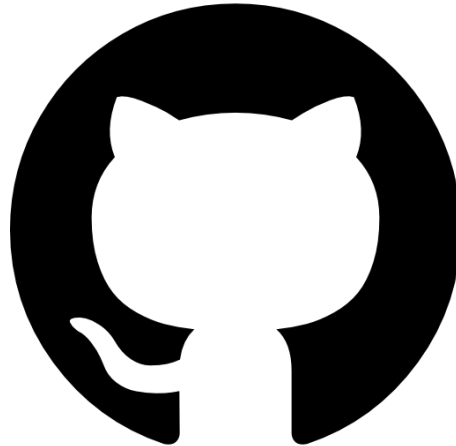
```
UserAgent1:es.upm.si.masChat.UserAgent;UserAg  
ent2:es.upm.si.masChat.UserAgent
```





# Ejemplo simple de interacción entre agentes

Este sistema está disponible en un repositorio de GitHub:



[dmjimenezbravo/MASChat](https://github.com/dmjimenezbravo/MASChat)

# Índice

1. ¿Qué es JADE?
2. ¿Cómo crear un agente con JADE?
3. ¿Cómo crear un comportamiento con JADE?
4. ¿Cómo se comunican los agentes en JADE?
5. Ejemplo simple de interacción entre agentes
6. **Ejemplo de MAS inteligente con Weka**
7. Recursos
8. Licencia

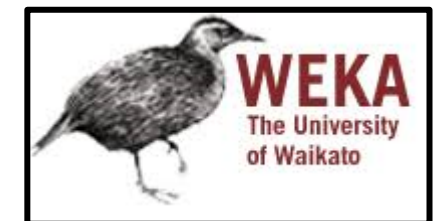
# Ejemplo de MAS inteligente con Weka

¿Qué es Weka?



**Weka** es una colección de **algoritmos de aprendizaje automático** para tareas de minería de datos. Contiene herramientas para la preparación de datos, clasificación, regresión, agrupación, reglas de asociación y visualización.

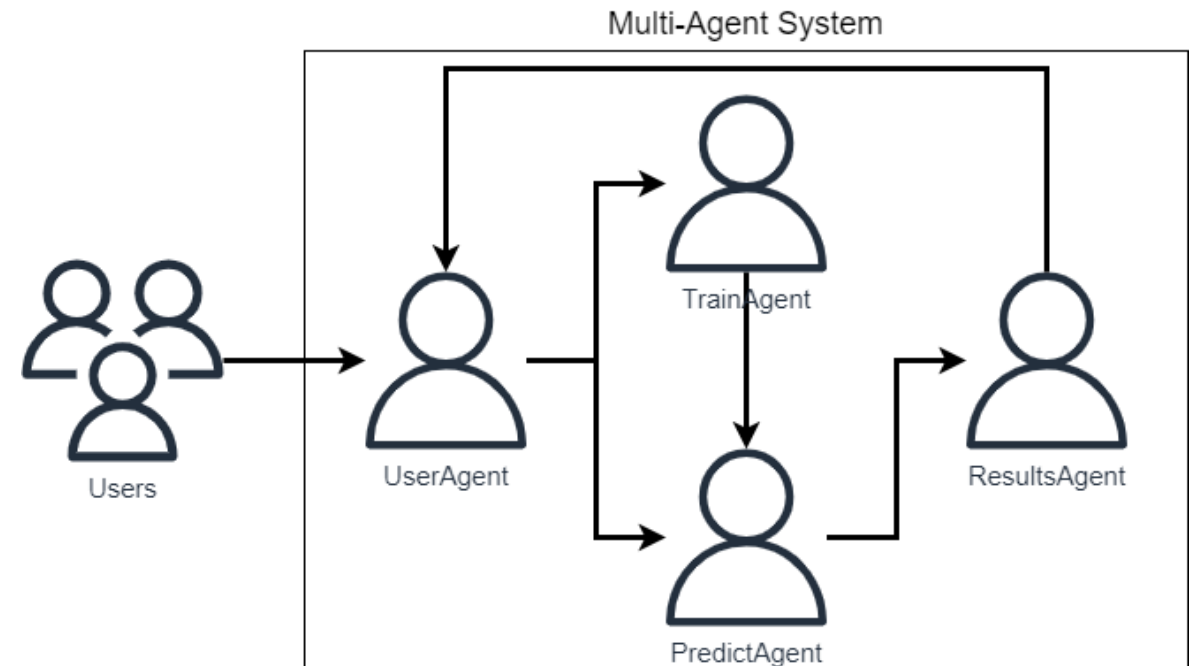
Fue creado por investigadores de la **Universidad de Waikato** (Nueva Zelanda). Proporciona [formación sobre aprendizaje automático](#).



# Ejemplo de MAS inteligente con Weka

Este ejemplo tiene los siguientes requisitos:

- MAS que sea capaz de entrenar un modelo y predecir instancias de un modelo dado.
- Cada agente tendrá una función (comunicación con los usuarios, entrenamiento, predicción y visualización de resultados).
- Los usuarios se comunicarán con el sistema a través de una *User Interface* (UI).

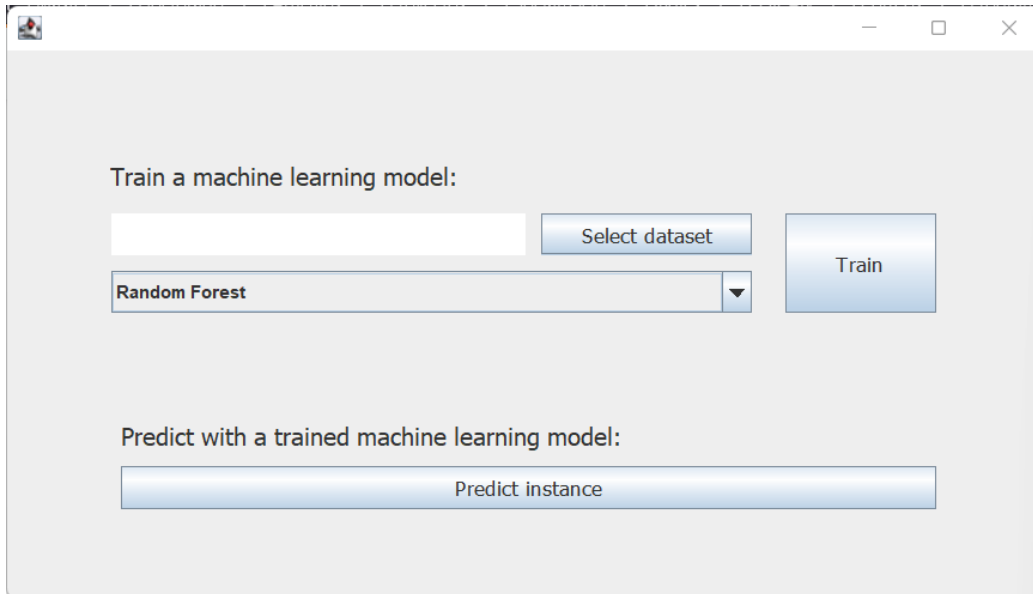


# Ejemplo de MAS inteligente con Weka

Como punto de partida,  
diseñaremos la UI.

# Ejemplo de MAS inteligente con Weka

Como punto de partida,  
diseñaremos la UI.

