

**DESARROLLO DE UN SIMULADOR DE ESCENARIOS  
PARA LA BÚSQUEDA DE RUTAS EN ROBOTICA MOVIL  
Development of a simulator stage for the search of routes in mobile robotics**

**RESUMEN**

Este artículo presenta el desarrollo de un simulador de escenarios para la búsqueda de rutas en robótica móvil, el cual se implementa mediante herramientas informáticas de última generación.

El simulador de escenarios es un componente de un proyecto de mayor envergadura denominado SIRUM (Simulador de escenarios para búsqueda en robótica móvil); proyecto de tesis de la Maestría en Instrumentación Física de la Universidad Tecnológica de Pereira.

**PALABRAS CLAVES:** Escenario, Búsqueda, Robótica, Simulador, Móvil.

**ABSTRACT**

This article presents the development of a simulator scenario for finding routes in mobile robotics, which is implemented through tools of last generation.

The simulator scenario is part of a larger project called SIRUM (Simulator scenarios for search on mobile robotics); thesis project at the Masters in Instrumentation Physics from the Universidad Tecnológica de Pereira.

**KEYWORDS:** Robotics, Mobile, Simulator, Scenario, Search.

**ALEJANDRO GONZÁLEZ OSPINA**

Ingeniero Industrial.

Universidad Tecnológica de Pereira

Especialista en administración de sistemas informáticos.

Universidad Nacional de Colombia – Sede Manizales.

Candidato a Magíster en Instrumentación Física.

Universidad Tecnológica de Pereira

softwaresalud@hotmail.com

**HUGO BALDOMIRO CANO GARZÓN**

Ingeniero Electricista.

Universidad Tecnológica de Pereira

Especialista en Gerencia de Tecnología.

Convenio de La Escuela de Administración de Negocios y la Universidad Tecnológica de Pereira

Candidato a Magíster en Instrumentación Física.

Universidad Tecnológica de Pereira

Profesor Catedrático Asistente.

Universidad Tecnológica de Pereira

hbcano@utp.edu.co

**WILLIAM ARDILA URUEÑA**

Licenciado en Física, M. Sc.

Profesor Titular

Universidad Tecnológica de Pereira

williamar@utp.edu.co

**1. INTRODUCCIÓN**

El simulador de escenarios desarrollado es un sistema informático que permite a cualquier usuario, generar de forma fácil y amigable ambientes en dos dimensiones usados en robótica móvil, para entre otros aspectos, probar diferentes estrategias de búsqueda en el trazado de rutas, evaluar el uso de diferentes tipos y configuraciones de sensores, y ahorrar costos antes de trabajar directamente con elementos físicos de la robótica. Los escenarios bidimensionales se construyen de forma arbitraria y de acuerdo a la imaginación del usuario, aunque posteriormente es posible llevar escenarios generados en aplicativos como AUTOCAD para ser integrados al sistema sin problemas. Algunos ejemplos de estos escenarios podrían ser, pistas automovilísticas, laberintos, plantas físicas industriales, etc., todos ellos generados de forma virtual, es decir, por medio de una emulación por software, percibiendo un ambiente como si se estuviera en él, de forma “real” e interactuando con una cantidad de elementos ya sean móviles o fijos.

Mediante una interfaz grafica simple, el usuario tiene la posibilidad de generar el ambiente de simulación, además

de ubicar en dicho ambiente íconos que representan elementos claves, como es el caso de un softbot [1] [4], representado por una figura móvil, que es quien articula todos los elementos del sistema. Las simulaciones virtuales, podrán almacenarse en medio magnético para posteriores sesiones.

La simulación de cualquier ambiente está conformada por una cantidad de elementos que tienen su paralelo en el mundo real, esto es, elementos fijos y/o móviles que pueden ser: paredes, puertas, muros, mobiliario de oficinas, carritos de reparto, vehículos, etc.

Ahora bien, por escenario se entiende aquel espacio en dos dimensiones (largo y ancho), donde los elementos del mundo físico son representados mediante iconos, figuras geométricas y colores.

Antes de iniciar la representación del escenario, se debe definir la Unidad Mínima de Representación (UMR), la cual puede ser por ejemplo, de 1\_cm<sup>2</sup>, 10\_cm<sup>2</sup>, 1\_mt<sup>2</sup>, etc.; posteriormente se realizan los trazados perimetrales, y la ubicación de los elementos internos. Un elemento dentro del escenario puede constar de una o más UMR's,

y el sistema podrá ser instruido para que lo maneje como un único elemento; mediante la interpretación de su codificación.

