



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“Análisis, Diseño e Implementación de un Juego Interactivo y Educativo que a través del uso de un Modelo Experto basado en Reglas de Inferencia permita a los usuarios soportar estrategias de prevención y reducción del Dengue”

TESINA DE SEMINARIO

Previo la obtención de los Títulos de:

**INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN
SISTEMAS DE INFORMACIÓN**

**INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN
SISTEMAS MULTIMEDIA**

**INGENIERO EN CIENCIAS COMPUTACIONALES ESPECIALIZACIÓN
SISTEMAS TECNOLÓGICOS**

Presentado por:

EDGAR LEONARDO HOLGUÍN FIGUEROA

ISAAC ADRIÁN MATAMOROS GALLEGOS

FABIÁN ROLANDO YAGUAL DEL PEZO

GUAYAQUIL – ECUADOR

2012

AGRADECIMIENTO

Agradezco a DIOS por estar siempre conmigo y haberme permitido culminar con éxito esta etapa de mi vida. A mi MADRE por brindarme su apoyo incondicional y motivarme a seguir adelante en cada paso que doy. A mi familia por confiar y creer en mí.

Edgar Holguín Figueroa

Agradezco a Dios por haberme dado el regalo de la perseverancia, que no me dejó rendirme en los tiempos difíciles. A mi familia por estar siempre a mi lado y soñar conmigo mis sueños. A mis profesores que dejaron de serlo hace algún tiempo y se convirtieron en mis amigos. A mis amigos por acompañarme durante todo el viaje hasta poder decir juntos llegamos a la meta.

Isaac Matamoros Gallegos

Mi infinito agradecimiento a Dios por estar siempre a mi lado, especialmente en esta etapa de mi vida. A mi esposa por qué gracias a su apoyo puedo superar cualquier barrera que se interponga en mi camino.

Fabián Yagual Del Pezo

DEDICATORIA

A mi MADRE por brindarme su infinito amor y ser mi mejor ejemplo a seguir. A mi PADRE que aunque ya no está conmigo sé que se siente orgulloso de mí. A mis hermanos por su cariño incondicional.

Edgar Holguín Figueroa

Dedico todo el trabajo a mi abuelo que no importaba cuan pequeñas o grandes fueran las hazañas que yo realizara siempre llenaban su corazón de orgullo y aunque no llegaste a ver el día en que me convirtiera en ingeniero, será lo primero que te cuente cuando al fin nos reencontremos.

Isaac Matamoros Gallegos

A mi MADRE por qué su ejemplo de esfuerzo y humildad me hacen crecer día a día. A mis HIJOS, los pilares de mi vida, por todo el amor que me brindan.

Fabián Yagual Del Pezo

TRIBUNAL DE SUSTENTACIÓN

Phd. Indira Nolivos

PROFESOR DE SEMINARIO

Ing. Marisol Villacrés.

PROFESOR DELEGADO DEL DECANO

DECLARATORIA EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la

Escuela Superior Politécnica del Litoral”

(Reglamento de Graduación de la ESPOL)

EDGAR LEONARDO HOLGUÍN FIGUEROA

ISAAC ADRIÁN MATAMOROS GALLEGOS

FABIÁN ROLANDO YAGUAL DEL PEZO

RESUMEN

Este trabajo describe el desarrollo e implementación de una aplicación multimedia que puede ser utilizada como herramienta educativa y de concientización para prevenir la incidencia del dengue.

Entre las características principales está la implementación del motor de inferencia en el lenguaje propio de la herramienta utilizada, el Game Maker Language, el cual se encarga de evaluar un conjunto de reglas propuestas y de generar las respectivas acciones a tomar dentro del juego, todo ello basándose en información proporcionada por un experto en el área .

La aplicación cuenta con una interfaz agradable que le permite al usuario conocer información básica sobre la enfermedad del dengue e interactuar en diferentes escenarios en donde el jugador decide qué medidas puede tomar como por ejemplo eliminar criaderos de mosquitos, tapar entradas a las viviendas, entre otros, todo esto con la finalidad de prevenir y controlar el contagio del dengue.

Finalmente en el presente trabajo se realiza un breve análisis del nivel de aceptación que tiene la aplicación implementada en cuanto a su interfaz gráfica y a su contenido.

INDICE GENERAL

RESUMEN.....	V
INDICE DE FIGURAS	IX
INDICE DE TABLAS	X
INTRODUCCIÓN.....	XI
CAPÍTULO 1	1
1. ANTECEDENTES Y JUSTIFICACIÓN.....	1
1.1. ANTECEDENTES	1
1.2. OBJETIVOS.....	2
1.3.1 <i>Objetivo General</i>	2
1.3.2 <i>Objetivos Específicos</i>	2
1.3. ALCANCE.....	2
1.4. JUSTIFICACIÓN	3
CAPÍTULO 2	5
2. FUNDAMENTOS TEÓRICOS.....	5
2.1. SISTEMA EXPERTO	5
2.2. SISTEMAS EXPERTOS BASADOS EN REGLAS.....	5
2.2.1. <i>Estructura Básica del Sistema Experto</i>	6
2.2.1.1 <i>Base de Conocimiento (BC)</i>	7
2.2.1.2 <i>Base de Hechos</i>	8
2.2.1.3 <i>Motor de Inferencia</i>	9
2.2.1.4 <i>Módulo de Justificación</i>	9
2.2.1.5 <i>Interfaz de Usuario</i>	9
2.3. MECANISMO DE INFERENCIA UTILIZADO.....	10
2.3.1 <i>Encadenamiento hacia Adelante (Forward Chaining)</i>	10
2.4. LA ENFERMEDAD	11
2.4.1. <i>El Dengue</i>	11
2.4.2. <i>Indicadores Entomológicos</i>	14
CAPÍTULO 3	17
3. MATERIALES Y MÉTODOS.....	17
3.1. INGENIERÍA DEL CONOCIMIENTO	17
3.1.1 <i>Definición del conocimiento</i>	17
3.1.2 <i>Variables y Reglas</i>	18
3.2. SELECCIÓN DE LA HERRAMIENTA	19
3.2.1 <i>El GameMaker</i>	19
3.2.2 <i>GameMakerLanguage (GML)</i>	20
3.3. EL MODELO DE LA PROGRAMACIÓN	20
3.4. DESCRIPCIÓN DE LA APLICACIÓN	22
3.5. EL MOTOR DE INFERENCIA	29

3.5.1	<i>Estructuras</i>	31
3.5.2	<i>Funciones</i>	32
3.5.3	<i>Ejemplo de aplicación del algoritmo</i>	33
3.6	METODOLOGÍA DE EVALUACIÓN DEL JUEGO.....	34
CAPÍTULO 4		37
4	RESULTADOS.	37
4.1	CASOS DE USO Y ESCENARIOS	37
4.1.1	<i>Especificación de Casos de Uso</i>	37
4.1.2	<i>Especificación de escenarios</i>	39
4.2	RESULTADOS DE EVALUACIÓN DEL JUEGO.....	42
4.3	ANÁLISIS Y DISCUSIÓN DE LOS RESULTADOS.....	45
CONCLUSIONES		
RECOMENDACIONES		
ANEXOS		
	ANEXO A.....	
	IMPLEMENTACIÓN DEL MOTOR DE INFERENCIAS EN GML.....	
	FUNCIONES IMPLEMENTADAS EN GML	
	ANEXO B	
	ENTREVISTA CON EL EXPERTO	
	ANEXO C	
	INSTRUMENTO UTILIZADO PARA MEDIR EL NIVEL DE ACEPTACIÓN DE LA APLICACIÓN.....	
	ANEXO D	
	FOTOS DE LA PRUEBA PILOTO.....	
	ANEXO E	
	REQUISITOS PARA INSTALAR LA APLICACIÓN	
GLOSARIO		
BIBLIOGRAFÍA		

INDICE DE FIGURAS

Figura 2-1. Componentes de un Sistema Experto	7
Figura 2-2. Fases de reproducción del mosquito	13
Figura 3-2. Escenario 1	24
Figura 3-3. Escenario 2	24
Figura 3-4. Escenario 2	25
Figura 3-5. Escenario 3	26
Figura 3-6. Escenario 4	27
Figura 3-7. Escenario 5	28
Figura 4-1. Prueba Piloto	35
Figura 4-2. Resultados Prueba Piloto.....	43
Figura 4-3. Resultados Prueba Piloto.....	43
Figura 4-4. Resultados Prueba Piloto.....	44

INDICE DE TABLAS

Tabla 2.1. Síntomas del Dengue	12
Tabla 3.1. Trazas Encadenamiento hacia adelante.....	33

INTRODUCCIÓN

Actualmente el dengue está considerado como uno de los problemas de salud pública más graves en el Ecuador debido a que anualmente la cantidad de personas afectadas en todo el país aumenta (ver anexo B).

Al parecer las estrategias y campañas de prevención que se han realizado durante los últimos años no han sido suficientes como para mitigar los altos índices de muerte que causa esta enfermedad. Los esfuerzos realizados por las autoridades no han logrado erradicar este mal que anualmente cobra muchas vidas, especialmente en las zonas costeras de nuestro país (ver anexo B).

A todo esto se suma la falta de información que existe en la comunidad en general, lo cual hace que no se tomen las medidas necesarias para prevenir y controlar la incidencia de la enfermedad.

Nosotros como miembros de la comunidad nos encontramos muy preocupados y deseamos aportar una herramienta que permita a la gente conocer información básica y las precauciones que deben

tomar con la finalidad de que todos seamos partícipes de una gran campaña para prevenir y erradicar el dengue en nuestros hogares.