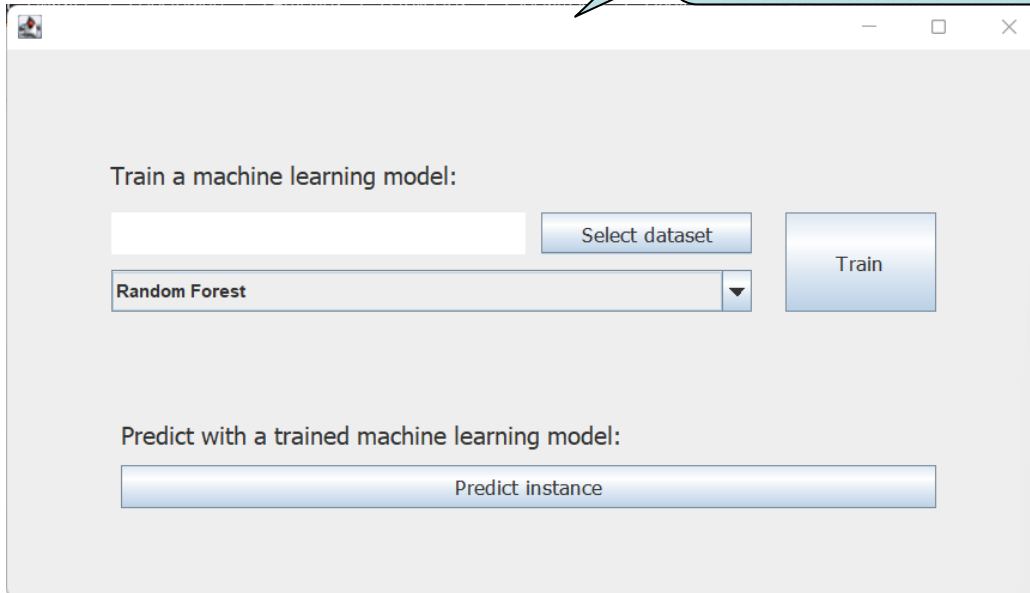


The screenshot shows a window titled "Train a machine learning model:" with a text input field and a "Select dataset" button. Below this is a dropdown menu currently showing "Random Forest" and a "Train" button. A second section, "Predict with a trained machine learning model:", contains a "Predict instance" button.

# Ejemplo de MAS inteligente con Weka

Como punto de partida,  
diseñaremos la UI.

Interfaz principal; entrena  
un modelo y lanza la  
pantalla de predicción.



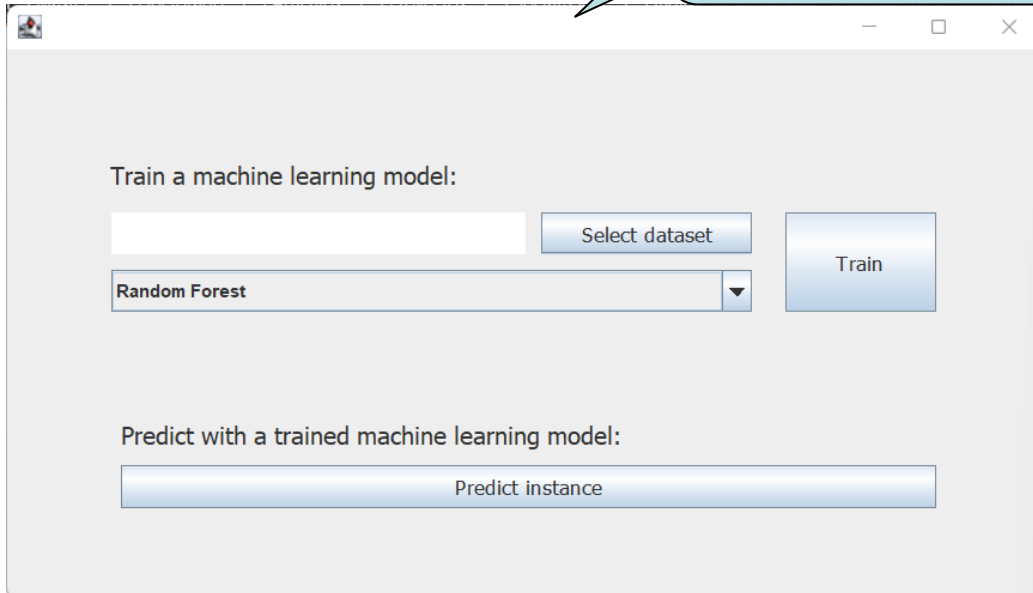
Train a machine learning model:

Predict with a trained machine learning model:

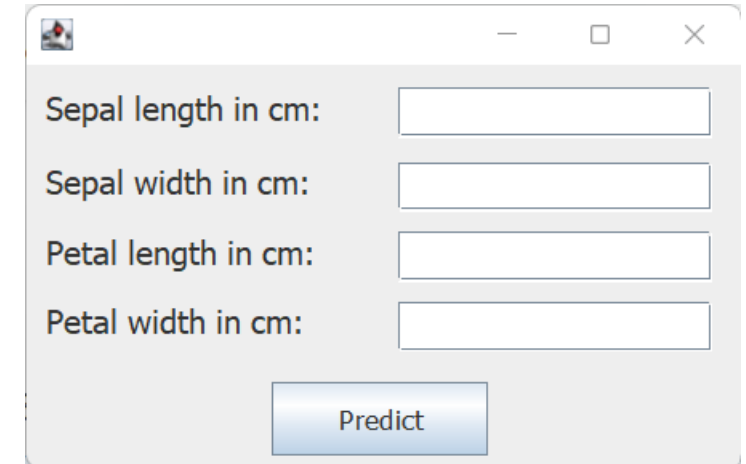
# Ejemplo de MAS inteligente con Weka

Como punto de partida,  
diseñaremos la UI.

Interfaz principal; entrena  
un modelo y lanza la  
pantalla de predicción.



The screenshot shows the main Weka GUI window. It has a light gray background and a standard Windows-style title bar. The window is divided into two main sections. The top section is titled "Train a machine learning model:" and contains a text input field, a "Select dataset" button, a dropdown menu currently showing "Random Forest", and a "Train" button. The bottom section is titled "Predict with a trained machine learning model:" and contains a single "Predict instance" button.



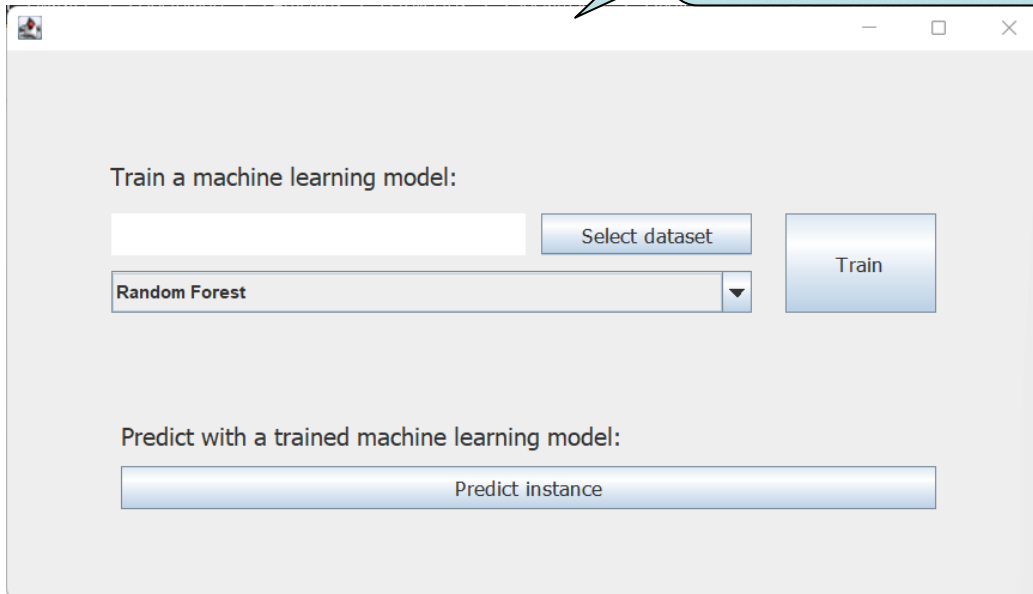
The screenshot shows a smaller dialog box titled "Predict" with a standard Windows-style title bar. It contains four input fields for numerical data: "Sepal length in cm:", "Sepal width in cm:", "Petal length in cm:", and "Petal width in cm:". Below these fields is a single "Predict" button.



