

Difu100ci@, Revista de Difusión Científica

Ingeniería y Tecnologías

Vol: 14

Num: 2

Publicación: Mayo - Agosto 2020

ISSN: 2007-3585

Reservas de Derechos al Uso Exclusivo: 04-2015-060212215100-203



Una publicación de la Universidad Autónoma de Zacatecas "Francisco García Salinas"

Directorio

Universidad Autónoma de Zacatecas “Francisco García Salinas”

Dr. Antonio Guzmán Fernández, *Rector*

Dr. Rubén de Jesús Ibarra Reyes, *Secretario General*

Dr. Luis Alejandro Aguilera Galavíz, *Secretario Académico*

Dr. Agustín Serna Aguilera, *Secretario Administrativo*

Dr. Manuel Reta Hernández, *Coord. Investigación y Posgrado*

Dra. Georgia Aralu González Pérez, *Coord. Depto. Editorial*

Editor en Jefe

Víktor Iván Rodríguez Abdalá, *Universidad Autónoma de Zacatecas, (México)*

Editores Asociados

Manuel Hernández Calviño, *Universidad de La Habana, (Cuba)*

José Ricardo Gómez Rodríguez, *Universidad Autónoma de Zacatecas, (México)*

Grupo Revisor

Jorge Flores Troncoso, *Universidad Autónoma de Zacatecas, (México)*

Remberito Sandoval Aréchiga, *Universidad Autónoma de Zacatecas, (México)*

Salvador Ibarra Delgado, *Universidad Autónoma de Zacatecas, (México)*

Juvenal Villanueva Maldonado, *CONACyT-UAZ, (México)*

Jorge Simón Rodríguez, *CONACyT-UAZ, (México)*

Israel Alejandro Arriaga Trejo, *CONACyT-UAZ, (México)*

Rodrigo Daniel Méndez Ramírez, *Centro de Investigación Científica y de Educación Superior de Ensenada, (México)*

Leonardo Acho Zuppa, *Universidad Politécnica de Catalunya, (España)*

Miguel Andrés, *Universidad de Valencia, (España)*

José Ángel González Fraga, *Universidad Autónoma de Baja California, (México)*

Ramón Parra Michel, *Centro de Investigación y de Estudios Avanzados del IPN, (México)*

Leonel Soriano Equigua, *Universidad de Colima, (México)*

José Luis Álvarez Flores, *Universidad de Colima, (México)*

Fermín Marcelo Rubén Maciel Barbosa, *Universidad de Colima, (México)*

Francisco Rubén Castillo Soria, *Universidad Autónoma de San Luis Potosí, (México)*

Cristian Eduardo Boyain y Goytia Luna, *Universidad Autónoma de Zacatecas, (México)*

Juan Pablo Morales Álvarez, *Instituto Tecnológico de La Paz, (México)*

Daniel Armando Hirles Valles, *Instituto Tecnológico de La Paz, (México)*

Jaime Sánchez García, *Centro de Investigación Científica y de Educación Superior de Ensenada, (México)*

Salvador Villarreal Reyes, *Centro de Investigación Científica y de Educación Superior de Ensenada, (México)*

Ernesto García Domínguez, *Universidad Autónoma de Zacatecas, (México)*

Rafael Villela Varela, *Universidad Autónoma de Zacatecas, (México)*

Jorge de la Torre y Ramos, *Universidad Autónoma de Zacatecas, (México)*

Manuel Reta Hernández, *Universidad Autónoma de Zacatecas, (México)*

Derechos reservados

Editorial

Universidad Autónoma de Zacatecas
“Francisco García Salinas”

Diseño de revista

Víktor Iván Rodríguez Abdalá
José Ricardo Gómez Rodríguez

Derechos Reservados

Universidad Autónoma de Zacatecas
“Francisco García Salinas”

ISSN 2007-3585

DIFU100ci@ (léase difuciencia) es una publicación cuatrimestral editada por la Universidad Autónoma de Zacatecas, “Francisco García Salinas”, a través de la Unidad Académica de Ingeniería Eléctrica. Jardín Juárez No. 147, Colonia Centro, Zacatecas, Zacatecas, C.P. 98000. Tel. (01 492) 925 6690.

Reservas de Derechos al Uso Exclusivo No. 04-2015-060212215100-203, ISSN: 2007-3585, ambos otorgados por el Instituto Nacional de Derecho de Autor.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación. Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización de la Universidad Autónoma de Zacatecas “Francisco García Salinas” a través de la Unidad Académica de Ingeniería Eléctrica.

<http://www.uaz.edu.mx>

<http://difu100cia.uaz.edu.mx>

difu100cia@uaz.edu.mx

Portada: https://wiki.ros.org/kinect_node

Plantilla de revista: ElegantLaTeX

Plantilla de artículos: Wenkeker Article

Carta del editor

Ante la situación actual que se vive a causa de la pandemia COVID-19, las plataformas digitales se han vuelto una parte integral como herramienta de difusión de contenidos, esto afecta también a la comunidad científica, ya que los artículos de investigación y difusión deben de realizarse bajo nuevos parámetros y lineamientos que permitan a las revistas digitales, como la revista Difu100ci@, continuar con la tarea de divulgar los trabajos que desarrollan los grupos de investigadores.



Aunado a la limitación al acceso a las áreas de trabajo y laboratorios de estos grupos de investigación por cuestiones de prevención y salud, la generación de conocimiento se ha visto afectado tanto en sus alcances como en los tiempos de desarrollo, siendo estos limitados o recortados respectivamente ya sea por cuestiones económicas o de personal disponible.

Sin embargo, esto ha permitido que varios grupos de investigación y desarrollo se unan en un solo objetivo, y es el de combatir a la pandemia en base a sus áreas de experiencia, ya sea por medio del desarrollo de nuevas tecnologías aplicadas en diversas áreas que van desde la educación hasta la industria así como el desarrollo mismo de la vacuna.

Por lo que hoy, esta generación cuenta con una oportunidad única de cambiar la forma de llevar a cabo la investigación y desarrollo de proyectos, permitiendo que las herramientas digitales maduren a tal punto que el acceso remoto a los instrumentos de medición y equipo de trabajo de los laboratorios sean una realidad. Esto permitirá la colaboración de diversos grupos de trabajo sin que implique poner en riesgo la salud de los colaboradores al desplazarse a los diferentes lugares donde puedan localizarse estos laboratorios.

Esta nueva realidad que se vive en la investigación y desarrollo de proyectos implica, para las revistas de divulgación científica, que los procesos para publicar los artículos sean claros, precisos y concisos, permitiendo de este modo que tanto el autor como la revista agilicen los tiempos de respuesta para la revisión y posterior publicación del trabajo de investigación, esto sin comprometer la calidad y veracidad del mismo.

Víctor Iván Rodríguez Abdalá
Editor en Jefe

Índice general



Directorio	I
Derechos reservados	II
Carta del editor	III
1. Navegación reactiva de un robot aplicando una estrategia bifásica en una Raspberry Pi 3	1
2. Diseño y elaboración de un sistema multimedia sobre los estados y fases de la materia	8

Navegación reactiva de un robot aplicando una estrategia bifásica en una Raspberry Pi 3

M. A. Navarrete-Sánchez¹, O. Vite-Chávez¹, Ro. Olivera-Reyna¹, Re. Olivera-Reyna¹, S. Mercado-Pérez¹, J. U. Muñoz-Minjares¹ y A. Marín-Hernández²

¹ Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Campus Jalpa
Lib. Jalpa Km. 156+380, Fracc. Solidaridad, Carretera Guadalajara-Saltillo, Jalpa, Zacatecas, México, CP 99601.

mnavarrete@uaz.edu.mx

² Centro de Investigación en Inteligencia Artificial - Universidad Veracruzana
Sebastián Camacho 5, Zona Centro, Xalapa-Enríquez, Veracruz, C.P. 91000

Resumen

Este trabajo presenta una estrategia bifásica, para la navegación reactiva en tiempo real, de un robot móvil que se desplaza en un ambiente estructurado. En la primera fase, se hace la adquisición de datos a través de un sensor Kinect, para ser procesados por la tarjeta Raspberry Pi 3 y detectar los elementos encontrados en el campo de visión del sensor. La segunda fase trabaja con la información obtenida para tomar las decisiones necesarias y evitar la colisión con los elementos detectados en el entorno. Finalmente, se muestran los resultados de las pruebas experimentales en entornos diferentes.

