

# PRÁCTICA 1: SISTEMAS DE PRODUCCIÓN

---

Mohamed El aakil Mesaoudi ([100405967@alumnos.uc3m.es](mailto:100405967@alumnos.uc3m.es))

Raúl Gimenez de Dios ([100405861@alumnos.uc3m.es](mailto:100405861@alumnos.uc3m.es))



# ÍNDICE

<b>INTRODUCCIÓN</b>	<b>3</b>
<b>MANUAL TÉCNICO</b>	<b>4</b>
Ontología	4
Reglas	5
Hechos	6
<b>MANUAL DE USUARIO</b>	<b>7</b>
SELECCIONAR JUEGO Y PUNTUACIÓN	7
SELECCIONAR PERSONALIDAD DEL PACIENTE	7
<b>PRUEBAS REALIZADAS</b>	<b>8</b>
ANÁLISIS DE RESULTADOS	8
<b>CONCLUSIONES</b>	<b>9</b>
<b>COMENTARIOS PERSONALES</b>	<b>9</b>

## INTRODUCCIÓN

En este documento se propone una solución al problema de la interacción entre un paciente y un robot NAO donde se llevarán a cabo una serie de juegos donde se deberá tomar acción dependiendo de las actitudes, personalidad y comportamientos del paciente. Para ello se usará el lenguaje CLIPS que nos permitirá ejecutar el sistema deseado.

Durante el documento se detalla el manual técnico donde aparecerá la ontología propuesta, detallando cada una de las clases, reglas, hechos e instancias que componen la solución. Posteriormente, se presentará el manual de usuario que dará las indicaciones para ejecutar el programa como se desee.

Más tarde, se describirán las pruebas que se han realizado para validar el sistema donde se podrán apreciar las distintas personalidades del paciente en ejecución y los distintos juegos programados.

Finalmente, se concluirá el documento con nuestras opiniones personales y técnicas sobre el trabajo realizado.

## MANUAL TÉCNICO

En este apartado se explicará el diseño de clases de la ontología, con sus atributos y posibles valores, por otro lado se especificará la funcionalidad de cada una de las reglas, y finalmente se describirán los hechos utilizados y la utilidad de los mismos.

### Ontología

Concepto	Características	Valores
JUGADOR	Nombre	String
	Puntuación	Integer
	Turno	allowed-values [1 - 2]
	Posicion	Integer
ROBOT (is-a JUGADOR)	-	
PACIENTE (is-a JUGADOR)	Personalidad	allowed-values [normal tramposo agresivo impaciente]
	Personalidad-controlada	allowed-values [si no]
JUEGO	Puntuación_maxima	Integer
	Turno	allowed-values [1 - 2]
	Ronda	Integer
	Juego-actual	allowed-values [trilleros rayuela ninguno]

## Reglas

### Personalidad

controlar-impaciente: Reduce la puntuación máxima de los juegos en una unidad para que los juegos sean más cortos y amenos para el niño, se ejecuta al principio del juego en el caso de que el niño tenga una personalidad impaciente.

controlar-tramposo: Se ejecuta cuando el niño va ganando por un punto y tiene una personalidad de tipo tramposo, el robot escarmienta al niño y le penaliza restándole un punto.

controlar-agresivo: Se ejecuta cuando el niño va perdiendo por un punto y tiene una personalidad de tipo agresivo, este no acepta ir perdiendo y golpea al robot, este le escarmienta y le penaliza restándole un punto.

### Rayuela

jugador-lanza-piedra-dentro: El jugador que es su turno lanza la piedra y esta cae en una casilla aleatoria. Esto activa los hechos de la dirección de movimiento a IDA y la posición de la piedra.

jugador-lanza-piedra-fuera: El jugador que es su turno lanza la piedra y esta cae fuera de las casillas o en una línea. Esto hace que pierda su turno.

jugador-va: Si la dirección del jugador es ida y no hay ninguna piedra en la posición siguiente, el jugador aumenta su valor de posición en una unidad.

jugador-esquiva-piedra: Si la dirección del jugador es ida y hay una piedra en la posición siguiente, éste la salta aumentando el valor de su posición en dos unidades.

jugador-pisa-piedra: Si la dirección del jugador es ida y hay una piedra en la posición siguiente, éste la pisa y pierde su turno, volviendo a la posición inicial.

jugador-debe-volver: Cuando un jugador llega a la posición 9 del tablero, da la vuelta cambiando su dirección y saltando a la posición 8.

jugador-vuelve: Si la dirección del jugador es vuelta y no hay ninguna piedra en la posición siguiente, el jugador disminuye su valor de posición en una unidad.

jugador-rocoge-piedra: Si la dirección del jugador es vuelta y hay una piedra en la posición siguiente, el jugador disminuye su valor de posición en una unidad y recoge la piedra.

jugador-completa-ronda: Cuando un jugador con la dirección vuelta llega a la posición 0, eso significa que ha acabado el recorrido, lo cual aumenta su puntuación en 1 y cambia el turno al otro jugador.