La herramienta a presentar en este trabajo es una aplicación interactiva muy sencilla que hace uso de técnicas de Inteligencia Artificial, de Ingeniería de Software y de Herramientas Multimedia para la creación de un juego por medio de la implementación de un sistema experto, que le brinde información al usuario sobre las medidas que debe tomar en diferentes escenarios para prevenir y controlar el Dengue.

En éste proyecto se plantea el uso de una aplicación sencilla (en este caso un juego) que sirva como herramienta educativa y de concientización de las consecuencias del dengue y de la incidencia dentro de la población.

CAPÍTULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN.

1.1. Antecedentes

Cada año los casos de dengue en nuestro país se incrementan, gran cantidad de personas acuden a los diferentes hospitales a nivel nacional presentando los síntomas de esta enfermedad, la cifras de personas infectadas especialmente en época invernal han sido cada vez más alarmantes lo cual genera preocupación en la población. Todo esto indica que las autoridades no están realizando campañas de información suficientes para que la gente tome las medidas adecuadas para prevenir esta enfermedad (Ver anexo B).

A causa de la desinformación que existe, los habitantes de diferentes sectores del país se ven seriamente afectados ya que al encontrarse habitando en sectores de algo riesgo no toman las precauciones necesarias lo que continúa generando consecuencias muy negativas y que el problema siga creciendo.

Debido a la alta incidencia de esta enfermedad en nuestro país, ha surgido la necesidad de generar una solución, algo que aporte para que la comunidad desde muy joven se informe de las medidas que se deben tomar para prevenir y controlar el dengue y así poder disminuir el número de casos que se reportan cada año.

1.2. **Objetivos**

1.3.1 **Objetivo General**

Desarrollar una aplicación que represente los factores de incidencia de la enfermedad, así como el impacto de las medidas que la comunidad puede tomar para reducir el riesgo de contagio.

1.3.2 **Objetivos Específicos**

- Desarrollar una aplicación que pueda ser utilizada como herramienta educativa y de concienciación.
- Desarrollar un juego sencillo e interactivo basado en un sistema experto.

1.3.3 **Alcance**

El producto permitirá a los usuarios comunes, tanto adultos como niños, interactuar con una aplicación multimedia sencilla, fácil de entender, con lo cual aprenderán como combatir la enfermedad

tomando medidas preventivas y correctivas en diferentes ambientes con la finalidad de reducir el riesgo de contagio del dengue.

La aplicación provee diferentes escenarios en los que el usuario será capaz de identificar cuáles son los ambientes propicios para la reproducción del mosquito transmisor del dengue, además podrá decidir qué medidas tomar para prevenir la presencia del mosquito en diferentes ambientes.

1.3. Justificación

Actualmente el índice de desinformación en la colectividad es muy elevado, la gente que habita en zonas de alto riesgo no se encuentra preparada ante como prevenir y evitar el contagio del dengue [1].

Factores como la variabilidad climática pueden producir incremento de lluvias en época invernal facilitando la creación de ambientes ideales para la reproducción del mosquito transmisor de esta enfermedad [1].

A esto se suma el mal manejo de desechos por parte de la gente, el fracaso de programas de prevención, la poca cooperación y participación de la comunidad en el desarrollo de actividades y estrategias que permitan mitigar los efectos de esta enfermedad.

Debido a esto surge la necesidad de incrementar las fuentes de información para que la ciudadanía conozca y participe de forma activa en una gran campaña de prevención y control del dengue.

Con nuestro proyecto, un juego basado en un sistema experto desarrollado con entorno amigable y divertido para el usuario, creado como una herramienta de concientización que permite a los usuarios ,de una forma divertida e intuitiva, tener una idea clara de cuáles son las principales medidas que se deben tomar para reducir la incidencia del dengue en nuestro país, se espera cooperar con las diferentes campañas de prevención que se han venido llevando a cabo durante los últimos años con la finalidad de mitigar las consecuencias que trae consigo este gran problema .

CAPÍTULO 2

2. FUNDAMENTOS TEÓRICOS.

En éste capítulo se hará una breve explicación de los conceptos relacionados con la implementación de nuestra aplicación, así como también algunas ideas básicas sobre la enfermedad del dengue.

2.1. Sistema Experto

La Inteligencia Artificial es la rama de las ciencias que se ocupa del diseño de sistemas de computación inteligentes, es decir sistemas que tratan de simular el comportamiento humano.

Un Sistema Experto es una herramienta de la Inteligencia Artificial que simula las actividades que realiza un ser humano para dar soluciones a problemas en diferentes áreas, puede definirse como un sistema computacional que resuelve problemas que requieren experiencia humana, mediante el uso de representación del conocimiento y procedimientos de decisión en un campo determinado [2].

2.2. Sistemas Expertos basados en Reglas

Los sistemas basados en reglas representan el conocimiento de resolución de un problema mediante el uso de expresiones lógicas como reglas if... then que representa el par condición-acción, también

llamado antecedente-consecuente o premisa-conclusión. Las premisas de la regla en la parte if corresponden a la condición, y la conclusión en la parte then corresponden a la acción. Cuando la condición es satisfecha, el sistema toma la acción haciendo la conclusión verdadera, los datos son guardados en la memoria de trabajo, el motor de inferencia implementa el ciclo de reconocimiento-acción [3].

Generalmente el formato de una regla es el siguiente y se lo codifica frecuentemente en un lenguaje de alto nivel.

If<condicion>then<acciones a ejecutar>

Las condiciones pueden involucrar constantes, atributos y operadores que pueden ser tanto lógicos como matemáticos.

En los sistemas basados en reglas encontramos dos elementos básicos que son los datos (hechos o evidencia) y el conocimiento (el conjunto de reglas almacenado en la base de conocimiento). El motor de inferencia usa ambos para obtener nuevas conclusiones o hechos [4].

2.2.1. Estructura Básica del Sistema Experto

Un sistema Experto está constituido básicamente por los siguientes elementos (Ver figura 2-1) [5].

2.2.1.1 Base de Conocimiento (BC)

Contiene el conocimiento modelado extraído del diálogo con el experto, las reglas, procedimientos y datos intrínsecos al área del problema. Consiste en algún tipo de codificación del dominio de experiencia del experto consultado.

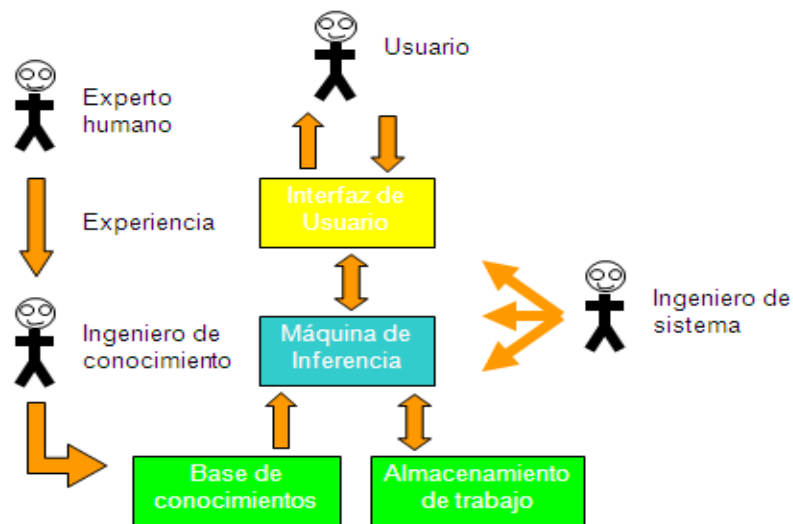


Figura 2-1. Componentes de un Sistema Experto [6]

El ingeniero de conocimiento es el encargado de modelar dicho conocimiento extraído con la finalidad de construir una base de datos que sirva para alimentar el motor de inferencias.

En la base de conocimiento se almacena el conocimiento en el campo a desarrollar. Existen varias formas de representar el conocimiento:

Marcos (frames). Son estructuras de datos que contienen una descripción general de un objeto, almacenan información de una situación específica, organizan el conocimiento en objetos y eventos que resultan útiles para determinadas situaciones.

Redes semánticas. Son una forma de representar el conocimiento gráficamente a través del uso de grafos orientados (nodos) que representan objetos, propiedades y arcos que se relacionan dichos objetos entre sí [7].

Reglas. Representan la forma de razonar. Tienen la forma IF <condición> THEN <acción/conclusión>.

2.2.1.2 Base de Hechos

También denominada memoria de trabajo (MT) contiene información de los datos u hechos relevantes de un problema en particular, representa el conocimiento del estado del sistema en un cierto instante, además contiene los hechos iniciales y los que se vayan obteniendo como consecuencias en el proceso inferencial. Su información está directamente enlazada con la base de conocimientos.

Un hecho puede producir el disparo o ejecución de una regla.

2.2.1.3 Motor de Inferencia

Componente principal de un sistema experto, consiste en un programa de control que modela el proceso de razonamiento humano y cuya función principal es seleccionar las reglas que satisfacen un determinado problema valiéndose para ello de ciertos mecanismos de inferencia, en nuestro caso se utilizó el método de encadenamiento hacia adelante (Forward Chaining). El motor de inferencia extrae el conocimiento de la base de conocimientos, para alcanzar una solución o conclusión determinada.

2.2.1.4 Módulo de Justificación

También denominado módulo de explicación es el que se encarga de explicar el razonamiento utilizado en el sistema para llegar a una determinada conclusión, se ejecuta cuando el usuario solicita la explicación de las conclusiones obtenidas por el sistema experto.

