

DESARROLLO DE UN SIMULADOR DE EDIFICIO CON PARÁMETROS VARIABLES

DEVELOPMENT OF A BUILDING SIMULATOR WITH A VARIETY OF PARAMETERS

Andrés Enrique, Rosso Mateus

*Ingeniero de Sistemas, Universidad Distrital Francisco José de Caldas
Laboratorio de Automática, Microelectrónica e Inteligencia Computacional, LAMIC
Bogotá, Colombia, andresrosso@gmail.com*

José Jairo, Soriano Méndez

*Ingeniero Electrónico Universidad Distrital, Ingeniero Químico Universidad Nacional de Colombia,
Magíster en Ingeniería Eléctrica con énfasis en Control Automático Universidad de los Andes, Doc-
torando Programa de Ingeniería Química Universidad Nacional de Colombia. Docente Universidad
Distrital Francisco José De Caldas de Bogotá, Colombia, josoriano@udistrital.edu.co*

Fecha de recepción: 16 de enero de 2009

Fecha de aprobación: 11 de junio de 2009

RESUMEN

Este artículo expone la metodología de implementación de un simulador de edificio teniendo en cuenta factores de desarrollo como la escalabilidad del software, su ciclo de desarrollo y los detalles matemáticos usados para el modelamiento del sistema. Muestra además una metodología sencilla y eficaz para la simulación de cualquier tipo de sistema, en la cual se exponen las librerías de programación que se encontraron más útiles para este tipo de desarrollo. Este simulador representa un edificio real con todas las características y variables que interesan para el caso y se pretende usar para probar algoritmos de control para grupos de elevadores [10]. La arquitectura del simulador permite acoplar cualquier algoritmo de control que siga el estándar propio del simulador, se muestran como ejemplo dos algoritmos de control los cuales son ingresados en el simulador para probar su desempeño [10]. El simulador permite generar reportes estadísticos del desempeño de los controladores usados, estos reportes son útiles a la hora de comprobar la eficiencia de estos y de comparar cuantitativamente el funcionamiento entre ellos.

Palabras clave: simulador para edificio, simulación, análisis sistémico, controlador para grupo de elevadores.

ABSTRACT

This article exposes the methodology of implementation of a building simulator bearing factors of development in mind as the scalability of the software, its cycle of development, and the mathematical details used for the modeling of the system. This article shows a simple methodology and in addition effective for the simulation of any type of system. There are exposed of the same form the libraries of programming that were more useful for this type of development. The Simulator represents a real building with all the characteristics and variables that are interested for the case. This simulator tries to be used to prove algorithms of control for groups of elevators, the architecture of the simulator allows to connect any algorithm of control that follows the own standard of the simulator, there appear as example two algorithms of control which are connected to the simulator to prove its performance[8]. The simulator allows generating statistical reports of the performance of the used controllers; these reports are useful at the moment of verifying the efficiency of these algorithms and to compare quantitatively the functioning between them.

Keywords: building simulator, simulation, systemic analysis, elevator group control.

INTRODUCCIÓN

Cuando se quieren desarrollar aplicaciones que tendrán lugar en el mundo real, estas deben probarse de forma iterativa para verificar así su desempeño e identificar sus fallencias, estas pruebas idealmente se deben llevar a cabo en un escenario real, pero en algunos casos simplemente se hace imposible llevar a cabo dichas pruebas. Es impensable por ejemplo probar en un edificio de 100 pisos un algoritmo de control desarrollado para disminuir el tiempo de espera de los pasajeros, esto traería inconvenientes para los usuarios ya que en un principio puede que el algoritmo no se comporte como se esperaba, además de esto se hace muy costoso pues para obtener información relevante acerca del desempeño del simulador es necesario contabilizar cuantas personas usan el servicio, hacia qué pisos se dirigen éstos y además repetir las pruebas durante todo el día y quizás durante meses para obtener un universo de pruebas aceptable. La simulación es una técnica que permite construir modelos de un sistema real y operarlos en hipotéticas condiciones exteriores con el objetivo de predecir el comportamiento esperado del sistema real a partir de la información generada en el modelo.

