



# Modelo de Casos de Uso

26 de noviembre de 2006

Versión C1.0.1

## Historia de revisiones

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
24/02/2006	E1.0.1	Primer versión del modelo de Casos de Uso del sistema FIBRA.	grupo FIBRA
02/03/2006	E1.0.2	Primer revisión del documento.	grupoFIBRA
03/02/2006	E1.0.3	Ultima revisión del documento.	grupo FIBRA
31/03/2006	C1.0.1	Corrección de Referencias	grupo FIBRA



Facultad de Ingeniería

Universidad de la República Oriental del Uruguay



# Índice

<b>1. Introducción</b>	<b>5</b>
1.1. Propósito . . . . .	5
1.2. Alcance . . . . .	5
1.3. Definiciones, Acrónimos y Abreviaciones . . . . .	5
<b>2. Actores del sistema</b>	<b>5</b>
2.1. Simulador SimuroSot . . . . .	5
<b>3. Casos de Uso</b>	<b>5</b>
3.1. Devolver velocidades . . . . .	5
3.1.1. Descripción . . . . .	5
3.1.2. Pre-condiciones . . . . .	5
3.1.3. Post-condiciones . . . . .	6
3.1.4. Flujo de eventos principal . . . . .	6
3.1.5. Flujo de eventos alternativo . . . . .	6



# 1. Introducción

## 1.1. Propósito

Se presentan los casos de uso que se desarrollan para cumplir con las funcionalidades de software del sistema descritas en el documento de Especificación de Requerimientos de Software [2]. Dado que se ha identificado una única funcionalidad principal en el documento mencionado, este documento presenta un único caso de uso que representa la realización de dicha funcionalidad.

## 1.2. Alcance

Se presentan los actores y los casos de uso de todo el sistema FIBRA. En este caso, el conjunto de actores comprende a un único actor que será el simulador SimuroSot, el cual disparará la funcionalidad del sistema. Cabe aclarar que el caso de uso sólo comprende la funcionalidad incluida dentro del alcance definido para el sistema, el cual se detalla en el documento de Alcance del Sistema [1].

## 1.3. Definiciones, Acrónimos y Abreviaciones

1. SimuRoSoT : Simulated Robot Soccer Tournament - Sumilador que utilizará el sistema a construir.
2. RoboSoccer : Robot Soccer - Fútbol de Robots.
3. DLL : Dynamic Link Library

# 2. Actores del sistema

## 2.1. Simulador SimuroSot

Como se indica en el documento [2], el sistema será utilizado por el simulador SimuroSot a través de una interfaz definida por dicho simulador. Éste será el único actor del sistema y accederá a la única funcionalidad del mismo.

# 3. Casos de Uso

A continuación se describen los casos de uso del sistema FIBRA.

## 3.1. Devolver velocidades

### 3.1.1. Descripción

Se reciben del simulador los datos que describen el estado actual del juego (*Environment*). Esta información contiene, entre otros datos, las posiciones de los robots (propios y oponentes), la posición de la pelota, quién posee la pelota actualmente, etc. En base a esta información, la estrategia del sistema calcula las mejores acciones a llevar a cabo para las condiciones dadas y define las velocidades que se deben asignar a cada rueda de cada robot propio para lograr dichas acciones. Se retorna al simulador el conjunto de velocidades que se asignan a las ruedas de cada robot.

### 3.1.2. Pre-condiciones

Este caso de uso será invocado a través de una interfaz definida. El acceso al sistema es a través de una DLL escrita en C++ como se detalla en [2]. Este caso de uso es disparado al invocarse el método *void Strategy ( Environment \*env )*. El simulador permanece bloqueado hasta que el sistema procese

la información contenida en el parámetro *env* y retorne las velocidades de las ruedas de cada robot, también a través del parámetro *env*.

Las posiciones de los objetos están indicadas en pulgadas.

### 3.1.3. Post-condiciones

Las velocidades de las ruedas de cada robot son devueltas en el parámetro *env* de la operación invocada por el simulador (*void Strategy ( Environment \*env )*), la cual dispara este caso de uso. Los valores de dichas velocidades pueden tomar valores entre -125 y 125.

### 3.1.4. Flujo de eventos principal

Dado que el sistema realizará una tarea de reconocimiento de patrones en la primera parte del juego y luego utilizará dicha información para adaptar su juego al del equipo contrario, el comportamiento del sistema se divide en dos etapas bien diferenciadas:

- La etapa de relevamiento de meta-información a partir del reconocimiento de patrones en las jugadas de ataque del adversario. Durante esta etapa, la estrategia no utiliza meta-información del juego del adversario para tomar decisiones.
- La etapa de predicción del juego del adversario. Durante esta etapa, la estrategia utiliza la meta-información relevada durante la primera parte del juego para intentar predecir el comportamiento del adversario.

El cambio de etapa se determina utilizando algún criterio basado en el estado del juego (por ejemplo, si el primer tiempo ha concluído). Por lo tanto, el flujo principal de este caso de uso puede tener dos comportamientos según el estado del juego.

1. Recibir información del estado de juego.
2. Determinar la etapa a partir de algún criterio basado en el estado de juego.
3. Si el sistema se encuentra en la etapa de relevamiento de meta-información:
  - a) Enviar información al módulo de reconocimiento de patrones.
  - b) Enviar información a la estrategia que no utiliza meta-información para tomar decisiones.
4. Si el sistema ha concluído la etapa de relevamiento de meta-información:
  - a) Enviar información a la estrategia que utiliza meta-información para tomar decisiones.
5. Obtener resultado de la estrategia correspondiente.
6. Retornar al simulador las velocidades de las ruedas de cada robot.

### 3.1.5. Flujo de eventos alternativo

En algunos casos podría decidirse que no se desea utilizar la meta-información obtenida en la primera etapa, más allá de que el relevamiento haya finalizado. En este caso el punto 4a del flujo principal se vería modificado, realizando la invocación a la estrategia del punto 3b.

4. Si el sistema ha concluído la etapa de relevamiento de meta-información
  - b) Enviar información a la estrategia que no utiliza meta-información para tomar decisiones.

## Referencias

- [1] Gustavo Armagno, Facundo Benavides, and Claudia Rostagnol. Alcance del sistema. Technical report, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay, 2006. 5
- [2] Gustavo Armagno, Facundo Benavides, and Claudia Rostagnol. Especificación de requerimientos de software. Technical report, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay, 2006. 5