

## Guía 1: Introducción al RoboSoccer y programación

El nombre "RoboSoccer" proviene de la unión de las palabras "Robot" y "Soccer". Esta disciplina consiste en el diseño y programación de máquinas autónomas capaces de jugar fútbol. Cada equipo de RoboSoccer posee múltiples jugadores cuyas características varían dependiendo de la liga en la que compiten.



Figura 1: Partido de RoboSoccer Small Size League.

Los robots utilizados en RoboSoccer están equipados con sensores, actuadores y algoritmos que les permiten percibir su entorno, tomar decisiones y actuar de manera autónoma durante el juego. Estos robots se comunican entre sí a través de un ordenador central que les proporciona información sobre el estado del juego y el entorno en tiempo real.

Esta disciplina se origina como una iniciativa para impulsar la investigación en robótica e inteligencia artificial. El fútbol ofrece un entorno complejo y cambiante que plantea problemas desafiantes para los robots y promueve el desarrollo de nuevas tecnologías.

La RoboCup, una de las mayores organizaciones de RoboSoccer, estableció la meta de que, para mediados del siglo XXI, un equipo de jugadores robóticos humanoides completamente autónomos sea capaz de ganar un partido de fútbol contra el equipo ganador de la última Copa del Mundo de la FIFA. Esta meta es conocida como el "Desafío RoboCup 2050".



Figura 2: Primera competencia RoboCup realizada en la IJCAI-97.

El juego de RoboSoccer se basa en las reglas del fútbol tradicional, realizando pequeñas adaptaciones con el fin de acomodar las necesidades de cada robot, dependiendo de sus características. Cada equipo está compuesto por 11 jugadores autónomos, aunque este número puede variar en algunas competencias.

En resumen, RoboSoccer combina el fútbol y la robótica, desafiando a los investigadores y desarrolladores a crear robots autónomos capaces de jugar fútbol de manera competente. A través de competencias y eventos, se promueve la colaboración y la competencia entre diferentes áreas tecnológicas, fomentando la innovación y el desarrollo de nuevas tecnologías en robótica e inteligencia artificial.

### 1.1 Ligas y asociaciones de RoboSoccer a nivel internacional.

En el contexto del RoboSoccer, existen diversas organizaciones y ligas que impulsan la competencia y el desarrollo en esta disciplina. Una de estas organizaciones es la Federación de Deportes Internacionales Asociación (Federation of International Sports Association - FIRA), que ha desempeñado un papel importante en la promoción y coordinación de competencias de RoboSoccer en diferentes categorías. Sin embargo, en este taller nos centraremos en la RoboCup, una de las principales y reconocidas organizaciones en el campo del RoboSoccer. La RoboCup se compone de varias ligas de fútbol robótico, cada una con sus propias características y desafíos. Estas ligas se dividen en categorías según el tipo de robots utilizados y las reglas específicas del juego. Las ligas pertenecientes a la categoría de fútbol robótico son:

#### 1.1.1 Liga de robots pequeños (Small Size League):

Conocida como la categoría más antigua de la copa, en ella se enfrentan dos equipos, cuya cantidad de jugadores se define dependiendo de la división en la que se participe (División A: máximo de 11 jugadores, División B: máximo de 6 jugadores), donde cada robot debe poseer un diámetro no superior a los 180 milímetros y una altura máxima de 15 centímetros. Esta disciplina se enfoca en la cooperación entre múltiples agentes y el control dentro de un ambiente altamente dinámico, ya que, debido a su tamaño, los robots utilizados son capaces de moverse a altas velocidades.



Figura 3: Robots Small Size League RoboCup.

### 1.1.2 Liga de Robots Medianos (Middle Size League)

En esta categoría, 5 robots autónomos por equipo compiten en un campo de 22 x 14 metros, utilizando un balón de tamaño estándar para la Federación Internacional de Fútbol Asociación (FIFA). Cada equipo es libre de diseñar hardware apropiado, pero cada sensor debe formar parte de la estructura general del robot y deben cumplir con un tamaño y peso estandarizado. El punto más importante para esta clase de competencia es generar habilidades para el diseño mecatrónico, cooperación y control de múltiples agentes de planta y percepción de niveles.

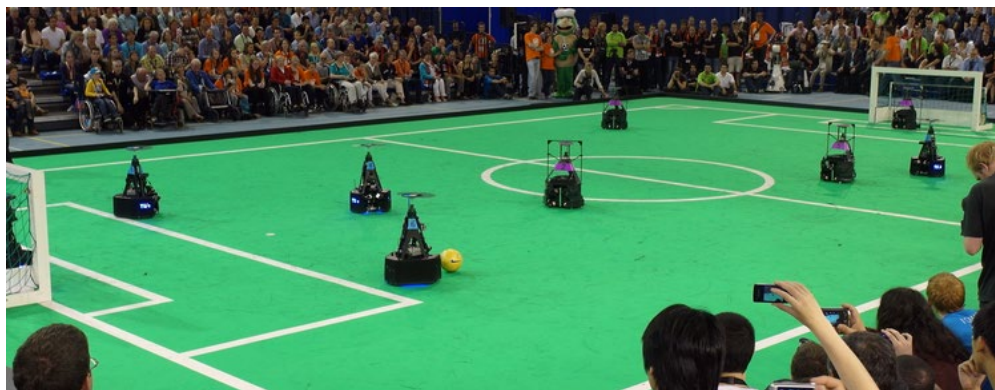


Figura 4: Partido Liga Middle Size RoboCup.