La premisa es que ninguna parte sustancial del universo es tan simple como para comprenderse y controlarse sin abstracción, es necesario reemplazar la parte del universo bajo consideración, por un modelo de estructura similar, pero más simple [4]. En este artículo se desarrolla un simulador gráfico de edificio, que representa o simula cualquier clase de edificación y se ajusta a cualquier tipo de tráfico en ella. El simulador tiene la capacidad de variar la velocidad de ejecución –esto es 1 segundo de simulación equivale a n segundos en la realidad, variando n-, esto con el fin de permitirle al investigador hacer

seguimiento al desarrollo de la simulación o avanzar el tiempo rápidamente y dirigirse a las estadísticas de la simulación. Para comprobar el funcionamiento del algoritmo de control bajo estudio, el simulador tiene la capacidad de generar informes de desempeño que muestran mediante gráficos estadísticos el funcionamiento y la eficiencia propias de este algoritmo de control, este desempeño toma valor en el tiempo de espera de un pasajero para dirigirse de un piso a otro, y además en la energía total consumida por el sistema. El desarrollo de este sistema implica un alto grado de escalabilidad y de independencia entre sus componentes. Para garantizar estas premisas se sigue una metodología propia, que lleva a los resultados esperados. Acerca de este aspecto se habla con más detenimiento en el artículo, exponiendo la arquitectura y justificando su uso.

1. MATERIALES Y MÉTODOS

El primer simulador de edificio fue desarrollado para probar el controlador de grupo de elevadores de la torre World Trade Center [9], en este entonces los controladores para grupos de ascensores eran evaluados con simuladores híbridos, los cuales consisten en modelos análogos y digitales [12]. En estos simuladores las tarjetas de control de los elevadores eran conectadas con el simulador, y de esta forma se obtenían las entradas del sistema. Más adelante estos vendrían dentro de mini computadores en donde eran estudiados varios fenómenos de tráfico [2]. Recientemente los adelantos en la informática han permitido desarrollar simuladores de edificios que permiten estudiar de forma certera el comportamiento de tráfico y la viabilidad de los algoritmos de control estudiados [5].

El desarrollo de simuladores para edificio es un tema de grandes corporaciones que invierten gran capital para estos desarrollos, las empresas más sobresalientes en esta área son: Mitsubishi (Japón), Kone (Finlandia), Schindler (Suiza) [6]. Cada una de estas empresas ha desarrollado su propio simulador, el cual por razones de competitividad es protegido y su código fuente no es libre. Entre estas empresas la que más divulga su información es KONE, su laboratorio permite conocer ciertos detalles del simulador Advanced Lift Traffic (ALTS) [9], el cual se basa en simulación de eventos continuos con un cronómetro que orquesta los eventos de acuerdo a su intervalo de aparición, el principal objetivo de este simulador es la investigación sobre nuevos principios de control.

Así como en el simulador propuesto, en el ALTS [8] se puede variar el número de pisos, el número de ascensores y el tráfico que se presenta en la edificación, el inconveniente es que ya que es un simulador industrial, el código fuente no es liberado y por lo tanto no es posible que éste se pueda usar en proyectos de investigación fuera de Kone.

Es necesario aclarar que no existe ningún simulador para grupo de elevadores de código libre. El simulador expuesto en este artículo es libre en su distribución y modificación, está desarrollado para ejecutarse en cualquier computador personal y sobre

cualquier sistema operativo que tenga instalada la máquina virtual de java. Para poder acoplar un controlador al simulador se desarrolló una API que permite la integración al sistema de nuevos algoritmos haciendo uso una interfaz genérica. Otra característica importante es la posibilidad de obtener información estadística detallada del comportamiento del algoritmo de control durante la simulación, estos reportes le permiten al investigador ajustar con información certera el controlador para su máxima eficacia.

1.1. MODELIZACIÓN

El proceso de simulación puede ser definido como la imitación de la operación en un proceso real o un sistema sobre el tiempo [4]. El proceso de simulación comprende tres elementos fundamentales [4]: Sistema real, Modelo, Ordenador. Estos elementos están conectados como se aprecia en la figura 1. Lo que se busca con esto es llegar a un modelo de ordenador simplificado partiendo de un sistema real, para alcanzar esto se definen las etapas que se observan en la figura 2.

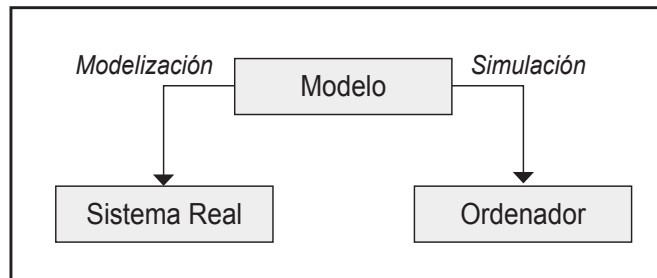


Figura 1. Elementos Básicos y relaciones en modelización y simulación [4].