2.2.1.5 Interfaz de Usuario

Es el medio de comunicación entre el usuario y el sistema experto, generalmente se realiza en un lenguaje natural.

El usuario provee información sobre el problema a resolver, mientras que el sistema retorna deducciones derivadas de la información almacenada en la base. Estas deducciones son generadas por el motor de inferencias luego de examinar la base de conocimiento.

2.3. Mecanismo de Inferencia Utilizado

El mecanismo de inferencia utilizado para la implementación de nuestro motor de inferencias es el denominado Forward Chaining.

2.3.1 Encadenamiento hacia Adelante (Forward Chaining).

Encadenamiento hacia adelante (denominado también forward chaining/ data driven/ eventdriven/ bottom-up) realiza comparaciones entre las reglas y los hechos disponibles de manera que se establezcan nuevos hechos hasta llegar al objetivo deseado, es decir es un razonamiento dentro de un sistema lógico que avanza desde los datos o premisas hacia las conclusiones. Se repite el proceso hasta que el programa alcanza su meta o se ejecuta de nuevas posibilidades. Esta técnica se utiliza típicamente para la búsqueda del estado-espacio o el razonamiento dirigido a los datos.

En un sistema con encadenamiento hacia adelante, el motor de inferencia dispara reglas cuyas premisas son emparejadas con la información contenida en la Memoria de Trabajo [8].

2.4. La Enfermedad

2.4.1. El Dengue

El dengue es una enfermedad viral, infecciosa transmitida por mosquitos del género *Aedes aegypti* considerado el vector (o portador) principal del virus del dengue.

Esta enfermedad se puede transmitir de personas enfermas a sanas a través de la picadura del mosquito hembra *Aedes*.

Los mosquitos son insectos dípteros, es decir que tienen un par de alas y necesitan sangre para vivir y reproducirse, nacen de huevos que la hembra pone en recipientes con agua.

Anualmente en el Ecuador el número de casos de dengue y de muertes por ésta enfermedad se incrementa de forma descontrolada por lo que constituye uno de los problemas de salud pública más importante del país (Ver anexo B).

Existen dos tipos de dengue:

- Dengue Clásico
- Dengue Hemorrágico (Síndrome de shock de dengue)

En la siguiente tabla se muestran cuales son los síntomas característicos para cada tipo de dengue, teniendo en consideración que al dengue hemorrágico se le suman los síntomas del dengue clásico.

Dengue Clásico	Dengue Hemorrágico
Fiebre alta.	Sangrado en diferentes partes del cuerpo
Dolor de cabeza.	Enrojecimiento de la cara.
Dolor intenso de músculos.	Vómito o diarrea.
Dolor de huesos y articulaciones.	Alteraciones de la presión.
Dolor retro ocular (en los ojos).	Falta de apetito.
Erupciones.	Palidez.
	Sudoración.
	Sueño.
	Shock.

Tabla 2.1. Síntomas del Dengue

2.4.1.2 ¿Cómo se reproduce el mosquito del dengue?

Por medio de huevos que son puestos en las paredes internas de los recipientes que almacenan agua limpia.

Estos huevos se transforman en larvas que viven en el agua y luego se transforman en pupas que finalmente se convierten en mosquitos adultos capaces de transmitir la enfermedad (Ver Figura 2-2).

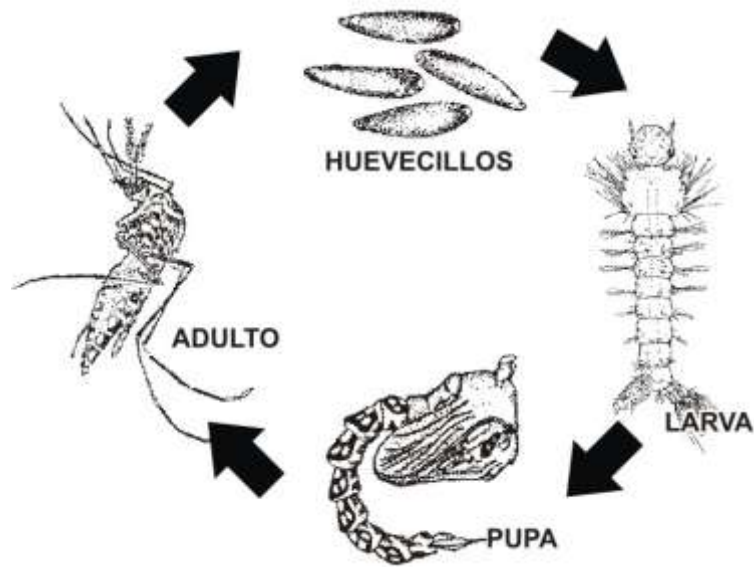


Figura 2-2. Fases de reproducción del mosquito [9]

2.4.1.3 ¿Dónde vive y se reproduce el mosquito del dengue?

Vive dentro y alrededor de nuestras casas. Se reproduce en todo tipo de recipientes que contengan agua tales como tanques, cisternas, piscinas, etc. Además aquellos en los que se puede acumular el agua de lluvia como llantas, tarros, latas, floreros, botellas, baldes, etc.

2.4.1.3 ¿Cuáles son las formas de prevenir y evitar el dengue?

La única medida efectiva para evitar la aparición del mosquito *Aedes aegypti* principal transmisor del dengue es la eliminación de los posibles criaderos dentro y fuera de las viviendas. Un criadero es todo recipiente artificial o natural que puede almacenar agua.

Entre las principales medidas que se deben tomar para prevenir el dengue están las siguientes:

- Impedir que los mosquitos depositen sus huevos tapando bien los tanques, cisternas, algibes y todo tipo de recipientes que almacenen agua por más de dos días.
- Eliminar los huevos limpiando y cepillando bien los tanques, cisternas y algibes al cambiar el agua.
- Eliminar las larvas cambiando el agua de los floreros y bebederos de animales domésticos cada cinco días.
- Voltear botellas, latas, botes, trastos, cubetas y todo aquello que pueda retener agua de lluvia.
- Agujerear recipientes como llantas, macetas para que el agua no se acumule y drene.

2.4.2. Indicadores Entomológicos

Existen varios índices o indicadores entomológicos de uso generalizado aplicables al estudio del Aedes con relación al dengue, los cuales manejan ciertas variables que son de vital importancia para el control del dengue (Ver anexo B).

Estás variables son las siguientes:

CASAS INFESTADAS: Representa el número de casas infestadas con larvas, pupas o ambos estados de desarrollo del *Aedes aegypti*.

CASAS INSPECCIONADAS: Representa el número de casas inspeccionadas en un determinado sector.

CONTENEDORES INSPECCIONADOS: Representa el número de contenedores (llantas, floreros, piletas, etc) con agua de los alrededores y del interior de la vivienda.

CONTENEDORES POSITIVOS: Representa el número de contenedores con agua examinados que contienen larvas, pupas de *Aedes aegypti*. Se registra positivo cuando al menos un contenedor de las casas inspeccionadas presenta larvas o pupas.

Los índices entomológicos de uso generalizado con respecto al desarrollo del dengue en un determinado sector más utilizados son:

ÍNDICE DE CASA: Es el porcentaje de casas infestadas con larvas, pupas o ambos estados de desarrollo del *Aedes aegypti*, determinado por la siguiente expresión:

$$IC = \frac{\text{Casas infestadas} \times 100}{\text{Casas inspeccionadas}}$$

ÍNDICE DE RECIPIENTE: Es el porcentaje de contenedores con agua examinados que contienen larvas, pupas de *Aedes aegypti*.

$$IR = \frac{\text{Num. de contenedores positivos} \times 100}{\text{Num. de contenedores inspeccionados}}$$

ÍNDICE DE BRETEAU : Es el número total de recipientes con agua examinados que contienen larvas, pupas o ambos de *Aedes aegypti* por cien casas.

$$IB = \frac{\text{Num. de contenedores positivos} \times 100}{\text{Casas inspeccionadas}}$$

La interpretación de los tres índices sirve como indicadores del nivel de riesgo epidemiológico al que se encuentra expuesto un determinado sector y dependiendo del valor obtenido de cada uno se ubicará en un determinado rango de valores que determinará en qué estado epidemiológico se encuentra una zona determinada y tomar así las medidas de control necesarias.

CAPÍTULO 3

3. MATERIALES Y MÉTODOS.

El juego del dengue es una aplicación interactiva que tiene como finalidad representar los factores de incidencia de la enfermedad del Dengue así como también el impacto de las medidas que la comunidad puede tomar para reducir el riesgo de contagio.

En este capítulo se hace una breve descripción de cuáles fueron las herramientas necesarias para la construcción y evaluación de nuestra aplicación.

3.1. Ingeniería del Conocimiento

3.1.1 Definición del conocimiento

La base de conocimiento utilizada para la construcción de nuestro sistema está basada en información proporcionada por el experto en epidemiología.

La adquisición del conocimiento se realizó por medio de entrevistas abiertas y semi-estructuradas con el experto (ver anexo B), a través de las cuales se pudo obtener ciertos datos que fueron analizados cuidadosamente con el fin de determinar las diferentes variables y

alternativas de acción. Dichas variables y acciones fueron incluidas en la construcción de reglas de inferencia para ser posteriormente ejecutadas por el motor de inferencia que se encarga de generar las respectivas deducciones que el sistema utilizará para realizar sugerencias de las acciones que el usuario debe tomar en el juego.

3.1.2 Variables y Reglas

Entre las variables más importantes que fueron utilizadas para armar el conjunto de reglas de inferencia que utiliza el motor está el clima, número de recipientes peligrosos, mosquitos infectados, mosquitos no infectados y nivel de peligro. El valor que puede tomar cada variable depende la información que se captura a medida que el usuario interactúa con la aplicación.