Palabras clave— Navegación Reactiva, Robot móvil, Estrategia bifásica.

I. Introducción

La robótica es un área de conocimiento que ha avanzado significativamente en los últimos años alcanzado grandes éxitos en la industria manufacturera. Tal es el caso de los brazos robóticos que realizan tareas repetitivas como soldar o manipular componentes eléctricos con alta precisión y a gran velocidad. Una de las principales desventajas de estos robots es la falta de movilidad. La incapacidad de desplazarse de un lugar a otro los limita a estar en una área de trabajo reducida y definida por su ubicación de instalación [1]. En contraste, un robot móvil es capaz de moverse libremente dentro de la planta, lo que le permite tener una mayor flexibilidad y eficacia al realizar una tarea [2].

En la literatura se pueden encontrar diversos trabajos diseñados para la navegación robótica, que utilizan técnicas de procesamiento digital de imágenes en ambientes parcialmente controlados, utilizando un sensor Kinect que incluye una cámara RGB, un sensor de profundidad 3D, un arreglo de micrófonos y motor de inclinación. Y que, en ciertas circunstancias se ha combinado con otros tipos de sensores. Por ejemplo, Chang, Y., *et al.* [3], utilizaron un sensor Kinect para capturar imágenes y el algoritmo Faster R-CNN para un aprendizaje profundo, incorporando el software middleware entre el Sistema Operativo Robótico (ROS, por sus siglas en inglés) y las aplicaciones que se ejecutan internamente, para identificar objetos con un robot móvil utilizando una Raspberry Pi de bajo costo.

Thapa V., *et al.* [4], por su parte evaluaron la efectividad de ROS, utilizando un Kinect como sensor de visión y un sistema Raspberry Pi. Los resultados generados indicaron que dentro del rango típico del sensor Kinect funciona con alta precisión, pero con un retraso mínimo del robot para evitar obstáculos.

Otro trabajo que hace uso del sensor Kinect es presentado por Martínez, C., *et al.* [5], donde desarrollaron un sistema de navegación reactiva difusa que emplea los datos de profundidad del sensor Kinect, algoritmos de visión por computadora y lógica difusa. Con esto generaron ángulos de giro suave para la navegación de un robot móvil realizando pruebas con la plataforma móvil ERA-MOBI, obtuvieron resultados de giros suaves con un porcentaje de evasión de obstáculos del 85.7%.

Por su parte Ruiz, J., *et al.* [6], describen la integración del sensor Kinect en un robot móvil para mejorar su navegación reactiva, gracias a su capacidad de detección de obstáculos que le permite desplazarse de forma más segura en entornos con obstáculos a diferentes alturas.

Mane, S., *et al.* [7], propusieron un algoritmo de detección

y evasión de obstáculos en tiempo real utilizando una cámara Kinect estereoscópica pasiva. La idea básica detrás del método de detección de obstáculos fue encontrar un mapa de profundidad de la imagen capturada por la cámara Kinect y mapear las coordenadas del mundo real. Fue probado en entornos interiores con una Raspberry Pi 2, el sistema fue simple, robusto y eficiente.

Este trabajo propone una estrategia de dos fases, la primera consiste en sensado y procesamiento; y la segunda fase es de navegación. Para la primera fase del sensado se utilizará un sensor único de tipo Kinect y la información se procesará por medio de una Raspberry Pi 3 de bajo costo. En cuanto a la fase de navegación, se implementará la técnica de navegación reactiva.

El artículo está organizado en: Sección II análisis de requerimientos. Sección III estrategia bifásica. Sección IV implementación. En la Sección V pruebas. Los resultados se presentan en la Sección VI y finalmente se presentan las conclusiones en la Sección VII.

II. Análisis de requerimientos

En la actualidad, una de las líneas de la investigación dentro de la robótica móvil que ha tenido mucho auge, es la robótica de servicio [8]. A nivel internacional se han obtenido grandes logros en aspectos como son: robots guías en museos, asistentes de asilos de ancianos, recolectores de datos en ambientes de producción y en actividades que ponen en peligro a los seres humanos. Independientemente del tipo de robot que se utilice, estos realizan actividades ya sean peligrosas o repetitivas para las cuáles su ayuda es muy valiosa. Pero la mayoría de estos robots aún siguen siendo teleoperados [9, 10] o programados para realizar tareas en específico, debido a la complejidad del ambiente al que son expuestos. La navegación de un robot en entornos no controlados es una tarea que tiene una complejidad elevada. Para que el robot se desplace de un punto a otro y parte de sus prioridades es no colisionar con los objetos que pudiera encontrarse en su camino, conlleva a que el robot tenga la capacidad de tomar sus propias decisiones sin tener que consultar a un ser humano.

Diferentes autores describen de manera distinta los esquemas de navegación, pero todos poseen un objetivo común, llevar el vehículo a su destino de forma segura. La capacidad de reacción ante situaciones inesperadas debe ser también considerada, ya que es parte fundamental para el cumplimiento de la tarea en cualquier entorno.

Las etapas involucradas para que un robot móvil realice una correcta navegación son:

- **La percepción del mundo.** Recopilación de información sobre el entorno a través de sus sensores.
- **La planificación de la trayectoria.** Utilización de metodologías para diseñar trayectorias libres de obstáculos para alcanzar el objetivo.
- **El seguimiento del camino.** Control del movimiento para seguir la trayectoria construida por la segunda etapa de forma continua.

La navegación en los robots móviles suele utilizar un mapa, que les permite realizar tareas de forma apropiada. Una de las

desventajas de este método es que supone que los objetos en el mapa permanecen estáticos. En la vida real es muy difícil de cumplir, debido a que en los ambientes los objetos no son totalmente estáticos y el robot móvil necesita realizar un proceso de mapeo continuo, lo que genera una pérdida de tiempo para que éste pueda realizar la tarea que le fue asignada. Además de las grandes cantidades de datos que son generados y que se tienen que almacenar para su posterior uso.

II.1. Requerimientos

El reconocimiento por visión tiene tareas como la extracción de información a partir de videos o imágenes, basada en la posición de la cámara, la detección y reconocimiento de objetos. En la propuesta que se plantea en este trabajo, consiste en una estrategia de navegación reactiva basada en la detección de objetos mediante un sensor Kinect sobre una plataforma Raspberry Pi 3. Las pruebas se realizaron dentro de un entorno estructurado y en uno no estructurado, ambos controlados. Se considera entorno estructurado y controlado aquel en el cual los objetos son seleccionados previamente para su detección en un área limitada de m^2 con un borde de 0.01 m de altura. Basándose en la tarea específica y con el conocimiento previo en el área, se establecieron las características de hardware y software:

Hardware

- Raspberry Pi 3: 1GB en RAM, procesador ARMv7 de cuatro núcleos 900 MHz, USB \times 4, video HDMI, Ethernet, alimentación 5 V y 1.5 A, slot Micro SD y GPIO 40 PINS, monitor VGA o HDMI, adaptador HDMI-VGA (en caso de que el monitor sea VGA), fuente de alimentación de 5 V y 1 A (mínimo).
 - Kinect: cámara RGB, captura de imagen a color con una resolución de 640 \times 480 pixeles, sensor de profundidad con patrones de puntos infrarrojos, 30 Hz, alcance de medición de 0.7 a 4 m.
 - Plataforma de robot móvil: cuatro motorreductores de CD, controlador para motores, tarjeta embebida Raspberry Pi 3, batería y sensor de visión Kinect.
- En la Fig. 1 se muestra la plataforma robótica estructurada.