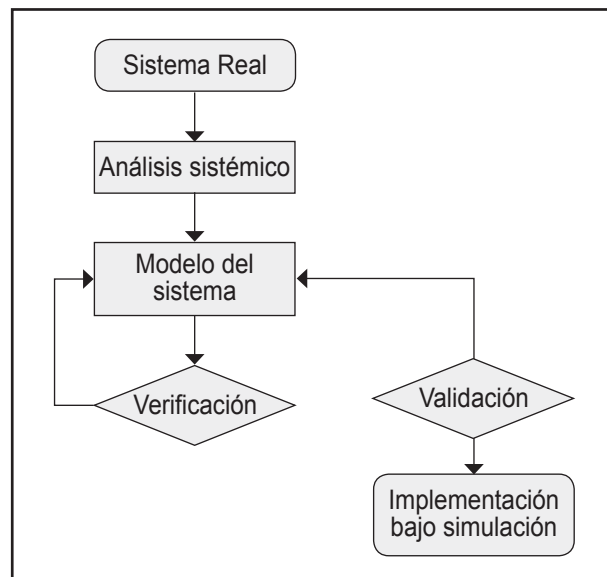


Figura 2. Diagrama de Flujo en la obtención del simulador de edificio [4].

El sistema real en este caso, es cualquier tipo de edificación, de ésta se obtienen las características esenciales y que son relevantes para la experimentación. El estudio de estas características se realiza usando técnicas de análisis sistémico con el cual se encuentran los elementos del sistema a simular y sus posibles relaciones. Una vez obtenido el modelo que representa el sistema real, es sometido a una verificación con respecto a su comportamiento particular, es posible que en el modelo se haya incurrido en errores de programación o errores en variables no controladas, lo que llevaría a corregir el modelo. Si el comportamiento del sistema modelado es consecuente con lo que se quería modelar, se pasa a la fase de validación en la cual se confrontará con el sistema real. Si esta etapa no es exitosa se debe corregir el modelo nuevamente.

1.2. CATEGORIZACIÓN DE LA SIMULACIÓN

El sistema de estudio hace que esta simulación sea de eventos continuos, dinámica y estocástica, a continuación se describen cada una de estas categorías.

Eventos Continuos: Este tipo de simulación [4] permite hacer el seguimiento de los procesos tal como si ocurrieran en tiempo real, es decir que el estado de las variables simuladas cambian paulatinamente de acuerdo al momento en el tiempo en el que se encuentre la simulación, no como en la simulación de eventos discretos [1] en la cual las variables tienen asociadas líneas de tiempo diferentes. En este modelo todas las variables siguen la misma línea de tiempo.

Dinámica: Ya que el interés es conocer el comportamiento del sistema a lo largo del tiempo, se utilizará la simulación dinámica [4].

Estocástica: El sistema tiene un comportamiento estocástico debido a que existen componentes aleatorios dentro del sistema. Una de las variables aleatorias que se encuentra es la tasa de arribo a determinado piso.

1.3. MODELO

El interés del simulador es representar de una manera sencilla y adaptable las características que más interesan de una edificación. Estas características se ingresan en un archivo XML como se ve en la figura 3, algunos de los parámetros ingresados en dicho archivo son: número de pisos del edificio, número de ascensores y el tráfico respecto al tiempo.

Estos datos alimentan el modelo de la simulación para crear el edificio con sus características y con la ayuda de un cronómetro se sincronizan los eventos que tengan lugar durante la simulación. Cuando la simulación se esté ejecutando, los datos de ésta, tales como las llamadas de ascensor y pasajeros generados, etc., se almacenan en una base de datos. Cuando el usuario del simulador desee generar algún reporte de la

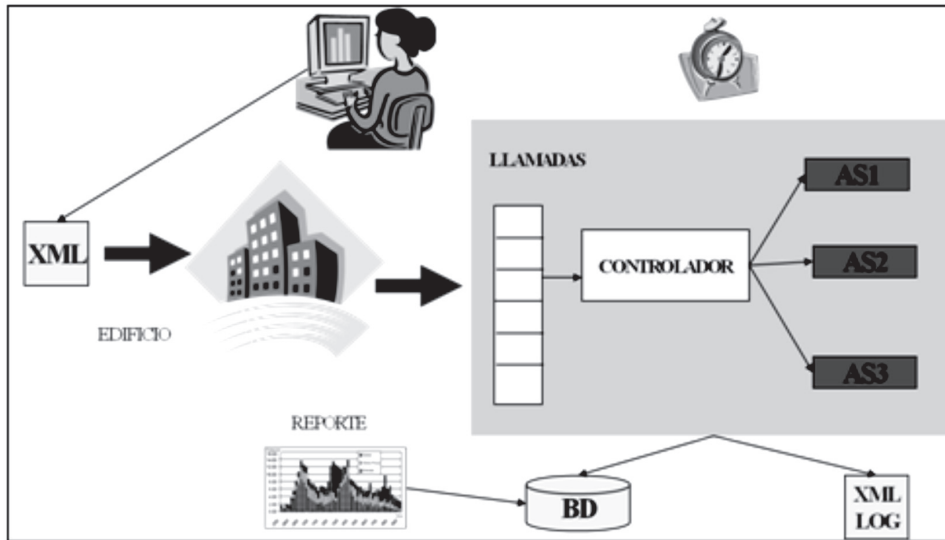


Figura.3. Modelo de la simulación.