# Ejemplo de MAS inteligente con Weka

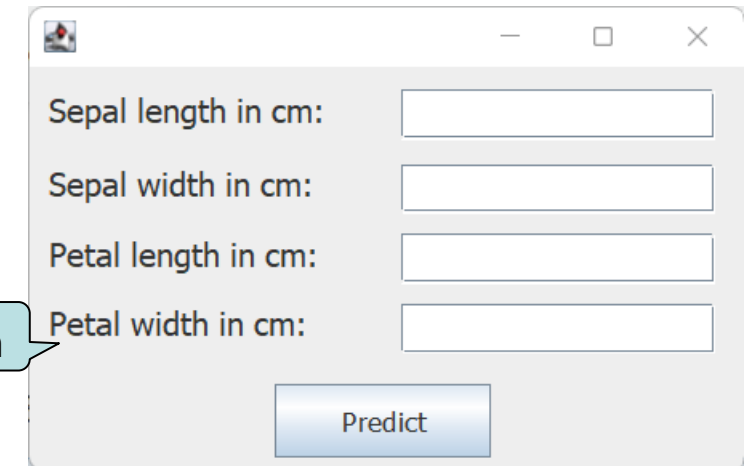
Como punto de partida,  
diseñaremos la UI.

Interfaz principal; entrena  
un modelo y lanza la  
pantalla de predicción.



The main Weka GUI window is titled 'Train a machine learning model:'. It features a text input field for dataset selection, a 'Select dataset' button, a dropdown menu currently showing 'Random Forest', and a 'Train' button. Below this, under the heading 'Predict with a trained machine learning model:', there is a 'Predict instance' button.

Pantalla de predicción

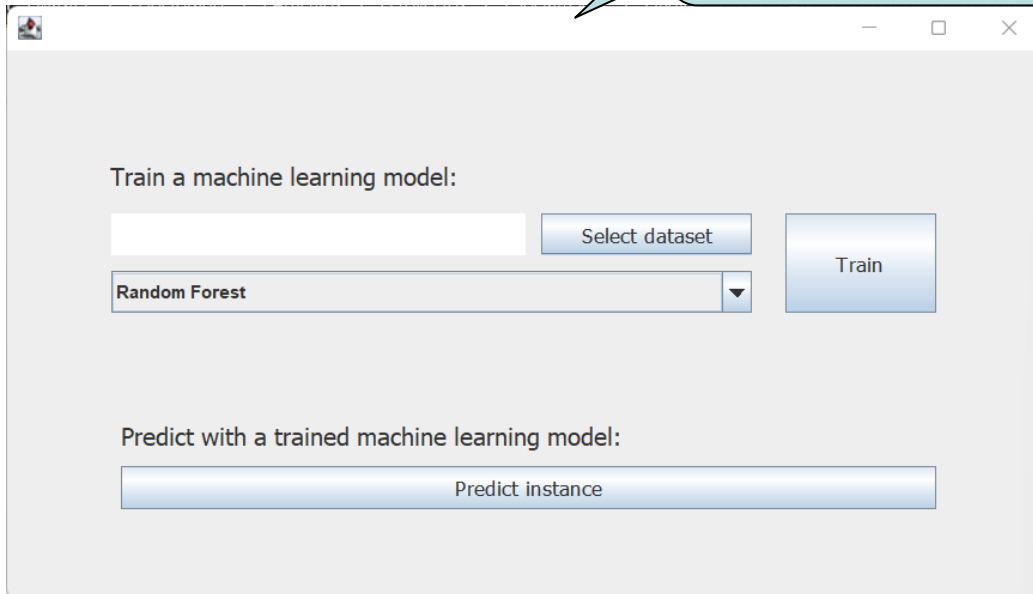


The prediction window is titled 'Predict'. It contains four input fields for feature values: 'Sepal length in cm:', 'Sepal width in cm:', 'Petal length in cm:', and 'Petal width in cm:'. A 'Predict' button is located at the bottom right of the window.

# Ejemplo de MAS inteligente con Weka

Como punto de partida,  
diseñaremos la UI.

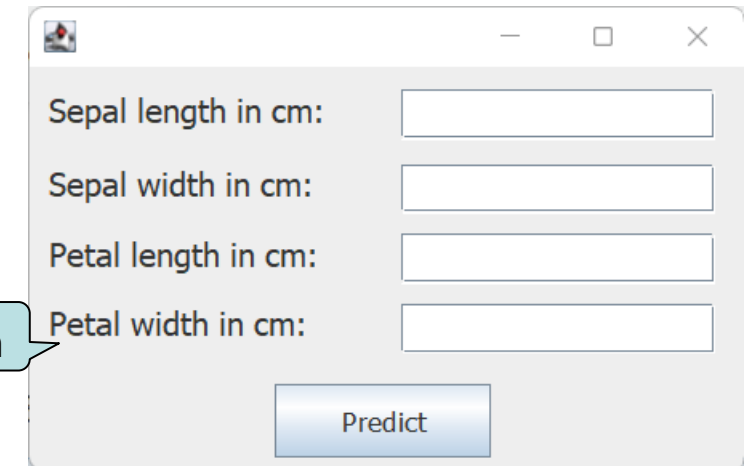
Interfaz principal; entrena  
un modelo y lanza la  
pantalla de predicción.



Train a machine learning model:

Predict with a trained machine learning model:

Pantalla de predicción

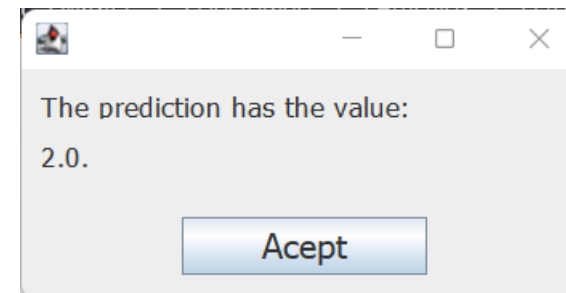


Sepal length in cm:

Sepal width in cm:

Petal length in cm:

Petal width in cm:

