PROTOTIPO DE UN JUEGO DE SIMULACIÓN SOCIO-POLÍTICO MULTIJUGADOR BASADO EN DINÁMICA DE SISTEMA

CÉSAR AUGUSTO POVEDA FLORÉZ
CHRISTOPHER MICHAEL CONTRERAS SERRANO

UNIVERSIDAD AUTONOMA DE BUCARAMANGA
INGENIERIA DE SISTEMAS
DINAMICA DE SISTEMAS
BUCARAMANGA
2007

PROTOTIPO DE UN JUEGO DE SIMULACIÓN SOCIO-POLÍTICO MULTIJUGADOR BASADO EN DINÁMICA DE SISTEMAS

CÉSAR AUGUSTO POVEDA FLORÉZ
CHRISTOPHER MICHAEL CONTRERAS SERRANO

Trabajo de grado presentado para optar el título de:
Ingeniero de Sistemas

Director:
PhD. Ricardo Sotaquira

UNIVERSIDAD AUTONOMA DE BUCARAMANGA
INGENIERIA DE SISTEMAS
DINAMICA DE SISTEMAS
BUCARAMANGA
2007

Nota de aceptación:

_____
_____
_____
_____
_____
_____
_____

_____
Firma del presidente del jurado

_____
Firma del jurado

_____
Firma del jurado

Bucaramanga, 15 de Abril de 2.010.

CONTENIDO

pág.

INTRODUCCIÓN 12

1. ANTECEDENTES 13

2. MARCO TEORICO Y ESTADO DEL ARTE 15

2.1. DINAMICA DE SISTEMAS 15

2.1.1. Juegos de Simulación basados en modelos de Dinámica de Sistemas.....15

2.1.1.1. Fish Banks.. 16

2.1.1.2. Beer Game.....16

2.1.1.3. B & B Enterprises Management Flight Simulator.. 16

2.1.1.4. International Oil Tankers.. 17

2.1.1.5. Greenhouse Gas Emissions Simulator.. 17

2.1.1.6. People Express.. 17

3. INGENIERIA DE SOFTWARE.....19

3.1. ETAPAS DEL CICLO DE VIDA DEL DESARROLLO SOFTWARE.....19

3.1.1. Comunicación.....19

3.1.2. Planeación.....20

3.1.3. Modelado.....20

3.1.4. Construcción.....20

3.1.5. Despliegue.....20

3.2. CICLOS DE VIDA DE UN PROYECTO SOFTWARE.....20

3.3. MODELO EN ESPIRAL 21

4. DISEÑO DE JUEGOS 22

4.1. JUGABILIDAD 22

4.2. NIVELES Y FASES DEL DISEÑO.....23

5. DISEÑO SOFTWARE.....24

5.1. UML 24

5.1.1. Diagramas de estructura.. 25

5.1.2. Diagramas de comportamiento.....25

5.1.3. Diagramas de Interacción.....25

6. CONEXIÓN A INTERNET 27

6.1. ARQUITECTURA DE PROTOCOLO TCP/IP 27

6.2. LAS CAPAS DE TCP/IP 27

6.2.1. Capa Física. 27

6.2.2. Capa de acceso a la Red. 28

6.2.3. Capa de Internet. 28

6.2.4. Capa de Transporte.....28

6.2.5. Capa de Aplicación.....28

6.3. FUNCIONAMIENTO DE TCP/IP 29

6.4. CONEXIÓN DEL JUEGO 29

7. EN QUE CONSISTE LA VPN? 32

8. CARACTERISTICAS JUEGO DE LA PIANGUA 33

8.1. INTERFAZ GRAFICA 33

8.2. DESCRIPCIÓN DEL JUEGO 36

8.3. FUNCIONALIDAD EN-LINEA 36

9. SOFTWARE DESAROLLADO (CLASES) 39

9.1. CLASE JUGADOR 39

9.2. CLASE PIANGUA 40

9.3. CLASE LARVA 42

9.4. CLASE JUVENIL 43

9.5. CLASE ADULTO 43

9.6. CLASE CONEXIÓN 45

10. CONCLUSIONES 47

BIBLIOGRAFÍA 49

LISTA DE TABLAS

pág.

Tabla 1. Características de los Juegos existentes 18

LISTA DE FÍGURAS

pág.

Figura 1. Modelo en Espiral 21

Figura 2 Menú Principal Juego de La Piangua 33

Figura 3. Interfaz Tablero del Juego 35

Figura 4. Interfaz Servidor Juego de La Piangua.....37

RESUMEN

Este proyecto nace de la necesidad de implementar una interfaz grafica que permita mejorar la interacción del usuario con un Juego basado en un modelo de dinámica de Sistemas, además que sea multijugador y no necesariamente sobre una misma máquina. Este modelo fue desarrollado por el Grupo de Pensamiento Sistémico de la UNAB, y consiste en la simulación de la situación actual de una comunidad residente en las costas de Colombia donde su supervivencia depende exclusivamente de la extracción de un recurso común llamado Piangua.

Para ello se implementa un motor de videojuego llamado StarTrooper, que nos permite llevar a cabo tanto la implementación como el manejo de la parte visual y la parte sonora del Juego. A su vez también se está desarrollando código para integrar la funcionalidad multijugador por medio del programa Hamachi que nos facilite la creación de una red virtual LAN, con la cual se pueda jugar en modo multijugador a través de internet sin la necesidad de que todos los jugadores estén ubicados en una misma máquina.

Lo que se busca finalmente con el desarrollo de este proyecto, es mejor la interacción con el juego por parte de los usuarios, y permitir así llevar a cabo un estudio de los resultados más próximo a la realidad; ya que gracias a esto no solo el juego se quedara en un modelo, sino que además se podrá entender la situación de una forma más clara ya que el jugador se verá más inmerso dentro del juego y no solo obteniendo datos numéricos sino visuales, y eso gracias a la interfaz grafica.

Línea de Investigación: Pensamiento Sistémico.

Palabras Claves: Piaguar, C# 2005, Dinámica de Sistemas, Ingeniería de Software

INTRODUCCIÓN

El grupo de pensamiento sistémico de la Universidad Autónoma de Bucaramanga, se encuentra desarrollando un juego basado en un modelo socio-político de una situación real que sucede en las costas de Colombia, donde una comunidad se sostiene en base a la extracción de un recurso común. Lo que se busca con este juego, es analizar el comportamiento de esta comunidad en base a unas políticas que establece alguna entidad gubernamental. Es necesario tener en cuenta que este simulador puede ser utilizado por personas poco conocedoras de la metodología aplicada, en este caso Dinámica de sistemas. Por ello, se ve la necesidad de implementar la ingeniería de software, para llevar a cabo el desarrollo de una interfaz gráfica, que complemente el juego y así mismo permita ver de manera más fácil, el comportamiento del sistema antes mencionado. Se busca a su vez tener una funcionalidad de multijugador y que se pueda utilizar por Internet. Este simulador está siendo desarrollado con un lenguaje de programación que soporta orientación a objetos llamado C# 2005 y con un motor de juegos llamado Startrooper facilitado por la empresa DigiPen.

