

Proyecto Forrest
Liga de Simulación 2D
RoboCup

Plan de Experimentación, Verificación y Validación

Versión 3.0

Histórico de Revisiones

Versión	Fecha	Resumen de cambios	Autor
1.0	15/02/2006	Versión inicial	Grupo <i>Forrest</i>
2.0	27/05/2006	Definición de los casos de pruebas	Grupo <i>Forrest</i>

Índice

1. Estrategia de evaluación del equipo <i>Forrest</i>	3
1.1. Tipos de pruebas	3
1.2. Indicadores de juego	5
1.3. Herramientas para la recolección y análisis de datos	6
1.4. Configuración del ambiente de ejecución	6
2. Estrategia de validación de la documentación del proyecto	8
2.1. Documentación a verificar	8
2.2. Elementos a verificar	8
3. Definición de casos	8
3.1. Pruebas funcionales	8
3.1.1. Verificación de la técnica grafo de coordinación (versión 4.0)	8
3.2. Pruebas de evolución funcional	9
3.2.1. Evolución funcional v3.1 - v4.0	10
3.2.2. Evolución funcional v3.1 - v5.0	10
3.2.3. Evolución funcional v4.0 - v6.0	10
3.3. Pruebas no funcionales	11
3.3.1. Performance	11
3.3.2. Carga o volumen	11
3.3.3. Esfuerzo	11
3.4. Otras pruebas	12

1. Estrategia de evaluación del equipo *Forrest*

El equipo *Forrest* es considerado un sistema de tiempo real que actúa sobre un ambiente dinámico y no determinista. Estas características, entre otras, determinaron que el plan de verificación y validación del equipo se hiciera en términos estadísticos. Es decir, el resultado esperado para cada prueba no se define en términos de un valor exacto conocido a priori, sino que los resultados se miden en términos promediales y estadísticos a partir de ejecuciones repetitivas de un mismo escenario. Debido al dinamismo y no determinismo del entorno, puede suceder que ciertas tendencias distorsionen los resultados finales de una prueba. Por ejemplo, un equipo puede tener el control del juego la mayor parte del tiempo y perder el partido por algún error aislado de su defensa o por alguna jugada afortunada del equipo contrario. Una forma de minimizar el impacto de este tipo de comportamiento es jugando una gran cantidad de partidos, evitando así que las estadísticas de los aspectos evaluados se vean afectadas.

En el fútbol no existe una única forma de demostrar la superioridad de un equipo. Generalmente, la evaluación se realiza a través de métricas o indicadores de juego, los cuales permiten realizar un análisis más profundo sobre el comportamiento del equipo. La cantidad y tipos de indicadores utilizados con este propósito se definen en función del aspecto o característica que se desee evaluar. Los puntos ganados por el equipo, la diferencia de goles o el promedio de posesión de la pelota, son algunos de los indicadores más utilizados. El principal objetivo de los experimentos realizados en este trabajo es evaluar si la combinación de técnicas utilizada en la estrategia de *Forrest* mejora el rendimiento general del equipo. Para lograr esto, la metodología de experimentación utilizada en cada prueba consiste en realizar n enfrentamientos, y para cada uno de ellos jugar m partidos. Al finalizar, se obtienen los valores estadísticos de cada indicador definido y se realiza el análisis correspondiente.

1.1. Tipos de pruebas

Uno de los problemas más importantes a resolver en el plan de verificación, es determinar las diferentes pruebas que se le deben realizar al sistema. En este sentido, la metodología de desarrollo utilizada para la construcción del equipo fue determinante. Se utilizó un proceso iterativo incremental, con el objetivo de que cada una de las versiones intermedias del equipo fuera evolucionando, hasta llegar finalmente a la propuesta planteada. En el Cuadro 1 se muestran las versiones del equipo *Forrest* [CANa].