Las reglas de inferencia están estructuradas con las variables anteriormente mencionadas de tal forma que el motor ejecuta la que cumple con las condiciones específicas para cada situación dentro del juego. Algunas de las reglas que se utilizan se muestran a continuación:

- R0 "Si Recipientes_Peligrosos>2 entonces Nivel_Peligro = Alto"
- R1 "Si Recipientes_Peligrosos>0 y Recipientes_Peligrosos<3 y Mosquitos_Infectados=0 y Mosquitos_no_Infectados=0 entonces Nivel_Peligro = Medio"
- R2 "Si Mosquitos_Infectados>0 y Mosquitos_no_Infectados>=0 entonces Nivel_Peligro = Alto"
- R3 "Si Mosquitos_Infectados=0 y Mosquitos_no_Infectados>0 y entonces Nivel_Peligro = Medio"
- R4 "Si Mosquitos_Infectados=0 y Mosquitos_no_Infectados=0 y Recipientes_Peligrosos=0 entonces Nivel_Peligro = Bajo"
- R5 "Si Clima=Lluvioso y Nivel_Peligro = Alto entonces Generar_Infectados=Muchos"
- R6 "Si Clima=Soleado ó Clima=Casa y Nivel_Peligro = Alto entonces Generar_Infectados=Pocos"
- R7 "Si Nivel_Peligro = Alto entonces Generar_no_Infectados= Muchos"
- R8 "Si Clima=Lluvioso y Nivel_Peligro = Medio entonces Generar_no_Infectados=Muchos"
- R9 "Si Clima=Soleado ó Clima=Casa y Nivel_Peligro = Medio entonces Generar_no_Infectados=Pocos"
- R10 "Si Clima=Lluvioso y Nivel_Peligro = Bajo entonces Generar_no_Infectados=Pocos"
- R11 "Si Nivel_Peligro = Alto ó Nivel_Peligro = Medio entonces Bonus_Tiempo=Decrementar"
- R12 "Si Nivel_Peligro = Bajo entonces Bonus_Tiempo=Incrementar"
- R13 "Si Nivel_Peligro = Alto entonces Personas_Infectadas=3"
- R14 "Si Nivel_Peligro = Medio entonces Personas_Infectadas=2"
- R15 "Si Nivel_Peligro = Bajo entonces Personas_Infectadas=1"

3.2. Selección de la herramienta

La herramienta seleccionada para el desarrollo de la aplicación fue GameMaker 8.0 distribuida en su versión gratuita por la compañía YoYoGames, debido a que es una herramienta de desarrollo rápido de aplicaciones con lo que nos facilita el trabajo del modelamiento del entorno gráfico de Juego del Dengue y nos permite la implementación de un motor de inferencia propio en lenguaje del programa (GML) para aplicar los métodos de los sistemas expertos.

3.2.1 El GameMaker

El GameMaker es concebido como una herramienta para desarrollar juegos en 2d o 3d sin necesidad de saber programar en primera instancia. Maneja una interfaz sencilla e intuitiva con lo que con unos cuantos arrastrar y soltar permite crear juegos rápidamente.

¿Por que decimos sin saber programar en primera instancia? Porque si necesitamos que nuestros juegos realicen algún tipo de acción mucho más personalizada utilizaremos scripts propios creados en GameMakerLanguage que es el lenguaje de programación interno del GameMaker. Con el GML logramos mucha más flexibilidad y control que las acciones estándares y predeterminadas del programa.

3.2.2 GameMakerLanguage (GML)

GML es un lenguaje de programación interpretado, desarrollado para usarse como complemento de la aplicación de desarrollo de juegos GameMaker. El lenguaje es utilizado en el programa para dar mayor potenciación al desarrollo de juegos u otros sistemas implementados con ayuda de esta herramienta [10]. Esto nos permite extender las posibilidades de manejo de las acciones o eventos dentro del entorno gráfico de desarrollo.

3.3. El modelo de la programación

El GameMaker utiliza la programación orientada a objetos, siendo este modelo el más usado y que ha probado ser muy exitoso.

Dentro de este entorno de desarrollo son considerados objetos por ejemplo el personaje en movimiento, los mosquitos infectados y sin infección, los recipientes, tachos y demás elementos del los escenarios.

Así mismo los eventos sobre dichos objetos o entre ellos como la interacción del personaje con los recipientes o con los mosquitos.

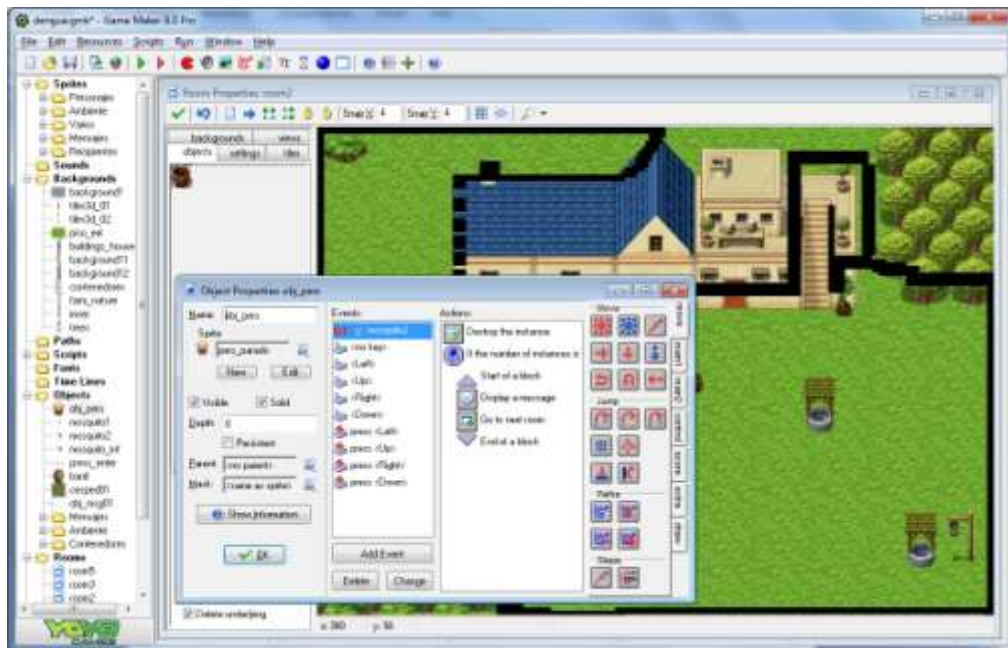


Figura 3-1. Entorno de desarrollo del GameMaker 8.0

También serán eventos las acciones del teclado que causen en este caso el movimiento del personaje dentro de determinado escenario.

Hay que dejar en claro la diferencia entre evento y acción que esta última es la que se realiza cuando llega a suceder determinado evento, por ejemplo el evento puede ser presionar la tecla direccional derecha lo que conlleva a la acción del personaje que se mueva hacia esta dirección.

Para culminar el modelo citado debemos referirnos a las propiedades que son las diferentes características que posee cada objeto como por ejemplo: nombre del objeto, color, ancho, largo, etc.

Sintetizando podemos tener:

Objeto: todo aquello sobre lo que puedo realizar una acción

Evento: algo que sucede sobre un objeto o provocado por él

Acción: cada orden que se le da al programa para que sea ejecutada cuando ocurre el evento en el que está definida.

Propiedades: cada una de las características de un objeto.

3.4. Descripción de la Aplicación

La aplicación del Juego del Dengue consiste en varios escenarios que el usuario deberá ir concluyendo dentro del tiempo establecido para poder avanzar al siguiente nivel. Dentro de cada escenario se describirán cuales son las acciones que el usuario deberá realizar para poder concluirlo de manera satisfactoria y acumular la mayor cantidad de puntos como recompensa. Las acciones que dicho usuario debe realizar sirven para ilustrar el impacto de la incidencia de la enfermedad así como las medidas que se pueden tomar para reducir el riesgo de contagio.

A través de los diferentes escenarios se espera que el usuario aprenda cuales son las medidas que debe tomar para prevenir y controlar el dengue en diferentes ambientes que tratan de simular situaciones reales en las que se puede propagar la enfermedad.

3.4.1 Escenario 1

El objetivo de este escenario es instruir al usuario la reacción ante los mosquitos y evitar la picadura de los mismos para evitar el posible contagio de la enfermedad.

En este escenario habrá mosquitos sin infectar que el usuario deberá liquidar evitando ser picado por los mosquitos que si se encuentran infectados

El tiempo límite para culminar este escenario es de 2 minutos.



Figura 3-2. Escenario 1

3.4.2 Escenario 2

El objetivo de este segundo escenario es hacerle conocer al usuario los posibles lugares exteriores donde el mosquito puede poner sus huevos, para destruir dichos lugares y así evitar la reproducción y propagación de los mosquitos y la enfermedad

El usuario deberá reconocer cuales son los posibles lugares de reproducción del mosquito y acercarse a ellos para poder destruirlos. Existirán lugares que no presentan riesgo de reproducción.

El clima en este escenario puede ser soleado o lluvioso (ver figura 3-3, 3-4 respectivamente).



Figura 3-3. Escenario 2



Figura 3-4. Escenario 2

3.4.3 Escenario 3

Al igual que el escenario 2, en este escenario el usuario deberá reconocer los posibles lugares donde el mosquito puede reproducirse pero ahora estos lugares se buscarán dentro de la vivienda.

También existirán lugares que no representen riesgo de reproducción del mosquito y siguiendo las reglas del escenario anterior si se destruyen estos lugares sin riegos se perderán puntos por estas acciones.