El objetivo principal del proyecto es la construcción del prototipo del juego ya mencionado que tenga como soporte el modelo socio-político proporcionado por el Grupo de Pensamiento Sistémico de la UNAB y que este prototipo permita participar a los usuarios a través de Internet, ya que este juego está orientado a una comunidad educativa y se busca que el mismo juego sin la necesidad de estar en una localidad, pueda ser jugado a través de Internet, como si estuviesen en el sitio de ejecución. Esto facilitaría al momento de demostrar el juego a diferentes personas ya que no se ve la necesidad de estar presente en un mismo cuarto todos los jugadores y sus decisiones van a ser más propias de él debido a que cada jugador no va poder ver las decisiones de los demás jugadores.

1. ANTECEDENTES

Este proyecto surge en base a un juego de simulación de carácter socio-político denominado *"El Juego de La Piangua",* que está siendo desarrollado por el grupo de investigación de pensamiento sistémico de la UNAB; donde se ve reflejado el comportamiento de una comunidad del pacífico colombiano que se sostiene en base a la explotación de un recurso común, para llevar a cabo su estudio y así obtener resultados sobre ello.[1]

A partir de lo anterior se ve una necesidad de complementar el juego, debido a que carece de un componente multi-jugador a través de Internet. Además se ve la utilidad de usar una interfaz gráfica apropiada, para llevar a cabo su ejecución con varias personas simultáneamente y así mejorando su interactividad; todo lo anterior aplicando ingeniería de software.

En este momento se tienen proyectos ya realizados como lo son el del Banco de Bagres [BANCOBA], un juego que implica el estudio de una problemática socio-económica, donde se veía la necesidad de llevar a cabo la implementación de la ingeniería de software para mejorar la interacción con el mismo. A diferencia del proyecto antes mencionado, esta tesis, involucra una problemática socio-política, que requiere de una investigación un poco diferente, debido a las decisiones que se deben tomar en la ejecución del modelo. Además, se espera poder realizar una adaptación Web que permita la función de multijugador; y desarrollarle una interfaz que cuente con un grado de interactividad mayor, que permita una presentación más ordenada y sencilla de los resultados obtenidos de la ejecución del juego,

[1] Ariza, Gerly Carolina; Serrano, Maria Cristina; Sotaquirá, Ricardo Una experiencia de diseño de juegos de simulación para el aprendizaje social 4o. Encuentro Colombiano de Dinámica de Sistemas Bogotá, 2006

teniendo en cuenta que actualmente contamos con más herramientas apropiadas para este tipo de desarrollos software.

Otra de las razones que se tuvo en cuenta para plantear este proyecto, es el interés sobre el desarrollo software, un área de la ingeniería de sistemas bastante competitiva en el mercado y sobre todo en un área especifica como lo es el desarrollo de juegos de computadora. Esta fue una de las mayores razones para plantear este proyecto, además porque a lo largo de la carrera se han estudiado varias temáticas relacionadas con el desarrollo software que tienen que ver con el uso de lenguajes de programación, tales como Java, C, C++, PHP y SQL.

2. MARCO TEORICO Y ESTADO DEL ARTE

En base a lo planteado para el desarrollo de este proyecto, es necesario tener en cuenta un conocimiento previo sobre ciertas temáticas para llevar a cabo su elaboración. Este proyecto se fundamenta básicamente en un área del conocimiento, que hace parte de la ingeniería de sistemas, la Ingeniería de software; más específicamente el diseño de juegos. Sin embargo también es necesario conocer sobre ciertas temáticas que apoyan este proyecto como lo es la dinámica de sistemas, conexión a internet y redes VPN. Para así, poder entender mejor el alcance del proyecto y el propósito de su desarrollo.

2.1. DINAMICA DE SISTEMAS

La dinámica de sistemas es un lenguaje de modelado de sistemas complejos existentes en la sociedad, que requieren de un estudio para analizar su comportamiento. Para llevar a cabo su aplicación utiliza como herramienta base la simulación por computadora y los modelos causales con el fin de analizar más fácilmente los posibles resultados.

Sin embargo, el objetivo principal con el uso de este lenguaje, es llevar a cabo un aprendizaje eficaz por parte de las organizaciones sociales, mediante el uso de modelos de la realidad, denominados micro mundos con los cuales se pueda experimentar y así analizar las principales variables que afectan al sistema.

2.1.1. Juegos de Simulación basados en modelos de Dinámica de Sistemas.

Algunos juegos que se encuentran disponibles en el mercado y que abarcan la

toma de decisiones mediante el uso de dinámica de sistemas, están relacionados en su mayoría con problemáticas empresariales o socio-económicas y algunos para ver el efecto de ciertas decisiones en alguna situación en particular. Algunos de esos juegos que se podrían mencionar con dichas características son los siguientes:

2.1.1.1. Fish Banks. Fish Bank es un juego de simulación por role en que cada equipo maneja una empresa pesquera. Cada participante busca aumentar sus activos que en este caso su flotilla a través de la explotación de un recurso renovable y la competencia. Este recurso explotado puede llegar a un estado crítico donde el recurso tarda mucho en regenerarse lo cual genera pérdidas en las empresas pesqueras.

2.1.1.2. Beer Game. The Beer Game es un juego de simulación por role desarrollado por la MIT donde se puede ver claramente la ventaja de un acercamiento integral para el manejo de una cadena de recursos; esto particularmente nos muestra la importancia de compartir la información a través de los componentes de esta dicha cadena.

2.1.1.3. B & B Enterprises Management Flight Simulator. Este juego consiste en la introducción de un nuevo producto al mercado, donde existe una competencia que quiere tomar posesión de la participación en el mercado de dicho producto. El jugador es responsable de cada decisión de la empresa tanto en publicidad como en producción. La interfaz de este juego consiste en la introducción de datos en un formulario para posteriormente ver un resultado a través de uno gráficos y además datos numéricos de los ingresos y egresos de la empresa.

2.1.1.4. International Oil Tankers. Es un juego que tiene una similitud con el Fish Bank [2] ya que el jugador debe tomar decisiones en base a un recurso común explotado. El recurso tratado en este juego es el petróleo; el jugador decide si comprar o no un nuevo barco que es el medio necesario para llevar a cabo la extracción del dicho recurso. El problema en este caso son los cambios imprevistos sobre la demanda del recurso al transcurrir del tiempo y a su vez la tardanza en la construcción de un barco. La interfaz de este juego es netamente formularios e información gráfica a través de un plano cartesiano.