Con el propósito de poder determinar el correcto funcionamiento de cada versión y su evolución con respecto a las anteriores, se planificaron y realizaron tres tipos de pruebas:

- *Pruebas funcionales*: Tienen como objetivo evaluar el correcto funcionamiento de la versión. Es decir, evaluar que la versión se comporte de acuerdo a su objetivo y a los cambios que se le introdujeron. Dadas las características de las técnicas utilizadas, uno de los principales objetivos de estas pruebas es determinar la sintonización y ajustes de los distintos componentes de la estrategia del equipo: función de utilidad, función de evaluación de acciones, variables de estado, reglas, etc.
- *Pruebas no funcionales*: Tienen el objetivo de medir el comportamiento del equipo con respecto a los requerimientos no funcionales definidos: performance, carga, esfuerzo, etc.
- *Pruebas de evolución funcional*: Tienen el objetivo de medir la evolución del equipo como consecuencia de la incorporación de una nueva técnica o variante. Es decir, las versiones que se comparan varían únicamente en el factor que se desee evaluar, entendiendo que una manera apropiada de medir el aporte de la variante x , es que el equipo que contiene x se evalúe contra otro equipo exactamente igual a diferencia de x . Pruebas de este tipo serán las que permitan evaluar como la combinación de técnicas propuesta modifica el comportamiento del equipo.

- *Otro tipo de pruebas*: las pruebas de esta categoría buscan determinar o evaluar diferentes problemas que se han detectado a lo largo del proyecto y que no entran en ninguna de las categorías anteriores.

Versión	Descripción
1.0	Se integra por primera vez la estrategia procedural utilizando el framework UvA Trilearn 2003. La misma se divide en dos partes según el estado del juego: <i>estrategia con pelota</i> y <i>estrategia sin Pelota</i> . Se incluyen conceptos importantes como: <i>función de utilidad</i> , <i>posicionamiento estratégico</i> y <i>mirada estratégica</i> .
2.0	Se rediseñan varios de los componentes desarrollados en la versión 1.0 con el objetivo de desacoplar el diseño del equipo del framework.
3.1	Es la primera versión estable del equipo. La toma de decisiones de los agentes se basa en la función de utilidad, eligiendo la acción que mejor se evalúe.
4.0	La toma de decisiones de los agentes se basa en la técnica de <i>grafos de coordinación</i> . La evaluación de cada regla es realizada a través de la función de utilidad definida en la versión 3.1.
5.0	La toma de decisiones de los agentes se basa en la técnica de <i>planificación en línea de dos niveles</i> . El método de evaluación utiliza la función de utilidad definida en la versión 3.1.
6.0	En esta versión se implementa la propuesta que se presenta en este trabajo. La toma de decisiones de los agentes es realizada a través la técnica de <i>grafos de coordinación</i> , donde la evaluación de cada regla es realizada a través de la <i>planificación en línea de dos niveles</i> .

Cuadro 1: Principales versiones del equipo *Forrest*.

En la figura 1 se muestra la evolución de versiones del sistema y los diferentes tipos de pruebas que se le aplican.

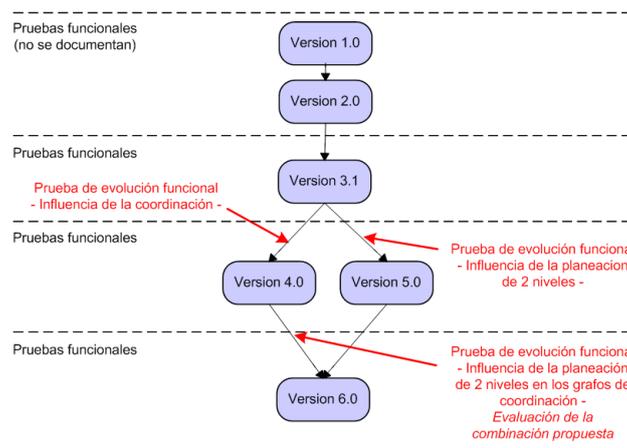


Figura 1: Evolución de las versiones del equipo.