simulación, tal como la media de los tiempos de espera de los usuarios, el componente de reportes obtiene la información de la base de datos y con esta generará el informe deseado, la simulación termina cuando el usuario quiera o hasta que termine el tiempo programado para ella.

El diagrama de flujo de la figura 4 representa la secuencia de ejecución del simulador. En la primera etapa -iniciar sistema- se obtienen los parámetros de la simulación, después de iniciada la simulación se evalúa si existen llamadas de ascensor encoladas, es decir, si hay pasajeros esperando por un ascensor, si nó, se crean pasajeros y estos se asignan según el tráfico respecto al tiempo, se aumenta seguidamente el tiempo de la simulación y si no ha llegado a su fin, vuelve el sistema a comprobar si hay llamadas encoladas. Como ya se generaron los pasajeros, estas llamadas se asignan a los ascensores por medio del controlador de tráfico, después de esto el ascensor es manipulado para moverse en el sentido de la llamada que tenga asignada, se guarda un registro estadístico en la base de datos con el estado de sistema y en seguida se grafica el edificio en el estado actual.

La simulación llega a su fin cuando el tiempo programado se acabe o cuando el usuario del simulador de la orden para detener la misma, el usuario puede en cualquier momento ver los reportes estadísticos de la simulación para comprobar la eficacia del algoritmo.

1.4. ARQUITECTURA

La consecución de aplicaciones de software complejas requiere de metodologías de análisis, diseño y desarrollo con las cuales se pueda ir incrementando la complejidad

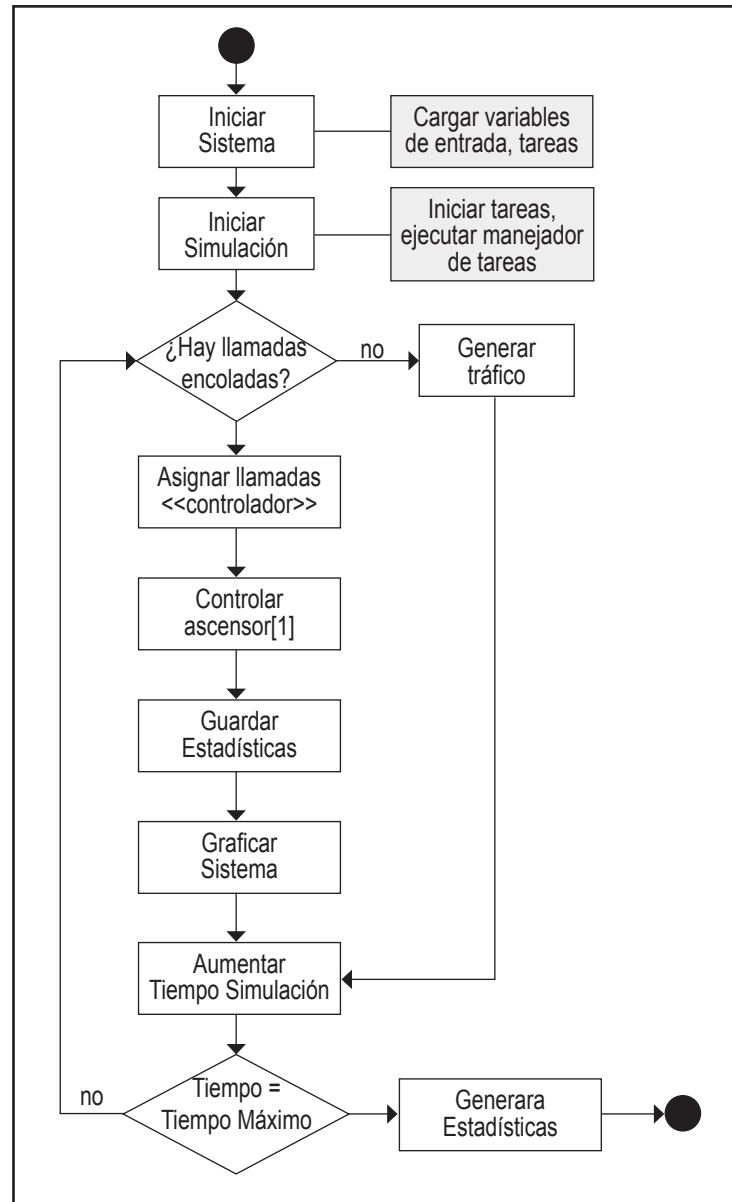


Figura 4. Diagrama de flujo de la simulación.