Además el jugador deberá identificar a las personas infectadas con dengue para poder curarlas y así poder evitar que se expanda el virus.



Figura 3-5. Escenario 3

3.4.4 ESCENARIO 4

El objetivo de este escenario es hacer que el usuario reconozca y elimine los posibles lugares que pueden ser los futuros criaderos de mosquitos *Aedes aegypti* en las calles de su ciudad con la finalidad de impedir que se genere una epidemia.

El usuario deberá reconocer cuales son los posibles lugares de reproducción del mosquito y acercarse a ellos para poder destruirlos. Existirán lugares que no presentan riesgo de reproducción, además debe tener mucho cuidado con los mosquitos infectados para no enfermarse.



Figura 3-6. Escenario 4

3.4.5 ESCENARIO 5

El objetivo de este escenario es concientizar al jugador sobre las acciones que debe tomar para evitar la entrada del mosquito a las viviendas y así prevenir el contagio del dengue como por ejemplo tapar ventanas con mallas metálicas, vaciar recipientes que contengan agua acumulada, tapar recipientes destapados, etc. En esta etapa del juego el usuario deberá identificar los errores que se cometen usualmente y deberá corregirlos, además podrá pedir un consejo al doctor experto para saber qué medidas tomar.



Figura 3-7. Escenario 5

3.5 El Motor de Inferencia

El mecanismo de inferencia que hemos utilizado en la implementación de nuestro motor es el llamado Forward Chaining.

Forward Chaining es un método top-down (de arriba hacia abajo), que toma los hechos disponibles y trata de obtener conclusiones (a partir de las reglas evaluadas exitosamente) que concluyen en acciones ejecutadas [11].

Nuestra aplicación es básicamente un sistema basado en reglas que utiliza el método Forward Chaining que dispara acciones, siempre y cuando dichas acciones correspondan a una regla cuyas condiciones son verdaderas, proceso que involucra las tareas de asignar valores a atributos, evaluar condiciones y chequear que todas las condiciones de la regla sean verdaderas [11].

El lenguaje de programación utilizado para implementar nuestro motor de inferencia es Game Maker Language (GML).

A continuación se detalla la secuencia del algoritmo [2] y cuáles son las funciones necesarias para la implementación del mismo.

Procedure Fordward;

```
    LeerValoresIniciales();
    fin1 <- 0
    While Not fin1
        Begin
            fin2 <- 0
            i <- 1
            While Not fin2
                Begin
                    IfNoUsada(i)
                Then
                    Begin
                        ExtraeAtrib(BC(i,1)+BC(i,2),A)
                        If TodosAtribValor(A)
                            Then
                                Begin
                                    If EVALUA(BC(i,2))
                                        Then
                                            Begin
                                                DetAtrib(i,C)
                                                PoneValorEnMT(Evalua(BC(i,1),C)
                                                MarcaRegla(i)
                                                fin2 <- 1
                                            End
                                        End
                                    End
                                End
                            End
                        End
                    End
                End
            End
        End
    Then
        Begin
            i <- i+ 1
            if i> NR
                Then
                    Begin
                        fin1 <- 1
                        fin2 <- 1
                    End
                End
            End
        End
    End
End
```

End

3.5.1 Estructuras

Para la implementación del mecanismo de inferencia utilizado en nuestro motor fue necesario disponer de las siguientes estructuras:

BC que constituye una matriz en la que se almacenan las reglas con las que va a trabajar el motor, la misma que contiene i filas (donde i corresponde al número de reglas cargadas en la base) y 3 columnas. En cada columna se guarda el consecuente, el antecedente y el estado de la regla (si ha sido o no utilizada) respectivamente.

MT que constituye una matriz que servirá como memoria de trabajo la cual irá tomando diferentes valores a medida que transcurra el tiempo en el juego. Esta matriz está conformada por tres filas y j columnas (donde j corresponde al número total de atributos existentes en la base). En cada fila se guardará los nombres de los atributos, los valores asignados para dichos atributos y el identificador de la regla por la cual ha sido probado cada atributo respectivamente

NR que es una variable que indica la cantidad de reglas de nuestro sistema.

3.5.2 Funciones

Las funciones necesarias para la implementación de nuestro motor de inferencias usando el mecanismo de Forward Chaining son las siguientes [2]:

LeerValoresIniciales(): Inicializa el valor de todas las variables en la memoria de trabajo del sistema.

NoUsada(i): Retorna verdadero si la *i*-ésima regla ha sido utilizada y falso en caso contrario.

ExtraeAtributos(EXP,A): Almacena en la lista *A* los atributos que aparecen como operandos en la expresión *EXP*.

TodosAtribValor(A): Retorna verdadero si todos los atributos almacenados en *A* poseen valor y falso en caso contrario.

Evalua(EXP): Evalúa la expresión *EXP*.

Extrae-Atrib(EXP,C): Coloca en *C* el atributo que se evalúa en la expresión *EXP*.

PoneValorEnMT(V,C): Actualiza la memoria de trabajo asignando el valor *V* al atributo *C*.

MarcaRegla(i): Marca la *i*-ésima regla como usada.

DetAtrib(i,C): Determina el atributo que se evalúa en la *i*-ésima regla y lo coloca en *C*.

3.5.3 Ejemplo de aplicación del algoritmo

Reglas Propuestas

R0 "Si Recipientes_Peligrosos>2 entonces Nivel_Peligro = Alto"
R1 "Si Recipientes_Peligrosos>0 y Recipientes_Peligrosos<3 y Mosquitos_Infectados=0 y Mosquitos_no_Infectados=0 entonces Nivel_Peligro = Medio"
R2 "Si Mosquitos_Infectados>0 y Mosquitos_no_Infectados>=0 entonces Nivel_Peligro = Alto"
R3 "Si Mosquitos_Infectados=0 y Mosquitos_no_Infectados>0 y entonces Nivel_Peligro = Medio"
R4 "Si Mosquitos_Infectados=0 y Mosquitos_no_Infectados=0 y Recipientes_Peligrosos=0 entonces Nivel_Peligro = Bajo"
R5 "Si Clima=Lluvioso y Nivel_Peligro = Alto entonces Generar_Infectados=Muchos"
R6 "Si Clima=Soleado ó Clima=Casa y Nivel_Peligro = Alto entonces Generar_Infectados=Pocos"
R7 "Si Nivel_Peligro = Alto entonces Generar_no_Infectados= Muchos"
R8 "Si Clima=LLuvioso y Nivel_Peligro = Medio entonces Generar_no_Infectados=Muchos"
R9 "Si Clima=Soleado ó Clima=Casa y Nivel_Peligro = Medio entonces Generar_no_Infectados=Pocos"
R10 "Si Clima=Lluvioso y Nivel_Peligro = Bajo entonces Generar_no_Infectados=Pocos"
R11 "Si Nivel_Peligro = Alto ó Nivel_Peligro = Medio entonces Bonus_Tiempo=Decrementar"
R12 "Si Nivel_Peligro = Bajo entonces Bonus_Tiempo=Incrementar"
R13 "Si Nivel_Peligro = Alto entonces Personas_Infectadas=3"
R14 "Si Nivel_Peligro = Medio entonces Personas_Infectadas=2"
R15 "Si Nivel_Peligro = Bajo entonces Personas_Infectadas=1"

Clima = "lluvioso"

Mosquitos Infectados = 2

Mosquitos no infectados =0

Recipientes peligrosos =0

Trazas Encadenamiento Hacia Adelante

Ciclo de Ejecución	Reglas aplicables	Regla Seleccionada	Hechos Derivados
1	2,5	2	Nivel Peligro = Alto
2	2,5	5	Generar Mosquitos = Muchos
3	2,5,13	13	Personas Infectadas = 3
4	Ya se ha llegado al Objetivo		

Tabla 3.1. Trazas Encadenamiento hacia adelante

3.5.3.1 Explicación

Al ejecutar el algoritmo con la base de conocimiento inicial planteada se deduce que:

En el primer ciclo de ejecución se dispara la regla 2 debido a que al reemplazar los valores en el antecedente de dicha regla ésta hace que la condición sea verdadera dando como resultado que el Nivel de Peligro = Alto.

En el segundo ciclo de ejecución se dispara la regla 5 debido a que al reemplazar los valores (Clima = lluvioso y Nivel Peligro = Alto) en el antecedente de dicha regla ésta hace que la condición sea verdadera dando como resultado que la variable Generar Mosquitos tome el valor de muchos.

En el tercer ciclo de ejecución se dispara la regla 13 debido a que al reemplazar los valores (Nivel Peligro = Alto) en el antecedente de dicha regla ésta hace que la condición sea verdadera dando como resultado que el número de personas infectadas es igual a tres.

3.6 Metodología de evaluación del juego.

Se aplicó una prueba piloto a un grupo de 34 estudiantes, 19 niños y 15 niñas de entre 13 y 15 años de edad pertenecientes a la unidad educativa Rubira del cantón Salinas de la provincia de Santa Elena (Ver Figura4-1) por pertenecer a un sector que es considerado vulnerable para la

propagación de la enfermedad y además por las facilidades que brindaron las autoridades de dicho plantel en cuanto a disponibilidad de equipos para la instalación del software y de estudiantes para ejecutar la prueba.