1.2. Indicadores de juego

La estrategia del equipo *Forrest* tiene como principal objetivo coordinar las acciones entre los agentes, priorizando aquellas que tengan mejores posibilidades de juego en el futuro. Como resultado de este comportamiento, se espera que el equipo mejore su rendimiento. Luego de analizar la mejor forma de medirlo, sabiendo que son muchos los factores que lo afectan y que resulta difícil medirlo con un único valor, se determinó un conjunto de indicadores que permitan realizar un análisis más profundo en este aspecto. A continuación se presentan cada uno de estos indicadores:

Puntos obtenidos (%PO)

El porcentaje de puntos obtenidos durante una serie de partidos resulta ser el indicador más representativo para medir el rendimiento de un equipo de fútbol. El principal problema al considerar este indicador es que se ve influenciado directamente por varios factores del equipo: organización de la defensa, capacidad ofensiva, etc. Si bien estos factores fueron considerados durante el desarrollo del equipo, no se encontraba dentro del alcance definido profundizar en estos aspectos y por lo tanto su implementación actual cuenta con varias limitaciones que disminuyen notoriamente su efectividad. Para las pruebas realizadas se otorgaron tres puntos por partido ganado, un punto por partido empatado y cero puntos por partido perdido.

Goles a favor (#GF) y Tiros al arco (#TA)

Estos indicadores permiten medir la capacidad ofensiva del equipo. Por un lado, #GF mide el promedio de goles convertidos por partido, resultando relevante su análisis cuando el indicador de puntos obtenidos no presenta diferencias entre los equipos evaluados. Por otro lado, #TA mide el promedio de tiros al arco realizados por partido. Este indicador permite analizar la capacidad ofensiva del equipo en función de la cantidad de oportunidades de gol que genere, dejando de lado los problemas de efectividad que pueda llegar a presentar la estrategia de definición del equipo.

Tiempo de la pelota en campo contrario (%TPC)

La cantidad de tiempo que pasa la pelota en el campo contrario resulta ser otro factor importante para medir la capacidad ofensiva del equipo. Este indicador, a diferencia de los anteriores, permite evaluar la disposición que presenta el equipo para atacar. El supuesto es que cuanto más tiempo pase la pelota en el campo contrario mayores van a ser las posibilidades de generar oportunidades de gol.

Control de la pelota (%CP)

Representa el porcentaje de tiempo que el equipo tiene el control de la pelota. Si bien un valor alto en este indicador podría ser deseable, no necesariamente representa una mejora en el rendimiento del equipo. Por ejemplo, se puede tener un alto porcentaje de control de la pelota como consecuencia de realizar acciones intrascendentes en el propio campo de juego sin generar peligro al equipo contrario. Por este motivo, es necesario considerarlo junto a otros indicadores que permitan realizar un análisis más certero acerca del beneficio real que presenta este factor, como puede ser el indicador %TPC.

Pases realizados (#P) y pases exitosos (%PE)

Estos indicadores se han elegidos para medir el efecto de aplicar la técnica grafos de coordinación sobre la estrategia del equipo. Las reglas que rigen la estrategia coordinan explícitamente los pases entre compañeros, por lo cual el promedio de pases exitosos resulta relevante para medir la efectividad de esta técnica.

1.3. Herramientas para la recolección y análisis de datos

Durante la ejecución de las pruebas, es necesario determinar la forma en de recolectar y analizar los indicadores de juego presentados en la sección anterior. Con este propósito, se utilizarán las herramientas *Statistics Proxy Server* [FRA], *FLACS* [CANb] y *Team Assistant 2003* [HED].

Statistics Proxy Server

Para recolectar los indicadores en cada uno de los partidos, se utiliza la herramienta *Statistics Proxy Server*. Durante la ejecución de un partido, esta herramienta se conecta al simulador como un cliente más, para consultar el estado del juego y generar estadísticas. Al final de cada partido genera un archivo con los valores de todos los indicadores que le fueron solicitados¹.