Figura 1: Robot móvil

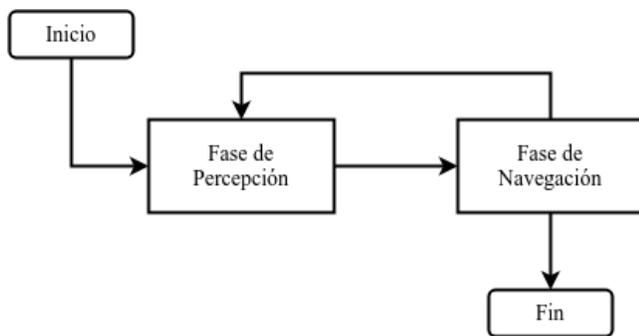


Figura 2: Diagrama de flujo general de navegación

Software

- Sistema operativo Raspbian GNU/Linux, librerías OpenCV, para la codificación se utilizó el conjunto de librerías y controladores Freenect (OpenKinect), por la flexibilidad que brinda el sensor Kinect en diversos sistemas operativos como Linux, Windows y OSX. Estas librerías están desarrolladas en lenguaje C, y permite trabajar utilizando otros lenguajes de programación como Python, Java, C# y C++ [5, 6].

III. Estrategia bifásica

Mediante el empleo de la estrategia bifásica se presenta una solución a la problemática de la navegación reactiva, (ver Fig. 2). Con base en un esquema de dos fases, en la primera fase, se hace la adquisición de datos a través del sensor Kinect, para ser procesados por la tarjeta Raspberry Pi 3. La segunda fase trabaja con la información obtenida para tomar las decisiones necesarias y evitar la colisión con los elementos detectados en el entorno.

III.1. Fase de percepción

Esta fase se encarga de percibir el entorno, esto es, obtener las imágenes y procesar la información para detectar los objetos que se encuentran en el entorno de acuerdo al diagrama de flujo que se muestra en la Fig. 3.

- **Datos de profundidad del Kinect.** Encargada de recibir los datos de profundidad que envía el sensor. El primer paso es obtener la información que es enviada por el sensor (cámara). Esta información es digitalizada para después continuar con la obtención de las transformaciones morfológicas, finalmente por medio de la función *erode* de OpenCV se obtienen los mapas de profundidad. En esta fase se lleva a cabo el proceso para obtener los mapas de profundidad que permite estimar la distancia de los objetos y poder aplicar las maniobras evasivas ante los objetos. Esta tarea se realiza con el siguiente proceso: primero se trazan los contornos dentro de los mapas de profundidad que se obtuvieron de la fase anterior, una vez trazados los contornos mediante diversas funciones se detectan los objetos y se almacena la información de los elementos detectados. Después de obtener los mapas

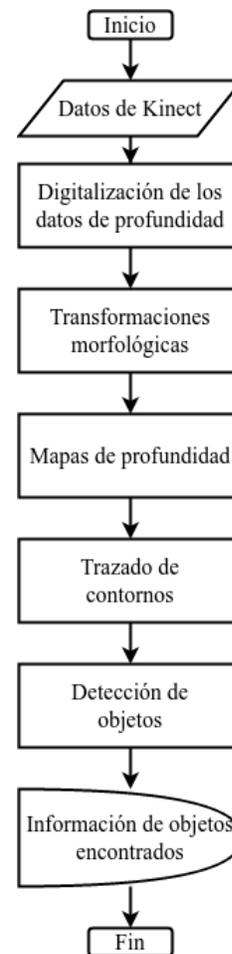


Figura 3: Diagrama de flujo de la fase de percepción

de profundidad el siguiente paso es detectar las cosas que se encuentran en el entorno.

- **Información de objetos detectados.** Una vez detectados los elementos dentro del mapa de profundidad se almacena la información de las cosas detectadas, como la dimensión, distancia, etc.

III.2. Fase de navegación

Esta fase trabaja con las coordenadas y las distancias de los objetos obtenidas de la fase de percepción; mediante una serie de condiciones, la plataforma robótica tendrá la capacidad de tomar decisiones para realizar movimientos de forma que esquivе los objetos evitando colisiones, como se muestra en la Fig. 4.

El proceso para tomar las decisiones comienza a partir de identificar el objeto más cercano considerando la información recibida de la fase de percepción. Después se obtiene la distancia a la cual se encuentra el objeto y ésta información es evaluada bajo las siguientes condiciones:

- Si la distancia es menor a 0.55 m , entonces se compara la coordenada x del objeto con el punto medio del eje x de la imagen. Si la coordenada x es mayor al punto

medio, entonces la plataforma girará a la derecha; de lo contrario, girará a la izquierda. El valor 0.55 m de la distancia a comparar está determinada por el límite del rango de distancias del sensor Kinect ($0.4\text{ m} - 8\text{ m}$), eligiendo 0.55 m para dar un mayor margen y así evitar la colisión.

- Si la distancia es igual a 10 m , con una tolerancia de $(\pm 0.05e)$ para los errores de medición y ruido que se pueda presentar en la lectura, entonces la plataforma retrocede y gira hacia la derecha. En este caso la distancia es igual a 10 m debido a que este dato se usa como referencia inicial en la función que encuentra el objeto más cercano.
- Si no se cumple ninguna de las condiciones anteriores entonces la plataforma se mueve hacia adelante.

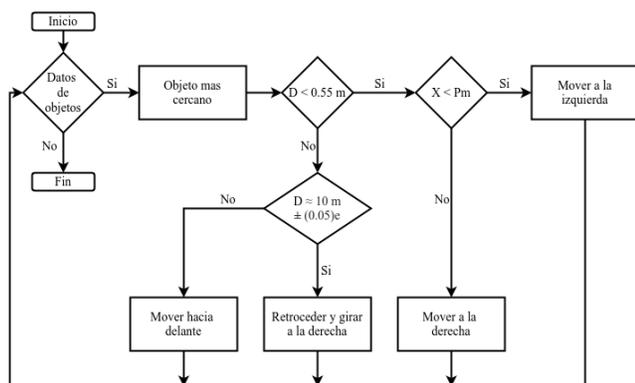


Figura 4: Diagrama de flujo de la fase de navegación

IV. Implementación en Raspberry Pi 3

En esta sección se describen los pasos que se siguieron para la implementación de cada una de las fases descritas anteriormente sobre la tarjeta Raspberry Pi 3, con base en la programación de cada uno de los algoritmos aquí descritos.

IV.1. Fase de percepción

Para las pruebas experimentales realizadas se utilizó una cabina y haciendo los ajustes necesarios se aprecia con facilidad el funcionamiento de la fase de percepción. La cabina de prueba se conformó por tres láminas de poliestireno expandido (unicel) de 1 m^2 cada una, y dentro de esta se colocó el sensor Kinect con los objetos para realizar las pruebas de percepción tal como se muestra en la Fig. 5.

En la fase de percepción, se adquieren los datos de profundidad del entorno mediante el sensor Kinect. Estos datos son procesados para crear el mapa de profundidad a través del Algoritmo 1. Los datos de profundidad se discretizan usando el método *binning* para después aplicar la función *erode* generando el resultado que se muestra en la Fig. 6.

El mapa de profundidad es procesado con la función que se muestra en el Algoritmo 2. La primera acción detecta los bordes y después los contornos de los objetos dentro del mapa. Posteriormente se evalúa cada contorno y se descartan aquellos

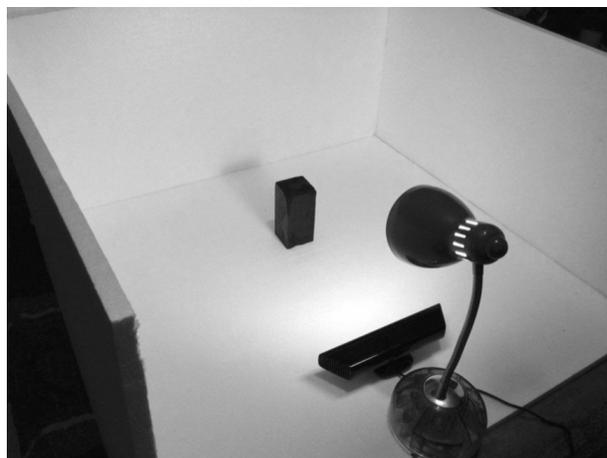


Figura 5: Cabina de pruebas para la fase de percepción

Algoritmo 1. Subfase de mapa de profundidad

Entrada: Datos de profundidad desde el sensor.