de la aplicación sin perder el control de la misma, bajo este enunciado fue conveniente usar un modelo de desarrollo en espiral en el cual cada iteración iba aportando la lógica necesaria para la construcción del todo como un sistema. Este modelo de desarrollo iterativo e incremental permite controlar el riesgo inherente en aplicaciones de gran tamaño, dividiendo en cada ciclo el objetivo a atacar para de esta manera reducir los puntos críticos más significativos. Los componentes del simulador más relevantes se listan en la Tabla 1.

Tabla 1. Componentes del simulador

COMPONENTE	OBJETIVO
Cronómetro	Es el reloj del sistema, la ejecución de las tareas están sincronizadas por él.
Generador de Tráfico	Genera los pasajeros dentro del edificio
Gráficos del Edificio	Mantiene un modelo visual de la simulación.
Controlador de Grupo	Algoritmo por el cual se asignan las llamadas de los pasajeros a los ascensores, su implementación está a discreción de cada investigador por lo que es posible cambiar el algoritmo de control. Se proponen en este caso un controlador difuso y uno convencional.[10]
Estadísticas	Módulo encargado de generar reportes para comprobar el funcionamiento de los algoritmos de control, estos reportes están definidos, pero se pueden ampliar.
Control de Movimiento	Encargado de ejecutar las tareas como: subir ascensor, bajar, abrir puerta de ascensor, cerrar puerta.
Manejador de Tareas	Componente encargado de administrar la ejecución y el ciclo de vida de las tareas.

Para hacer un desarrollo modular en espiral, es necesario hacer que sus componentes sean independientes y que la implementación de estos se lleve a cabo en diferentes fases. Para lograr esta independencia y no perder el control de los mismos se ideó un modelo basado en tareas, en donde cada tarea será supervisada y administrada por otro componente que será el manejador de tareas.

Una tarea es un concepto lógico establecido para reunir alguna funcionalidad en él y que se ejecuta cada cierto periodo de tiempo. El concepto de tarea permite reducir la complejidad y desacoplar los componentes identificados en una aplicación de software dividiendo la lógica sistémica en tareas organizadas y administradas por un manejador de tareas.

Esta arquitectura se aprecia en la figura 5, se observa que de la clase padre *Tarea* heredan *Cronómetro* y *Generador Tráfico*, esta clase padre tiene como atributos el tiempo de activación, el cual consulta el manejador de tareas para saber cada cuánto tiempo se debe ejecutar determinada tarea.

El manejador de tareas no debe saber qué lógica va a ejecutar cada una de las tareas previamente agregadas, el sólo invoca el método a ejecutar y dentro de este método estará el código que realice la acción deseada, para ilustrar este proceso se incluye un fragmento del código de la clase cronómetro y su función a ejecutar (figura 6).

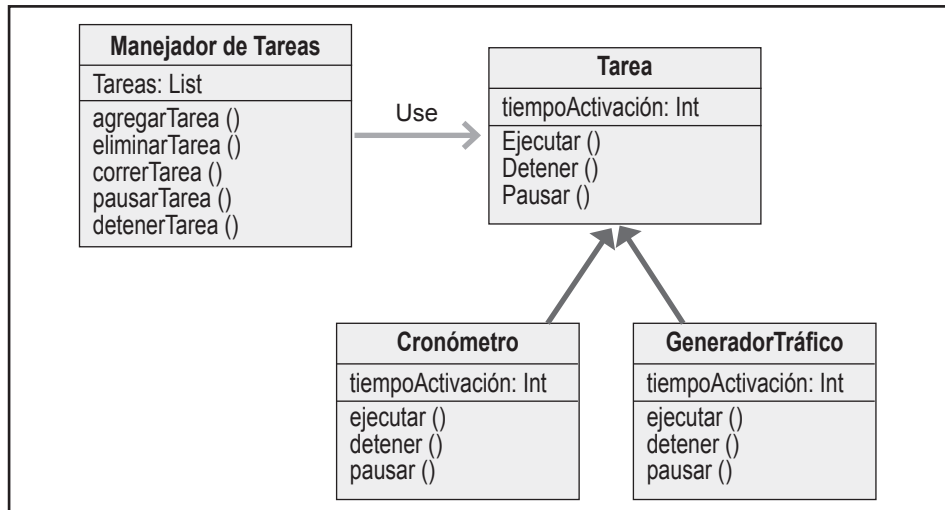


Figura 5. Diagrama de clases modelo por tareas

```

public class Cronometro extends Task{

    int segundos=0;
    int tiempoActivacion=1;

    void ejecutar() {
        segundos++;
    }
}
  
```

Figura 6. Fragmento de código para la ejecución de tareas

Estos componentes hacen uso de las librerías de java, todas ellas open source:

Base De Datos: El motor Hipersonic Hsqldb, de alto desempeño y puede ser empujado en la aplicación.