FLACS - Forrest Log Analyzer Consultancy Service

Para analizar la información recolectada durante las pruebas, se utiliza una herramienta OLAP llamada *FLACS*. Una vez terminada la sesión de pruebas, *FLACS* permite graficar y realizar consultas sobre toda la información generada a través de una carga previa de los datos.

La información analizada por *FLACS* proviene de dos fuentes distintas. Por un lado, contiene los indicadores de juego generados durante las sesiones de prueba a través de la herramienta *Statistics Proxy Server*. Por otro lado, cada agente del equipo *Forrest* fue diseñado para generar un archivo de log en donde se indican todas las acciones que realizaron durante un partido. El análisis de esta información permite, entre otras cosas, evaluar la coordinación del equipo al utilizar la técnica grafos de coordinación. Por ejemplo, si un agente i decide pasarle la pelota a otro agente j en una dirección d , debe ocurrir que en el mismo ciclo el agente j tome la decisión de moverse en la dirección d . En caso de no suceder esto se considera un error en la coordinación del pase.

Team Assistant 2003

En varias ocasiones es necesario analizar detenidamente la ejecución de un partido para determinar los ajustes que se le deben realizar al equipo. Con este propósito, generalmente se utilizan asistentes de visualización para facilitar este tipo de tareas. Durante las pruebas realizadas al equipo se utiliza la herramienta *Team Assistant 2003*. Esta aplicación es un asistente multi-función que permite analizar y extraer información elaborada de los partidos jugados a través del simulador. Algunas de sus características son: permite reproducir partidos en varios modos (pausa, retroceso, avance rápido y lento, etc.), brinda información del estado de todos los objetos del campo de juego, actúa como coach del equipo y genera diversos reportes.

1.4. Configuración del ambiente de ejecución

Todos los partidos realizados durante las pruebas serán ejecutados en un ambiente controlado, donde muchas de sus condiciones difieren de las utilizadas durante las competencias oficiales de la liga. Por un lado, el simulador es configurado para que los agentes de ambos equipos tengan visión global sobre el campo de juego. Esta condición es necesaria para que el algoritmo de *eliminación de variables* aplicado en este trabajo funcione correctamente. Por otra parte, también se configura el simulador para que funcione en modo sincrónico. Este modo determina que el simulador espere a que todos los agentes envíen sus acciones antes de avanzar de ciclo y actualizar el entorno. De esta forma, se tiene la seguridad que la acción determinada por cada agente del equipo se ejecute en el ciclo adecuado, por más que el

¹Esta herramienta brinda un conjunto predefinido de indicadores.

tiempo para determinar la acción a ejecutar sea mayor del que permiten las condiciones de competencia (modo asincrónico con ciclos de simulación de 100 ms). Por último, se configura el simulador para que funcione de forma autónoma, logrando así que no interrumpa el juego al finalizar el primer tiempo y que termine su ejecución automáticamente cuando finalice el partido. Este cambio es necesario para realizar automáticamente la ejecución de varios partidos.

Todas las pruebas serán realizadas en un único equipo P4 con 512 MB de memoria RAM y 2.0 GHz de velocidad de procesador, utilizando la plataforma Linux Susse 9.2. La versión utilizada del simulador fue la 10.x.

Modos de ejecución

Otro aspecto a resolver en lo que refiere a la configuración del ambiente, es la forma en la cual se ejecutan las pruebas. Se utilizaron dos mecanismos de ejecución:

- *Ejecución manual*: Este tipo de ejecución permite evaluar visualmente algunas características puntuales del funcionamiento del equipo: coordinación en los pases, toma de decisiones, definición de jugadas, etc. Estas pruebas fueron realizadas con la ayuda de la herramienta *Team Assistant 2003*.
- *Ejecución automática*: Este tipo de ejecución se utiliza para evaluar el comportamiento general del equipo con respecto a una versión anterior o a otro equipo. Estas pruebas consisten en la repetición de una serie de partidos de similares características con el objetivo de recolectar suficiente información estadística para luego realizar el análisis correspondiente. Para este tipo de pruebas se definió un shell script que permite configurar los distintos equipos que se van a enfrentar y la cantidad de partidos que se juegan entre ellos. Este script también se ocupa de cargar las estructuras de datos utilizadas por *FLACS*. Se utiliza la herramienta *Statistics Proxy Server* para generar los indicadores de juego presentados en la sección 1.2 y la herramienta *FLACS* para analizar los resultados. En la Fig. 2 se muestra el esquema de la estructura de archivos que se genera durante una sesión de pruebas del equipo.

