



# UNIVERSIDAD DE GRANADA

## GRADO EN INGENIERÍA INFORMÁTICA

DESARROLLO BASADO EN AGENTES  
CURSO 2023 / 2024

---

### Memoria – Práctica 3

*Comunicación entre agentes.*

---

Cano Flores, Julia María  
Muñoz Gómez, Sergio  
Reyes García, Teresa Fernanda  
Rincón Otero, Marta  
Velázquez Ortuño, Diego

20 de diciembre, 2023

# ÍNDICE

---

<b>1. Planificación del proyecto.</b>	<b>2</b>
<b>2. Estructura del proyecto.</b>	<b>4</b>
Protocolo de comunicación	5
Diagrama de clases	8
<b>3. Interfaz Gráfica</b>	<b>9</b>

---

Para el desarrollo del proyecto hemos hecho uso de GitHub como plataforma para la gestión grupal de nuestro código ya que es una plataforma muy cómoda para la colaboración en equipo. Hemos hecho uso de diferentes ramas para las funcionalidades del proyecto como es el entorno, mapa, agente, etc. Además, hemos usado un README para hacer una descripción breve del proyecto.

- **GitHub:** <https://github.com/teresitarg22/Desarrollo-Agentes>
- **ID\_GRUPO:** 206

## 1. Planificación del proyecto.

El **Product Backlog** es una lista priorizada que contiene todas las funcionalidades y tareas que necesitamos para la creación de nuestro producto final. Esta lista nos permite organizar y planificar de manera eficiente todo lo que debemos implementar en el proyecto, lo que resulta fundamental para una gestión efectiva del tiempo que disponemos.

PRODUCT BACKLOG			
1	Diseño inicial del proyecto.		
	División en tareas.		
	ID	Descripción	Estimación
	1.1	Diseño conceptual de la estructura de clases.	1 <sub>h</sub>
2	1.2	Reparto y asignación de tareas.	40 <sub>min</sub>
	Modificación y adaptación de clases a este proyecto.		
	División en tareas.		
	ID	Descripción	Estimación
3	2.1	Revisión de la clase 'Mapa'	30 <sub>min</sub>
	2.2	Modificación de la clase 'MapaGUI'	1 <sub>h</sub>
	2.3	Modificación de la clase 'Entorno'	1 <sub>h</sub>
	2.4	Adaptación del main al proyecto.	30 <sub>min</sub>
	Creación de los agentes.		
	División en tareas.		
	ID	Descripción	Estimación
	3.1	Creación del agente 'Buscador'	15 <sub>min</sub>
	3.2	Creación del agente 'SantaClaus'	15 <sub>min</sub>
	3.3	Creación del agente 'Rudolph'	15 <sub>min</sub>
	3.4	Implementación de los métodos de cada agente.	30 <sub>min</sub>




4	Modificación y adaptación de los comportamientos.		
	División en tareas.		
	ID	Descripción	Estimación
	4.1	Adaptación de las interfaces y enumerado.	30 <sub>min</sub>
	4.2	Modificación del comportamiento 'Mover'	30 <sub>min</sub>
	4.3	Implementación del movimiento de los agentes.	1 <sub>h</sub>
	4.4	Revisión del comportamiento 'Visión'	30 <sub>min</sub>
5	4.5	Corrección de errores performativas FIPA y mejora de implementación.	3 <sub>h</sub>
	Desarrollador		
	Marta		
	Teresa		
	Diego		
	Diego		
	Diego Marta Teresa		
6	Implementación del protocolo de movimiento FIPA.		
	División en tareas.		
	ID	Descripción	Estimación
	5.1	Diseñar el protocolo de movimiento.	1 <sub>h</sub>
	5.2	Implementar las performativas FIPA correspondientes.	2 <sub>h</sub>
	5.3	Creación de un diagrama visual del protocolo.	1 <sub>h</sub>
	Desarrollador		
7	Teresa		
	Teresa Sergio Julia		
	Teresa		
	Pruebas y documentación.		
	División en tareas.		
	ID	Descripción	Estimación
	6.1	Redacción de la memoria del proyecto.	3 <sub>h</sub>
8	6.2	Creación de la presentación del proyecto.	2 <sub>h</sub>
	6.3	Realizar pruebas de funcionamiento y depuración.	2 <sub>h</sub>
	Desarrollador		
	Teresa		
	Teresa Sergio Julia		
	Todos		