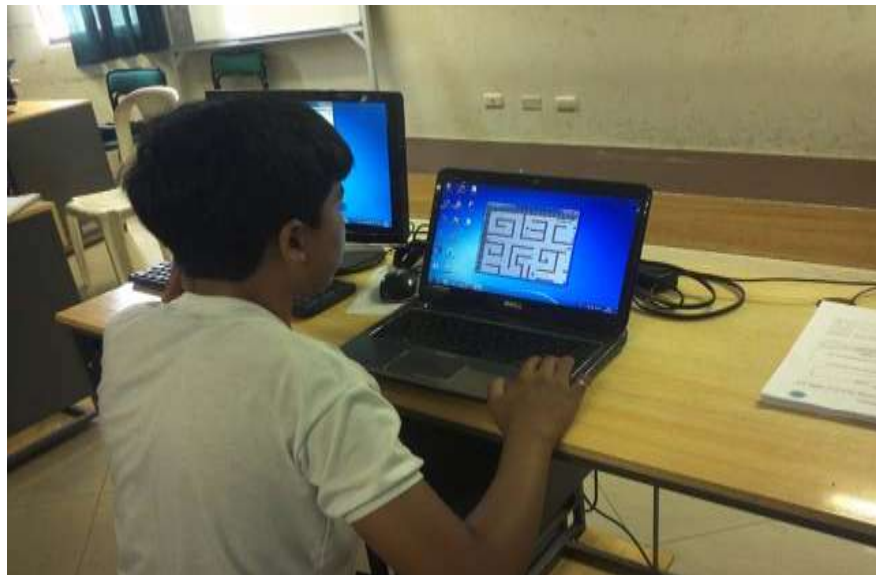


Figura 4-1. Prueba Piloto

Aunque el sistema está diseñado para ser utilizado por usuarios de cualquier edad, se seleccionó un grupo de individuos en el rango de edades anteriormente mencionado porque consideramos que las personas dentro de este grupo poseen mayor facilidad para utilizar este tipo de herramienta y un menor grado de conocimiento sobre las diferentes medidas que se deben tomar para prevenir y controlar el dengue, además de que el juego desarrollado constituye un método atractivo de aprendizaje.

El objetivo de esta prueba fue medir el nivel de aceptación de la aplicación, tanto de la interfaz gráfica como de la información mostrada en el mismo. Como instrumento de medición se utilizó un cuestionario (ver anexo C) que

consta de 14 preguntas cerradas cuyo valor fue ponderado utilizando la escala de Likert, en este se evaluó la percepción de los estudiantes sobre las principales características del juego tales como navegabilidad, facilidad de uso, claridad de la información, sonido, además si la herramienta servía como herramienta de aprendizaje para prevenir y controlar el dengue.

CAPÍTULO 4

4 RESULTADOS.

4.1 Casos de Uso y Escenarios

CU 01: Iniciar juego.
CU 02: Eliminar mosquitos no infectados.
CU 03: Eliminar mosquitos infectados.
CU 04: Eliminar recipientes peligrosos.
CU 05: Tapar entradas.
CU 06: Salir del juego.

4.1.1 Especificación de Casos de Uso

Caso de Uso 01: Iniciar juego
Actores: Jugador
Descripción: El jugador deberá presionar el botón Enter para iniciar el juego.
Notas: <ul style="list-style-type: none">Solo puede jugar un usuario a la vez.El usuario debe tener instalado todos los componentes necesarios para ejecutar la aplicación.
Excepciones: <ul style="list-style-type: none">Falla al ejecutar la aplicación.
Escenarios: <ol style="list-style-type: none">1.1 El jugador inició el juego con éxito.1.2 El jugador no inició el juego con éxito

Caso de Uso 02: Eliminar mosquitos no infectados
Actores: Jugador
Descripción: El jugador deberá eliminar los mosquitos no infectados en los diferentes niveles.
Notas: <ul style="list-style-type: none">Solo puede jugar un usuario a la vez.
Excepciones: <ul style="list-style-type: none">Falla al ejecutar la aplicación.

Escenarios:

- 2.1 El jugador eliminó todos los mosquitos no infectados.
- 2.2 El jugador no eliminó todos los mosquitos no infectados

Caso de Uso 03: Eliminar mosquitos infectados**Actores:** Jugador**Descripción:** El jugador deberá eliminar los mosquitos infectados en los diferentes niveles.**Notas:**

- Solo puede jugar un usuario a la vez.

Excepciones:

- Falla al ejecutar la aplicación.

Escenarios:

- 3.1 El jugador eliminó todos los mosquitos infectados.
- 3.2 El jugador no eliminó todos los mosquitos infectados

Caso de Uso 04: Eliminar recipientes peligrosos**Actores:** Jugador**Descripción:** El jugador deberá eliminar los recipientes que representan peligro.**Notas:**

- Solo puede jugar un usuario a la vez.

Excepciones:

- Falla al ejecutar la aplicación.

Escenarios:

- 4.1 El jugador eliminó todos los recipientes peligrosos.
- 4.2 El jugador no eliminó todos los recipientes peligrosos.

Caso de Uso 05: Salir del juego
Actores: Jugador
Descripción: El jugador deberá dar clic en el botón cerrar para salir de la aplicación
Notas: <ul style="list-style-type: none"> Solo puede jugar un usuario a la vez.
Excepciones: <ul style="list-style-type: none"> Falla al cerrar la aplicación.

4.1.2 Especificación de escenarios

Caso de Uso 01: Iniciar Juego
Escenario 1.1 : El jugador inició juego con éxito
Descripción de Pasos: <ul style="list-style-type: none"> El jugador presiona el botón ENTER para iniciar el juego. El sistema muestra la pantalla de inicio.
Suposiciones/Asunciones: <ul style="list-style-type: none"> No hubo problemas al ejecutar la aplicación.
Resultados: <ul style="list-style-type: none"> El jugador inicio el juego correctamente.

Caso de Uso 01: Iniciar Juego
Escenario 1.2 : El jugador no inició el juego con éxito
Descripción de Pasos: <ul style="list-style-type: none"> El jugador presiona el botón ENTER para iniciar el juego. La aplicación muestra un mensaje de ERROR.
Suposiciones/Asunciones: <ul style="list-style-type: none"> No si instalaron los componentes necesarios para ejecutar la aplicación, tales como la base de datos que utilizará posteriormente el motor de inferencias.
Resultados: <ul style="list-style-type: none"> El jugador no pudo iniciar el juego correctamente.

Caso de Uso 02: Eliminar mosquitos no infectados
Escenario 2.1 : El jugador eliminó todos los mosquitos no infectados.
Descripción de Pasos: <ul style="list-style-type: none"> El jugador inicia el juego correctamente. El jugador recoge pócimas de curación. El jugador solicita información al anciano sabio. El sistema muestra la información solicitada. El jugador pide un consejo al doctor experto.

- El sistema muestra información solicitada.
- El jugador elimina todos los mosquitos no infectados.

Suposiciones/Asunciones:

- El jugador acumula puntos y realiza las acciones deseadas.

Resultados:

- El motor de inferencias obtiene información para establecer las condiciones del siguiente nivel.

Caso de Uso 02: Eliminar mosquitos no infectados

Escenario 2.2 : El jugador no eliminó todos los mosquitos no infectados.

Descripción de Pasos:

- El jugador inicia el juego correctamente.
- El jugador recoge pócimas de curación.
- El jugador no elimina todos los mosquitos no infectados.

Suposiciones/Asunciones:

- El jugador no solicita información al sistema.
- El tiempo de juego se agotó.
- El jugador no acumula puntos.

Resultados:

- El jugador no eliminó todos los mosquitos no infectados.
- El motor de inferencias obtiene información para establecer las condiciones del siguiente nivel.

Caso de Uso 03: Eliminar mosquitos infectados

Escenario 3.1 : El jugador eliminó todos los mosquitos infectados.

Descripción de Pasos:

- El jugador inicia el juego correctamente.
- El jugador recoge pócimas de curación.
- El jugador solicita información al anciano sabio.
- El sistema muestra la información solicitada.
- El jugador pide un consejo al doctor experto.
- El sistema muestra información solicitada.
- El jugador elimina todos los mosquitos infectados.

Suposiciones/Asunciones:

- El jugador acumula puntos y pasa al siguiente nivel.

Resultados:

- El motor de inferencias obtiene información para establecer las condiciones del siguiente nivel.

Caso de Uso 03: Eliminar mosquitos infectados
Escenario 3.2 : El jugador no eliminó todos los mosquitos infectados.
Descripción de Pasos: <ul style="list-style-type: none"> • El jugador inicia el juego correctamente. • El jugador recoge pócimas de curación. • El jugador no elimina todos los mosquitos infectados.
Suposiciones/Asunciones: <ul style="list-style-type: none"> • El jugador no solicita información al sistema. • El tiempo de juego se agotó. • El jugador no acumula puntos. • El jugador es picado por un mosquito infectado.
Resultados: <ul style="list-style-type: none"> • El jugador no eliminó todos los mosquitos infectados. • El motor de inferencias obtiene información para establecer las condiciones del siguiente nivel.

Caso de Uso 04: Eliminar recipientes peligrosos
Escenario 4.1 : El jugador eliminó todos los recipientes peligrosos.
Descripción de Pasos: <ul style="list-style-type: none"> • El jugador inicia el juego correctamente. • El jugador recoge pócimas de curación. • El jugador solicita información al anciano sabio. • El sistema muestra la información solicitada. • El jugador pide un consejo al doctor experto. • El sistema muestra información solicitada. • El jugador elimina todos los recipientes peligrosos.
Suposiciones/Asunciones: <ul style="list-style-type: none"> • El jugador acumula puntos.
Resultados: <ul style="list-style-type: none"> • El motor de inferencias obtiene información para establecer las condiciones del siguiente nivel.