Salida: Mapa de profundidad inicio.

```

1 inicio
2 # Obtención de datos del sensor
3 datos = Kinect.profundidad
4 # Se discretizan los datos de profundidad mediante el método binning
5 mapa = binning(datos)
6 # Transformaciones morfológicas
7 mapaf = erode(mapa)
8 fin
9 return mapaf
    
```



Figura 6: Mapa de profundidad a partir del sensor Kinect

que no cumplan con las condiciones de evaluación. Los contornos que cumplen con dichas condiciones, serán los objetos que se encuentran dentro del entorno, de los cuales se guardarán los datos tales como las coordenadas (x, y) de su ubicación dentro de la imagen, su dimensión y la distancia con respecto al sensor Kinect.

Algoritmo 2. Subfase de detección de objetos

Entrada: Mapa de profundidad.

Salida: Información de objetos detectados.

```

1 inicio
2   edges = canny(mapa)
3   contornos = findContours(edges)
4   i = 0
5   for obj in contornos do
6     x, y, radio = enclosingCircle(obj)
7     distancia = mapa[x, y]
8     lista[i] = (x, y, radio, distancia)
9     i++
10    ,
11  end
12 fin
13 return lista

```

IV.2. Fase de navegación

La implementación se realizó mediante la creación de dos funciones principales. La primera función se encarga de encontrar el objeto más cercano dentro del arreglo que se recibe en la fase de percepción. La función se basa en dos variables donde se guardarán los valores del objeto más cercano. La variable x_c guarda la coordenada x del objeto dentro de la imagen mientras que la segunda variable, d_c , almacena el valor de la distancia del objeto con respecto a la plataforma y se encuentra inicializada con una tolerancia de $\pm 0.05e$ (error aproximado). Si el objeto mas cercano detectado se localiza demasiado lejos de la plataforma, su distancia seguirá siendo menor a 10, debido a que es un valor que esta fuera del rango de distancias del sensor Kinect.

El arreglo de objetos se va recorriendo y comparando con el valor de la distancia de cada objeto y la variable d_c , al ser menor la distancia del objeto se guardan los datos de distancia y la coordenada x . El sensor dejará de detectar objetos cuando esté demasiado cerca a una pared o un objeto, en este caso la función regresará los valores iniciales de las variables x_c y d_c , tal como se muestra en el Algoritmo 3.

Algoritmo 3. Función objetoCercano()

Entrada: Información de objetos detectados.

Salida: Coordenada x y distancia del objeto cercano inicio.

```

1 inicio
2    $x_c = 0$ 
3    $d_c = 10$ 
4   for obj in contornos do
5     if ob.distancia <=  $d_c$  then
6        $d_c = ob.distancia$ 
7        $x_c = ob.x$ 
8     end
9   end
10 fin
11 return  $x_c, d_c$ 

```

La segunda función muestra la integración de la fase de navegación descrito en el Algoritmo 4. Esta función tiene como tarea la toma de decisiones para que la plataforma no colisione con los objetos detectados. Primero se encuentran los datos del objeto más cercano y posteriormente el dato de distancia se evalúa con las condiciones definidas dentro de la función.

Algoritmo 4. Fase de navegación reactiva

```

1 inicio
2    $x_c, d_c = objetoCercano()$ 
3   if  $d_c <= 0.55$  then
4     if  $d_c <= pm$  then
5       girar a la izquierda
6     end
7     else
8       girar a la derecha
9     end
10  end
11  else if  $d_c == 10$  then
12    retroceder
13    girar a la derecha
14  end
15  else
16    avanzar
17  end
18 fin

```

V. Pruebas

Una vez concluidas las etapas de desarrollo y programación de la estrategia en la plataforma, se procedió a realizar las pruebas experimentales dentro de dos entornos diferentes. En el primer entorno se considera un ambiente estructurado, este consta de un cuadrado de $2m^2$ con una altura de $0.5m$, tal como se muestra en Fig. 7a. Se consideraron tres objetos de diferentes tamaños como obstáculos, lo cual facilitó que se identificaran rápidamente los objetos dentro del rango de visión del sensor. En el segundo entorno, Fig. 7b, se considera un escenario no estructurado (no representa restricción de espacio ni de objetos) colocados de forma aleatoria.

Identificación de:

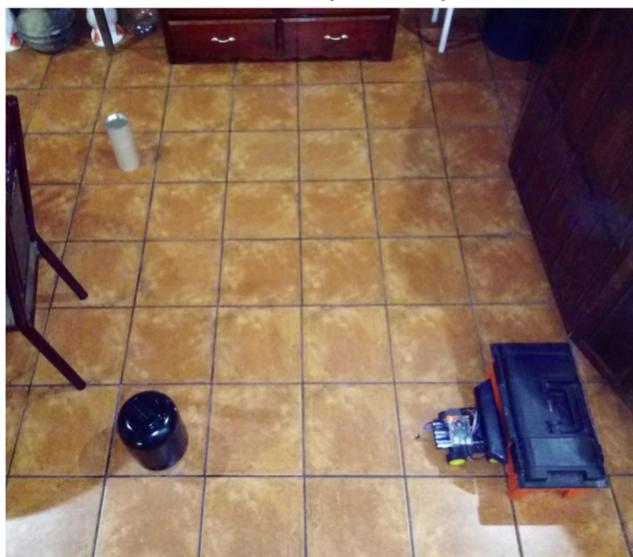
Zonas: Los objetos dentro del entorno se distribuyeron de manera que se identifican dos zonas, tal como se observa en la Fig. 7a.

Objetos: Se realizó con una distribución de objetos como se muestra en la Fig. 7a. Durante esta prueba se evaluaron 170 fotogramas. La distribución de los objetos dentro del entorno permitió que la plataforma pudiera moverse con mayor libertad, en relación a la distribución de la primera prueba. Al igual que en el caso anterior, no se presentaron colisiones y el rango de los objetos que se identificaron fue el mismo, de 0 a 7 unidades.

Entorno no estructurado: La tercera prueba se realizó en un entorno no estructurado, se utilizó la distribución de objetos que se muestra en la Fig. 7b. Una posición inicial



(a) Entorno estructurado con objetos e identificación de zonas



(b) Entorno no estructurado con objetos aleatorios

Figura 7: Diseño del ambiente para pruebas experimentales

de la plataforma junto de un objeto. Para esta prueba se evaluaron 200 fotogramas.

VI. Resultados

Se presentan los resultados arrojados en relación al número de objetos y el tiempo de reacción. A partir de las tres pruebas realizadas, las primeras dos se llevaron a cabo dentro del entorno estructurado y la tercera en ambiente no estructurado.

Identificación de:

Zonas: La gráfica de la Fig. 8a, muestra los objetos identificados (línea azul) y el tiempo de reacción (línea roja) en el transcurso de las pruebas, donde el dispositivo móvil llevó a cabo la captura de 150 imágenes. El rango de los objetos que este identificó van desde 0 a 7 unidades y el tiempo que tardó el proceso en enviar la instrucción de control fue de 0.108 s a los 0.794 s. Como se puede observar en las

gráficas, el tiempo se incrementa en relación al número de objetos detectados; así también se puede observar que los picos más altos se dan cuando el número de objetos detectados igual a cero. Esto es debido al comportamiento mismo de la función *findContours()* que deberá de evaluar cada contorno para descartar aquellos que no cumplen con las condiciones de evaluación, existan o no obstáculos, aumentando tiempo de procesamiento.

Objetos: Los resultados son en base al tiempo de reacción de 0.109 s a los 0.293 s, tal como se muestra en la gráfica de la Fig. 8b.

Entorno no estructurado: Los resultados se pueden observar a pesar de encontrarse cerca de un objeto al inicio, realiza las maniobras para evitar el contacto con el objeto, pero no evita sufrir colisiones y específicamente en las capturas 1 y 2, posteriormente en las capturas 125 a la 135, debido a la posición de la plataforma con respecto al objeto donde el sensor no pudo percibir al objeto. El rango de objetos que se identificaron en este caso fue de 0 a 8 unidades y el tiempo de reacción estuvo en el rango de los 0.109 s a 0.794 s, (ver Fig. 8c).