## 2. Estructura del proyecto.

Contamos con varios elementos principales en este proyecto: los agentes, el entorno y la vista. Para las clases **agentes**, hemos implementado tres comportamientos de comunicación específicos que son ComunicacionBuscador, ComunicacionSantaClaus y ComunicacionRudolph, que implementan el protocolo de comunicación FIPA y el paso de mensajes entre los agentes. Además, contamos con el comportamiento de movimiento Moverse que fusiona los tres comportamientos de la práctica anterior. Por otro lado, la clase **entorno** proporciona una representación del entorno en el que el agente realiza sus acciones en el mundo bidimensional. Cuenta con variables como los sensores, que almacenan información sobre las celdas adyacentes al agente, el mapa o las posiciones del agente y el objetivo. Finalmente, hemos implementado una **interfaz gráfica** que muestra el comportamiento de los agentes durante la realización de la misión.

Al igual que la práctica anterior, hemos hecho uso de **interfaces** para escuchar eventos relacionados con la interfaz gráfica. Contamos con un listener para el entorno y otro para el main. Por otro lado, para la gestión de los posibles movimientos disponibles para el agente (contando con diagonales), en un entorno bidimensional, hemos hecho uso de un **enumerado**. El **directorio de mapas** contiene archivos de texto que representan diferentes configuraciones de mapas, desde entornos sin obstáculos hasta entornos con obstáculos complejos.

En el marco de la práctica realizada, hemos desarrollado un sistema multiagente que involucra la interacción de tres agentes principales: 'Buscador', 'Santa Claus' y 'Rudolph'. El **objetivo** de esta práctica es la comunicación entre estos agentes para llevar a cabo una misión específica, recuperar los renos de Santa Claus para que estén listos para la Navidad.

	El <b>agente Buscador</b> es el encargado de la misión, si Santa piensa que es apto. Recorrerá el entorno con las indicaciones de Rudolph para encontrar a todos los renos que necesitan ser rescatados.
	El <b>agente Santa Claus</b> es el encargado de valorar si el buscador es apto para la misión y quedará a la espera de recibir sus renos de vuelta, con lo que contestará HoHoHo!
	El <b>agente Rudolph</b> es el encargado de proporcionar al Buscador las coordenadas de los renos perdidos hasta que no queden más por buscar.

## Protocolo de comunicación

Hemos establecido un **protocolo de comunicación** basado en los estándares FIPA (Foundation for Intelligent Physical Agents) que realiza lo siguiente:

### 1. Propuesta del Buscador a Santa Claus.

El agente Buscador hace una performativa **PROPOSE** a Santa preguntando si es apto para la misión. Pensamos que es correcto hacer uso de esta performativa porque se va a realizar una negociación entre los agentes donde Santa evalúa si el Buscador es apto (80%) o no apto (20%).

### 2. Posibles respuestas de Santa Claus ante la propuesta.

Ante la propuesta, Santa Claus tiene dos posibles respuestas:

- **ACCEPT\_PROPOSAL** – Santa Claus acepta la propuesta y proporciona el código secreto al Buscador. Además informa a Rudolph, mediante la performativa **INFORM**, de que se va a llevar a cabo la misión.
- **REJECT\_PROPOSAL** – Santa Claus rechaza la propuesta. El Buscador realiza un `'doDelete()'`. Además informa a Rudolph, mediante la performativa **INFORM**, de que no se realiza la misión y ambos agentes restantes realizan un `'doDelete()'` también.

### 3. Comunicación con Rudolph.

Si se acepta la performativa anterior, el Buscador realiza un **REQUEST** a Rudolph pasándole como ConversationID el código secreto, que solo conocen Santa y Rudolph, para que le de las coordenadas del reno a rescatar. Pensamos que el uso de esta performativa es lo mejor ya que el emisor le está pidiendo al receptor que ejecute una acción, en este caso el Buscador solicita las coordenadas de un reno a Rudolph.