### 1.1.3 Liga de Robots Humanoides (Humanoid League):

En esta liga, robots autónomos con un plan corporal y sentidos similares a los humanos juegan al fútbol unos contra otros. Los robots humanoides de esta liga, solo pueden implementar sensores que simulen el alcance humano. Además de los enfrentamientos de fútbol, se llevan a cabo desafíos técnicos, como caminar, correr y patear el balón de forma dinámica manteniendo el equilibrio, la percepción visual del balón, de otros jugadores y del campo, la auto localización y el juego, entre otros. Un partido lo jugarán dos equipos, cada uno de los cuales estará formado por un máximo de dos jugadores.



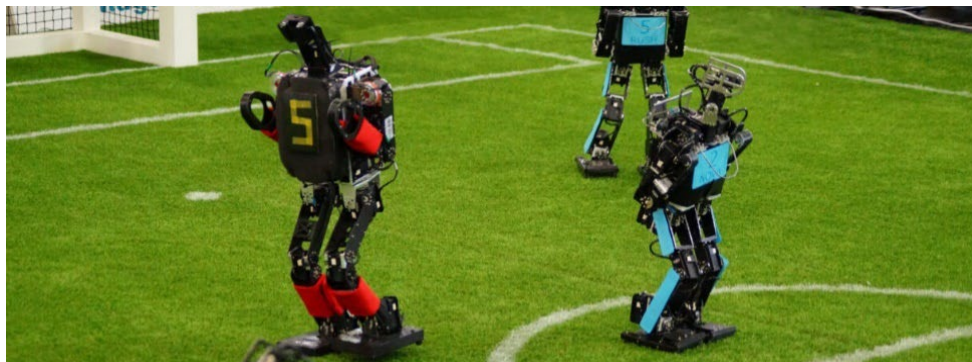


Figura 5: Partido Liga Humanoide RoboCup.

#### 1.1.4 Plataforma Estandarizada (Standard Platform League):

Esta liga consiste en equipos donde todos los participantes utilizan el mismo modelo de robot, llamado NAO robot de la compañía SoftBank Robotics, de ahí se deriva su nombre Plataforma Estandarizada. Su estilo de juego es totalmente autónomo, es decir, que las decisiones que toma cada robot son realizadas independientemente de su equipo, pero, aun así, es necesario que el equipo juegue como tal, al utilizar algoritmos de comunicación eficientes. Otro punto que agrega dificultad a este tipo de competencia es el hecho de que tanto el campo como el balón utilizados asemejan condiciones reales de juego sin considerar colores que sean más sencillos de reconocer al momento de procesar cada imagen.



Figura 6: Partido Liga Standard Platform RoboCup.

#### 1.1.5 Liga de Simulación (Simulation League):

Esta liga se enfoca en la aplicación de inteligencia artificial y estrategia en equipo. En ella, los jugadores actúan de forma individual jugando en un computador que simula un campo virtual, existiendo subcategorías de simulación en 2 y 3 dimensiones. Uno de sus mayores atractivos es el hecho de que no es necesario crear un modelo de jugador real, teniendo de esta forma un costo de implementación muy bajo en comparación con las otras ligas.



Figura 7: : Partido Liga Simulation en 2 y 3 dimensiones RoboCup.

## 1.2 Liga de simulación en dos dimensiones.

En la Liga de Simulación 2D, el objetivo principal es crear un entorno que simule de manera realista las condiciones variables de un campo de fútbol. En esta liga, dos equipos formados por 11 jugadores (agentes) son controlados por programas de software autónomos. El campo de juego, los jugadores y el balón se representan utilizando un servidor central llamado SoccerServer. Cada jugador puede ejecutar comandos básicos, como correr, girar y patear el balón. Estos comandos se pueden ingresar manualmente o programarse previamente en el algoritmo principal que controla a cada jugador.

Visualmente, cada jugador se representa como un disco en el plano que simboliza el campo de juego. Es importante destacar que el campo de juego simulado debe tener las dimensiones correspondientes a un campo de fútbol oficial de la FIFA, que son de 105 metros de largo por 68 metros de ancho. Esta representación se utiliza para emular la perspectiva captada por una cámara central en las competiciones de la Liga Small Size.



Figura 8: Visualización de un juego simulado en 2D.

### 1.2.1 RCSSSERVER

El software RCSSServer permite la competencia de múltiples equipos en un juego de fútbol en un entorno bidimensional. Utiliza un enfoque cliente-servidor a través de sockets UDP/IP para la comunicación entre el servidor y cada cliente, que son procesos independientes. Cada agente de RoboCup cuenta con tres sensores distintos: el sensor aural para detectar mensajes, el sensor visual para obtener información visual y actuar como sensor de proximidad, y el sensor corporal para detectar el estado físico del jugador. Estos sensores trabajan juntos para brindar al agente una visión completa del entorno.