La figura\_1 es la representación mediante diagrama de bloques del proyecto SIRUM, en el cual el generador de escenarios es uno de sus componentes.

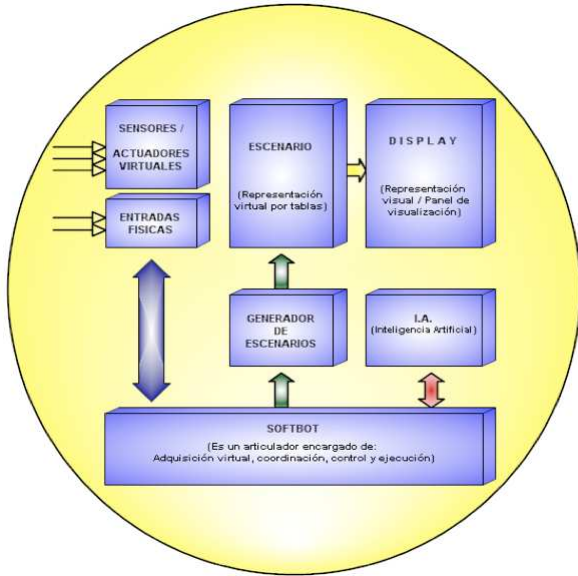


Figura 1. SIRUM

**2. FUNCIONAMIENTO: DESCRIPCIÓN GENERAL**

Como requisitos mínimos de hardware, para el funcionamiento del simulador se tendrían:

- PC con procesador Intel Pentium 4, 1.8 MHz (o similar).
- 512 MB de memoria principal.
- 20 MB de espacio libre en disco duro.
- Sistema operativo Windows XP o posterior.

A nivel de desarrollo se requiere del uso de una herramienta avanzada cómo puede ser el Visual-FoxPro o el lenguaje de programación C-Sharp (C#) bajo entorno de desarrollo Visual Estudio .Net [2].

Para la creación de una simulación pueden manejarse dos modos: a) Escenario previamente construido y almacenado en el sistema de archivos, b) Generación desde cero de un nuevo proyecto.

El tener un proyecto guardado, mejora los tiempos de simulación, por lo que los elementos están creados y no hay limitaciones físicas ni posibles casos no permitidos, como sería la ubicación de elementos en posiciones no válidas (Una silla en medio de un muro).

Se tendrá un panel de opciones donde para el caso de una simulación nueva tendrá que asignarse un nombre para ser guardado en el sistema de archivos, permitiendo además ingresar elementos como si se tuviera un editor de imágenes normal.

**3. AMBIENTE GENERADO:**

El generador de escenarios es un software que permite representar un área delimitada dentro de la cual se encuentran:

**Objetos fijos:**

Representan paredes, máquinas, equipos de oficina, etc.

**Objetos que involucran fenómenos físicos:**

Por ejemplo un horno y el área anexa con diferentes gradientes de temperatura.

**Señalizaciones:**

De evacuación, prohibición, información y “conocimiento” [1].

**Objetos móviles:**

Son todos aquellos que cambian su localización en un instante de tiempo “t” determinado. Ejemplo de objeto móvil es el icono que representa un Softbot (Robot de Software), el cual se desplaza dentro del escenario desde un punto inicial a una meta, ambas fijadas de manera previa.

La construcción del escenario se realiza mediante tablas de información como la de la Figura\_2, la cual contiene en sus celdas el código con información que representa un escenario cualquiera:

Figura 2. Tabla con información codificada

Cada celda de la tabla contiene información codificada que representa un elemento del escenario o que forma parte de él. La información contenida en una celda podría ser:

- Un carácter.(X, O, R, Z, V, A, etc. )
- Una cadena alfanumérica. (C1542TKU43349000)
- Sin información. ( “ ” )

Esta tabla contiene una representación de todos y cada uno de los elementos que conforman el escenario. Así

mismo, es posible manejar un conjunto de tablas que interactúan complementando la información.

En la Tabla\_1 siguiente, se observan algunos ejemplos con información de elementos básicos que conformarían un ambiente construido arbitrariamente.

Dado que el escenario es un modulo dentro del proyecto de búsqueda de rutas optimas, la información codificada en las celdas debe facilitar el procesamiento algorítmico de los datos.