- **AGREE** – Rudolph confirma que el código es correcto y proporciona las coordenadas del siguiente reno a rescatar. Rudolph mantiene un contador de renos por rescatar, por lo que cada vez que pasa unas coordenadas este contador disminuye. Usamos esta performativa ya que el receptor acepta ejecutar la acción.

Si la respuesta por parte de Rudolph es **AGREE**, el Buscador se desplaza a las coordenadas indicadas, rescata al reno y vuelve a Rudolph. Se repite el proceso hasta que no hayan más renos por rescatar.

- **INFORM** – Rudolph informa al Buscador de dos posibles situaciones...

- Si la respuesta es **INFORM** con "*Código incorrecto*", el Buscador realiza un `doDelete()` y se borran todos los agentes.
- Si la respuesta es **INFORM** con "*No quedan más renos*", el Buscador realiza un **REQUEST** a Santa para obtener sus coordenadas.

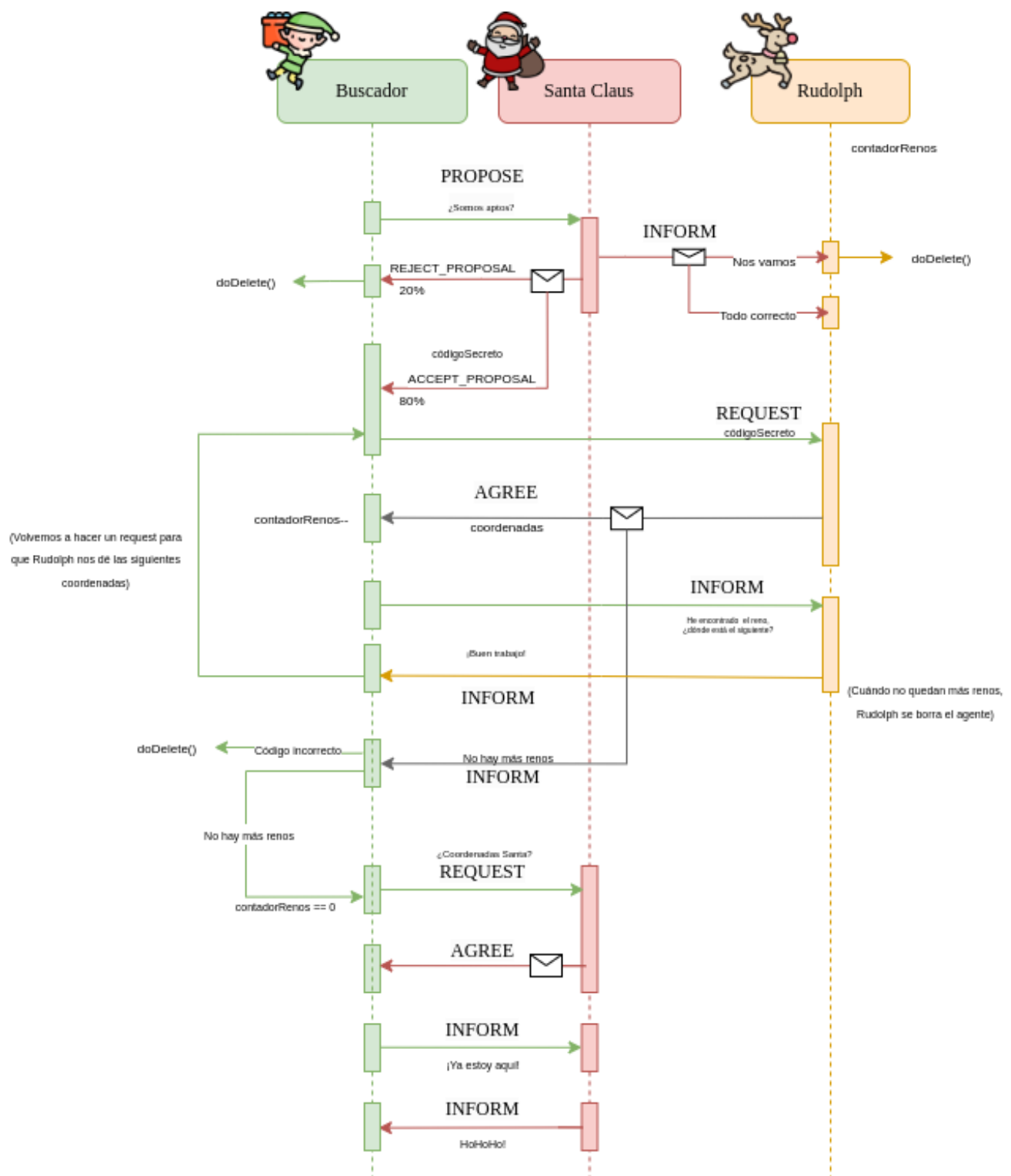
Utilizamos esta performativa ya que el objetivo es pasar información.