Gráficas Estadísticas: JFREECHAT, permite crear gráficos estadísticos de una forma sencilla y con un diseño enriquecido.

Reportes: IREPORT, permite manejar reportes profesionales separando el diseño y los datos de estos mismos.

Motor Matemático: COLT, especializada en métodos matemáticos.

Motor XML: Xerces, implementación parser XML.

En la figura 7 se ve un diagrama con los componentes del simulador y la jerarquía en nivel de estos mismos.



Figura 7. Arquitectura del sistema.

1.5. GENERADOR DE TRÁFICO

Una parte muy importante en la simulación es representar la forma en la que arriban las personas al sistema, el componente que lleva a cargo esta tarea es el generador de tráfico el cual se activa cada cierto tiempo generando pasajeros para todos los pisos de acuerdo al tráfico presente. Es ya conocido, que un proceso estocástico de la vida real tal como la tasa de arribo sigue la distribución de Poisson [4] y su distribución de probabilidad está dada por la ecuación (1):

$$P = \frac{(\lambda T)^n e^{-\lambda T}}{n!}, T=0, 15, 30, 600 \quad (1)$$

Probabilidad de arribo (P) en cierto tiempo T, siendo λ la tasa de arribo de los usuarios al sistema. El tiempo que existe entre el arribo de dos pasajeros sigue una distribución exponencial [11], ecuación 2.

$$\frac{-\ln(r)}{\lambda}, \lambda > 0 \quad (2)$$

En donde r es un número aleatorio entre 0 y 1 y λ es la tasa de arribo expresada como un porcentaje de la población total.

Después de estimar el tiempo de arribo del siguiente pasajero, la siguiente tarea consiste en establecer el piso origen y el piso destino de la llamada, estas variables aleatorias siguen una distribución normal. De esta forma se generan los siguientes datos: número de pasajeros para los siguientes 5 minutos, tiempo de arribo de estos pasajeros en el intervalo de 5 minutos, piso origen de estos pasajeros, piso destino de estos pasajeros.

2. RESULTADOS Y ANÁLISIS

EDIFICACIÓN SIMULADA. Uno de los objetivos de este trabajo es simular una edificación real, gracias a un estudio realizado por la empresa Kone y a que facilitaron los

datos que representan el tráfico en una edificación real, fue posible obtener la información del tráfico de la torre Tour Europe (Paris) [9]. Estos datos muestran la tendencia que hay en los desplazamientos de las personas dentro de un edificio a lo largo de un día. Los patrones de tráfico se dividen como aparecen en la Figura 8.

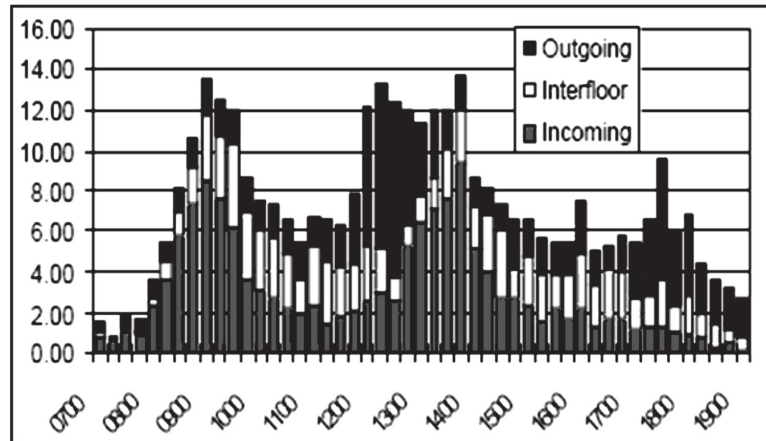


Figura 8. Tráfico Consolidado [3]

Tráfico de entrada (incoming): Es el tráfico que se presenta cuando la tendencia es desplazarse desde los pisos que se entra a la edificación, hacia los pisos altos.

Tráfico de salida (outgoing): Si la tendencia es de los pisos altos hacia los pisos de salida.

Tráfico entrepisos (interfloor): Si la tendencia es movilizarse en pisos intermedios.