Caso de Uso 04: Eliminar mosquitos infectados
Escenario 4.2 : El jugador no eliminó todos los recipientes peligrosos.
Descripción de Pasos: <ul style="list-style-type: none"> • El jugador inicia el juego correctamente. • El jugador recoge pócimas de curación. • El jugador no elimina todos los recipientes peligrosos.
Suposiciones/Asunciones:

- | |
|---|
| <ul style="list-style-type: none">• El jugador no solicita información al sistema.• El tiempo de juego se agotó.• El jugador no acumula puntos. |
| Resultados: <ul style="list-style-type: none">• El jugador no eliminó todos los recipientes peligrosos.• El motor de inferencias obtiene información para establecer las condiciones del siguiente nivel. |

4.2 Resultados de evaluación del juego

La figura 4-4 presenta los resultados obtenidos para cada ítem del cuestionario aplicado al grupo de estudiantes que participaron en la prueba piloto, excepto para los ítems de las preguntas 13 y 14 que se encuentran representados en las figuras 4-2 y 4-3.

Gráficamente (ver figura 4-4) se puede determinar que el nivel de aceptación de la aplicación es alto, características del juego como la música, colores, información proporcionada presentan una aceptación que supera el 90 %. Sin embargo factores como facilidad de uso muestra que a un grupo pequeño de usuarios (aproximadamente 10%) les resulto algo complicado utilizar el juego.

También encontramos que un grupo significativo de usuarios volvería a utilizar la aplicación y que además la recomendaría a sus amigos, esto se aprecia en la figuras 4-2 y 4-3.

¿Recomendarías este juego a tus amigos?

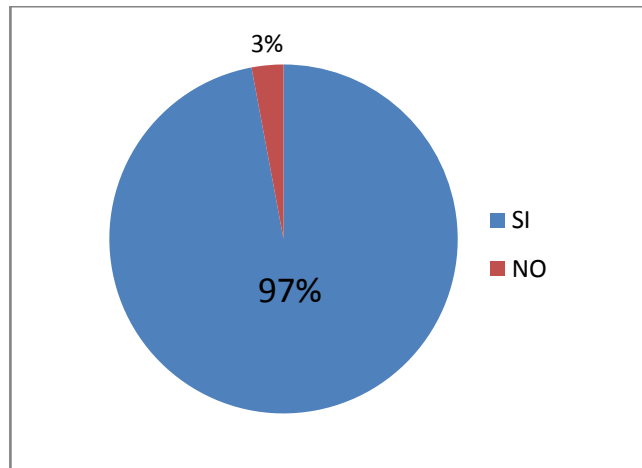


Figura 4-2. Resultados Prueba Piloto

¿Volverías a utilizar este juego?

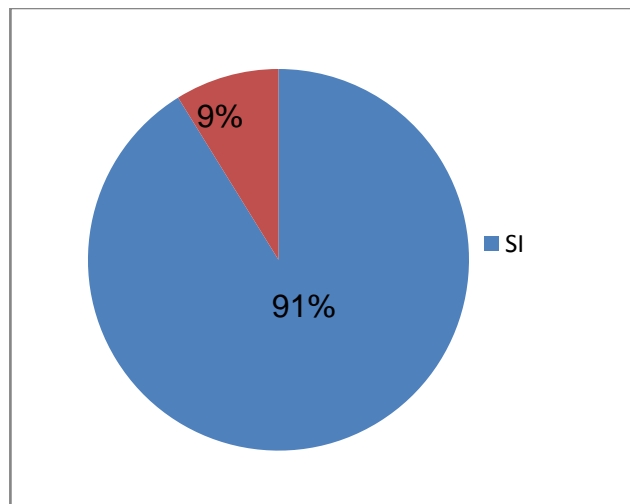


Figura 4-3. Resultados Prueba Piloto

NIVEL DE ACEPTACIÓN DEL JUEGO

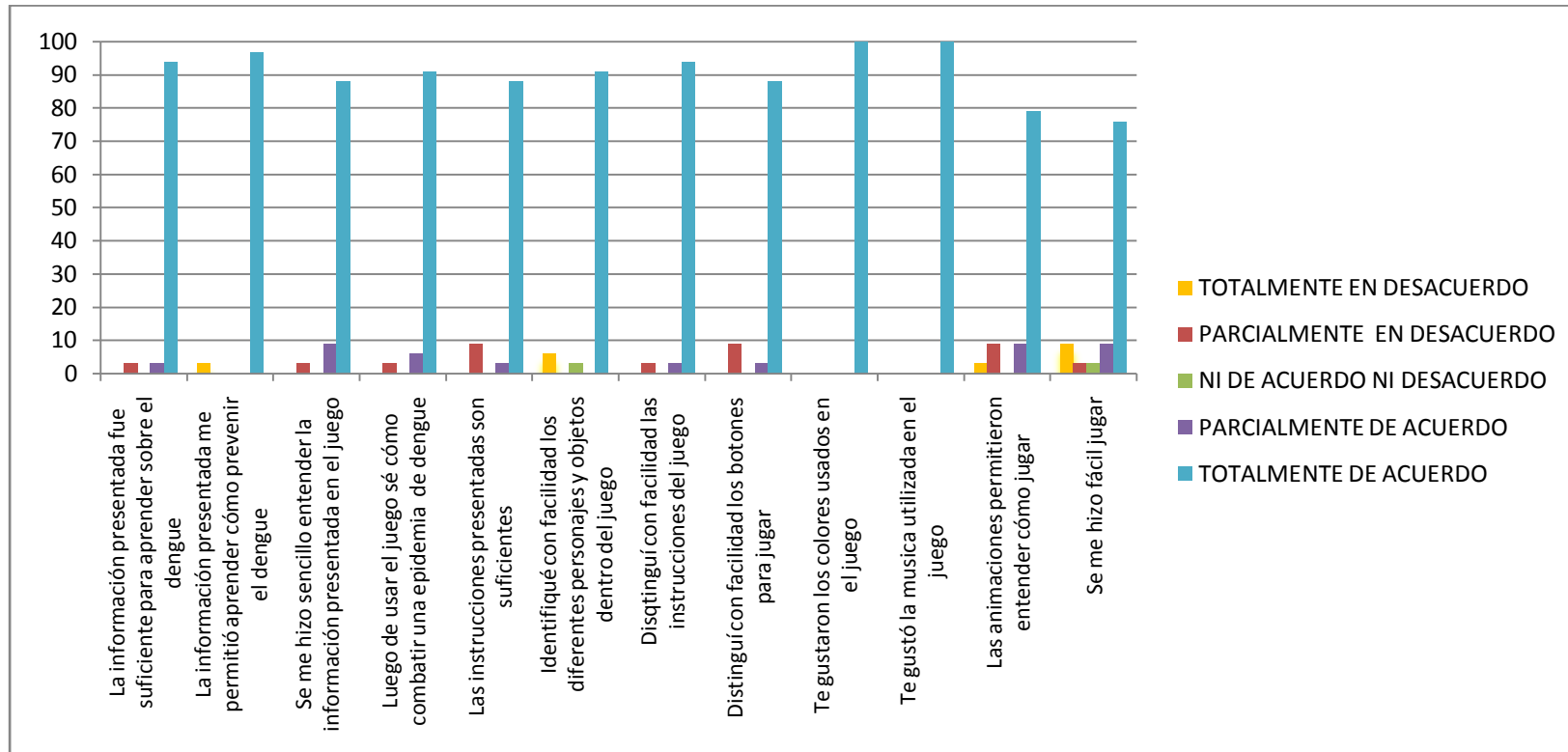


Figura 4-4. Resultados Prueba Piloto

4.3 Análisis y discusión de los Resultados

Los resultados obtenidos en la prueba piloto muestran que el nivel de aceptación de las características principales del juego que hacen que se cumplan los objetivos inicialmente planteados tales como la información sobre cómo prevenir el dengue mostrada en el juego, facilidad de uso, sonido coinciden con los esperados ya que superan el 90%, sin embargo será necesario realizar algunas mejoras que permitan a los usuarios obtener una mayor satisfacción al momento de jugar. Además se muestra que para un grupo significativo de usuarios fue algo complicado usar la aplicación, esto puede deberse a que no todos los equipos en los que se instaló el software para realizar la prueba cuentan con los requerimientos necesarios de funcionalidad del mismo como por ejemplo el uso de parlantes los cuales proporcionan una retroalimentación muy importante al usuario al momento de jugar.

Por otro lado cabe recalcar que por limitaciones de tiempo en el desarrollo de nuestro proyecto no se pudo realizar un estudio más profundo sobre el nivel de aceptación que tendría la aplicación en diferentes grupos de usuarios para así realizar una comparación y poder generalizar los resultados obtenidos anteriormente. También sería interesante conocer si los usuarios poseen conocimientos previos sobre la enfermedad y si no es así realizar una pequeña

campaña informativa al grupo de usuarios seleccionados lo que permitiría realizar un estudio pre-test y post-test y comparar los resultados obtenidos para verificar si la información adquirida a través del juego antes y después de dicha campaña fue suficiente para cumplir con los objetivos planteados.

Sin embargo, los resultados obtenidos en este estudio muestran que la aplicación es muy buena y que puede ser utilizada como herramienta de aprendizaje ya que cumple con las expectativas tanto para los usuarios (en este caso niños de escuelas) como para los creadores del juego cuyo objetivo principal fue construir una herramienta de concientización que permita a dichos usuarios saber cuáles son las medidas que se deben tomar para prevenir el contagio del dengue.