#### 4. Comunicación final con Santa Claus.

Tras el **REQUEST** del Buscador solicitando las coordenadas de Santa, este se quedará a la espera de recibir una respuesta con las mismas. Santa Claus enviará una respuesta **AGREE** con sus coordenadas y quedará a la espera. Volvemos a usar esta performativa ya que el emisor le está pidiendo al receptor que ejecute una acción, en este caso el Buscador solicita las coordenadas de Santa.

- El Buscador se mueve hasta la posición e informa a Santa Claus con un **INFORM** de que ha llegado a su posición.
- Finalmente, Santa Claus responde con un **INFORM** "HoHoHo!", para indicar el éxito de la misión.

Utilizamos esta performativa ya que el objetivo es pasar información.

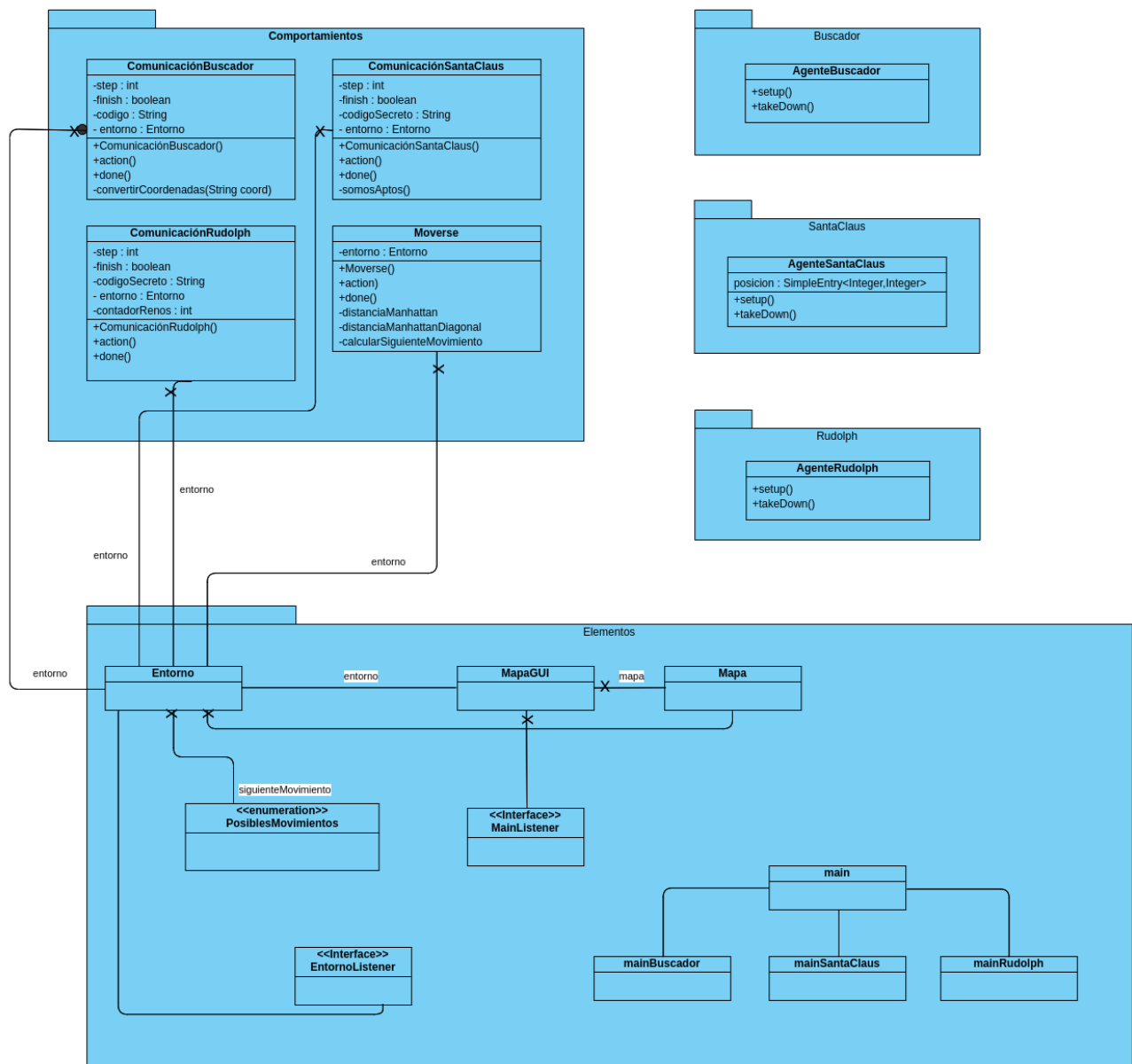