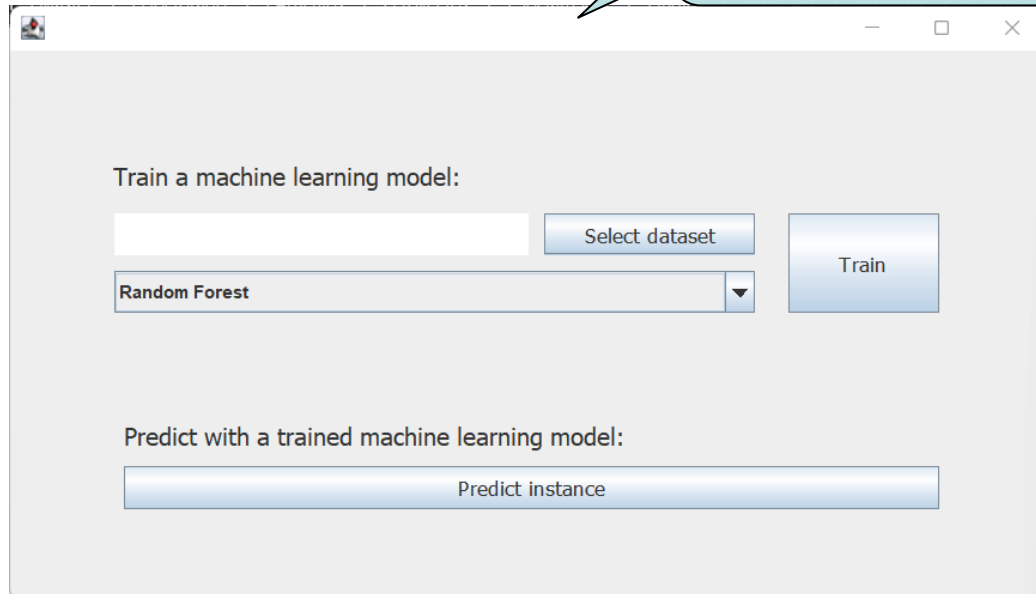


The prediction has the value:  
2.0.

# Ejemplo de MAS inteligente con Weka

Como punto de partida,  
diseñaremos la UI.

Interfaz principal; entrena  
un modelo y lanza la  
pantalla de predicción.

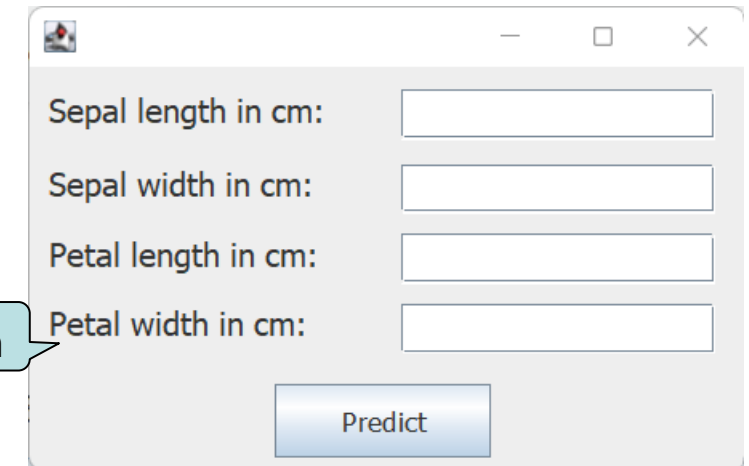


Train a machine learning model:

Random Forest

Predict with a trained machine learning model:

Pantalla de predicción



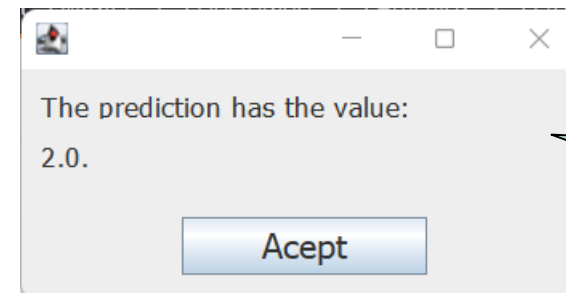
Sepal length in cm:

Sepal width in cm:

Petal length in cm:

Petal width in cm:

Pantalla de resultados.



The prediction has the value:  
2.0.

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

```
public class UserAgent extends Agent{
    private static final long serialVersionUID = 1L;
    private boolean canPredict;
    private boolean canTrain;
    private File file;
    private String classificationMethod;
    private Classifier classifier;
    private UserJFrame userJFrame;
    private ConfigurePredictionJFrame configurePredictionJFrame;
    protected CyclicBehaviourSendTrainPredict trainPredictBehaviour;
    protected CyclicBehaviourLaunchWindow launchBehaviour;

    public UserAgent(){
        super();
        this.canPredict = false;
        this.canTrain = false;
        this.userJFrame = new UserJFrame(this);
        this.configurePredictionJFrame = new ConfigurePredictionJFrame(this);
        this.trainPredictBehaviour = new CyclicBehaviourSendTrainPredict();
        this.launchBehaviour = new CyclicBehaviourLaunchWindow();
    }
}
```

...

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

```
public class UserAgent extends Agent{  
    private static final long serialVersionUID = 1L;  
    private boolean canPredict;  
    private boolean canTrain;  
    private File file;  
    private String classificationMethod;  
    private Classifier classifier;  
    private UserJFrame userJFrame;  
    private ConfigurePredictionJFrame configurePredictionJFrame;  
    protected CyclicBehaviourSendTrainPredict trainPredictBehaviour;  
    protected CyclicBehaviourLaunchWindow launchBehaviour;  
  
    public UserAgent(){  
        super();  
        this.canPredict = false;  
        this.canTrain = false;  
        this.userJFrame = new UserJFrame(this);  
        this.configurePredictionJFrame = new ConfigurePredictionJFrame(this);  
        this.trainPredictBehaviour = new CyclicBehaviourSendTrainPredict();  
        this.launchBehaviour = new CyclicBehaviourLaunchWindow();  
    }  
}
```

Extensión de la clase Agent de JADE.

...

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

```
public class UserAgent extends Agent{
    private static final long serialVersionUID = 1L;
    private boolean canPredict;
    private boolean canTrain;
    private File file;
    private String classificationMethod;
    private Classifier classifier;
    private UserJFrame userJFrame;
    private ConfigurePredictionJFrame configurePredictionJFrame;
    protected CyclicBehaviourSendTrainPredict trainPredictBehaviour;
    protected CyclicBehaviourLaunchWindow launchBehaviour;

    public UserAgent(){
        super();
        this.canPredict = false;
        this.canTrain = false;
        this.userJFrame = new UserJFrame(this);
        this.configurePredictionJFrame = new ConfigurePredictionJFrame(this);
        this.trainPredictBehaviour = new CyclicBehaviourSendTrainPredict();
        this.launchBehaviour = new CyclicBehaviourLaunchWindow();
    }
}
```

Extensión de la clase Agent de JADE.

Atributos, será necesario crear los *getters* y los *setters*. Obsérvese que en esta clase no tiene un parámetro para el JFrame.

...

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

```
public class UserAgent extends Agent{  
    private static final long serialVersionUID = 1L;  
    private boolean canPredict;  
    private boolean canTrain;  
    private File file;  
    private String classificationMethod;  
    private Classifier classifier;  
    private UserJFrame userJFrame;  
    private ConfigurePredictionJFrame configurePredictionJFrame;  
    protected CyclicBehaviourSendTrainPredict trainPredictBehaviour;  
    protected CyclicBehaviourLaunchWindow launchBehaviour;
```

Extensión de la clase Agent de JADE.

Atributos, será necesario crear los *getters* y los *setters*. Obsérvese que en esta clase no tiene un parámetro para el JFrame.

```
    public UserAgent(){  
        super();  
        this.canPredict = false;  
        this.canTrain = false;  
        this.userJFrame = new UserJFrame(this);  
        this.configurePredictionJFrame = new ConfigurePredictionJFrame(this);  
        this.trainPredictBehaviour = new CyclicBehaviourSendTrainPredict();  
        this.launchBehaviour = new CyclicBehaviourLaunchWindow();  
    }  
    ...
```

Constructor con la llamada a la función `super()`.



# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

```
...
protected void setup() {
    DFAgentDescription dfd = new
    DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new
    ServiceDescription();
    sd.setName("User communication");
    sd.setType("User communication");
    sd.addLanguages(new
    SLCodec().getName());
    dfd.addServices(sd);

    try {
        DFService.register(this, dfd);
    } ...
}
```

```
...
        catch(FIPAException e) {
            System.err.println("Agent " +
            this.getLocalName() + ": " + e.getMessage());
        }

        this.addBehaviour(this.getTrainPredictB
        ehaviour());
        this.addBehaviour(this.getLaunchBehavio
        ur());
        this.getUserJFrame().setVisible(true);
    }
}
```

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

Método setup().

```
...
protected void setup() {
    DFAgentDescription dfd = new
DFAgentDescription();
    dfd.setName(getAID());
    ServiceDescription sd = new
ServiceDescription();
    sd.setName("User communication");
    sd.setType("User communication");
    sd.addLanguages(new
SLCodec().getName());
    dfd.addServices(sd);

    try {
        DFService.register(this, dfd);
    } ...
}
```

```
...
        catch(FIPAException e) {
            System.err.println("Agent " +
this.getLocalName() + ": " + e.getMessage());
        }

        this.addBehaviour(this.getTrainPredictB
ehaviour());
        this.addBehaviour(this.getLaunchBehavio
ur());
        this.getUserJFrame().setVisible(true);
    }
}
```

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

Método setup().

```
...  
protected void setup() {  
    DFAgentDescription dfd = new  
DFAgentDescription();  
    dfd.setName(getAID());  
    ServiceDescription sd = new  
ServiceDescription();  
    sd.setName("User communication");  
    sd.setType("User communication");  
    sd.addLanguages(new  
SLCodec().getName());  
    dfd.addServices(sd);  
  
    try {  
        DFService.register(this, dfd);  
    } ...
```

Configuración y  
registro del servicio  
del agente.

```
...  
        catch(FIPAException e) {  
            System.err.println("Agent " +  
this.getLocalName() + ": " + e.getMessage());  
        }  
  
        this.addBehaviour(this.getTrainPredictB  
ehaviour());  
        this.addBehaviour(this.getLaunchBehavio  
ur());  
        this.getUserJFrame().setVisible(true);  
    }  
}
```

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

Método setup().

```
...  
protected void setup() {  
    DFAgentDescription dfd = new  
DFAgentDescription();  
    dfd.setName(getAID());  
    ServiceDescription sd = new  
ServiceDescription();  
    sd.setName("User communication");  
    sd.setType("User communication");  
    sd.addLanguages(new  
SLCodec().getName());  
    dfd.addServices(sd);  
  
    try {  
        DFService.register(this, dfd);  
    } ...
```

Configuración y  
registro del servicio  
del agente.

```
...  
    catch(FIPAException e) {  
        System.err.println("Agent " +  
this.getLocalName() + ": " + e.getMessage());  
    }  
  
    this.addBehaviour(this.getTrainPredictB  
ehaviour());  
    this.addBehaviour(this.getLaunchBehavio  
ur());  
    this.getUserJFrame().setVisible(true);  
    }  
}
```

Añadimos los comportamientos.

# Ejemplo de MAS inteligente con Weka

Diseño del agente UserAgent.

Método setup().

```
...  
protected void setup() {  
    DFAgentDescription dfd = new  
DFAgentDescription();  
    dfd.setName(getAID());  
    ServiceDescription sd = new  
ServiceDescription();  
    sd.setName("User communication");  
    sd.setType("User communication");  
    sd.addLanguages(new  
SLCodec().getName());  
    dfd.addServices(sd);  
  
    try {  
        DFService.register(this, dfd);  
    } ...
```

Configuración y  
registro del servicio  
del agente.

```
...  
    catch(FIPAException e) {  
        System.err.println("Agent " +  
this.getLocalName() + ": " + e.getMessage());  
    }  
  
    this.addBehaviour(this.getTrainPredictB  
ehaviour());  
    this.addBehaviour(this.getLaunchBehavio  
ur());  
    this.getUserJFrame().setVisible(true);  
    }  
}
```

Añadimos los comportamientos.

Hacemos visible el JFrame.

# Ejemplo de MAS inteligente con Weka

A la hora de definir el UserJFrame es importante apuntar algunas cosas.

# Ejemplo de MAS inteligente con Weka

A la hora de definir el UserJFrame es importante apuntar algunas cosas.

```
...
JButton btnSelectDataset = new JButton("Select dataset");
btnSelectDataset.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            JFileChooser fileChooser = new JFileChooser();
            fileChooser.showOpenDialog(fileChooser);
            String path = fileChooser.getSelectedFile().getAbsolutePath();
            File file = new File(path);
            textPaneDataset.setText(file.getPath());
            userAgent.setFile(file);
            datasetLoaded = true;
        }
        catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
});
btnSelectDataset.setFont(new Font("Tahoma", Font.PLAIN, 14));
btnSelectDataset.setBounds(347, 86, 141, 28);
panel.add(btnSelectDataset);
...
```



# Ejemplo de MAS inteligente con Weka

A la hora de definir el UserJFrame es importante apuntar algunas cosas.

```
...
JButton btnSelectDataset = new JButton("Select dataset");
btnSelectDataset.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            JFileChooser fileChooser = new JFileChooser();
            fileChooser.showOpenDialog(fileChooser);
            String path = fileChooser.getSelectedFile().getAbsolutePath();
            File file = new File(path);
            textPaneDataset.setText(file.getPath());
            userAgent.setFile(file);
            datasetLoaded = true;
        }
        catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
});
btnSelectDataset.setFont(new Font("Tahoma", Font.PLAIN, 14));
btnSelectDataset.setBounds(347, 86, 141, 28);
panel.add(btnSelectDataset);
...
```

Lanzar el seleccionador de archivos para localizar nuestro archivo entre los directorios del sistema.

# Ejemplo de MAS inteligente con Weka

A la hora de definir el UserJFrame es importante apuntar algunas cosas.

```
...
JButton btnSelectDataset = new JButton("Select dataset");
btnSelectDataset.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            JFileChooser fileChooser = new JFileChooser();
            fileChooser.showOpenDialog(fileChooser);
            String path = fileChooser.getSelectedFile().getAbsolutePath();
            File file = new File(path);
            textPaneDataset.setText(file.getPath());
            userAgent.setFile(file);
            datasetLoaded = true;
        }
        catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
});
btnSelectDataset.setFont(new Font("Tahoma", Font.PLAIN, 14));
btnSelectDataset.setBounds(347, 86, 141, 28);
panel.add(btnSelectDataset);
...
```

Lanzar el seleccionador de archivos para localizar nuestro archivo entre los directorios del sistema.

Carga de un fichero de datos mediante Java.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourSendTrainPredict.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourSendTrainPredict.

```
public class CyclicBehaviourSendTrainPredict extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUE
ST));

            if(msg != null) {
                if(msg.getContent().equals("Train")) {
                    TrainComponents trainComponents = new
TrainComponents();
                    trainComponents.setClassificationMethod(((UserAgent
) this.myAgent).getClassificationMethod());
                    trainComponents.setDataset(((UserAgent)
this.myAgent).getFile());

                    ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                    msg2.addReceiver(new AID("TrainAgent",
AID.ISLOCALNAME));
                    msg2.setContentObject(trainComponents);
                    this.myAgent.send(msg2);
                }
            }
        }
    }
}
```

```
...
        else {
            if(msg.getContentObject() instanceof
PredictComponets) {
                ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                msg2.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
                msg2.setContentObject(msg.getContentObject());
                this.myAgent.send(msg2);
            }
        }
    }else {
        this.block();
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourSendTrainPredict.