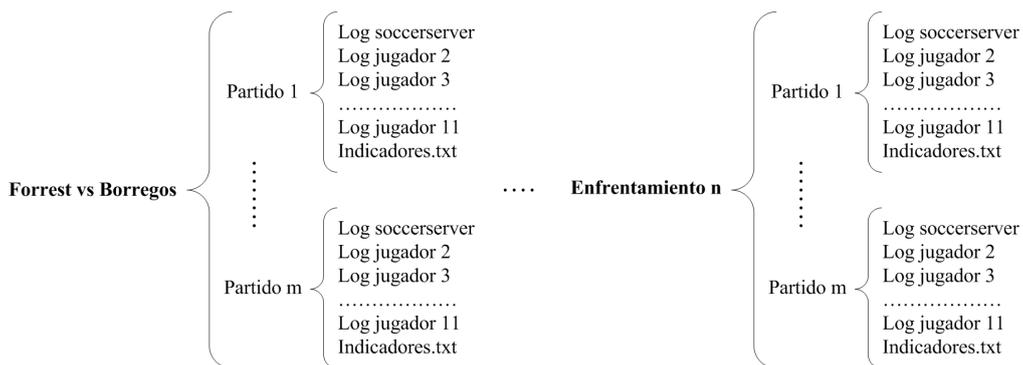


Figura 2: Archivos generados durante una sesión de pruebas del equipo *Forrest*.

2. Estrategia de validación de la documentación del proyecto

2.1. Documentación a verificar

Se debe verificar toda la documentación generada durante el proyecto. La misma será verificada por los integrantes del grupo en las instancias que se crean oportunas. Además, para el caso de algunos documentos de vital importancia en el proyecto (Estado del arte, Informe final, Presentación de la defensa del proyecto) se realizará la verificación a través de personas ajenas al proyecto.

2.2. Elementos a verificar

Entro de los principales aspectos a verificar se encuentran:

- ortografía
- presentación del documento
- correcto uso de las referencias
- comprensión del documento
- poder de expresión
- calidad del contenido

3. Definición de casos

3.1. Pruebas funcionales

3.1.1. Verificación de la técnica grafo de coordinación (versión 4.0)

Caso 1	
Objetivo	Evaluar la correctitud del algoritmo de eliminación.
Estrategia	Imprimir en archivo valores intermedios y luego analizarlos. Cantidad de partidos: 1 Equipo Oponente: Forrest v3.1

Caso 2	
Objetivo	Evaluar la correctitud del algoritmo de asignación de roles.
Estrategia	Imprimir en archivo valores intermedios y luego analizarlos. Cantidad de partidos: 1 Equipo Oponente: Forrest v3.1

Caso 3	
Objetivo	Evaluar la correctitud del algoritmo de instanciación del grafo.
Estrategia	Imprimir en archivo valores intermedios y luego analizarlos. Cantidad de partidos: 1 Equipo Oponente: Forrest v3.1

Caso 4	
Objetivo	Evaluar la correctitud del funcionamiento del grafo de coordinación. Verificar que todos los agentes encuentren la misma acción óptima conjunta.
Estrategia	Imprimir en archivo valores intermedios y luego analizarlos. Cada vez que un agente determina una acción que es coordinada, corroborar que el resto de los agentes involucrados decidan lo mismo. Cantidad de partidos: 1 Equipo Oponente: Forrest v3.1