El ingreso de los parámetros de la simulación se hace por medios del archivo xml del que se habló anteriormente, en la Tabla 2 se muestra el parámetro y el valor para la simulación que se va a realizar.

Tabla 2. Parámetros de entrada de la simulación

PARÁMETRO	VALOR	DESCRIPCIÓN
Número de pisos	28	Total pisos
Pisos portal	1	Pisos de entrada y salida a la edificación
Número de ascensores	8	Total de ascensores del grupo
Velocidad del ascensor	1	Velocidad dada en pisos por segundo
Capacidad del Elevador	30	Capacidad máxima en número de personas
Controlador	Controlador Inteligente	Controlador a usar en la simulación.
Factor de avance	10	Velocidad a la que avanza la simulación, 1 si la simulación avanza 1 a 1, es decir 1 segundo de simulación es un segundo en la realidad.

Después de ingresar los parámetros en el simulador, el sistema genera el modelo del sistema real y prepara la simulación para empezar. Cuando el usuario da clic en el menú de iniciar simulación, se abren dos ventanas ver figura 9, en la ventana de la parte izquierda se gráfica el edificio en el estado actual, esta gráfica animada muestra la posición de los ascensores y el número de pasajeros dentro de cada ascensor, en la ventana del lado derecho se tienen dos gráficos estadísticos, el de arriba muestra el tráfico presente en el edificio y en el de abajo se ve la media del tiempo de espera de los usuarios. Este gráfico sirve para monitorear en tiempo real el comportamiento del principio de control en estudio.

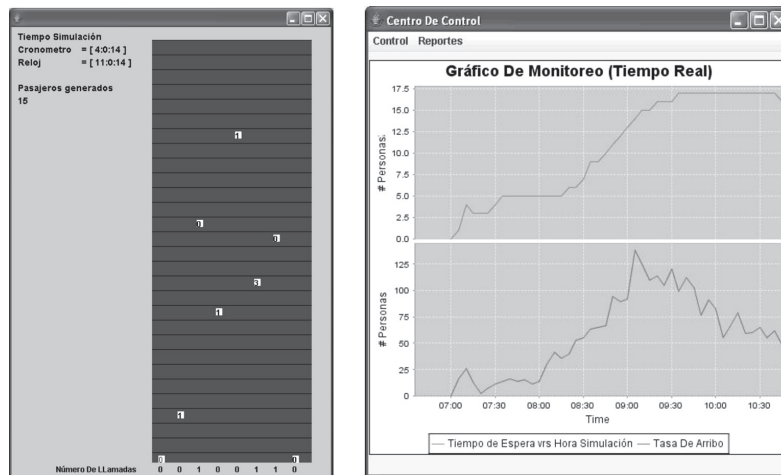


Figura 9. Simulador de Edificio.

El simulador además del reporte en tiempo real, permite generar diversos reportes en cualquier instante de la simulación, en la figura 10, se tiene un reporte estadístico del tiempo de espera de los usuarios a lo largo de la simulación, este informe provee datos importantes como la media, la varianza y la desviación estándar encontrada en la gráfica.

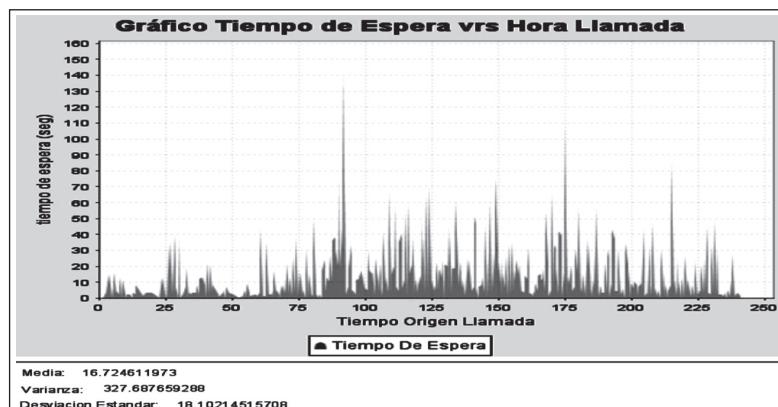


Figura 10. Reporte tiempo de espera.

Otro reporte que permite ponderar la eficiencia del principio de control se ve en la Figura 11. En este informe se compara la energía consumida por cada uno de los elevadores que pertenecen al grupo, la energía se mide en pisos recorridos y se entiende que entre menor cantidad de pisos recorridos menor es el consumo de energía, esto dado que el número de pisos que sube un ascensor es proporcional al número de pisos que baja, este reporte permite conocer qué tanta energía se ahorra usando determinado principio de control, además de esto el informe incluye datos relevantes como la media, varianza, desviación estándar y el total de energía que consume el sistema. Como en el caso anterior con este informe se puede comparar la utilidad de un algoritmo de control contra un algoritmo convencional [10].