```
public class CyclicBehaviourSendTrainPredict extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUE
ST));

            if(msg != null) {
                if(msg.getContent().equals("Train")) {
                    TrainComponents trainComponents = new
TrainComponents();
                    trainComponents.setClassificationMethod(((UserAgent
) this.myAgent).getClassificationMethod());
                    trainComponents.setDataset(((UserAgent)
this.myAgent).getFile());

                    ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                    msg2.addReceiver(new AID("TrainAgent",
AID.ISLOCALNAME));
                    msg2.setContentObject(trainComponents);
                    this.myAgent.send(msg2);
                }
            }
        }
    }
}
```

Se recibe el mensaje.

```
...
        else {
            if(msg.getContentObject() instanceof
PredictComponets) {
                ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                msg2.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
                msg2.setContentObject(msg.getContentObject());
                this.myAgent.send(msg2);
            }
        }
    }else {
        this.block();
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourSendTrainPredict.

```
public class CyclicBehaviourSendTrainPredict extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUE
ST));

            if(msg != null) {
                if(msg.getContent().equals("Train")) {
                    TrainComponents trainComponents = new
TrainComponents();
                    trainComponents.setClassificationMethod(((UserAgent
) this.myAgent).getClassificationMethod());
                    trainComponents.setDataset(((UserAgent)
this.myAgent).getFile());

                    ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                    msg2.addReceiver(new AID("TrainAgent",
AID.ISLOCALNAME));
                    msg2.setContentObject(trainComponents);
                    this.myAgent.send(msg2);
                }
            }
        }
    }
}
```

Se recibe el mensaje.

Caso en el que se desee entrenar un modelo.

```
...
        else {
            if(msg.getContentObject() instanceof
PredictComponets) {
                ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                msg2.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
                msg2.setContentObject(msg.getContentObject());
                this.myAgent.send(msg2);
            }
        }
    }else {
        this.block();
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourSendTrainPredict.

```
public class CyclicBehaviourSendTrainPredict extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUE
ST));

            if(msg != null) {
                if(msg.getContent().equals("Train")) {
                    TrainComponents trainComponents = new
TrainComponents();
                    trainComponents.setClassificationMethod(((UserAgent
) this.myAgent).getClassificationMethod());
                    trainComponents.setDataset(((UserAgent)
this.myAgent).getFile());

                    ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                    msg2.addReceiver(new AID("TrainAgent",
AID.ISLOCALNAME));
                    msg2.setContentObject(trainComponents);
                    this.myAgent.send(msg2);
                }
            }
        }
    }
}
```

Se recibe el mensaje.

Caso en el que se desee entrenar un modelo.

Envío de contenidos.

```
...
        else {
            if(msg.getContentObject() instanceof
PredictComponets) {
                ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                msg2.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
                msg2.setContentObject(msg.getContentObject());
                this.myAgent.send(msg2);
            }
        }
    }else {
        this.block();
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourSendTrainPredict.

```
public class CyclicBehaviourSendTrainPredict extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUE
ST));

            if(msg != null) {
                if(msg.getContent().equals("Train")) {
                    TrainComponents trainComponents = new
TrainComponents();
                    trainComponents.setClassificationMethod(((UserAgent
) this.myAgent).getClassificationMethod());
                    trainComponents.setDataset(((UserAgent)
this.myAgent).getFile());

                    ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                    msg2.addReceiver(new AID("TrainAgent",
AID.ISLOCALNAME));
                    msg2.setContentObject(trainComponents);
                    this.myAgent.send(msg2);
                }
            }
        }
    }
}
```

Se recibe el mensaje.

Caso en el que se desee entrenar un modelo.

Envío de contenidos.

```
...
        else {
            if(msg.getContentObject() instanceof
PredictComponets) {
                ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                msg2.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
                msg2.setContentObject(msg.getContentObject());
                this.myAgent.send(msg2);
            }
        }
    }else {
        this.block();
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

Caso en el que se desee predecir una estancia.



# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourSendTrainPredict.

```
public class CyclicBehaviourSendTrainPredict extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUE
ST));

            if(msg != null) {
                if(msg.getContent().equals("Train")) {
                    TrainComponents trainComponents = new
TrainComponents();
                    trainComponents.setClassificationMethod(((UserAgent
) this.myAgent).getClassificationMethod());
                    trainComponents.setDataset(((UserAgent)
this.myAgent).getFile());

                    ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                    msg2.addReceiver(new AID("TrainAgent",
AID.ISLOCALNAME));
                    msg2.setContentObject(trainComponents);
                    this.myAgent.send(msg2);
                }
            }
        }
    }
}
```

Se recibe el mensaje.

Caso en el que se desee entrenar un modelo.

Envío de contenidos.

```
...
        else {
            if(msg.getContentObject() instanceof
PredictComponets) {
                ACLMessage msg2 = new
ACLMessage(ACLMessage.REQUEST);
                msg2.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
                msg2.setContentObject(msg.getContentObject());
                this.myAgent.send(msg2);
            }
        }
    }
    }else {
        this.block();
    }
}
catch (Exception e)
{
    e.printStackTrace();
}
}
```

Caso en el que se desee predecir una estancia.

Envío de contenidos.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

```
public class CyclicBehaviourTrain extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg = this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUEST));

            if(msg != null) {
                TrainComponents trainComponets = new TrainComponents();
                trainComponets = (TrainComponents) msg.getContentObject();
                ((TrainAgent) this.myAgent).setFile(trainComponets.getDataset());
                ((TrainAgent) this.myAgent).setClassificationMethod(trainComponets.getClassificationMethod());
                this.train();
            }else {
                this.block();
            }
        }
        catch (Exception e) {
            //e.printStackTrace();
        }
    }
}
```

...

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

Extensión de CyclicBehaviour.

```
public class CyclicBehaviourTrain extends CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg = this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUEST));

            if(msg != null) {
                TrainComponents trainComponets = new TrainComponents();
                trainComponets = (TrainComponents) msg.getContentObject();
                ((TrainAgent) this.myAgent).setFile(trainComponets.getDataset());
                ((TrainAgent) this.myAgent).setClassificationMethod(trainComponets.getClassificationMethod());
                this.train();
            }else {
                this.block();
            }
        }
        catch (Exception e) {
            //e.printStackTrace();
        }
    }
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

```
public class CyclicBehaviourTrain extends CyclicBehaviour{  
    private static final long serialVersionUID = 1L;  
  
    public void action() {  
        try {  
            ACLMessage msg = this.myAgent.receive(MessageTemplate.MatchPerformative(ACLMessage.REQUEST));  
  
            if(msg != null) {  
                TrainComponents trainComponets = new TrainComponents();  
                trainComponets = (TrainComponents) msg.getContentObject();  
                ((TrainAgent) this.myAgent).setFile(trainComponets.getDataset());  
                ((TrainAgent) this.myAgent).setClassificationMethod(trainComponets.getClassificationMethod());  
                this.train();  
            }else {  
                this.block();  
            }  
        }  
        catch (Exception e) {  
            //e.printStackTrace();  
        }  
    }  
}
```

Extensión de CyclicBehaviour.