2.1.1.5. Greenhouse Gas Emissions Simulator. Juego informativo que trata de mostrar la relación entre la emisión de gases efecto invernadero y la concentración de gases efecto invernadero en la atmósfera. Además también se puede percibir unas ciertas tasas con respecto al calentamiento global. El jugador debe experimentar con diferentes situaciones para luego ver el comportamiento de los datos introducidos en el experimento.

2.1.1.6. People Express. Juego donde el participante decide como es la tasa de crecimiento, los precios, y la cantidad de publicidad respecto a la gestión de una aerolínea. Dependiendo de estos datos el participante puede incrementar la demanda o perder ante la competencia. El verdadero objetivo de este juego es no ganar o perder sino dar una perspectiva más amplia de la complejidad de la coordinación en un mercado creciente. También se puede ver la dinámica entre la firma y sus competidores. Este simulador es considerado como un laboratorio donde uno puede tomar varias estrategias sin arriesgar la fortuna de una firma. La interfaz usada por este juego es muy sencilla en el sentido de mostrar la

[2] Meadows, D., Fiddaman, T., Shannon, D., "Fish Banks, Ltd. A microcomputer-assisted simulation teaching the principles for sustainable management of renewable resources," The International Network for Resource Information Centers. 1987.

información debido a que hace uso de formularios y gráficas en plano cartesiano para mostrar los resultados de dicha estrategia tomada.

A continuación podremos ver las características resumidas de cada juego en la siguiente tabla:

Tabla 1. Características de los Juegos existentes

Juegos	Caracteristcas							
	Disponible Web	En Linea	Por Internet	Formularios	Graficas Plano Cartesiano	Multi-Jugador	Interfaz 2 Dimensiones	Animacion
The Beer Game V. 3.0	SI	NO	SI	SI	SI	SI	NO	NO
Fish Bank	SI	NO	NO	SI	SI	SI	SI	SI
People Express	SI	NO	NO	SI	SI	NO	NO	NO
B&B	SI	NO	NO	SI	SI	NO	NO	NO
International Oil Tankers	NO	NO	NO	SI	SI	NO	NO	NO
Greenhouse Gas Emissions Simulator	SI	NO	SI	SI	SI	NO	NO	NO
Juego de la Piangua	SI	SI	NO	SI	NO	SI	SI	SI

Realizado por autores del proyecto a partir de investigación previa.

3. INGENIERIA DE SOFTWARE

Para llevar a cabo el desarrollo de cualquier aplicación software, es importante tener en cuenta un área de la ingeniería de sistemas como lo es la Ingeniería de Software. Que se encuentra definida como “el establecimiento y uso de principios sólidos de la ingeniería para obtener un software confiable y que funcione de modo eficiente en maquinas reales”. Básicamente la ingeniería de software nos permite llevar a cabo la gestión del proyecto a lo largo de su desarrollo, teniendo en cuenta unas técnicas de diseño y de aseguramiento de calidad. Además según lo que plantea la ingeniería de software se debe tener presente un marco de trabajo general que consta de 5 etapas que se aplican de acuerdo a un ciclo de vida escogido para llevar a cabo el desarrollo. Dichas etapas se encuentran planteadas de la siguiente manera: Comunicación, Planeación, Modelado, Construcción y Despliegue.

3.1. ETAPAS DEL CICLO DE VIDA DEL DESARROLLO SOFTWARE

A continuación se enumeran de forma general las etapas que hacen parte del ciclo de vida de un desarrollo software; las cuales deben ser tenidas en cuenta para cualquier desarrollo software.

3.1.1. Comunicación. Esta actividad del marco de trabajo implica una intensa colaboración y comunicación con sus clientes; además abarca la investigación de requisitos y otras actividades relacionadas.

3.1.2. Planeación. Esta actividad establece un plan para el trabajo de la ingeniería de software. Describe las tareas técnicas que deben realizarse, los riesgos probables, los recursos que serán requeridos, los productos del trabajo que han de producirse y un programa de trabajo.

3.1.3. Modelado. Esta actividad abarca la creación de modelos que permiten al desarrollador y al cliente entender mejor los requisitos del software y el diseño que lograra satisfacerlos.

3.1.4. Construcción. Esta actividad combina la generación de código y la realización de pruebas necesarias para descubrir errores en el código.

3.1.5. Despliegue. El software (como una entidad completa o un incremento completo de manera parcial) se entrega al cliente, quien evalúa el producto recibido y proporciona la información basada en su evaluación.

3.2. CICLOS DE VIDA DE UN PROYECTO SOFTWARE

"Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas".

Dicho en otras palabras, el ciclo de vida consiste en el proceso necesario que se debe llevar a cabo para el desarrollo de la aplicación software. En la actualidad, existen diversas propuestas o modelos planteados como lo son: Modelo en

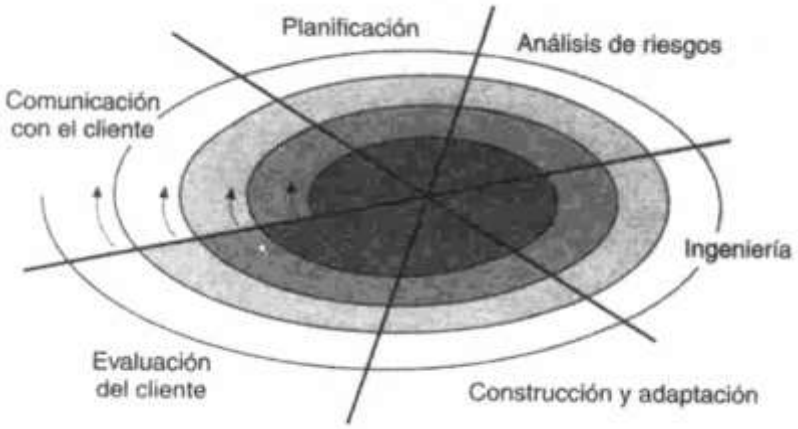
cascada, Modelo en espiral, Modelo de prototipos y Modelo De Desarrollo Incremental.

Para llevar a cabo el desarrollo de este proyecto se escogió el modelo en espiral debido a las especificaciones y retroalimentación que se desea hacer sobre el mismo al cabo de su desarrollo.

3.3. MODELO EN ESPIRAL

"Según Boehm, es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la construcción de prototipos con los aspectos controlados y sistemáticos del modelo en cascada". Este tipo de modelado se basa en una serie de entregas evolutivas, donde cada vez más se producen versiones más completas sobre el sistema desarrollado. Ver fig.1

Figura 1. Modelo en Espiral



PRESSMAN, Roger S. Ingeniería del Software: un enfoque práctico. 5 ed.