### 1.2.2 RCSSMONITOR

RCSSMonitor es una herramienta de interfaz visual basada en el Frameview de Artur~Merke que permite observar y controlar el desarrollo del juego dentro del servidor de la RoboCup. Proporciona información detallada como los nombres de los equipos, el marcador, el tiempo transcurrido, los límites del campo y las posiciones de los jugadores y el balón. Además, ofrece opciones para interactuar con el servidor, como iniciar el juego manualmente. El monitor también permite ajustar la escala, color y tamaño de los componentes, así como mostrar datos relevantes de los agentes, como su estamina, área de tacleo, área de visión, entre otros. Estas opciones se pueden configurar desde la interfaz gráfica o mediante la terminal, brindando flexibilidad al usuario para personalizar el programa según sus necesidades y preferencias.

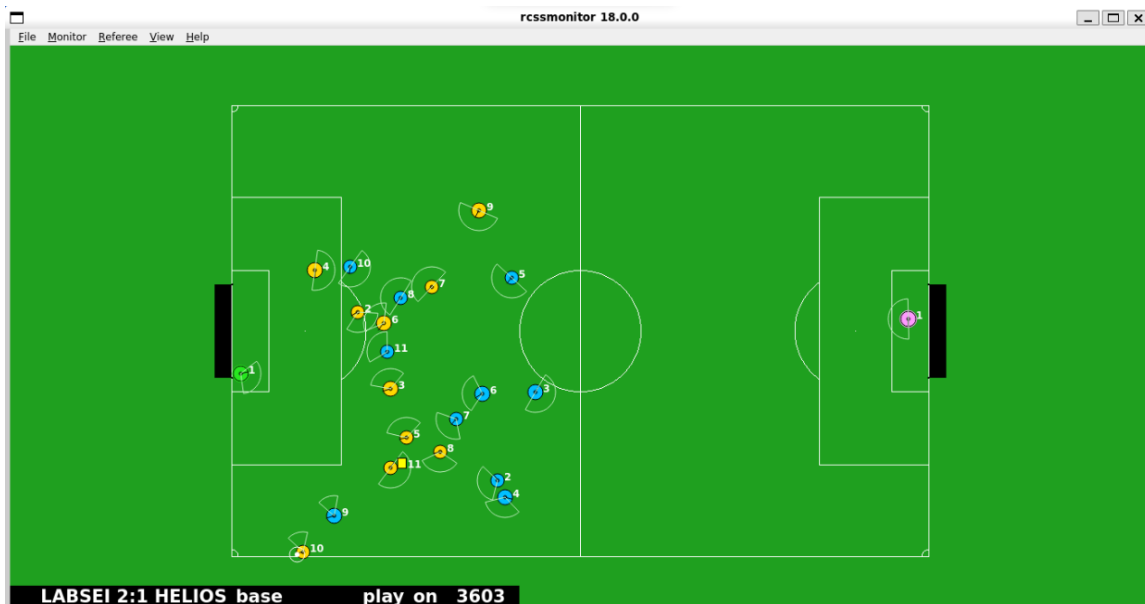


Figura 9: Interfaz gráfica durante el partido.

### 1.2.3 Cliente

Los "clientes" son algoritmos que interactúan con el servidor de simulación y controlan los agentes virtuales en el juego. El programa rcssclient es un cliente de texto básico utilizado para comprender los comandos del servidor. Se establece la conexión enviando comandos de inicialización y reconexión al puerto UDP del jugador en el servidor. Los comandos y respuestas son cadenas de caracteres encerradas

entre paréntesis. Cada jugador se presenta al servidor al inicio del juego y opcionalmente en el medio tiempo para la reconexión. La reconexión permite modificar el programa del jugador sin reiniciar el juego. Existe un comando de despedida para desconectarse del servidor y eliminar al jugador del campo.

## 2 Actividad 1: Instalación de aplicaciones necesarias.

Duración aproximada: 30 minutos

En esta actividad, se llevará a cabo la instalación de las aplicaciones necesarias para realizar un partido de simulación en el sistema operativo Linux. Esta instalación es crucial para poder continuar el taller y aprovechar al máximo las capacidades de simulación.

Los objetivos de esta actividad son:

- Instalar el software requerido para la simulación del partido en el sistema operativo Linux.
- Familiarizarse con el entorno de simulación en Linux.

A continuación, recibirá una sucesión de instrucciones las cuales debe seguir para completar la instalación del software necesario. Es importante notar que desde ahora en adelante las instrucciones dadas hacen referencia a la aplicación de la terminal del sistema operativo Linux de preferencia. Es decir, que no es apropiado utilizar un sistema de terminal en base de Windows o IOS. Cabe destacar también, que para completar con éxito esta actividad es necesario haber seguido las instrucciones detalladas en la Guía 0 de este taller.

### 2.1 librcsc

La librería librcsc es una librería básica dedicada al desarrollo de un equipo de soccer simulado para el servidor SoccerServer de la RoboCup.

#### Paso 1:

Descargar el fichero correspondiente a la librería a través del siguiente enlace.