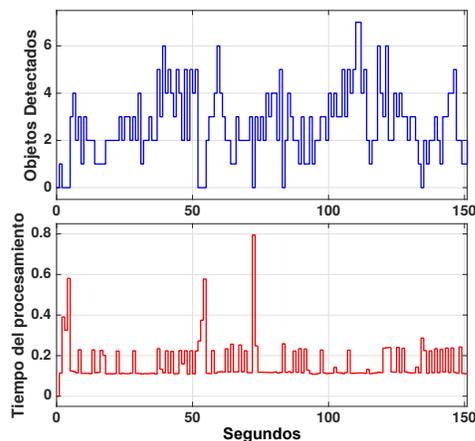
VII. Conclusiones

La integración entre un sensor de visión como lo es el Kinect y una tarjeta la Raspberry Pi 3 ambos de bajo costo, es posible. Se menciona en los trabajos relacionados, el uso del sensor Kinect, diferentes plataformas para el procesamiento como el sistema operativo robótico ROS, o plataformas robóticas complejas y algoritmos robustos para la navegación reactiva, con la necesidad de almacenar previamente mapas del entorno. A pesar de las ventajas que presentan estos trabajos, destacamos en el nuestro la no creación de un mapa para la navegación. La navegación reactiva que utilizamos es sencilla y cumple con el objetivo de este trabajo, permite evitar posibles colisiones con obstáculos. Se utiliza como único sensor de visión el dispositivo Kinect y el procesamiento de la información es llevado a cabo por la tarjeta Raspberry Pi 3.

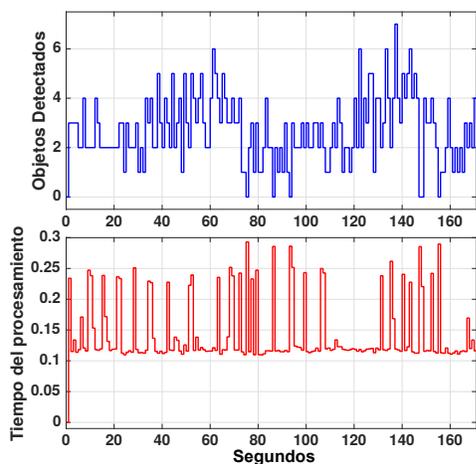
A pesar de la simplicidad de la estrategia de dos fases, proporciona resultados satisfactorios. Se obtuvo un 0 % de colisiones dentro de un entorno controlado y un 5.5 % en un entorno no controlado. Estas fueron dadas por las limitantes del sensor debido al rango de visibilidad y las distancias con las que trabaja el Kinect las cuales son 0.5m de valor mínimo y 3m de valor máximo, impidiendo que detecte objetos fuera de este rango de distancias. Cabe destacar que debido a la forma en que trabaja el Kinect para obtener el mapa de profundidad, el sensor puede detectar objetos en ambientes con oscuridad total por lo cual se descartó el uso de una iluminación ideal.

Se presentó un retardo de tiempo entre el proceso de detección de objetos y el proceso de localización que solo perjudica al tiempo de reacción y acción de movimiento del robot, ya que requiere de tiempo para determinar la dirección a seguir. Debido a que no se consideró el tiempo de respuesta del proceso de detección de objetos, que era mayor que el del proceso de localización, esto provocó que el sistema tenga que esperar que la tarea de búsqueda de objetos finalice.

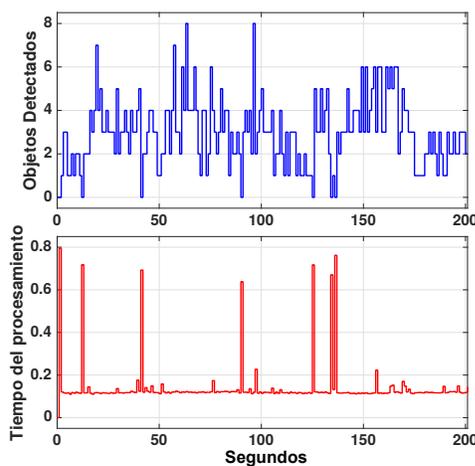
Estos primeros resultados dan la pauta para continuar con nuestra investigación. Como trabajos futuros se busca una na-



(a) En la identificación de zonas



(b) Por la identificación de objetos



(c) En el entorno no estructurado

Figura 8: Tiempo de procesamiento por objetos detectados

vegación libre de colisiones de cualquier robot móvil. Además, se pretende reducir los tiempos de procesamiento y el almacenamiento de información de manera significativa.

Referencias

- [1] Roland Siegwart, Illah Reza Nourbakhsh y Davide Scaramuzza. *Introduction to autonomous mobile robots*. MIT press, 2011.
- [2] Aníbal Ollero Baturone. *Robótica: manipuladores y robots móviles*. Marcombo, 2005.
- [3] Yeong-Hwa Chang, Ping-Lun Chung y Hung-Wei Lin. «Deep learning for object identification in ROS-based mobile robots». En: *2018 IEEE International Conference on Applied System Invention (ICASI)*. IEEE. 2018, págs. 66-69.
- [4] Vikas Thapa y col. «Obstacle avoidance for mobile robot using rgb-d camera». En: *2017 International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE. 2017, págs. 1082-1087.
- [5] Claudia Cruz Martínez y col. «Sistema de Navegación Reactiva Difusa para Giros Suaves de Plataformas Móviles Empleando el Kinect». En: *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica* 1.1 (2016).
- [6] JR Ruiz-Sarmiento y col. «Navegación reactiva de un robot móvil usando kinect». En: *Actas ROBOT 2011* (2011).
- [7] Sunil B Mane y Sharan Vhanale. «Real time obstacle detection for mobile robot navigation using stereo vision». En: *2016 International Conference on Computing, Analytics and Security Trends (CAST)*. IEEE. 2016, págs. 637-642.
- [8] Ifr. *Service Robots*. URL: <https://ifr.org/service-robots/>.
- [9] Riyadh A El-laithy, Jidong Huang y Michael Yeh. «Study on the use of Microsoft Kinect for robotics applications». En: *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. IEEE. 2012, págs. 1280-1288.
- [10] Miguel Angel García-Sánchez. «Diseño de estructuras para vehículos aéreos no tripulados». En: *Di-fu100ci@ Revista en Ingeniería y Tecnología*, UAZ 10.1 (2016).

Diseño y elaboración de un sistema multimedia sobre los estados y fases de la materia

Eduardo García Sánchez¹, José Manuel Cervantes Viramontes¹, Luis Eduardo Bañuelos García¹ y Verónica Torres Cosío¹

¹Universidad Autónoma de Zacatecas (UAZ), Unidad Académica de Ingeniería Eléctrica,
Av. López Velarde 801, Col. Centro, Zacatecas, Zac., México, 98000.
eduardogarciasanchez@gmail.com

Resumen

En este trabajo, se desarrolló, verificó y validó un sistema multimedia sobre los estados y fases de la materia. La metodología que se siguió para la generación del software multimedia sobre los estados y fases de la materia fue MEDESME, cuyas etapas o fases permiten generar un sistema multimedia educativo de calidad. En las fases de concepto o pre-producción, análisis y diseño se efectuó el diseño instruccional, físico y comunicacional propuesto para la generación del sistema. A continuación, se desarrolló e implementó el sistema multimedia, permitiendo con esto llevar a cabo la verificación de la aplicación, encontrando que el sistema multimedia cumplía con los requerimientos planteados al inicio. Para llevar a cabo la verificación y validación, se efectuaron pruebas piloto, y después se validó el sistema multimedia al evaluar el entorno general, pedagógico y técnico/estético con base a los formatos propuestos por la Fundación Bolivariana de Informática y Telemática.

Palabras clave— Sistema Multimedia, Estados de Agregación, Desarrollo de Software

I. Introducción

En virtud de la crisis económica que se vive en México, el gobierno ha ido disminuyendo el financiamiento a la educación superior, lo que conlleva que las universidades públicas no tengan los recursos suficientes para llevar a cabo sus actividades sustantivas. Varias de las necesidades que presentan dichas universidades es la cantidad, calidad y/o falta de equipamiento adecuado de sus laboratorios, los cuales, son

pilares estratégicos que apoyan y fortalecen las acciones propias de la docencia, la investigación y extensión que se llevan a cabo en ellos. Ante esto, surge en la actualidad la necesidad en las universidades públicas de solventar de alguna manera esta falta de equipamiento mediante soluciones alternativas.