4. DISEÑO DE JUEGOS

El diseño de juegos es una metodología que define las etapas para diseñar el contenido, los antecedentes y las restricciones del juego como lo son las reglas del mismo. Los diseñadores de juegos profesionales se enfocan principalmente en ciertas clases de juegos tales como: Juego de mesa, juegos de cartas o videojuegos.[3]

4.1. JUGABILIDAD

La jugabilidad es parte del jugador cuando realiza su acción, que a vez se encuentra controlado por la interacción de otras mecánicas del juego. La jugabilidad busca como objetivo hacer el juego divertido e interesante para cada uno de los participantes tanto desde el punto de vista de los jugadores como el espectador, además de la diversión proporcionada busca también desafiar a los jugadores de diferentes formas tanto sicológicamente como físicamente.[4]

---

[3] WIKIPEDIA, Fundación Wikimedia [online]. Available from Internet: <URL:http://es.wikipedia.org/wiki/Dise%C3%B1o_de_juegos.html>.

[4] FEDERICO PEINADO'S WEBSITE, Federico Peinado [online]. Available from Internet: <URL:http://www.fdi.ucm.es/profesor/fpeinado/courses/gamedesign/index_es.html>

4.2. NIVELES Y FASES DEL DISEÑO

En la labor del diseñador encontramos tres niveles. Un nivel alto, donde se denotan las características atractivas del juego para atraer a los clientes, un nivel medio y bajo donde se definen especificaciones formales de acuerdo a los implementaciones del juego.

También encontramos tres fases de diseño: una fase de concepto, donde se define las diferentes características el juego tales como: el género, las reglas, los retos y la cantidad de niveles. Una fase de preproducción en la que se empieza el diseño de los niveles y todas las secciones del juego como son los niveles y las demás características mencionadas en la fase de concepto y como ultimo una fase de producción donde se implementa todo lo necesario para la buena ejecución del juego [5].

[5] FEDERICO PEINADO'S WEBSITE, Federico Peinado [online]. Available from Internet: <URL:http://www.fdi.ucm.es/profesor/fpeinado/courses/gamedesign/index_es.html>

5. DISEÑO SOFTWARE

Para llevar a cabo el diseño del proyecto, es necesario llevar a cabo en primer lugar un diseño de ingeniería de software, de tal manera que se muestre la parte funcional del software dependiendo del paradigma de programación que se esté utilizando.

Para nuestro caso en particular, nuestro paradigma es el orientado a objetos. Es decir, trabajamos sobre un lenguaje de programación que se encuentra orientado a objetos; lo cual significa que la base fundamental es trabajar con clases que serian la representación en lenguaje de programación de un objeto.

Cuando hablamos de objetos en programación, no es más que representar cualquier cosa del mundo real, con una identificación, unos atributos, y unos comportamientos. Cuando hablamos de la identificación, hacemos referencia al nombre del objeto, que sería la forma de llamarlo dentro del programa. Sus atributos, que serian las características que lo hacen único con respecto a otro objeto. Y finalmente unos comportamientos que hacen referencia a lo que puede hacer el objeto dentro del programa; es decir la función que tiene en el mismo.

5.1. UML

El lenguaje unificado de modelado es un lenguaje grafico muy utilizado en el mundo de desarrollo de software donde se puede visualizar, definir y componer una documentación para un software.

En UML tenemos diferentes tipos de diagramas:

5.1.1. Diagramas de estructura. En este diagrama describimos los elementos que deben de existir en el sistema que se está modelando.

5.1.2. Diagramas de comportamiento. Diagrama donde se define lo que va suceder en el sistema.

5.1.3. Diagramas de Interacción. Este diagrama se encarga de describir el comportamiento sobre el flujo de control y de los datos entre cada miembro del sistema.

El diagrama usado para el análisis y diseño de un sistema basado en clases se llama **Diagrama de clases**. En este diagrama uno enseña las clases del sistema con sus relaciones. Además de definir en forma general la clase también se profundiza tanto en sus atributos como sus operaciones. El diagrama de clases hace de los diagramas de estructura del UML.

Entre las definiciones de diagramas de clase tenemos:

- Las **propiedades** que también se conocen por el término de atributos, estos son valores que pueden ser color, cantidad, peso, etc. Es una información detallada del objeto con respecto a su característica física, en el caso de la programación se refiere a lo que está representando el objeto. Por ejemplo podemos tener un objeto carro en el cual tiene diferentes atributos como puede ser cantidad de puertas, color, motor, etc.

- Las **operaciones** son las acciones que pueden ejecutar un objeto ósea deben ser verbos como abrir, avanzar, cerrar, cargar, etc. Su denotación es en minúscula si solo está compuesta por una palabra en dado caso que fuese dos entonces la primera es en minúscula y la segunda pegada con su primer letra en mayúscula. Por ejemplo: avanzarAuto, abrirBaúl, etc.

- La **interfaz** permite al objeto comportarse de una cierta forma por medio de operaciones o propiedades definidas.

- **La herencia** es la reutilización de un objeto padre para extender la funcionalidad de un objeto hijo. Cuando un hijo hereda del padre el tiene en si las propiedades y operaciones del padre.

6. CONEXIÓN A INTERNET

Para llevar a cabo la conexión a la red por parte del Juego y que sea multi-jugador a través de la misma, se ha decidido implementar los protocolos TCP/IP. Esto se debe a algunas características que nos facilitan la interacción con el juego, y su funcionalidad.

6.1. ARQUITECTURA DE PROTOCOLO TCP/IP

Básicamente veremos de qué manera se hace la comunicación a través de la Red por medio de estos protocolos, y específicamente en que nos beneficia para el desarrollo de nuestro proyecto.

6.2. LAS CAPAS DE TCP/IP

En primer lugar es importante mencionar que el modelo de comunicación TCP/IP estructura el problema de la comunicación en cinco capas: capa física, de acceso a red, de internet, de transporte y de aplicación.

6.2.1. Capa Física. Hace referencia a la parte física que conecta a nuestra maquina (computador) con la red a la cual está conectado. En nuestro caso especifico, hace referencia al modem, o tarjeta de red que se encuentre conectada a la máquina de cada de cada Jugador y que le permita tener acceso a internet.

6.2.2. Capa de acceso a la Red. Hace referencia al software responsable del intercambio de datos entre el sistema final (Administrador del juego) y la red a la cual está conectado (internet) para posteriormente mandar los datos necesarios a cada Jugador.