- <https://github.com/helios-base/librcsc>

O desde la terminal usando el comando:

```
1 git clone https://github.com/helios-base/librcsc.git
```

#### Paso 2:

Es necesario ejecutar los siguientes comandos en la terminal para asegurar el entorno de desarrollo básico.

```
1 sudo apt update
2 sudo apt install build-essential libboost-all-dev autoconf automake libtool
```

**Paso 3:**

Ingresar los siguientes comandos en la terminal en el orden establecido:

```
1 sudo unzip librcsc-master.zip
2 cd librcsc-master
3 ./bootstrap
4 ./configure --disable-unit-test
5 make
6 sudo make install
```

```
make[1]: Entering directory '/home/labsei/librcsc-master/example'
make[2]: Entering directory '/home/labsei/librcsc-master/example'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/labsei/librcsc-master/example'
make[1]: Leaving directory '/home/labsei/librcsc-master/example'
Making install in src
make[1]: Entering directory '/home/labsei/librcsc-master/src'
make[2]: Entering directory '/home/labsei/librcsc-master/src'
/usr/bin/mkdir -p '/usr/local/bin'
/bin/bash ./libtool --mode=install /usr/bin/install -c rclmscheduler rclmtableprinter rcg2csv rcg2txt rcgrenameteam rcgreverse rcgverconv rcgversion '/usr/local/bin'
libtool: install: /usr/bin/install -c rclmscheduler /usr/local/bin/rclmscheduler
libtool: install: /usr/bin/install -c rclmtableprinter /usr/local/bin/rclmtableprinter
libtool: install: /usr/bin/install -c .libs/rcg2csv /usr/local/bin/rcg2csv
libtool: install: /usr/bin/install -c .libs/rcg2txt /usr/local/bin/rcg2txt
libtool: install: /usr/bin/install -c .libs/rcgrenameteam /usr/local/bin/rcgrenameteam
libtool: install: /usr/bin/install -c .libs/rcgreverse /usr/local/bin/rcgreverse
libtool: install: /usr/bin/install -c .libs/rcgverconv /usr/local/bin/rcgverconv
libtool: install: /usr/bin/install -c .libs/rcgversion /usr/local/bin/rcgversion
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/labsei/librcsc-master/src'
make[1]: Leaving directory '/home/labsei/librcsc-master/src'
make[1]: Entering directory '/home/labsei/librcsc-master'
make[2]: Entering directory '/home/labsei/librcsc-master'
/usr/bin/mkdir -p '/usr/local/bin'
/usr/bin/install -c librcsc-config librcscenv '/usr/local/bin'
/usr/bin/mkdir -p '/usr/local/lib/pkgconfig'
/usr/bin/install -c -m 644 librcsc.pc '/usr/local/lib/pkgconfig'
make[2]: Leaving directory '/home/labsei/librcsc-master'
make[1]: Leaving directory '/home/labsei/librcsc-master'
labsei@Fenrir:~/librcsc-master$
```

Figura 10: Interfaz luego de ingresados los comandos de instalación.

Este proceso es automático, luego de seguir estos pasos su interfaz debería mostrar algo parecido a lo que ilustra la Figura 10.

## 2.2 RCSSSERVER

Los siguientes pasos corresponden a las acciones necesarias para instalar el servidor de simulación:

**Paso 1:**

Descargar el archivo rcssserver-xx.x.x.tar.gz desde el enlace:

- <https://github.com/rcsoccersim/rcssserver/releases>.

Este archivo debe estar posicionado en la dirección donde se desee instalar el servidor. De preferencia es mejor utilizar la dirección home/(nombreusuario)

**Paso 2:**

Es necesario ejecutar los siguientes comandos en la terminal con el fin de resolver todos los problemas de dependencia que puedan generarse al instalar el servidor.

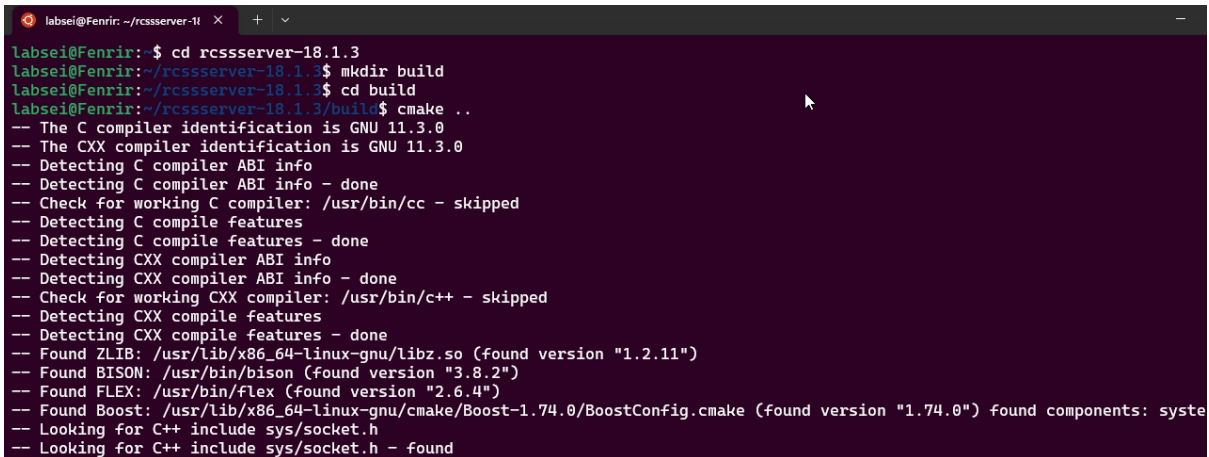
```
1 sudo apt update
2 sudo apt install build-essential automake autoconf libtool flex bison libboost-all-dev
```