Dado que existe un gran auge en el uso de las Tecnologías de la Información y Comunicación (TIC's), estas se pueden aprovechar para poder generar aplicaciones multimedia que sirvan como prácticas de laboratorio virtuales y esto compense por un lado, la falta de equipamiento, y por otro, que favorezcan y faciliten los procesos de enseñanza-aprendizaje de las temáticas que se abordan en el ámbito educativo, dando como resultado en los alumnos un incremento en los niveles de percepción, atención, retención y asimilación de los contenidos, conocimientos y habilidades que deben adquirir durante su formación académica [1, 2].

En el Laboratorio de Química de la Unidad Académica de Ingeniería Eléctrica (UAIE) se carece de equipo de laboratorio que permita llevar a cabo todas las prácticas de laboratorio correspondientes a dicha materia, específicamente aquellas referentes a los temas relacionados a los estados y fases de la materia, por esta razón en este trabajo se tiene como objetivo generar un sistema multimedia educativo referente a este tema en particular y así describir todo el proceso de documentación que se debe seguir para generar un sistema multimedia educativo de calidad.

II. Metodología

Tomando en consideración que en la actualidad existe una gran cantidad de metodologías reportadas en la literatura que difieren en sus fases y etapas, así como los subproductos generados en ellas [1, 2, 3, 4, 5, 6, 7, 8, 9]; en este trabajo se eligió la metodología MEDESME [10] para el desarrollo del sistema

multimedia, debido a que dicha metodología es una de las más completas, lo cual permitirá generar un sistema multimedia educativo de calidad. Dicha metodología comprende ocho fases: concepto o pre-producción, análisis, diseño, desarrollo, implementación, evaluación y validación del programa, producción y elaboración de material complementario. Los subproductos y la documentación de la aplicación de esta se describen en la Sección III.

Es importante mencionar que para llevar a cabo la generación de un sistema multimedia educativo de calidad, se requiere tener diferentes agentes implicados en una o varias etapas de la metodología, además de llevar a cabo todo un proceso de documentación que permita en su momento llevar a cabo el desarrollo, la verificación y validación del sistema, razón por la cual, desde el inicio de la generación del sistema multimedia se empiezan a generar resultados de cada una de las actividades.

III. Resultados y Discusión

Para la generación del sistema multimedia, en la primera fase denominada concepto o pre-producción, se decidió generar un sistema multimedia para suplir la práctica de laboratorio referente a las propiedades físicas y químicas de la materia que no se podía llevar a cabo por falta de equipamiento. Los agentes implicados en la generación de dicho sistema multimedia fueron los autores de este trabajo, además, considerando que se contaban con todos los recursos y tiempo para el desarrollo del mismo, esto hizo posible que el proyecto de generación del sistema multimedia fuera viable y justificable.

En la fase de análisis se planteó como objetivo generar un sistema multimedia sobre los estados y fases de la materia, pero, para tener un sistema multimedia de calidad, este fue generado siguiendo la metodología MEDESME [10]. Los contenidos para dicho sistema multimedia son: las características y propiedades de los estados de la materia (sólido, líquido, gaseoso, plasma), las características y propiedades de las fases y mesofases (sólido amorfo, sólido cristalino, sólido plástico, nemático, colestérico, esméctico, discótico e isotrópico) así como sus transiciones de fase correspondientes, también se definió que tipo de elementos debían tener (imágenes, texto, audios, videos, botones y animaciones), así como las tareas y actividades de aprendizaje a realizar (cuestionario y simulador).

Además, para poder generar el sistema multimedia se debe considerar que procesos son esenciales para que el usuario estimule su aprendizaje, esto con base a las necesidades que presenta el usuario. Teniendo en cuenta lo anterior se consideró que el usuario pudiera realizar los siguientes procesos cognitivos: analizar, observar, identificar, comprender y memorizar, siendo estos requeridos para efectuar las tareas y actividades de aprendizaje del sistema multimedia, tomando en cuenta el perfil de los usuarios (alumnos de la UAIE que cursen la materia de Química y Laboratorio de Química). El entorno de aprendizaje en donde se instalará el sistema multimedia para su uso será en el Laboratorio de Cómputo de la UAIE, que cuenta con 100 computadoras con las características siguientes: Dell optiplex 755, Windows 7 y XP, 6Gb Ram, Pentium Dual Core E2160 1.9 Gb, Disco Duro de 160 Gb, Windows Media Player, Driver video Intel Q35 Express Chipsey, Red Gigabit Intel 82566 DM-2 y High Definition Audio.

En la fase de diseño se generó la ficha general en la cual se pueden plasmar las principales características del recurso didáctico, esto con la finalidad de que el usuario de forma rápida tenga una idea general del contenido (ver Tabla 1). Dentro de esta fase se presenta el diseño lógico que es una descripción de los requisitos funcionales del sistema, además se definen las características didácticas de la aplicación, las reglas y políticas específicas del entorno educativo y el enfoque a utilizar (inductivo, deductivo, aprendizaje por descubrimiento, etc.).

Para esta parte de la etapa de diseño, se generó la ficha técnica psicopedagógica, la cual comprende: los objetivos didácticos (generales y específicos) propuestos al generar elementos multimedia, tareas y actividades de aprendizaje (generar un simulador, archivos de texto, un cuestionario y video, que permitan analizar, observar, identificar, comprender, memorizar y evaluar el grado de conocimientos adquiridos sobre los estados y fases de la materia), los conocimientos que deben ser adquiridos con el sistema, y la forma de evaluación con base a las variables de medición (calificación generada al efectuar el cuestionario).

Otra parte importante dentro de esta etapa de diseño es el diseño funcional, en el cual se define la función instructiva y pedagógica para la que está destinada la aplicación, esto permite darle sentido al objetivo del producto que debe ser funcional para el usuario final y los requerimientos del cliente. Para ello se generó la ficha educativa-técnica la cual consta de la siguiente información: título del recurso, a quién está dirigido, requerimientos técnicos, los contenidos, nivel educativo (licenciatura, aunque puede ser implementado en secundaria y preparatoria), la metodología (se planteó una pequeña sesión para entrega de material adicional, una explicación del funcionamiento del sistema multimedia y del diagrama de análisis de tareas, descrito más adelante en este trabajo, la aplicación del sistema multimedia y la evaluación del mismo) y los autores del sistema multimedia.

En la fase de diseño se definió de manera gráfica la estructura del sistema multimedia en la cual se deben cumplir con los objetivos didácticos, los elementos multimedia, contenidos y las tareas y actividades de aprendizaje.

Para ello se generó el storyboard (ver Figura 1) que muestra el conjunto de las ilustraciones y anotaciones propuestas que fueron utilizadas para el diseño del sistema multimedia y que permite tener una guía para comprender la secuencia y estructura del sistema a generar, dando con esto, una idea inicial de los elementos que se requiere para generar el diseño de la interfaz, describiendo la ubicación de los elementos en las escenas que serán incorporados, así como la distribución de los botones, pantallas de menú y la localización de las imágenes.

Por otro lado, en el esquema o diagrama de navegación (ver Figura 2) se visualiza la estructura y la descripción general de los contenidos correspondientes a los módulos que conforman el sistema multimedia. El mapa de navegación (ver Figura 3) representa todas las relaciones de jerarquía y secuencia presentes en el sistema, el diagrama de análisis y tareas (ver Figura 4) propone una secuencia de instrucción al usuario, para que este cumpla con los objetivos didácticos para los que fue diseñado el sistema.