### Trilleros

eleccion-rojo-jugador: Si el juego es trilleros, el jugador elige un vaso que contiene una bola roja, lo que hace que pierda un punto y cambie de turno.

eleccion-verde-jugador: Si el juego es trilleros, el jugador elige un vaso que contiene una bola verde, lo que hace que gane un punto y cambie de turno.

eleccion-vacia-jugador: Si el juego es trilleros, el jugador elige un vaso que está vacío, no ocurre nada y vuelve a repetir la elección.

jugador-gana: Cuando un jugador llega a la puntuación máxima, ha ganado el juego y acaba la simulación.

Para la generalización del código se ha procedido a colapsar todas las reglas de robot y paciente en una única de jugador. Para solventar los problemas relacionados con la alternancia de los turnos se han utilizado dos hechos (alternativa robot paciente) (alternativa paciente robot).

### Hechos

**Hecho (casilla [A] [B])** donde [A] puede tomar los valores *dentro* y *fuera*, y donde [B] puede tomar los valores 1, 2, 3, 4, 5, 6, 7, 8 para *dentro* y *línea* y *fuera* para *fuera*. Sirve para representar las posibles posiciones de la piedra en el juego de la rayuela.

**Hecho (bola [A] [B])** donde [A] puede tomar los valores *rojo* y *verde* y donde [B] puede tomar los valores 1, 2 y 3. Sirve para representar las posibles combinaciones de bolas debajo de los vasos.

**Hecho (elección [A][B])** donde [A] puede tomar los valores *robot* y *paciente* y donde [B] puede tomar los valores 1, 2 y 3. Sirve para representar las elecciones de cada jugador.

**Hecho (alternativa robot paciente) y hecho (alternativa paciente robot)**, que se utilizan para alternar el turno entre ambos personajes.

## **MANUAL DE USUARIO**

### **SELECCIONAR JUEGO Y PUNTUACIÓN**

Para seleccionar el juego nos deberemos ir a las instancias, específicamente a la instancia que se corresponde con el juego, y en el atributo denominado juego-actual pondremos el valor del juego al que deseamos jugar, dichos valores son: rayuela y trilleros. En esta misma instancia también se puede cambiar la puntuación máxima del juego, siendo de 6 para el juego de trilleros y 3 para el juego de la rayuela.

### **SELECCIONAR PERSONALIDAD DEL PACIENTE**

Para determinar la personalidad del paciente debemos irnos nuevamente al apartado correspondiente a las instancias del juego y tendremos que modificar el atributo personalidad en la instancia de paciente otorgando uno de los siguientes valores: tramposo, agresivo, impaciente y normal.

## PRUEBAS REALIZADAS

A continuación se presenta una tabla con las distintas configuraciones que se han probado:

PRUEBAS	PERSONALIDAD	JUEGO	VERIFICADO
Prueba 1	normal	trilleros	si
Prueba 2	tramposo	trilleros	si
Prueba 3	agresivo	trilleros	si
Prueba 4	impaciente	trilleros	si
Prueba 5	normal	rayuela	si
Prueba 6	tramposo	rayuela	si
Prueba 7	agresivo	rayuela	si
Prueba 8	impaciente	rayuela	si

## ANÁLISIS DE RESULTADOS

Se verifica que el flujo del juego varía en función de la personalidad del niño, siendo así que el robot responde adecuadamente a estas corrigiendo dichos comportamientos. Por otro lado vemos que el juego que se ejecuta también se realiza correctamente sin ninguna traba en su ejecución de acuerdo a las distintas posibles personalidades. Sin embargo pueden darse casos donde no se llegue a disparar las reglas que controlan el comportamiento del niño debido a que no se cumplan las precondiciones, esto por ejemplo puede ser el caso de que el niño sea agresivo pero que siempre gane, por tanto, no se enfadará y no se disparará la regla. Se considera que este planteamiento tiene sentido debido a que en una situación real la personalidad del niño no siempre tiene que afectar a la sesión sino que afectará dadas unas condiciones específicas. Por otro lado, se permite que tanto el paciente como el robot puedan tener una puntuación negativa, es decir, si alguno de los dos tiene una puntuación de 0 y pierde, la puntuación consecuentemente disminuirá a -1, sin embargo, esto podría controlarse fácilmente en las reglas si fuese necesario.



## CONCLUSIONES

Se trata de un proyecto interesante que nos ha permitido llevar más allá los conocimientos adquiridos de CLIPS y de adquisición del conocimiento, conceptualización e implementación de la jerarquía de clases final. Si bien es cierto que el manejo de clips es un poco arcaico y que en ocasiones se puede hacer engorroso, se ha manejado de manera satisfactoria.

## COMENTARIOS PERSONALES

Debido a la complejidad del problema se ha decidido seguir un proceso de implementación gradual con una progresiva implementación de funcionalidades, desde el diseño modular de los juegos, pasando por la implementación de las personalidades y finalmente generalizando las reglas que no contenían condiciones específicas. Esto ha hecho el proceso más llevadero y que ha acabado resultando en un SBC funcional.