6.2.3. Capa de Internet. Tiene que ver con las diferentes subredes que se encuentran interconectadas entre sí, y que nos permiten comunicarnos con otras maquinas sin necesidad de estar conectados directamente. Para ello se utilizan los llamados en caminadores, que son procesadores que conectan dos redes y cuya función principal es retransmitir datos de una red a otra siguiendo la ruta adecuada para alcanzar al destino.

6.2.4. Capa de Transporte. La función principal de esta capa es asegurar que los datos se intercambien de forma fiable. Para ello se centra en que todos los datos lleguen a la aplicación destino y en segundo lugar en el mismo orden en que fueron enviados.

6.2.5. Capa de Aplicación. Es la capa que contiene toda la lógica necesaria para posibilitar las distintas aplicaciones de usuario. En este caso en particular, estará dentro del Juego dicha lógica encargada de posibilitar la transferencia de datos entre los jugadores y el administrador.

6.3. FUNCIONAMIENTO DE TCP/IP

Antes que nada, para llevar a cabo la explicación del funcionamiento del protocolo TCP/IP, es necesario tener en cuenta ciertos conceptos claves como lo son el de subred. Cuando hablamos de subredes, nos referimos al conjunto de redes que conforman la comunicación en un momento dado. Es necesario tener en cuenta que para llevar a cabo la comunicación entre un computador y una subred existen ciertos protocolos que permiten hacerlo tales como Ethernet. La función de dichos protocolos es la de enviar datos a través de la subred a otro computador, o a un dispositivo de encaminamiento que los retransmita IP. Hay que tener en cuenta que TCP se implementa solamente en los sistemas finales, donde es necesario supervisar los datos para asegurar que todos se entregan de forma fiable a la aplicación apropiada.

Es necesario tener en cuenta además, que existen 2 niveles de direccionamiento. Cada computador en una subred debe tener una dirección de internet única que permita enviar los datos al computador adecuado, y cada proceso que se ejecute dentro de un computador debe tener, a su vez una dirección que sea única dentro del mismo. Para nuestro caso especifico, el juego como tal en cada computador manejara su propia dirección, y cada computador de cada jugador una dirección para la sub red.

6.4. CONEXIÓN DEL JUEGO

Para llevar a cabo la conexión entre los Jugadores y el administrador del mismo, se ve la necesidad de emplear además un software gratuito que nos facilita llevar a cabo dicha labor. Para ello se decidió entonces emplear Hamachi, que

básicamente es un software que nos permite crear una red virtual LAN, es decir simular una red de computadores conectados directamente entre sí.

Hamachi es un sistema central VPN (Mas adelante se explicara en detalle) que se centra en un servidor que es manejado por el distribuidor y el software del cliente que es instalado en el usuario final (cada jugador).

Básicamente para llevar a cabo su funcionamiento, el software del cliente (jugador y administrador) crea una interfaz VPN en el computador que sirve para interceptar tráfico entrante y saliente tipo VPN. El tráfico saliente del sistema operativo es enviado a través de la interfaz al software cliente, que es el encargado de autenticar y autentificarlo para luego enviarlo al nodo VPN para una conexión UDP.

Cada cliente (en este caso cada Jugador y el administrador) establece y mantiene una conexión controlada con el servidor. Cuando se establece la conexión, el cliente pasa por una secuencia de login, seguido de un proceso de descubrimiento y un estado de sincronización. En la secuencia de login se autentifica el cliente con el servidor y viceversa. El paso de descubrimiento es usado para determinar la topología de la conexión del cliente, específicamente para detectar la presencia de un NAT y un dispositivo firewall en la ruta hacia La Internet. En la parte de sincronización trae una visión de la red privada del cliente en sincronización con otros miembros de esta red.

Cuando un miembro de la red se conecta o se desconecta, el servidor le instruye a los otros nodos de la red establecer o destruir los túneles hacia ese usuario. Cuando se establece nodos, Hamachi usa una técnica transversal asistida por un servidor NAT. Información detallada de cómo trabaja esta técnica no ha sido publica aun.

Cada cliente Hamachi es asignado con una dirección IP desde la red 5.0.0.0/8. Esta dirección es asignada cuando un cliente se logea al sistema por primera vez y se asocia de aquí en adelante con la llave publica crypto del cliente. Siempre y cuando el cliente mantenga esta llave, puede logearse al sistema y usar esta dirección IP.

Esto permite a Hamachi ser una herramienta muy usada en los juegos y administración remota.

7. EN QUE CONSISTE LA VPN?

La **VPN** es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet.

El ejemplo más común es la posibilidad de conectar dos o más sucursales de una empresa utilizando como vínculo Internet, permitir a los miembros del equipo de soporte técnico la conexión desde su casa al centro de cómputo, o que un usuario pueda acceder a su equipo doméstico desde un sitio remoto, como por ejemplo un hotel. Todo esto utilizando la infraestructura de Internet.

Para nuestro caso en particular, nos va permitir la conexión entre los participantes del juego y el administrador. Simulando lo que nosotros denominamos una red LAN virtual.

8. CARACTERISTICAS JUEGO DE LA PIANGUA

8.1. INTERFAZ GRAFICA

El juego de la Piangua posee una interfaz grafica que cuenta con un menú principal donde el jugador tiene varias opciones para elegir tal y como se puede observar en la Figura 2.

Figura 2. Menú Principal Juego de La Piangua

Tomado del Software en ejecución.

En la Figura 2 se muestra el menú principal del juego, en el cual se puede apreciar un manglar que es el escenario principal, una de las ventajas de este menú es que es muy sencillo de entender y manejar ya que primero él color distintivo indica en donde está parado el jugador para así elegir su opción. Además la parte grafica del menú principal cuenta con una música de fondo muy suave para darle un enfoque más divertido que no solo capte los sentidos de la vista sino también los del oído.

Para llevar a cabo el desarrollo del juego el usuario podrá escoger entre un jugador y multijugador; para nuestro caso en particular la opción multijugador es la que permite llevar a cabo el desarrollo del juego a través de la red. Una vez dada la opción multijugador el usuario se va encontrar con una interfaz en donde en primera instancia debe consignar la dirección IP donde está el servidor y esperar a que se le asigne el turno. Finalmente teniendo en cuenta el turno que le correspondió podrá participar en el juego en el momento adecuado, es decir que no podrá jugar hasta que no sea su turno.

Figura 3. Interfaz Tablero del Juego

Tomado del Software en ejecución.

En la Figura 3 se puede apreciar el tablero de Juego que cada participante vera en su máquina y sobre el cual podrá participar dependiendo del turno asignado. Además podrá observar otras variables que son fundamentales para las decisiones que tome a lo largo del juego, tales como las variables de bienestar y de salud.