Además se llevó a cabo el diseño de la interfaz, esto a tra-

Tabla 1: Ficha General del sistema multimedia SIMEFAM

FICHA GENERAL	
Título del recurso	Sistema Multimedia SIMEFAM
Tipo de recurso	Software Multimedia Educativo
Área de conocimiento	Ciencias Básicas
A quién esta dirigido	Estudiantes nivel secundaria, preparatoria y licenciatura
Objetivo general	Rediseñar el sistema multimedia SIMEFAM siguiendo la metodología MEDESM de manera interactiva para que los usuarios puedan tener una mejor comprensión sobre el tema
Objetivos específicos	<ul style="list-style-type: none"> Rediseñar un software que pueda ser utilizado para la enseñanza, aportando conocimientos básicos a los estudiantes de los niveles mencionados, referentes a los “Estados y fases de la materia” Hacer un diagnóstico de los elementos que contendrá dicha aplicación Utilizando las herramientas que ofrece Macromedia Flash Professional 8, Macromedia Fireworks 8, Adobe Audition, Neobook 5.6.4, Shalom Help Maker v0.6.1, Astro Flash Creator y Compilador Microsoft Help Workshop
Contenidos	<p>Una descripción de los estados y fases de la materia mediante apartados como:</p> <ul style="list-style-type: none"> Introducción, Objetivos, Temario, Simulador, Glosario y Actividades de repaso
Descripción general del software	<p>El sistema multimedia SIMEFAM es una herramienta que aporta contenidos de los estados y fases de la materia en formato de texto, imágenes, video y audio. Cuenta con una herramienta para evaluar los conocimientos adquiridos, brinda ayuda en caso de ser necesaria, al final ofrece sitios web con contenido extra a los del sistema</p>

comprenderá el sistema multimedia (que son: de inicio, introducción, objetivos, temario, simulador, glosario, actividades de repaso, sitios de interés, bibliografía, ayuda, requerimientos mínimos del sistema), en las cuales se definieron las zonas de la pantalla, el significado de lo que el usuario deberá observar en dichas zonas, el nombre de la zona, el color y tipo de letra, y finalmente, se generó el diseño de los elementos de control, esto, definiendo las acciones que llevan a cabo cada uno de los botones y la descripción de cada uno de ellos, especificando su tamaño, forma, tipo de letra y color.

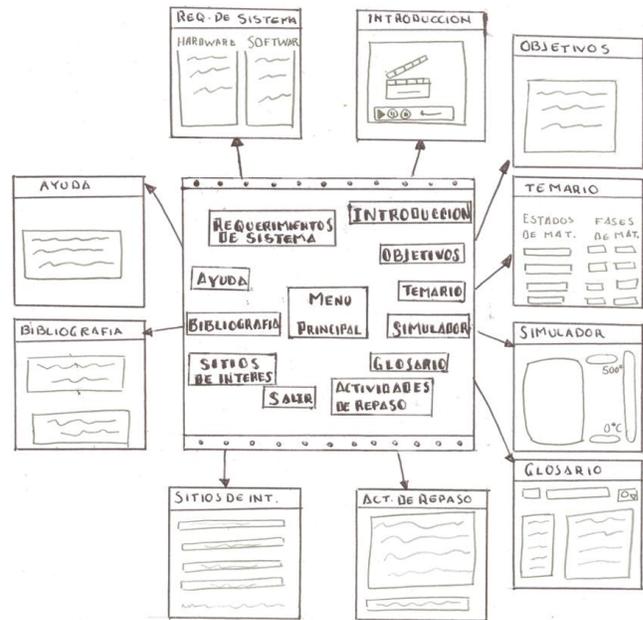


Figura 1: Storyboard del sistema multimedia referente a los estados y fases de la materia

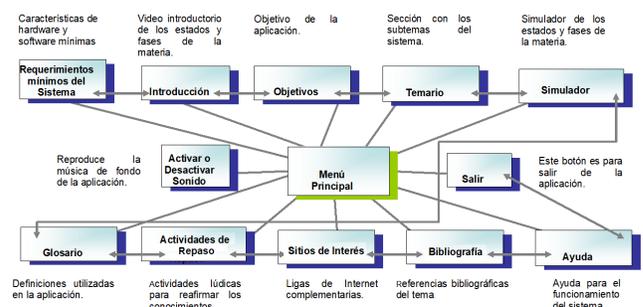


Figura 2: Esquema o diagrama de navegación del sistema multimedia

En la etapa de diseño se llevó a cabo el diseño de la interfaz, en el cual se definió el aspecto de las pantallas que comprenden al sistema multimedia. En cada pantalla se describen las zonas de la pantalla y el significado de lo que el usuario debe observar en dichas zonas, el nombre de la zona, el color, etc. En la Figura 5 se puede apreciar la pantalla de inicio del sistema multimedia. Además, el sistema multimedia tiene las pantallas de: intro-

vés del diseño de cada una de las plantillas o pantallas que

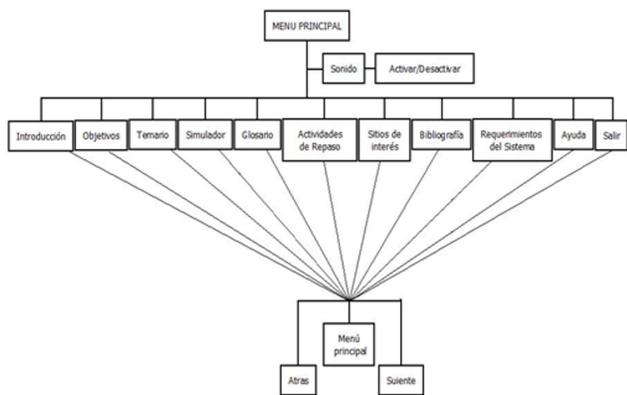


Figura 3: Mapa de navegación del sistema multimedia



Figura 4: Diagrama de análisis de tareas del sistema multimedia

ducción, objetivos, temario, simulador, glosario, repaso, sitios de interés, ayuda, requerimientos del sistema y bibliografía. Adicionalmente, en la fase de diseño se llevó a cabo la fase de diseño comunicacional donde se describe la interfaz del usuario, la cual está definida inicialmente por la pantalla del ordenador, el espacio de trabajo en el ordenador que permite interactuar con el sistema multimedia (interfaz gráfica comunicacional). Para completar la interfaz del usuario se requiere adaptar el teclado y el ratón, estos permiten tener acceso a los controles de navegación y funcionalidad del sistema, al ser accionados con el clic izquierdo del ratón o en el teclado.

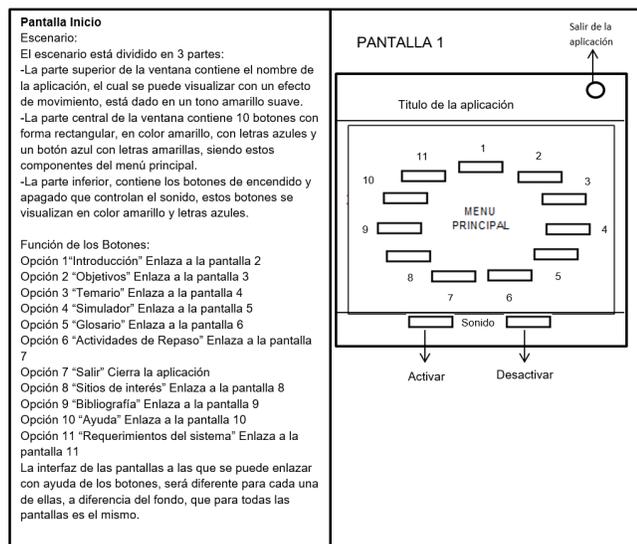


Figura 5: Pantalla de Inicio del Sistema Multimedia

Para llevar a cabo la exploración a través de los distintos nodos de la aplicación se consideró el uso de botones, los cuales deben tener sentido con respecto a las acciones que se efectuarán y sean entendidas de forma intuitiva por medio de su representación gráfica o por su texto. Además, se mantuvo su ubicación a lo largo de la aplicación, para conservar la homogeneidad y consistencia, facilitando con esto su ubicación por parte del usuario. La descripción y funcionalidad de cada uno de los botones que se presentan en la aplicación se puede apreciar en la Tabla 2.

Por otro lado, los colores utilizados en la aplicación fueron combinaciones de verde y naranja, y azul y amarillo. Para el uso de texto se utilizó la menor cantidad de letras en el contenido, el tipo de letra fue Times New Roman en tamaño 12 para una mejor comprensión de la lectura.