Caso 5	
Objetivo	Verificar que la transformación de las acciones determinadas por las agentes es transformada a los comandos de ejecución de forma correcta.
Estrategia	Imprimir en archivo valores intermedios. Reproducir los logs del simulador en <i>Team Assistant</i> y analizar, en determinados ciclos, si lo que efectivamente se ejecuta en el simulador son las acciones que los agentes determinaron a través del GC. Cantidad de partidos: 1 Equipo Oponente: Forrest v3.1

3.2. Pruebas de evolución funcional

Se tienen dos formas de realizar estas pruebas. Por un lado, se juegan series de partidos entre las versiones evaluadas con el objetivo de evaluar directamente la evolución del equipo entre estas versiones. Por otro lado, se comparan las versiones a través de un tercer equipo como forma de validar los resultados obtenidos en las pruebas anteriores. En este caso, se eligió al equipo Borregos.

La elección de *Borregos* se debe principalmente a las similitudes que presenta con el equipo *Forrest*. Ambos utilizan como base de su implementación la distribución del equipo *UvA Trilearn*. Esto permite afirmar que las habilidades básicas de los agentes son iguales en ambos equipos, diferenciándose en la estrategia de alto nivel que utilizan. Por otra parte, la versión del equipo *Borregos* utilizada para estas pruebas es el resultado de un trabajo de postgrado cuya dedicación y esfuerzo fueron similares al de este trabajo. Los equipos más destacados de la liga presentan un promedio de 5 años de experiencia, logrando un nivel de competitividad muy alto que dificulta la comparación directa con *Forrest*. Por último, y luego de analizar su comportamiento, se detectó que *Borregos* realiza una marca *hombre a hombre*² en su zona defensiva, dificultando así las posibilidades de juego de su adversario. Se consideró interesante enfrentar a un equipo con estas dificultades ya que permite evaluar la capacidad y efectividad al coordinar acciones ante situaciones adversas como esta, principalmente al realizar los pases.

²Cada agente de la defensa se ocupa de marcar a un solo jugador adversario.

3.2.1. Evolución funcional v3.1 - v4.0

Caso 6	
Objetivo	Determinar el efecto de la técnica de GC sobre el equipo Forrest. Como se va a evaluar el objetivo: Se va a evaluar a través del efecto que tenga esta técnica sobre el rendimiento del equipo. El rendimiento se evalúa a través de los indicadores de la sección 1.2.
Estrategia	Enfrentamientos: Forrest v3.1 vs Forrest v4.0 Cantidad de partidos: 20

Caso 7	
Objetivo	Es el mismo objetivo que el caso 6 pero a través de un tercer equipo.
Estrategia	Enfrentamientos: Forrest v3.1 vs Borregos y Forrest v4.0 vs Borregos. Cantidad de partidos: 20 (de cada serie)

3.2.2. Evolución funcional v3.1 - v5.0

Caso 8	
Objetivo	Determinar el efecto de la técnica de planificación en línea de dos niveles sobre el equipo <i>Forrest</i> . Se va a evaluar a través del efecto que tenga esta técnica sobre el rendimiento del equipo. El rendimiento se evalúa a través de los indicadores de la sección 1.2.
Estrategia	Enfrentamientos: Forrest v3.1 vs Forrest v5.0 Cantidad de partidos: 20

Caso 9	
Objetivo	Es el mismo objetivo que el caso 8 pero a través de un tercer equipo.
Estrategia	Enfrentamientos: Forrest v3.1 vs Borregos y Forrest v5.0 vs Borregos. Cantidad de partidos: 20 (de cada serie)

3.2.3. Evolución funcional v4.0 - v6.0

Caso 10	
Objetivo	Determinar el efecto de la técnica de Planificación de 2 niveles sobre el equipo Forrest coordinado con GC. Se va a evaluar a través del efecto que tenga la combinación sobre el rendimiento del equipo. El rendimiento se evalúa a través de los indicadores de la sección 1.2.
Estrategia	Enfrentamientos: Forrest v4.0 vs Forrest v6.0 Cantidad de partidos: 20