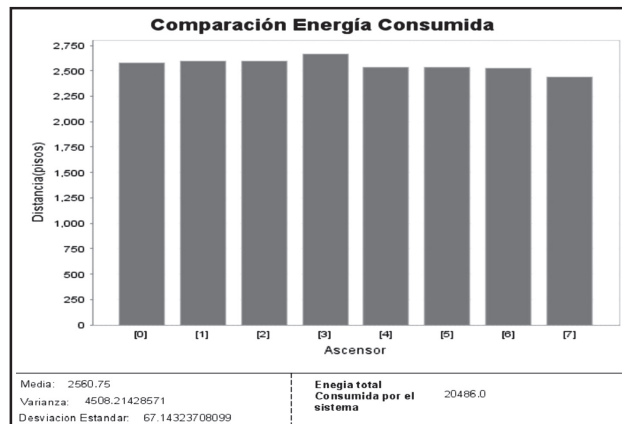


Figura 11. Reporte Energía Consumida

3. CONCLUSIONES

El simulador desarrollado en este trabajo puede ejecutarse en cualquier computador personal que tenga instalada la máquina virtual de java. Este simulador permite tener un ambiente controlado que representa las características más importantes del ambiente real y simplifica su representación, facilitando así, su mantenimiento, adaptación y reutilización.

Se hace fácil modelar una edificación real debido a la forma de ingreso de los datos, estos parámetros de entrada pueden ajustarse a cualquier tipo de edificio y además se puede incluir cualquier tipo de tráfico. Debido a que la simulación se hace gráficamente es fácil hacer el monitoreo del sistema, además como el factor de avance del tiempo es parametrizable es posible correr la simulación a una velocidad muy alta 1000X si se quiere, o correrla a tiempo real 1X.

El componente de reportes es bastante útil para los investigadores que quieran comprobar la eficiencia de su algoritmo de control ya que este provee la información necesaria para hacer su evaluación. El marco de trabajo basado en tareas brinda la flexibilidad, simplicidad y adaptabilidad necesarias para que sea utilizado en otro tipo de simulación.

Como trabajo futuro se propone mejorar el simulador para que pueda recibir órdenes desde una consola, para así poder generar un pasajero, eliminar una llamada, etc. Se deja un marco de trabajo definido para que los investigadores puedan probar sus algoritmos de control en este simulador el proyecto está disponible para toda la comunidad [7], allí encontrara la implementación hablada en este artículo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] BANKS, J. "Discrete-event system simulation". 3rd edition, Prentice Hall. (2000).
- [2] BARNEY, G.C.; COOPER, D.A. and INGLIS, J.. 1997. Elevator & Escalator Micropedia. IAEE Publications, Heron Syke, Brigsteer, Kemdal, Cumbria, England.
- [3] BROWNE, J.J. and KELLY, J.J.. 1968. "Simulation of elevator system for world's tallest buildings". Transportation Science, Volume 2, No. 1. pp. 35-56.
- [4] INSUA RIOS, David; INSUA RIOS, Sixto; MARTIN Jacinto. "Simulación métodos y aplicaciones". Edit. Alfaomega. 2000. 371 pag.
- [5] LUSTIG, A. "Lift simulation program, elevator world", 1986. No. 3, pp. 74-79.
- [6] MIRAVETE, Antonio; LARRODE, Emilio. "El Libro del transporte vertical". Editado por servicio de publicaciones Centro Politécnico Superior Universidad de Zaragoza. 1996. 485 págs.
- [7] ROSSO, M. Andres. "Sitio Web del proyecto Elevator". <http://sourceforge.net/projects/elevator/>
- [8] SIIKONEN Marja-liisa. "Planning and control models for elevators in high-rise buildings", Doctoral thesis, Helsinki University of Technology, Systems Analysis Laboratory, Research Reports A68 October 1997.
- [9] SIIKONEN Marja-liisa "Elevator simulation and control". Kone elevators 2004.
- [10] SORIANO Jairo; ROSSO Andres. "Sistema de control inteligente para un grupo de elevadores". Revista Ciencia e Ingenieria Neogranadina, Vol. 2, Diciembre 2008.
- [11] SWARUP, K., GUPTA, P.K., and MOHAN, M., "Operations research", Sultan Chand & Sons Pub., 1978. pp. 504-510.
- [12] WEINBERGER, W. "A lift traffic simulator", Födern und Heben, 1967. Vol. 17, No. 16, pp. 923-925.