En la fase de desarrollo se seleccionó las herramientas de desarrollo Neobook 5.6.4, Astro Flash Creator, Macromedia Flash Professional 8, Macromedia Fireworks 8, Shalom Help Maker v0.6.1, Compilador Microsoft Help Workshop y Adobe Audition para generar el sistema multimedia. En esta fase se define el esqueleto de la aplicación a ensamblar en forma esquemática, organizada y artesanal, las estrategias de enseñanza con los elementos de instrucción y los recursos de presentación, y a cada uno de los elementos multimedia propuestos en la fase de análisis y diseño. Esto en base a las características, restricciones y compatibilidad (entre ellos y con el sistema operativo en el cual será instalado el sistema multimedia) que se requiere.

El producto final de esta fase fue la recopilación de la información de todos los objetos para construir un prototipo de la aplicación, también se generó un manual técnico que contiene cada una de las instrucciones que se requieren para la generación del sistema multimedia, y finalmente, en esta fase se crearon todos los elementos multimedia que fueron incorporados al sistema multimedia.

En la fase de implementación, en una primera etapa, se generó la versión preliminar del sistema multimedia referente a los estados y fases de la materia, la cual en etapas subsecuentes el sistema multimedia sufrió modificaciones con base a los errores detectados y a propuestas de modificación planteadas en la fase de verificación y validación, dicho proceso cíclico se llevó a cabo hasta que se tuvo la versión final del sistema multimedia referente a los estados y fases de la materia. En la Figura 6 se presenta únicamente la ventana principal del sistema multimedia referente a los estados y fases de la materia.

En la fase de evaluación y validación del programa, mediante pruebas piloto se encontraron algunos errores que fueron debidamente corregidos en la fase de implementación, después se validó el contenido y se procedió a evaluar el entorno general, pedagógico y técnico-estético en base a formatos propuestos por la Fundación Bolivariana de Informática y Telemática [11].

A continuación se presentan los resultados que se obtuvieron al comparar los elementos mínimos requeridos para elaborar software educativo. Para el caso del entorno general, aunque no son puntos irrelevantes, se puede comentar que faltó la identificación de cada página según un formato establecido, además del cumplimiento de las pautas del formato para tamaño de hojas, numeración de páginas, tipo y tamaño de fuente, sangría y extensión del título.

En lo que respecta al entorno pedagógico, la única consi-

Tabla 2: Descripción de los botones del sistema multimedia SIMEFAM

Tipo de Botón	Descripción de las acciones de los botones	Descripción de los botones
Botón de Texto	Botones para dar acceso a las actividades principales del sistema multimedia, para la pantalla “Principal”	Diez botones en forma rectangular en un tono color amarillo y letras azul fuerte
Botón de Texto	Botón que permite cerrar la aplicación, en la pantalla “Principal”	Un botón en forma rectangular en un tono color amarillo y letras azul fuerte
Botón de Texto	Botones que permiten activar o desactivar el sonido de fondo de la aplicación	Dos botones en forma rectangular en un tono color amarillo y letras azul fuerte
Texto e Imágenes	Botones propios de la plantilla del software que permiten minimizar, maximizar y cerrar la pantalla del sistema multimedia.	Botón que aparece en forma determinada del software en el que se genere el sistema multimedia
Imágen	Botones para reproducción de videos, para la pantalla “Introducción”	Botones que muestran imágenes universales de los diferentes controles para reproducción de video
Botón de Texto	Botones que permiten ir a la plantilla anterior o a la plantilla siguiente dentro de las actividades principales de la aplicación multimedia	Botón en forma rectangular en un tono color sepia, con bordes blancos y letras amarillas
Botón de Texto	Botón que permite ir al a plantilla principal correspondiente al menú principal, para las pantallas del menú principal	Botón en forma terminador en un tono gris trasparente y letras café
Botón de Texto	Botones que permiten ir a la plantillas estados o fases de la materia, para la pantalla “Temario”	Trece botones rectangulares tono transparente al fondo con letra color azul y roja alternadamente en los botones
Botón de Texto	Botones que aumentan y disminuyen la temperatura, para la plantilla “Simulador”	Botones con una combinación degradada de color azul y blanco, rojo y blanco en forma rectangular con las esquinas redondeadas, la letra en color amarillo con forma degradada
Botón de Texto	Botones para modificar y buscar palabras dentro del glosario	Botones rectangulares color azul con letras transparentes al fondo
Botón de Texto	Botón para iniciar cuestionario	Botón rectangular color transparente al fondo, letras en tono verde
Botón de Texto	Botones que permiten ir a los sitios de interer relacionados con el tema “Estados y Fases de la Materia”	Botones rectangulares tono transparente al fondo, letras en tono verde

deración fue que faltó la especificación sobre el conjunto de valores que se pretenden fomentar a través del recurso, y finalmente para el caso del entorno técnico-estético no se tuvieron omisiones con base a los elementos mínimos requeridos.

Con lo cual, se puede decir que el software multimedia cumple adecuadamente con los estándares fijados en este trabajo, además, se encontró que el software es muy fácil de usar y cumple con los objetivos para los que fue propuesto. Después se pasó a la fase de producción en la cual el sistema multimedia fue instalado en el Laboratorio de cómputo de la UAIE, finalmente, se llevó a cabo la fase de elaboración del material complementario, específicamente se generaron el manual de usuario y el manual o guía de actividades didácticas del sistema multimedia.

IV. Conclusiones

En este trabajo se generó un software multimedia referente a los estados y fases de la materia SIMEFAM, apegado a las necesidades, requerimientos y objetivos planteados en la práctica de laboratorio sobre las propiedades físicas y químicas de la materia que se debe llevar a cabo en la materia de Química y Laboratorio considerando estrategias que facilitarían al usuario un mayor aprendizaje. Dentro de estas se encuentran: analizar, observar, identificar, comprender y memorizar, generando con esto además un sistema multimedia usable que sea utilizada como una herramienta educativa alternativa para apoyar en el proceso enseñanza-aprendizaje.

Las recomendaciones de las actividades que se tienen que llevar a cabo para el sistema multimedia es que:

- Se pueda evaluar si el sistema multimedia permite faci-



Figura 6: Ventana del Menú Principal del sistema multimedia

litar e incrementar los conocimientos adquiridos por los usuarios.

- Que dicha aplicación funcione para sistemas Android.
- Se pueda subir a la nube o en un repositorio institucional para que cualquier usuario pueda tener acceso a él desde cualquier ubicación.
- Efectuar el trámite de los Derechos de Autor del programa de cómputo correspondiente.

Referencias

- [1] Gladys Benigni. «Una metodología orientada a objetos para la producción de software multimedia». En: *Saber* 16.1 (2004).
- [2] Eric J Braude. *Ingeniería de software*. Alfaomega, 2003.
- [3] A Galvis. *ingeniería de software educativo 2da edición Universidad de los andes ediciones UNIANDES*. 2000.
- [4] Ingrith Marcano y Gladys Benigni. «Análisis de alternativas metodológicas para el desarrollo de software educativo». En: *SABER. Revista Multidisciplinaria del Consejo de Investigación de la Universidad de Oriente* 26.3 (2014), págs. 297-304.
- [5] Pere Marques Graells. *Software educativo: guía de uso y metodología de diseño*. E. estel, 1995.
- [6] P Marquès. «Modelos de Desarrollo de MDCS». En: *La Metodología de Pere Marqués. Disponible en línea en: <http://www.peremarques.net/disdesa.htm> (Acceso 10.03. 2013)* (2005).
- [7] Marina Polo. «Aproximación a un Modelo de Diseño: ADITE». En: *Docencia Universitaria* 1.4 (2003), págs. 67-83.
- [8] R. S Pressman. *Ingeniería de software. Un enfoque práctico*. McGrawHill, 2010.
- [9] I Sommerville. *Ingeniería de software*. España: Pearson y Addison Wesley, 2007.
- [10] Eduardo García Sánchez y col. «Metodología para el desarrollo de software multimedia educativo MEDESME». En: *CPU-e. Revista de Investigación Educativa* 23 (2016), págs. 216-226.
- [11] Fundabit. *Orientaciones generales para la elaboración de recursos didácticos apoyados en las TIC's*. <http://cmappublic3.ihmc.us/rid=1N60C257X-22PR6F7-1777/OrientGralesElabRecDidacTIC.pdf>. 2006.