Caso 11	
Objetivo	Es el mismo objetivo que el caso 10 pero a través de un tercer equipo.
Estrategia	Enfrentamientos: Forrest v4.0 vs Borregos y Forrest v6.0 vs Borregos. Cantidad de partidos: 20 (de cada serie)

3.3. Pruebas no funcionales

3.3.1. Performance

Caso 12	
Objetivo	Medir el tiempo de ejecución de la estrategia de la versión 4.0, 5.0 y 6.0
Estrategia	Modificar el programa forrestJugador para que, usando las bibliotecas de tiempo de C/C++, haga una marca de tiempo al inicio del algoritmo y al final de forma de medir el tiempo buscado. Para el correcto funcionamiento, el servidor debe estar seteado para funcionar en modo sincronizado. Enfrentamientos: Forrest v(6.0, 5.0, y 3.1) vs Borregos Cantidad de partidos: 1 (para cada enfrentamiento)

3.3.2. Carga o volumen

Caso 13	
Objetivo	Determinar el comportamiento del equipo frente de diferentes cargas del sistema.
Estrategia	Jugar una serie de partidos en una única máquina vs. jugar la misma serie de partidos en un red formada por dos máquinas. Enfrentamientos: Forrest v6.0 vs Forrest v3.1 Cantidad de partidos: 1

3.3.3. Esfuerzo

Caso 14	
Objetivo	Determinar cuales son los errores del equipo cuando el equipo se ve limitado por recursos del sistema, en este caso tiempo de cómputo.
Estrategia	En función del tiempo medido de la ejecución de la estrategia del equipo, setear el ciclo de ejecución del simulador a un tiempo menor y jugar un partido. La idea es poder conocer los límites del equipo. Enfrentamientos: Forrest v6.0 vs Forrest v3.1 Cantidad de partidos: 1

Caso 15	
Objetivo	Determinar cuales son los errores del equipo cuando el equipo se ve limitado por recursos del sistema, en este caso tiempo de cómputo.
Estrategia	Jugar un partido y observar el uso de memoria de los procesos en cuestión y analizar el comportamiento del equipo luego de que esta se agota. Enfrentamientos: Forrest v6.0 vs Forrest v3.1 Cantidad de partidos: 1

3.4. Otras pruebas

Caso 16	
Motivación	.Determinar el estado de Forrest en comparación con los equipos actuales
Objetivo	Evaluar la posición de Forrest en el ranking mundial. Se mide en función de los puntos obtenidos en el conjunto de encuentros jugados en cada caso.
Estrategia	Determinar los equipos de prueba, equipo1, equipo2, equipo3, donde: Equipo1= equipo entre los primeros 10 del ranking mundial Equipo2= equipo entre la posición 11 y 20 del ranking mundial Equipo3= equipo entre la posición 21 y 30 del ranking mundial Enfrentamientos: Se juego Forrest v6.0 contra cada uno de estos equipos. Cantidad de partidos: 5

Referencias

- [CANa] Canales, R.; Casella, S.; Rodríguez, P. "*Descripción de Versiones del Sistema*". Tejera, G. (tutor). Trabajo de Grado. Facultad de Ingeniería. Instituto de Computación - Montevideo, Marzo 2006.
- [CANb] Canales, R.; Casella, S.; Rodríguez, P. "*FLACS - Forrest Log Analyzer Consultancy Service*". Tejera, G. (tutor). Trabajo de Grado. Facultad de Ingeniería. Instituto de Computación - Montevideo, Julio 2006.
- [FRA] Frank, I.; Katsuto A.; Matsubara, H.; Tanaka-Ishii, K. "*The statistics proxy server*". En: Peter Stone, Tucker Balch, and Gerhard Kraetschmar, RoboCup-2000: Robot Soccer World Cup IV, pag. 303-308. Springer Verlag, Berlin, 2001.
- [HED] Hedayati, A; Samimi, R.; Tabari, E.; Zareian, A. *Team Assistant 2003*. Disponible en: <http://www.sbcee.net/pres/index.htm> [Consulta: febrero 2006].