**Paso 3:**

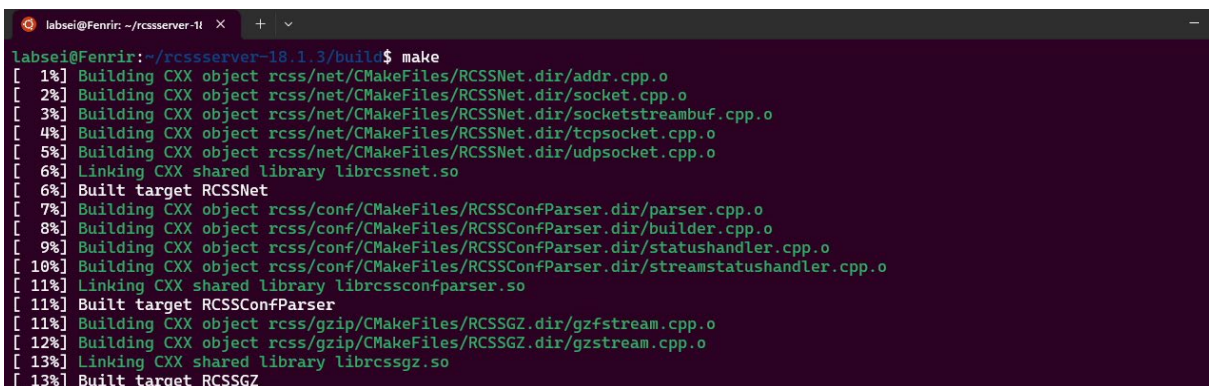
Ingresar los siguientes comandos en la terminal en el orden establecido. Los símbolos xx.x.x deben ser reemplazados por la versión del servidor descargada, como es posible apreciar en la Figura 11.

```
1 tar xzvf rcssserver-xx.x.x.tar.gz
2 cd rcssserver-xx.x.x
3 mkdir build
4 cd build
5 cmake ..
6 sudo make
```



```
labsei@Fenrir: ~/rcssserver-11
labsei@Fenrir:~/rcssserver-11$ cd rcssserver-18.1.3
labsei@Fenrir:~/rcssserver-18.1.3$ mkdir build
labsei@Fenrir:~/rcssserver-18.1.3$ cd build
labsei@Fenrir:~/rcssserver-18.1.3/build$ cmake ..
-- The C compiler identification is GNU 11.3.0
-- The CXX compiler identification is GNU 11.3.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found ZLIB: /usr/lib/x86_64-linux-gnu/libz.so (found version "1.2.11")
-- Found BISON: /usr/bin/bison (found version "3.8.2")
-- Found FLEX: /usr/bin/flex (found version "2.6.4")
-- Found Boost: /usr/lib/x86_64-linux-gnu/cmake/Boost-1.74.0/BoostConfig.cmake (found version "1.74.0") found components: system
-- Looking for C++ include sys/socket.h
-- Looking for C++ include sys/socket.h - found
```

Figura 11: Interfaz luego de ingresar el comando "cmake .."



```
labsei@Fenrir: ~/rcssserver-11
labsei@Fenrir:~/rcssserver-18.1.3/build$ make
[ 1%] Building CXX object rcss/net/CMakeFiles/RCSSNet.dir/addr.cpp.o
[ 2%] Building CXX object rcss/net/CMakeFiles/RCSSNet.dir/socket.cpp.o
[ 3%] Building CXX object rcss/net/CMakeFiles/RCSSNet.dir/socketstreambuf.cpp.o
[ 4%] Building CXX object rcss/net/CMakeFiles/RCSSNet.dir/tcpsocket.cpp.o
[ 5%] Building CXX object rcss/net/CMakeFiles/RCSSNet.dir/udpsocket.cpp.o
[ 6%] Linking CXX shared library librcssnet.so
[ 6%] Built target RCSSNet
[ 7%] Building CXX object rcss/conf/CMakeFiles/RCSSConfParser.dir/parser.cpp.o
[ 8%] Building CXX object rcss/conf/CMakeFiles/RCSSConfParser.dir/builder.cpp.o
[ 9%] Building CXX object rcss/conf/CMakeFiles/RCSSConfParser.dir/statushandler.cpp.o
[ 10%] Building CXX object rcss/conf/CMakeFiles/RCSSConfParser.dir/streamstatushandler.cpp.o
[ 11%] Linking CXX shared library librcssconfparser.so
[ 11%] Built target RCSSConfParser
[ 11%] Building CXX object rcss/gzip/CMakeFiles/RCSSGZ.dir/gzfstream.cpp.o
[ 12%] Building CXX object rcss/gzip/CMakeFiles/RCSSGZ.dir/gzstream.cpp.o
[ 13%] Linking CXX shared library librcssgz.so
[ 13%] Built target RCSSGZ
```

Figura 12: Interfaz luego de ingresar el comando "make"

**Paso 4:**

Luego, es necesario ejecutar el algoritmo de configuración del servidor y completar la instalación a través de los comandos:

```
1 cd build //Este comando solo es necesario en caso de no encontrarse en el fichero build.
2 cmake ..
3 sudo make install
```

## 2.3 RCSSMONITOR

Este software nos permite visualizar lo que sucede durante el partido, por lo que no es imperante al momento de desarrollar un encuentro de RoboSoccer, pero si entrega una forma mucho más clara y sencilla de entender las acciones implementadas por cada jugador.