\* Se pondrá a disposición del profesor el archivo PNG del diagrama de comunicación para una mejor visualización del contenido.



## Diagrama de clases

El **diagrama de clases** es una herramienta que nos va a permitir representar visualmente la estructura de clases que hemos seguido en nuestro proyecto, identificando sus clases (y cómo las clases se relacionan entre ellas), métodos y atributos.



\* Se pondrá a disposición del profesor el archivo PNG del diagrama de clases para una mejor visualización del contenido.

### 3. Interfaz Gráfica

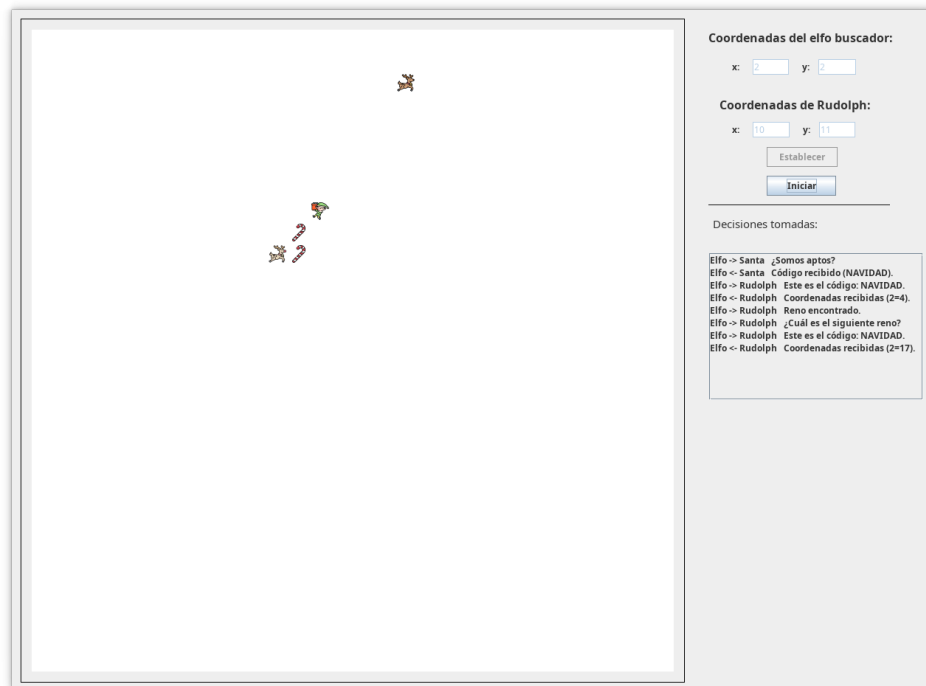
La **interfaz** sigue el prototipo de malla de celdas por las que pasa el agente buscador, pero se diferencia en la visualización ya que será un mapa en blanco sin líneas divisorias entre celdas simulando un campo de nieve.