Se extrae los componentes encapsulados en el mensaje.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

```
...
    public void train() {
        try {
            Instances data;
            data = new Instances(new BufferedReader(new
FileReader(((TrainAgent) this.myAgent).getFile())));
            data.setClassIndex(data.numAttributes() - 1);

            Classifier localClassifier = null;
            if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("Random
Forest")) {
                localClassifier = new RandomForest();
            }
            else if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("JRip")) {
                localClassifier = new JRip();
            }
        }
    }
    ...
```

```
...
        else { //Bayes Net
            localClassifier = new BayesNet();
        }

        localClassifier.buildClassifier(data);
        Evaluation evaluation = new Evaluation(data);
        evaluation.crossValidateModel(localClassifier, data, 10,
new Random(1));
        ((TrainAgent)
this.myAgent).setClassifier(localClassifier);

        ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
        msg.setContentObject((Serializable) ((TrainAgent)
this.myAgent).getClassifier());
        msg.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
        this.myAgent.send(msg);
    } catch (Exception e) {
        //e.printStackTrace();
    }
}
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

```
...
public void train() {
    try {
        Instances data;
        data = new Instances(new BufferedReader(new
FileReader(((TrainAgent) this.myAgent).getFile())));
        data.setClassIndex(data.numAttributes() - 1);

        Classifier localClassifier = null;
        if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("Random
Forest")) {
            localClassifier = new RandomForest();
        }
        else if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("JRip")) {
            localClassifier = new JRip();
        }
    }
}
...
```

Carga de instancias y  
selección de la clase.

```
...
else { //Bayes Net
    localClassifier = new BayesNet();
}

localClassifier.buildClassifier(data);
Evaluation evaluation = new Evaluation(data);
evaluation.crossValidateModel(localClassifier, data, 10,
new Random(1));
((TrainAgent)
this.myAgent).setClassifier(localClassifier);

ACLMMessage msg = new ACLMessage(ACLMMessage.INFORM);
msg.setContentObject((Serializable) ((TrainAgent)
this.myAgent).getClassifier());
msg.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
this.myAgent.send(msg);
} catch (Exception e) {
    //e.printStackTrace();
}
}
```



# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

```
...
public void train() {
    try {
        Instances data;
        data = new Instances(new BufferedReader(new
FileReader(((TrainAgent) this.myAgent).getFile())));
        data.setClassIndex(data.numAttributes() - 1);

        Classifier localClassifier = null;
        if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("Random
Forest")) {
            localClassifier = new RandomForest();
        }
        else if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("JRip")) {
            localClassifier = new JRip();
        }
    }
}
...
```

Carga de instancias y selección de la clase.

Configuración del modelo en función de la selección del usuario.

```
...
else { //Bayes Net
    localClassifier = new BayesNet();
}

localClassifier.buildClassifier(data);
Evaluation evaluation = new Evaluation(data);
evaluation.crossValidateModel(localClassifier, data, 10,
new Random(1));
((TrainAgent)
this.myAgent).setClassifier(localClassifier);

ACLMMessage msg = new ACLMessage(ACLMMessage.INFORM);
msg.setContentObject((Serializable) ((TrainAgent)
this.myAgent).getClassifier());
msg.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
this.myAgent.send(msg);
} catch (Exception e) {
    //e.printStackTrace();
}
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

```
...
public void train() {
    try {
        Instances data;
        data = new Instances(new BufferedReader(new
FileReader(((TrainAgent) this.myAgent).getFile())));
        data.setClassIndex(data.numAttributes() - 1);

        Classifier localClassifier = null;
        if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("Random
Forest")) {
            localClassifier = new RandomForest();
        }
        else if(((TrainAgent)
this.myAgent).getClassificationMethod().equals("JRip")) {
            localClassifier = new JRip();
        }
    }
}
...
```

Carga de instancias y selección de la clase.

Configuración del modelo en función de la selección del usuario.

```
...
else { //Bayes Net
    localClassifier = new BayesNet();
}

localClassifier.buildClassifier(data);
Evaluation evaluation = new Evaluation(data);
evaluation.crossValidateModel(localClassifier, data, 10,
new Random(1));
((TrainAgent)
this.myAgent).setClassifier(localClassifier);

ACLMMessage msg = new ACLMessage(ACLMMessage.INFORM);
msg.setContentObject((Serializable) ((TrainAgent)
this.myAgent).getClassifier());
msg.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
this.myAgent.send(msg);
} catch (Exception e) {
    //e.printStackTrace();
}
}
```

Entrenamiento del modelo.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourTrain.

```
...
public void train() {
    try {
        Instances data;
        data = new Instances(new BufferedReader(new
        FileReader(((TrainAgent) this.myAgent).getFile())));
        data.setClassIndex(data.numAttributes() - 1);

        Classifier localClassifier = null;
        if(((TrainAgent)
        this.myAgent).getClassificationMethod().equals("Random
        Forest")) {
            localClassifier = new RandomForest();
        }
        else if(((TrainAgent)
        this.myAgent).getClassificationMethod().equals("JRip")) {
            localClassifier = new JRip();
        }
    }
}
...
```

Carga de instancias y  
selección de la clase.

Configuración del modelo en función  
de la selección del usuario.

```
...
else { //Bayes Net
    localClassifier = new BayesNet();
}

localClassifier.buildClassifier(data);
Evaluation evaluation = new Evaluation(data);
evaluation.crossValidateModel(localClassifier, data, 10,
new Random(1));
((TrainAgent)
this.myAgent).setClassifier(localClassifier);

ACLMMessage msg = new ACLMessage(ACLMMessage.INFORM);
msg.setContentObject((Serializable) ((TrainAgent)
this.myAgent).getClassifier());
msg.addReceiver(new AID("PredictAgent",
AID.ISLOCALNAME));
this.myAgent.send(msg);
} catch (Exception e) {
    //e.printStackTrace();
}
}
```

Entrenamiento del  
modelo.

Envío del  
modelo  
entrenado al  
PredictAgent.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourPredict.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourPredict.

```
public class CyclicBehaviourPredict extends
CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(AC
LMessage.REQUEST));

            if(msg != null) {
                PredictComponets predictComponets =
(PredictComponets) msg.getContentObject();
                Instances data =
predictComponets.getDataset();
                Instance instance =
predictComponets.getInstance();
                data.setClassIndex(data.numAttributes() -
1);

                instance.setDataset(data);
            }
        }
    }
}
```

```
...
        Double prediction = ((PredictAgent)
this.myAgent).getClassifier().classifyInstance(instance);

        ACLMessage msg2 = new
ACLMessage(ACLMessage.INFORM);
        msg2.addReceiver(new AID("ResultsAgent",
AID.ISLOCALNAME));
        msg2.setContentObject(prediction);
        this.myAgent.send(msg2);
    }else {
        this.block();
    }
}
catch (Exception e) {
    e.printStackTrace();
}
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourPredict.

```
public class CyclicBehaviourPredict extends  
CyclicBehaviour{  
    private static final long serialVersionUID = 1L;  
  
    public void action() {  
        try {  
            ACLMessage msg =  
this.myAgent.receive(MessageTemplate.MatchPerformative(AC  
LMessage.REQUEST));  
  
            if(msg != null) {  
                PredictComponets predictComponets =  
(PredictComponets) msg.getContentObject();  
                Instances data =  
predictComponets.getDataset();  
                Instance instance =  
predictComponets.getInstance();  
                data.setClassIndex(data.numAttributes() -  
1);  
                instance.setDataset(data);  
            }  
        }  
    }  
}
```

Se recibe el mensaje.