8.2. DESCRIPCIÓN DEL JUEGO

Dentro del juego cada jugador tendrá como objetivo lograr que su familia tenga el mejor bienestar posible. Teniendo en cuenta que únicamente podrá intervenir eligiendo un esfuerzo de trabajo para llevar a cabo el sustento de la misma.

Dicho esfuerzo tendrá un intervalo entre 1 y 8, donde 1 será el mínimo esfuerzo posible que puede hacer la familia y 8 el máximo. Al hablar de bienestar, nos referimos a una serie de aspectos que son vitales en la familia para su supervivencia tales como: alimentación, salud, educación y recreación. Los cuales serán representados en el juego, mediante unos iconos que nos muestran si dicho aspecto lo hemos alcanzado o no. Todo esto mediante un ingreso que se recibe a cambio del trabajo realizado con el esfuerzo.

Además un aspecto adicional del juego será la política que se implementara a partir de la ronda 10 el cual limita al jugador a no realizar un esfuerzo mayor al asignado por el servidor del juego, el cual será administrado por el coordinador del juego; aleatoriamente se escogerán a partir de la ronda 10 dos jugadores si alguno de ellos incumple la política será penalizado.

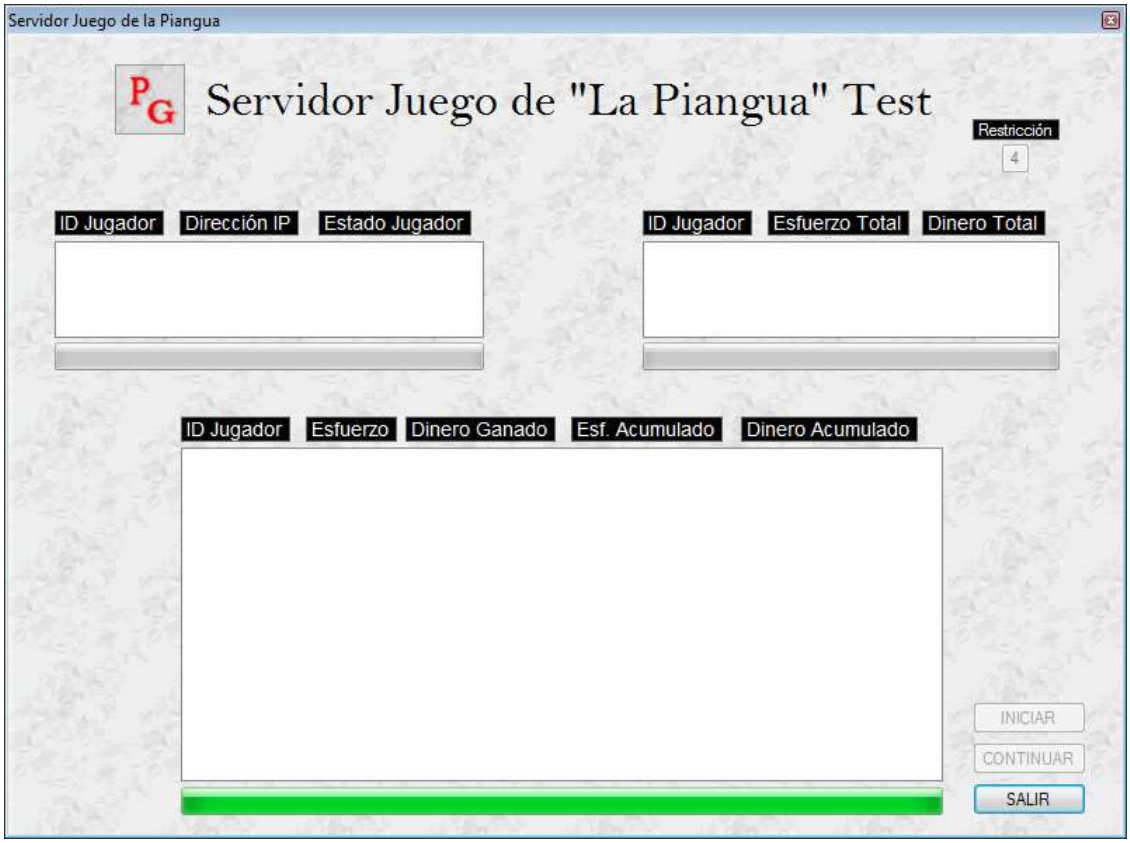
Finalmente es importante tener en cuenta que el juego se llevara a cabo durante 20 rondas, y que el mejor será quien quede con el mejor bienestar posible.

8.3. FUNCIONALIDAD EN-LINEA

Como se menciono en la sección de la interfaz grafica, el jugador le será asignado su turno por medio del servidor, esta característica está asociada a la

funcionalidad de Red que posee el juego. El servidor es otra aplicación desarrollada que estará manejando un coordinador del juego y que por ende deberá estar instalada en otra máquina aparte de la de los 5 jugadores. En esta pequeña aplicación se podrán observar en detalle todas las decisiones tomadas por los jugadores y el desempeño de ellos en cada una de las rondas. Además el servidor es el encargado de sincronizar la conexión entre los 5 jugadores con el juego.

Figura 4. Interfaz Servidor Juego de La Piangua



Tomada de la interfaz del Servidor del Juego de la Piangua.

Esta funcionalidad está incorporada en el juego para no ser necesario jugar en una misma máquina todos los jugadores sino que sin importar el lugar donde estén puedan hacerlo; esto con ayuda de un programa llamado Hamachi que es gratis en su versión básica y una conexión de Internet, más específicamente en una red virtual gracias a las características del programa Hamachi; teniendo en cuenta obviamente la ejecución del servidor del juego, ya que sin él no habría comunicación entre los jugadores.

9. SOFTWARE DESAROLLADO (CLASES)

Para el desarrollo del Juegos se programaron las siguientes clases:

9.1. CLASE JUGADOR

Esta clase es la encargada de hacer las operaciones correspondientes a las acciones realizadas por el jugador, como son las de calcular el pago y el esfuerzo.

Los atributos que pertenecen a esta clase son los siguientes:

- **pagoTotal** es una variable de tipo doublé de nivel de acceso público y es la encargada de almacenar el pago total acumulado del Jugador.

- **pagoActual** es una variable de tipo doublé de nivel de acceso público y es la encargada de almacenar el pago actual del Jugador.

- **esfActuales** es un vector de tipo entero que permite almacenar todos y cada uno de los esfuerzos realizados por el jugador durante todas las rondas del juego.

- **pagosActuales** es un vector de tipo entero que permite almacenar todos y cada uno de los pagos recibidos por el jugador durante todas las rondas del juego.

- **esfuerzoTotal** es una variable de tipo entero con un nivel de acceso público y es la encargada de acumular el esfuerzo de todos los jugadores durante la ronda actual para así tener el esfuerzo total de la ronda.