### Paso 1:

Descargar el archivo rcssmonitor-xx.x.x.tar.gz desde el enlace:

- <https://github.com/rcsoccersim/rcssmonitor>

### Paso 2:

Ejecutar los siguientes comandos en la terminal con el fin de resolver todos los problemas de dependencia que puedan generarse al instalar el servidor.

```
1 sudo apt update
2 sudo apt install build-essential libfontconfig1-dev libaudio-dev libxt-dev libglib2.0-dev libxi-
  dev libxrender-dev
3 sudo apt-get install qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools
```

### Paso 3:

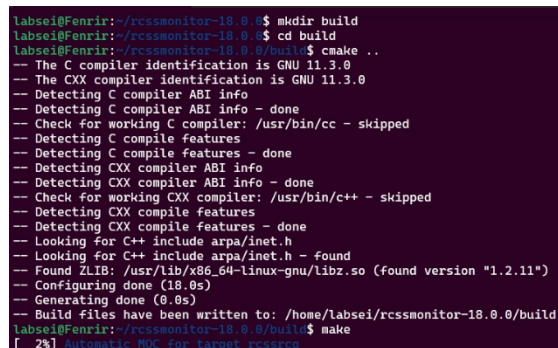
Ingresar los siguientes comandos en la terminal en el orden establecido. Los símbolos xx.x.x deben ser reemplazados por la versión del monitor descargada.

```
1 tar xzvfp rcssmonitor-x.x.x.tar.gz
2 cd rcssmonitor-x.x.x
3 mkdir build
4 cd build
5 cmake ..
6 sudo make
```

### Paso 4:

Luego, es necesario ejecutar el algoritmo de configuración del servidor y completar la instalación a través de los comandos:

```
1 cmake ..
2 sudo make install
```



```
labsei@Fenrir:~/rcssmonitor-18.0.0$ mkdir build
labsei@Fenrir:~/rcssmonitor-18.0.0$ cd build
labsei@Fenrir:~/rcssmonitor-18.0.0/build$ cmake ..
-- The C compiler identification is GNU 11.3.0
-- The CXX compiler identification is GNU 11.3.0
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for C++ include arpa/inet.h
-- Looking for C++ include arpa/inet.h - found
-- Found ZLIB: /usr/lib/x86_64-linux-gnu/libz.so (found version "1.2.11")
-- Configuring done (18.0s)
-- Generating done (0.0s)
-- Build files have been written to: /home/labsei/rcssmonitor-18.0.0/build
labsei@Fenrir:~/rcssmonitor-18.0.0/build$ make
[ 2%] Automatic MOC for target rcssmon
```

Figura 13: Interfaz de la terminal luego de ingresar los comandos presentados.

Al completar esta actividad, se espera haber logrado los siguientes resultados:

- Todas las aplicaciones necesarias estarán instaladas en el sistema operativo Linux.
- Se habrá adquirido la base necesaria para realizar partidos de simulación en el entorno correspondiente.

La instalación exitosa de las aplicaciones necesarias establece las bases para las etapas posteriores del taller, permitiendo aprovechar al máximo las capacidades de simulación. Esta actividad es fundamental para garantizar un avance fluido en el taller y proporcionar a los participantes las herramientas necesarias para el desarrollo de las actividades futuras.

### 3 Actividad 2: Comandos básicos de simulación

Duración aproximada: 10 minutos

En esta actividad, pondremos en práctica los conocimientos adquiridos previamente al utilizar comandos básicos de simulación en el entorno de RoboSoccer.

Los objetivos de esta actividad son:

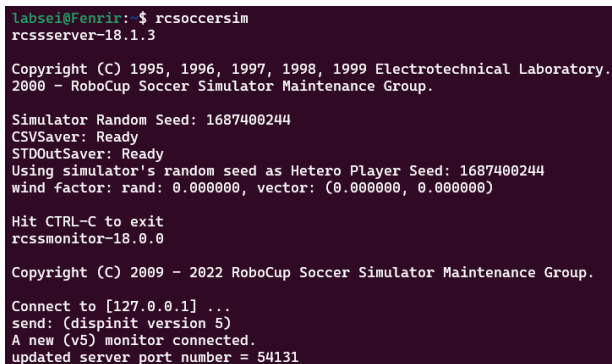
- Aplicar los conocimientos adquiridos para utilizar comandos básicos de simulación.
- Familiarizarse con la interacción mediante comandos en la terminal.
- Inicializar un jugador en el campo de juego simulado.
- Aplicar los comandos de movimiento de un jugador desde la terminal.

Esta actividad consiste en inicializar un jugador, posicionarlo en el campo, iniciar el partido y desplazar al jugador de manera que cree un cuadrado alrededor del centro del campo. Todo esto debe ser realizado ingresando comandos a través de la terminal. A continuación, se entregará una serie de pasos que deben ser completados para lograr este objetivo.

Para poder ejecutar el servidor y monitor se ingresará el siguiente comando:

```
1 rcsoccersim
```

Al ejecutar este comando se debería apreciar en la terminal lo ilustrado en la Figura 14.



```
labsei@Fenrir:~$ rcsoccersim
rcssserver-18.1.3

Copyright (C) 1995, 1996, 1997, 1998, 1999 Electrotechnical Laboratory.
2000 - RoboCup Soccer Simulator Maintenance Group.

Simulator Random Seed: 1687400244
CSVSAver: Ready
STDOutSAver: Ready
Using simulator's random seed as Hetero Player Seed: 1687400244
wind factor: rand: 0.000000, vector: (0.000000, 0.000000)

Hit CTRL-C to exit
rcssmonitor-18.0.0

Copyright (C) 2009 - 2022 RoboCup Soccer Simulator Maintenance Group.

Connect to [127.0.0.1] ...
send: (dispinet version 5)
A new (v5) monitor connected.
updated server port number = 54131
```

Figura 14: Salida en terminal al inicializar el servidor.