```
...  
        Double prediction = ((PredictAgent)  
this.myAgent).getClassifier().classifyInstance(instance);  
  
        ACLMessage msg2 = new  
ACLMessage(ACLMessage.INFORM);  
        msg2.addReceiver(new AID("ResultsAgent",  
AID.ISLOCALNAME));  
        msg2.setContentObject(prediction);  
        this.myAgent.send(msg2);  
    }else {  
        this.block();  
    }  
}  
catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourPredict.

```
public class CyclicBehaviourPredict extends  
CyclicBehaviour{  
    private static final long serialVersionUID = 1L;  
  
    public void action() {  
        try {  
            ACLMessage msg =  
this.myAgent.receive(MessageTemplate.MatchPerformative(AC  
LMessage.REQUEST));  
  
            if(msg != null) {  
                PredictComponets predictComponets =  
(PredictComponets) msg.getContentObject();  
                Instances data =  
predictComponets.getDataset();  
                Instance instance  
predictComponets.getInstance();  
                data.setClassIndex(data.numAttributes() -  
1);  
                instance.setDataset(data);  
            }  
        }  
    }  
}
```

Se recibe el mensaje.

Se extrae la instancia y el dataset.

```
...  
        Double prediction = ((PredictAgent)  
this.myAgent).getClassifier().classifyInstance(instance);  
  
        ACLMessage msg2 = new  
ACLMessage(ACLMessage.INFORM);  
        msg2.addReceiver(new AID("ResultsAgent",  
AID.ISLOCALNAME));  
        msg2.setContentObject(prediction);  
        this.myAgent.send(msg2);  
    }else {  
        this.block();  
    }  
}  
catch (Exception e) {  
    e.printStackTrace();  
}  
}
```



# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourPredict.

```
public class CyclicBehaviourPredict extends
CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(AC
LMessage.REQUEST));

            if(msg != null) {
                PredictComponets predictComponets =
(PredictComponets) msg.getContentObject();
                Instances data =
predictComponets.getDataset();
                Instance instance =
predictComponets.getInstance();
                data.setClassIndex(data.numAttributes() -
1);
                instance.setDataset(data);
            }
        }
    }
}
```

Se recibe el mensaje.

Se extrae la instancia  
y el dataset.

Es importante darle un "contexto" a  
la instancia que se quiere predecir.

```
...
        Double prediction = ((PredictAgent)
this.myAgent).getClassifier().classifyInstance(instance);

        ACLMessage msg2 = new
ACLMessage(ACLMessage.INFORM);
        msg2.addReceiver(new AID("ResultsAgent",
AID.ISLOCALNAME));
        msg2.setContentObject(prediction);
        this.myAgent.send(msg2);
    }else {
        this.block();
    }
}
catch (Exception e) {
    e.printStackTrace();
}
}
```



# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourPredict.

```
public class CyclicBehaviourPredict extends  
CyclicBehaviour{  
    private static final long serialVersionUID = 1L;  
  
    public void action() {  
        try {  
            ACLMessage msg =  
this.myAgent.receive(MessageTemplate.MatchPerformative(AC  
LMessage.REQUEST));  
  
            if(msg != null) {  
                PredictComponets predictComponets =  
(PredictComponets) msg.getContentObject();  
                Instances data =  
predictComponets.getDataset();  
                Instance instance  
predictComponets.getInstance();  
                data.setClassIndex(data.numAttributes() -  
1);  
                instance.setDataset(data);  
                ...  
            }  
        }  
    }  
}
```

Se recibe el mensaje.

Se extrae la instancia  
y el dataset.

Es importante darle un "contexto" a  
la instancia que se quiere predecir.

```
...  
        Double prediction = ((PredictAgent)  
this.myAgent).getClassifier().classifyInstance(instance);  
  
        ACLMessage msg2 = new  
ACLMessage(ACLMessage.INFORM);  
        msg2.addReceiver(new AID("ResultsAgent",  
AID.ISLOCALNAME));  
        msg2.setContentObject(prediction);  
        this.myAgent.send(msg2);  
    }else {  
        this.block();  
    }  
}  
catch (Exception e) {  
    e.printStackTrace();  
}  
}
```

Clasificación de  
la instancia.

# Ejemplo de MAS inteligente con Weka

Definición del CyclicBehaviourPredict.

```
public class CyclicBehaviourPredict extends
CyclicBehaviour{
    private static final long serialVersionUID = 1L;

    public void action() {
        try {
            ACLMessage msg =
this.myAgent.receive(MessageTemplate.MatchPerformative(AC
LMessage.REQUEST));

            if(msg != null) {
                PredictComponets predictComponets =
(PredictComponets) msg.getContentObject();
                Instances data =
predictComponets.getDataset();
                Instance instance =
predictComponets.getInstance();
                data.setClassIndex(data.numAttributes() -
1);
                instance.setDataset(data);
            }
        }
    }
}
```

Se recibe el mensaje.

Se extrae la instancia y el dataset.

Es importante darle un "contexto" a la instancia que se quiere predecir.

```
...
        Double prediction = ((PredictAgent)
this.myAgent).getClassifier().classifyInstance(instance);

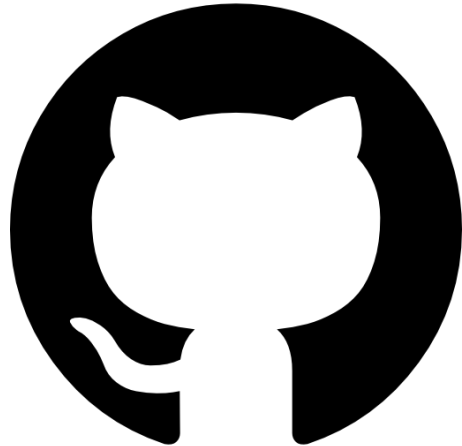
        ACLMessage msg2 = new
ACLMessage(ACLMessage.INFORM);
        msg2.addReceiver(new AID("ResultsAgent",
AID.ISLOCALNAME));
        msg2.setContentObject(prediction);
        this.myAgent.send(msg2);
    }else {
        this.block();
    }
}
catch (Exception e) {
    e.printStackTrace();
}
}
```

Clasificación de la instancia.

Envío de la predicción a ResultsAgent.

# Ejemplo de MAS inteligente con Weka

Este sistema está disponible en un repositorio de GitHub:



[dmjimenezbravo/IntelligentMASPlatform](https://github.com/dmjimenezbravo/IntelligentMASPlatform)

# Índice

1. ¿Qué es JADE?
2. ¿Cómo crear un agente con JADE?
3. ¿Cómo crear un comportamiento con JADE?
4. ¿Cómo se comunican los agentes en JADE?
5. Ejemplo simple de interacción entre agentes
6. Ejemplo de MAS inteligente con Weka
7. **Recursos**
8. Licencia

# Recursos

- Ian H. Witten, Eibe Frank and Mark A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series, 2017.
- [Weka courses](#).
- [Weka wiki](#).
- Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood. Developing Multi-Agent Systems with JADE. John Wiley & Sons, Ltd, 2007.
- [JADE API](#).

# Índice

1. ¿Qué es JADE?
2. ¿Cómo crear un agente con JADE?
3. ¿Cómo crear un comportamiento con JADE?
4. ¿Cómo se comunican los agentes en JADE?
5. Ejemplo simple de interacción entre agentes
6. Ejemplo de MAS inteligente con Weka
7. Recursos
8. **Licencia**

# Licencia

Usted es libre de:

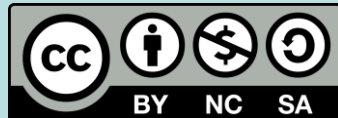


copiar, distribuir y comunicar públicamente la obra.



hacer obras derivadas.

Bajo las siguientes condiciones:







Más información en <https://creativecommons.org/licenses/by-nc-sa/4.0/>.



# Creación de sistemas multiagentes inteligentes con JADE



-  Sistemas Inteligentes
-  Grado en Ingeniería Informática
-  Diego M. Jiménez Bravo
-  [djimenez@fi.upm.es](mailto:djimenez@fi.upm.es)
-  13 de mayo de 2022