Elemento / Cadena	Significado
X	Muro fijo
S	Objeto móvil (Softbot)
M	Objeto móvil (Vehículo, Plataforma, etc.)
H	Huella o señalización.
000255000	Color verde.
255000000T0800C	Punto del escenario donde se tiene una temperatura (T) de 800 grados centígrados (C) representada con color rojo mediante el código internacional.
S0##	Softbot en el origen
S1##	Meta del softbot
S2##	Softbot en la meta
EP03670240	Obstáculo reportado por Escaneo de Proximidad en Y=0367, X=0240

Tabla 1. Tabla con información de elementos Básicos

La información de la primera columna de la tabla\_1 se alimenta o deposita mediante el formulario de diagramación y captura mostrado en la Figura\_3, en el cual contamos con listas desplegables de objetos (Tanto móviles como fijos), que pueden ser escogidos y ubicados en el lugar deseado.

Lo primero que se define es el largo y el ancho del área del rectángulo que representa el escenario. Con base en esta información el software representa en el Panel de visualización (Display) el área aproximada del escenario y genera la matriz de M filas por N columnas (MxN) requerida para guardar la información de todas y cada una de las unidades mínimas de representación.

Una unidad mínima de representación equivale a un área de  $n \times n$  centímetros, donde  $n$  puede tomar cualquier valor entero.

En la figura\_3 se muestra un proyecto donde se ha simulado un escenario arbitrario con unos muros de frontera en color negro, muros internos en rojo, un objeto móvil en azul y verde.

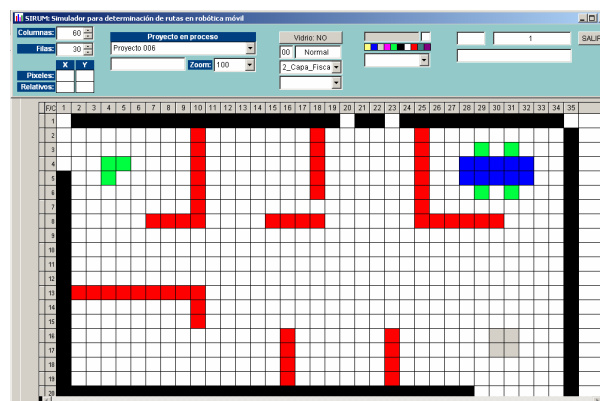


Figura 3. Construcción de un escenario

Es importante señalar que esta es una herramienta de gestión gráfica automatizada, con la cual a medida que se van dibujando o colocando objetos en el ambiente (entorno), se van escribiendo las codificaciones que las representan en tablas / capas asociadas al proyecto.

Cada cuadro de la figura\_3 está asociado a una celda codificada en la tabla de la figura\_2.

Es de absoluta importancia profundizar un poco en un aspecto que tiene que ver con el diseño del formulario de captura y gestión mostrado en la figura\_3:

Inicialmente la matriz de MxN cuadros fue implementada como un conjunto de MxN objetos, lo que para un escenario de tan solo 100 filas y 500 columnas, equivale a un total de 50.000 objetos; cantidad que produjo un procesamiento y gestión muy lenta para un procesador normal.

Como alternativa, en lugar de definir objetos individuales, se maneja una tabla en la que cada celda y la codificación allí contenida, representan un objeto.

La carga a memoria y posterior procesamiento se logra en tiempos muy breves, en razón a que los datos son cargados en memoria RAM, a medida que se van necesitando para su procesamiento y/o representación; de la misma manera se evacúan de memoria los registros (filas) o columnas, que ya no se necesitan.

### Representación por capas y/o niveles:

Dada la gran cantidad de información y la heterogeneidad de la misma, se plantea un modelo de gestión de información por capas de datos afines, como el de la

Figura\_4, que permiten clasificar la información y obtener diferentes niveles de representación y/o abstracción.