- **esfuerzoActual** es una variable de tipo entero con un nivel de acceso público y es la encargada de tener el valor del esfuerzo actual del Jugador.

- **Ubicación** es una variable de tipo entero con un nivel de acceso público y es la encargada de llevar el número de la ronda en la cual está participando cada jugador.

La operación o método que pertenece a esta clase es:

- **Pago** es una operación que recibe como parámetros la cantidad capturada de pianguas (***cantCapturaTotal***) que es de tipo doublé, el esfuerzo total de la ronda (***esfuerzoTot***) que es de tipo flotante y como último parámetro el indicador de ronda con penalización (***penalizado***) que es de tipo bool. Con parámetros el método pago se encarga de calcular el pago actual y total del jugador teniendo en cuenta si ha sido penalizado o no.

9.2. CLASE PIANGUA

Esta clase es la encargada de almacenar la información de cualquier tipo de Piangua.

Atributos que pertenecen a esta clase son los siguientes:

- **cantidad** es una variable de tipo double de nivel de acceso público y es la encargada de definir la cantidad de Piangua que hay en un momento determinado.

Las operaciones o métodos que pertenecen a esta clase son:

- **Constructor** vacio que no hace ninguna operación.

- **Constructor** que recibe como parámetros la densidad (***densidad***) que es de tipo doublé, el área (***area***) que es de tipo doublé y la tasa (***tasa***) que es de tipo doublé. Con estos parámetros no hace ninguna operación.

Los dos constructores anteriores no realizan ninguna operación pero van a ser necesarios al momento de ser heredados por otra clase.

- **Constructor** que recibe como parámetros la densidad (***densidad***) que es de tipo doublé, el área (***area***) que es de tipo doublé, cantidad de larvas (***cantLarva***) que es de tipo doublé y la cantidad de pianguas juveniles (***cantJuvenil***) que es de tipo doublé. Con estos parámetros el constructor define la cantidad de pianguas adultas.

9.3. CLASE LARVA

Esta clase es la encargada de almacenar la información inicial del tipo de Piangua larva específicamente. Esta clase hereda de la clase Piangua descrita anteriormente.

Los Atributos que pertenecen a esta clase son los siguientes:

- **tasaNacimiento** es una variable de tipo doublé de nivel de acceso público y es la encargada de definir el nivel de nacimiento de la Piangua.

Las operaciones o métodos que pertenecen a esta clase son:

- **Constructor** que es heredado de la clase Piangua que recibe como parámetros la densidad (***densidad***) que es de tipo doublé, el área (***area***) que es de tipo doublé y la tasa (***tasa***) que es de tipo doublé. Este constructor se encarga de inicializar los valores que pertenecen al tipo de piangua Larva.

- **Transformación** que recibe como parámetro la cantidad de pianguas adultas (***cantAdulto***) que es de tipo doublé. Esta operación se encarga de controlar la cantidad de Pianguas larvas que se transforman en juveniles y la cantidad que nacen.

9.4. CLASE JUVENIL

Esta clase es la encargada de almacenar la información inicial del tipo de Piangua juvenil específicamente. Esta clase hereda de la clase Piangua.

Las operaciones o métodos que pertenecen a esta clase son:

- **Constructor** que es heredado de la clase Piangua que recibe como parámetros la cantidad de larvas (***cantidad***) que es de tipo doublé. Este constructor se encarga de inicializar los valores que pertenecen al tipo de Piangua Juvenil.

- **Transformación** que recibe como parámetro la cantidad de Piangua tipo larva (***cantLarva***) que es de tipo doublé. Esta operación se encarga de controlar la cantidad de Piangua de tipo juvenil que se transforman en adulta y la cantidad que se transforma a juvenil.

9.5. CLASE ADULTO

Esta clase es la encargada de almacenar la información inicial del tipo de Piangua Adulto específicamente. Esta clase hereda de la clase Piangua.

Los atributos que pertenecen a esta clase son los siguientes:

- **tasaMortCaptura** es una variable de tipo doublé de nivel de acceso prívate y es la encargada de definir la tasa de mortalidad por captura.

- **cantMuerteCaptura** es una variable de tipo double de nivel de acceso público y es la encargada de almacenar la cantidad de Pianguas muertas por captura.

Las operaciones o métodos que pertenecen a esta clase son:

- **Constructor** que es heredado de la clase Piangua que recibe como parámetros la densidad (***densidad***) que es de tipo doublé, el área (***area***) que es de tipo doublé, cantidad de larvas (***cantLarva***) que es de tipo doublé y la cantidad de pianguas juveniles (***cantJuvenil***) que es de tipo doublé. Este constructor realiza la operación de su padre de quien fue heredado.

- **Transformación** que recibe como parámetro la tasa de mortalidad natural (***tasaMortNatural***) que es de tipo doublé y recibe como segundo parámetro la cantidad de Piangua de tipo juvenil actual (***cantJuvenil***) que es de tipo doublé. Esta operación se encarga de controlar la cantidad de Pianguas juveniles que se transforman en adultas y la cantidad que crecen a adultas.

- **Captura** que recibe como parámetro el esfuerzo total de la ronda (**esfTotal**) que es de tipo flotante. Esta operación se encarga de calcular la cantidad de Piangua Adulta muerta por captura.

9.6. CLASE CONEXIÓN

Esta clase es la encargada de habilitar la función de comunicarse con el servidor del juego.

Los atributos que pertenecen a esta clase son los siguientes:

- **alSockets** es una variable de tipo ArrayList de nivel de acceso público y a su vez estático y es la encargada de almacenar los sockets.

- **DireIp** es una variable de tipo String de nivel de acceso público y es la encargada de almacenar la dirección IP del servidor.

- **archivoRec** es una variable de tipo String de nivel de acceso público y es la encargada de guardar el nombre del archivo que fue recibido.

- **esperando** es una variable de tipo booleano con nivel de acceso público y es se utiliza de bandera para saber si un proceso está en espera.

Las operaciones o métodos que pertenecen a esta clase son:

- **Constructor** que no tiene parámetros pero es encargado de captar la dirección del host y almacenar esta información en la variable ***DireIp***.

- **enviarArchivo** que recibe como parámetros el archivo que se va enviar (***archivo***) de tipo String y la dirección a donde se envía (***Ip***) que es de tipo String. Esta operación es la encargada de enviar los archivos necesarios para la ejecución del juego y su interacción con el servidor.

- **listenerThread** no recibe ningún parámetro. Esta operación es encargada del hilo que se encarga de manejar el listener de la conexión TCP.