The screenshot shows a graphical user interface for a search simulation. On the left is a large, empty rectangular area representing the map. To the right of the map are two control panels. The top panel, titled 'Coordenadas del elfo buscador:', contains input fields for 'x:' and 'y:'. The bottom panel, titled 'Coordenadas de Rudolph:', also contains input fields for 'x:' and 'y:', followed by 'Establecer' and 'Iniciar' buttons. Below these panels is a section labeled 'Decisiones tomadas:' with an empty rectangular box for displaying results.

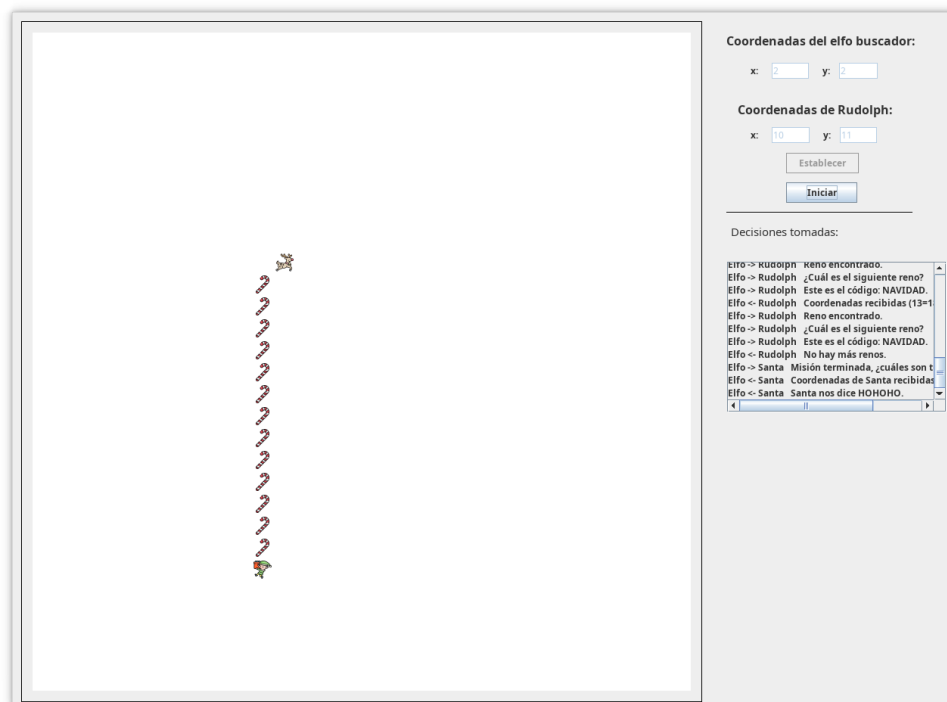
Podremos establecer la posición del buscador y de Rudolph, mientras que la posición de Santa Claus se definirá aleatoriamente y se dará a conocer al momento de solicitar sus coordenadas.

This screenshot shows the same GUI as the previous one, but with visual elements added to the map. A small icon of Santa Claus is positioned in the top-left corner of the map area, and a small icon of Rudolph is positioned in the center. The 'Coordenadas del elfo buscador:' panel now shows 'x:' with the value '2' and 'y:' with the value '2'. The 'Coordenadas de Rudolph:' panel shows 'x:' with the value '10' and 'y:' with the value '11'. The 'Establecer' and 'Iniciar' buttons remain visible below the input fields.

Cuando el buscador se mueva por el mapa irá soltando caramelos indicando el camino por el que ha pasado, limpiándose este camino cada vez que el buscador le dice a Rudolph que ha encontrado un reno.



Finalmente, cuando el buscado ha terminado la misión, le solicita a Santa Claus su ubicación y se dirige hacia ella.



Como se puede observar, a modo de diario de acciones, se indica en el panel de decisiones tomas las comunicaciones que se realizan entre el agente buscador y los agentes Rudolph y Santa Claus.