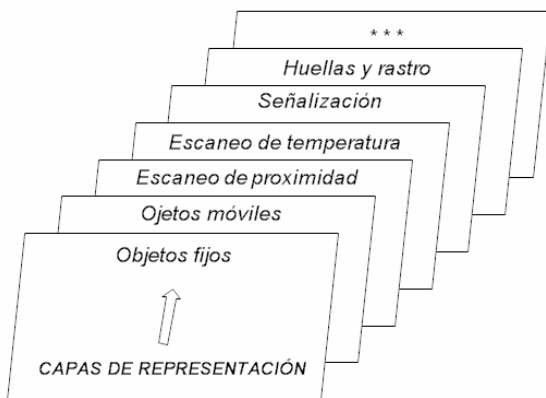


Figura 4. Esquema en capas

La representación de la información mediante capas (tablas) separadas por cada tipo de información afín, permite un diseño que ahorra espacio en las celdas de las tablas. Nótese como por ejemplo, para representar una temperatura, se requiere más espacio que para representar un muro.

Otra consideración importante es al momento de representar las variaciones de estado que en el transcurrir del tiempo se puedan dar: Si entre un instante “ $t_1$ ” y un instante “ $t_2$ ” lo único que varía es un cambio de temperatura de una región del escenario o de posición de un objeto móvil, bastaría con representar dichas variaciones como pequeños vectores de cambio asociados a la capa específica que contiene la información donde se de la variación. (Este es un principio de compresión de video muy utilizado en la actualidad [3]).

Lo anteriormente expresado, significa que un determinado estado “ $X_n$ ”, se puede representar como el equivalente al estado anterior “ $X_{n-k}$ ” más la sumatoria de variaciones entre dicho estado y el estado “ $X_n$ ”.

#### 4. PANEL DE VISUALIZACION (Display):

El panel de visualización es el lugar donde se representan gráficamente tanto el escenario como los eventos que en el ocurren. Así mismo, es el área de trabajo con la cual interactúa el usuario del simulador.

El panel de visualización se descompone en:

- Visualizador de tablas: Permite ver cómo se van representando todos los objetos y eventos involucrados. De tal forma que el contenido de las diferentes tablas (capas), se van actualizando dinámicamente conforme pasa el tiempo y ocurren eventos.
- Representación dinámica del escenario, softbot y rutas:

Cuando el usuario del simulador desea incorporar al ambiente de trabajo un mueble (Escritorio, silla, etc.), un objeto especial -como el móvil que representa el softbot- o una huella o señalización (un símbolo de pare, de información, etc.), basta con hacerle “clic” al icono deseado del almacén de objetos, y arrastrarlo para colocarlo en el lugar deseado. El simulador se encarga automáticamente de ubicar su representación en la tabla de la capa correspondiente.

En la figura\_5 se muestra un escenario simulado con diferentes elementos tanto fijos como móviles.

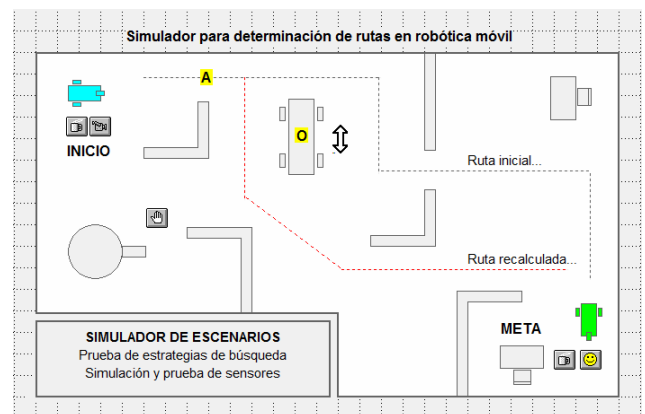


Figura 5. Escenario Simulado

#### 5. CONCLUSIONES Y/O RECOMENDACIONES

El proyecto presentado sirve para probar estrategias de búsqueda en inteligencia artificial, de forma que se use para encontrar una ruta para ir de un punto a otro del escenario, teniendo presente las codificaciones encontradas en las diferentes capas de representación.

Un escaneo virtual (censado) de las celdas y la codificación en ellas contenidas, no será otra cosa que la realización de ciclos computacionales tipo For\_Next o Do\_While\_End Do, mediante los cuales se recorren las tablas (capas) de representación.

#### BIBLIOGRAFÍA

- [1] Inteligencia Artificial, Un enfoque moderno  
Russell y Norvig, Ed. Prentice Hall.
- [2] Microsoft C#  
Ceballos, Francisco Javier Ed. Alfaomega – Ta-Ma
- [3] Revista PC PRO Número 35 Página 74
- [4] Inteligencia artificial y minirobots  
Delgado, Alberto. Ed. ECOE