- **handlerThread** que recibe como parámetro el archivo que va manipular (***Archivo***) que es de tipo String. Esta operación es encargada de manejar información necesaria capturada por la conexión con el servidor para ser manipulada con un archivo.

- **recibirArchivo** tiene como parámetro el archivo que va ser guardado (***nombreArchivo***) es de tipo String. Esta operación es la encargada de asignar el valor de la variable **archivoRec.**

10. CONCLUSIONES

La interfaz grafica puede ser vista como una forma de expresión o lenguaje universal, que nos permite llevar el juego a la práctica sin tener en cuenta el idioma del Usuario. Es decir, la interacción usuario – juego no se limita al lenguaje del programador sino que es entendible para cualquier persona. Ya que el modo gráfico permite a los usuarios interactuar con mayor sencillez la aplicación en este caso el Juego de la Piangua, sin necesidad de entender mucho sobre el idioma natal sobre el que se desarrollo el mismo.

Por tanto, la interfaz va permitir que los usuarios utilicen de una forma más adecuada el juego y el aprendizaje sea mayor ya que el usuario tiene posibilidad de visualizar lo que está ocurriendo en el juego lo cual es un valor agregado para el mismo.

Otra ventaja importante que se vio con el desarrollo de la interfaz grafica del juego, consiste en el buen entendimiento del juego por parte de los participantes. Es decir con la implementación de esta interfaz los jugadores pueden fácilmente entender la situación problemática y cuál es su función en el juego. Si bien antes había una explicación, y el jugador podía imaginarse dicha situación, ahora puede observarla y entenderla de una mejor forma, ya que se encuentra explicita en la interfaz grafica. No solo entendiendo la problemática, sino que además el resultado inmediato de sus decisiones; objetivo fundamental del juego de la piangua “la toma de decisiones ante una situación problemática”.

Al llevar a cabo el desarrollo de la interfaz grafica del juego, se cambio un poco la temática con la que inicialmente se invento, con el fin de lograr un aprendizaje mas de carácter empírico que teórico; es decir que no esté implícito en el objetivo

del juego, en este caso llegar a la cooperación, sino que se aprenda con la experiencia del mismo.

Por lo tanto se decidió plantear un objetivo donde cada jugador debe buscar su bienestar para lograr su beneficio; sin embargo, con la participación en el juego, las experiencias vividas, y los conocimientos previos del jugador; creemos que el jugador puede llegar a concluir que dicho fin, solo se logra si se cuenta con la cooperación tanto de él como de los demás jugadores, que es la idea del juego original. Por lo tanto creemos, que aunque se desvía un poco el objetivo original del juego, dicho aprendizaje que se plantea en el juego original se logra al finalizar la participación en el mismo. Con la ayuda de los resultados obtenidos, y la retroalimentación que se pueda hacer al final con la tabla de resultados.

Finalmente podemos concluir que una de las grandes mejoras logradas en el juego con el desarrollo de la interfaz grafica es el uso de la red (internet) para llevar a cabo su ejecución, ya que gracias a esta mejora el juego no solo se limita a jugadores de la localidad sino que amplía el aprendizaje pudiéndolo utilizar a distancia con diferentes personas en diferentes lugares. El uso de la red nos permite no solo ampliar el conocimiento adquirido por el juego, sino que llevar aprendizaje a más personas con mucho menos esfuerzo, lo cual lo convierte en una herramienta portable de aprendizaje muy eficaz.

BIBLIOGRAFÍA

ANDRADRE SOSA, Hugo H.; DYNER R., Isaac; ESPINOSA, Angela; LÓPEZ GARAY, Hernán y SOTAQUIRA, Ricardo. Pensamiento Sistémico: diversidad en búsqueda de unidad. Bucaramanga: Universidad Industrial de Santander, 2001. 423 p.

[PRE02] PRESSMAN, Roger S. Ingeniería del Software: un enfoque práctico. 5 ed. Madrid: McGraw – Hill / Inter Americana de España, S. A. U., 2002. 601 p.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACIÓN. Compendio: Tesis y otros trabajos de grado. 5 ed. Bogotá: ICONTEC, 2002. 23 p.

[MAS99] Multi-AGENT SYSTEM TO MODEL THE FISHBANKS PLAY PROCESS [online]. Available from Internet:
<URL:http://cormas.cirad.fr/pdf/ceemas99.pdf>

[FB87] Meadows, D., Fiddaman, T., Shannon, D., "Fish Banks, Ltd. A microcomputer-assisted simulation teaching the principles for sustainable management of renewable resources," The International Network for Resource Information Centers. 1987.

[MASY76] MA SYSTEM [online]. Available from Internet:

<URL:http://www.masystem.com/beergame>

[WIKI06a] WIKIPEDIA, Fundación Wikimedia [online]. Available from Internet: <URL:http://es.wikipedia.org/wiki/Modelado.html>.

[WIKI06b] WIKIPEDIA, Fundación Wikimedia [online]. Available from Internet: <URL:http://es.wikipedia.org/wiki/Sistema_complejo.html>.

[VON76] VON BERTALANFFY, Ludwig. Teoría General de Sistemas: Fundamentos, Desarrollo y Aplicaciones. Bogota: Fondo de Cultura Económica. 1976. 312 p.

[WIKI06c] WIKIPEDIA, Fundación Wikimedia [online]. Available from Internet: <URL:http://es.wikipedia.org/wiki/Dise%C3%B1o_de_juegos.html>.

[FED06] FEDERICO PEINADO'S WEBSITE, Federico Peinado [online]. Available from Internet:
<URL:http://www.fdi.ucm.es/profesor/fpeinado/courses/gamedesign/index_es.html
>

[DPE06] DEPARTAMENTO DE ELECTRONICA E INFORMATICA [online]. Available from Internet:
<URL:http://www.dei.uc.edu.py/tai99/introsimulacion/simucon.html>

[GPS06a] Ariza, Gerly Carolina; Serrano, Maria Cristina; Sotaquirá, Ricardo Una experiencia de diseño de juegos de simulación para el aprendizaje social 4o. Encuentro Colombiano de Dinámica de Sistemas Bogotá, 2006

[GPS06b] Ariza, Gerly Carolina; Serrano, Maria Cristina; Sotaquirá, Ricardo Una experiencia de diseño de juegos de simulación para el aprendizaje social 4o. Encuentro Colombiano de Dinámica de Sistemas Bogotá, 2006

[GPS06c] Gerly Carolina; Modelo del juego de la Piangua

[BG05] Li, Michael; Simchi-Levi, David. The MIT Beer Game V.3.0.
Available from Internet:
<URL:http://beergame.mit.edu/default.htm>

[MBG02] Li, Michael; Simchi-Levi, David. Beer Game Guide.
Available from Internet:
<URL:http://beergame.mit.edu/guide.htm>