Por otro lado, al iniciar el servidor utilizando este comando, se abrirá inmediatamente la interfaz del monitor para la simulación, como es posible apreciar en la Figura 15.



Figura 15: Interfaz de aplicación de monitor.

Para iniciar un partido se debe presionar la opción Kick-Off en el menú de Referee.

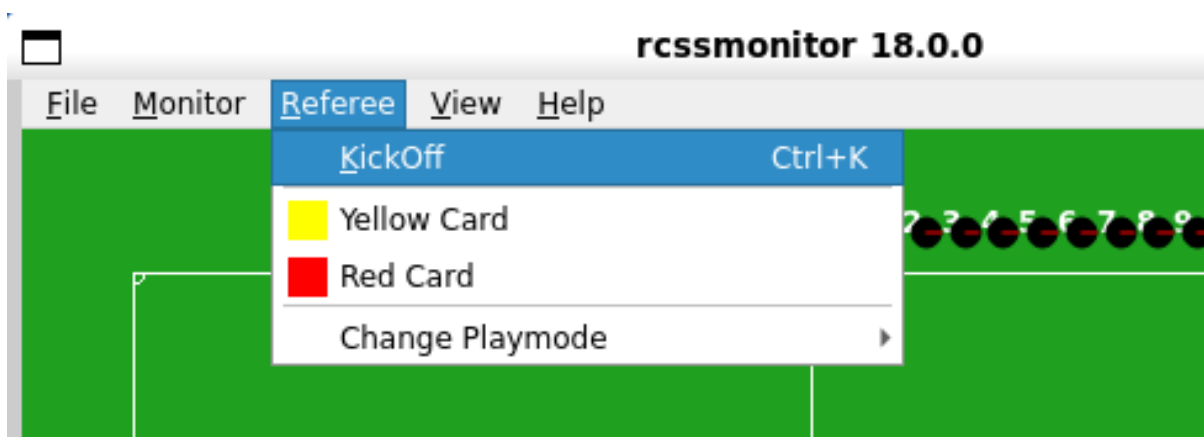


Figura 16: Patada inicial.

La Tabla 3-1 muestra los comandos más básicos al momento de inicializar y mover a un jugador.



Tabla 3-1: Comandos básicos del jugador

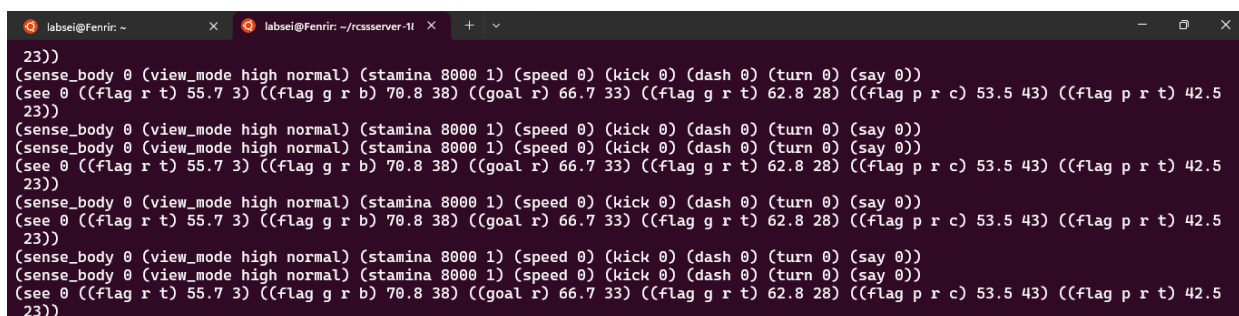
Comando	Función
(init NombreEquipo)	Este comando entrega un mensaje de "saludo" al servidor, inicializando así al jugador.
(move X Y)	Este comando solo puede ser ejecutado antes del saque inicial y después de un gol. Su función es mover al jugador a la posición exacta de X (entre -54 y 54) e Y (entre -32 y 32) en un ciclo de simulación.
(dash <i>Potencia</i> )	Acelera al jugador en la dirección de su cuerpo (no en la dirección de la velocidad actual). La potencia se encuentra dentro de un rango definido por el servidor, con valores máximos (maxpower) y mínimos (minpower) establecidos (por defecto, -100 y 100).
(turn_neck <i>Angulo</i> )	Este comando permite girar el cuello del jugador según el ángulo indicado con relación al ángulo previo. Es importante destacar que el ángulo resultante del cuello estará comprendido entre minneckang (por defecto: -90) y maxneckang (por defecto: 90) con respecto a la dirección del cuerpo del jugador.
(bye)	Este comando manda un mensaje de "despedida" al servidor, desconectando de esa forma al jugador.

Se recomienda alterar el factor max\_dash\_power para completar más fácilmente esta primera actividad. Para ello se debe modificar esta variable en el archivo server.conf localizado en la carpeta .rcssserver.

El servidor RCSSSERVER, posee un sistema de cliente de demostración el cual permite probar estos comandos básicos. Para iniciar este sistema es necesario seguir los siguientes comandos.

```
1 cd rcssserver-18.1.3/src
2 ./rcssclient
```

Al inicializar un jugador el servidor enviará toda la información entregada por los sensores, por lo que la terminal se encontrará sobrecargada de información. Esto puede ser observado en la Figura 17.



```
labsei@Fenrir: ~
labsei@Fenrir: ~/rcssserver-18.1.3/src
23))
(sense_body 0 (view_mode high normal) (stamina 8000 1) (speed 0) (kick 0) (dash 0) (turn 0) (say 0))
(see 0 ((flag r t) 55.7 3) ((flag g r b) 70.8 38) ((goal r) 66.7 33) ((flag g r t) 62.8 28) ((flag p r c) 53.5 43) ((flag p r t) 42.5 23))
(sense_body 0 (view_mode high normal) (stamina 8000 1) (speed 0) (kick 0) (dash 0) (turn 0) (say 0))
(sense_body 0 (view_mode high normal) (stamina 8000 1) (speed 0) (kick 0) (dash 0) (turn 0) (say 0))
(see 0 ((flag r t) 55.7 3) ((flag g r b) 70.8 38) ((goal r) 66.7 33) ((flag g r t) 62.8 28) ((flag p r c) 53.5 43) ((flag p r t) 42.5 23))
(sense_body 0 (view_mode high normal) (stamina 8000 1) (speed 0) (kick 0) (dash 0) (turn 0) (say 0))
(see 0 ((flag r t) 55.7 3) ((flag g r b) 70.8 38) ((goal r) 66.7 33) ((flag g r t) 62.8 28) ((flag p r c) 53.5 43) ((flag p r t) 42.5 23))
(sense_body 0 (view_mode high normal) (stamina 8000 1) (speed 0) (kick 0) (dash 0) (turn 0) (say 0))
(sense_body 0 (view_mode high normal) (stamina 8000 1) (speed 0) (kick 0) (dash 0) (turn 0) (say 0))
(see 0 ((flag r t) 55.7 3) ((flag g r b) 70.8 38) ((goal r) 66.7 33) ((flag g r t) 62.8 28) ((flag p r c) 53.5 43) ((flag p r t) 42.5 23))
```

Figura 17: Salida de terminal al ejecutar la aplicación de demostración de cliente.

Por lo que se recomienda utilizar una de estas dos opciones al momento de ejecutar esta aplicación:

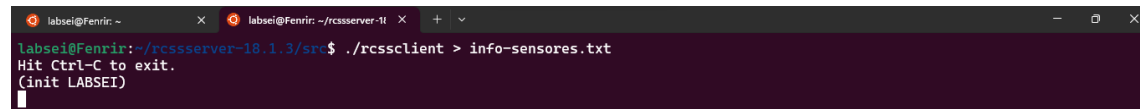
- Opción 1: Utilizar el comando `./rcssclient > /dev/null` el cual elimina la información entregada por la terminal.



```
labsei@Fenrir: ~
labsei@Fenrir: ~/rcssserver-18.1.3/src$ ./rcssclient > /dev/null
Hit Ctrl-C to exit.
(init LABSEI)
```

Figura 18: Salida al aplicar el comando `./rcssclient > /dev/null`

- Opción 2: Utilizar el comando `./rcssclient > info-sensores.txt` el cual enviara a un archivo de texto toda la información entregada por los sensores.



```
labsei@Fenrir: ~
labsei@Fenrir: ~/rcssserver-18.1.3/src$ ./rcssclient > info-sensores.txt
Hit Ctrl-C to exit.
(init LABSEI)
```

Figura 19: Salida al aplicar el comando `./rcssclient > info-sensores.txt`

En este caso la información esta siendo almacenada en el archivo de texto `info-sensores.txt`.

Una vez ingresados los comandos anteriores el jugador será inicializado en el campo, por lo que es necesario utilizar los comandos indicados en la Tabla 3-1, para posicionarlo en el área de juego y moverlo a través de la zona central.

Al completar esta actividad, se espera haber logrado los siguientes resultados:

- El jugador será inicializado en el equipo.
- El jugador se posicionará en la zona central del campo.
- El partido se iniciará correctamente.
- El jugador se moverá de manera que forme un cuadrado en la zona central.

Las siguientes imágenes ilustran los resultados esperados:



Figura 20: Jugador luego de ser inicializado y posicionado en el campo de juego.



Figura 21: Jugador moviéndose por el campo después de la patada inicial.



Figura 22: Jugador después de girar y continuar moviéndose en línea recta.

Esta actividad nos permite aplicar el conocimiento adquirido y familiarizarnos con el uso de comandos desde la terminal. Conocer y dominar estas habilidades es esencial al momento de interactuar con un entorno virtual, permitiendo así conocer mejor el sistema de juego y preparándonos para desafíos futuros.

## Referencias

- [1] RoboCup Federation. (s.f.). *RoboCup Federation official website*. (RoboCup Federation) Recuperado el 2 de Julio de 2023, de <https://robocup.org>
- [2] RoboCup Federation. (s.f.). *RoboCup Small Size League - Principles and Goals*. Recuperado el 18 de Abril de 2023, de <https://ssl.robocup.org/principles-and-goals/>
- [3] The RoboCup Soccer Simulator Maintenance Committee. (s.f.). *The robocup soccer simulator users manual*. Recuperado el 02 de Julio de 2023, de <https://rscsoccersim.readthedocs.io>