CONCLUSIONES

Con respecto a la herramienta utilizada para la implementación de nuestro motor de inferencia y de la interfaz gráfica de la aplicación podemos concluir lo siguiente:

1. El GameMaker probó ser una herramienta poderosa y flexible en la implementación tanto del motor de inferencia como de la interfaz gráfica del juego del dengue, permitiendo desarrollar e integrar ambas partes en una aplicación que brinda un entorno interactivo, intuitivo y agradable para el usuario, tanto en efectos visuales como en la información proporcionada para el aprendizaje.

Con respecto a los resultados obtenidos en la prueba piloto de la aplicación podemos concluir lo siguiente:

2. El nivel de aceptación tanto del entorno gráfico como de la información proporcionada en el juego es EXCELENTE, supera el 90%, por lo que se puede concluir que la aplicación desarrollada en este proyecto satisface los objetivos propuestos, permite a los usuarios aprender de una manera divertida a través de un entorno gráfico sencillo, interactivo e intuitivo cuáles son las principales medidas que se deben tomar para prevenir la incidencia de la enfermedad.

RECOMENDACIONES

Las recomendaciones son:

1. Promover el uso de esta aplicación en escuelas y colegios con la finalidad de crear en los estudiantes una cultura de prevención del dengue.
2. Instalar la aplicación en ordenadores que cuenten con los componentes (ver anexo E) necesarios para su ejecución, entre los cuales está el uso de audio que es muy importante para crear un entorno divertido.
3. Realizar una prueba piloto en adultos para determinar cuál es el nivel de aceptación de la aplicación en este grupo de usuarios.
4. Añadir para futuras versiones escenarios que involucren otro tipo de ambientes para el contagio del dengue como por ejemplo en zonas rurales la existencia de pozos de agua, zanjas, etc. representan peligro.

ANEXOS

ANEXO A

IMPLEMENTACIÓN DEL MOTOR DE INFERENCIAS EN GML

A continuación se presenta la implementación del código del motor de inferencias en Game Maker Language.

FORWARD CHAINING EN GML

```
/*
 * Nombre      : ForwardChaining
 * Prototipo   : List ForwardChaining(List)
 * Uso        : NT=ForwardChaining(Argumentos);
 * Descripción: Esta función ejecuta el algoritmo de Forward Chaining y
 * retorna la lista NT para el analisis de resultados.
 */
NR=0;
BC=ds_list_create();
ds_list_clear(BC);
NR=create_BC(BC);
MT=ds_list_create();
ds_list_clear(MT);

//A=ds_list_create();
LeerValoresIniciales(MT,argument0);
fin1=0;
while(not fin1)
{
    fin2=0;
    indice=0;
    while(not fin2)
    {
        if(NoUsada(ds_list_find_value(BC,indice)))
        {
            con=ds_list_find_value(ds_list_find_value(BC,indice),0);
            an=ds_list_find_value(ds_list_find_value(BC,indice),1);
            A=ExtraeAtributos(an);
            if(TodosAtribValor(A,MT))
            {
                if(Evalua(an,MT))
                {
                    C=DetAtrib(ds_list_find_value(BC,indice));
                    FoneValorEnMT(MT,Evalua(con,MT),C,indice);
                    MarcaRegla(ds_list_find_value(BC,indice));
                    fin2=1;
                }
            }
        }
        if(not fin2)
        {
            indice+=1;
            if(indice>=NR)
            {
                fin1=1;
                fin2=1;
            }
        }
    }
}
return MT;
```

FUNCIONES IMPLEMENTADAS EN GML

```
/*
 * Nombre      : LeerValoresIniciales
 * Prototipo   : void LeerValoresIniciales(List,List)
 * Uso         : LeerValoresIniciales(MT,argument1);
 * Descripción: En esta función se llena la lista MT con la lista de valores *
 * enviados y se llena los nombres de los atributos.
 */
Nombres=ds_list_create();
Valores=ds_list_create();
IdRegla=ds_list_create();

ds_list_add(Nombres,"BT"); //ID=0 - BONUS TIEMPO
ds_list_add(Nombres,"CL"); //ID=1 - CLIMA
ds_list_add(Nombres,"RP"); //ID=2 - RECIPIENTES PELIGROSOS
ds_list_add(Nombres,"MI"); //ID=3 - MOSQUITOS INFECTADOS
ds_list_add(Nombres,"MNI"); //ID=4 - MOSQUITOS NO INFECTADOS
ds_list_add(Nombres,"NP"); //ID=5 - NIVEL DE PELIGRO
ds_list_add(Nombres,"PI"); //ID=6 - PERSONAS INFECTADAS
ds_list_add(Nombres,"GI"); //ID=7 - GENERAR INFECTADOS
ds_list_add(Nombres,"GNI"); //ID=8 - GENERAR NO INFECTADOS
for(i=0;i<ds_list_size(Nombres);i+=1)
{
    ds_list_add(Valores,-1);
    ds_list_add(IdRegla,-1);
}

for(i=0;ds_list_find_value(argument1,i)!=-1;i+=1)
{
    value=ds_list_find_value(argument1,i)div 10;
    index=ds_list_find_value(argument1,i)mod 10;
    ds_list_replace(Valores,index,value);
}
ds_list_add(argument0,Nombres);
ds_list_add(argument0,Valores);
ds_list_add(argument0,IdRegla);

/*
 * Nombre      : NoUsada
 * Prototipo   : int NoUsada(List)
 * Uso         : if(NoUsada(ds_list_find_value(BC,indice)))
 * Descripción: Esta función revisa dentro de un item de la lista BC si una *
 * regla ha sido utilizada retornando 1 si la regla no ha sido usada y 0 de *
 * lo contrario.
 */
if(ds_list_find_value(argument0,2)==0)
    return 1;
else
    return 0;
```



```

/*****
* Nombre      : ExtraeAtributos
* Prototipo   : List ExtraeAtributos(String)
* Uso        : A=ExtraeAtributos(an);
* Descripción: Esta función extrae los nombres de todos los atributos
* o variables de un String dado y los guarda dentro de la lista A
*****/
tmp=argument0;
A=ds_list_create();
while(string_length(tmp)!=0)
{
    i=string_pos("&&",tmp);
    o=string_pos("||",tmp);
    if(i!=0&&o!=0)
    {
        if(i<o)
        {
            atrib=string_copy(tmp,0,i-1);
            tmp=string_delete(tmp,1,i+1);
        }
        else
        {
            atrib=string_copy(tmp,0,o-1);
            tmp=string_delete(tmp,1,o+1);
        }
    }
    else if(i!=0||o!=0)
    {
        if(i!=0)
        {
            atrib=string_copy(tmp,0,i-1);
            tmp=string_delete(tmp,1,i+1);
        }
        else
        {
            atrib=string_copy(tmp,0,o-1);
            tmp=string_delete(tmp,1,o+1);
        }
    }
    else
    {
        atrib=string_copy(tmp,0,string_length(tmp));
        tmp=string_delete(tmp,1,string_length(tmp));
    }
    ds_list_add(A,atrib);
}
for(i=0;i<ds_list_size(A);i+=1)
{
    tmp=ds_list_find_value(A,i);
    menor=string_pos("<",tmp);
    mayor=string_pos(">",tmp);
    igual=string_pos("=",tmp);
    if(menor!=0)
    {
        atrib=string_copy(tmp,0,menor-1);
        ds_list_replace(A,i,atrib);
    }
    else if(mayor!=0)
    {
        atrib=string_copy(tmp,0,mayor-1);
        ds_list_replace(A,i,atrib);
    }
    else
    {
        atrib=string_copy(tmp,0,igual-1);
        ds_list_replace(A,i,atrib);
    }
}
return A;

```

```

/*****
* Nombre      : TodosAtribValor
* Prototipo   : bool TodosAtribValor(List,List)
* Uso         : if(TodosAtribValor(A,MT))
* Descripción: Esta función verifica que todos los valores de la lista A
* tengan valor dentro de la lista MT, retorna false de encontrar un valor
* de A que no exista dentro de MT
*****/
for(i=0;i<ds_list_size(argument0);i+=1)
{
    idkey=ds_list_find_value(argument0,i);
    idr=ds_list_find_index(ds_list_find_value(argument1,0),idkey);
    if(ds_list_find_value(ds_list_find_value(argument1,1),idr)==-1)
        return false;
}
return true;

/*****
* Nombre      : Evalua
* Prototipo   : bool/int Evalua(String,List)
* Uso         : if(Evalua(an,MT))
* Descripción: Esta función hace la evaluación logica de un String que
* representa una expresión de una regla ya sea el argumento o el consecuente*
* y retorna el valor de verdad o valor de asignación respectivamente
*****/
tmp=argument0;
valores=ds_list_create();
operadores=ds_list_create();
while(string_length(tmp)!=0)
{
    i=string_pos("&&",tmp);
    o=string_pos("||",tmp);
    if(i!=0&&o!=0)
    {
        if(i<o)
        {
            ds_list_add(operadores,0);
            atrib=string_copy(tmp,0,i-1);
            menor=string_pos("<",atrib);
            mayor=string_pos(">",atrib);
            igual=string_pos("=",atrib);
            if(menor!=0)
            {
                valor1=string_copy(atrib,0,menor-1);
                valor2=string_copy(atrib,menor+1,string_length(atrib));
                idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
                valor1l=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
                if(valor1l<real(valor2))
                    ds_list_add(valores,true);
                else
                    ds_list_add(valores,false);
            }
        }
    }
}

```

```

else if(mayor!=0)
{
    valor1=string_copy(atrib,0,mayor-1);
    valor2=string_copy(atrib,mayor+1,string_length(atrib));
    idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
    valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
    if(valor11>real(valor2))
        ds_list_add(valores,true);
    else
        ds_list_add(valores,false);
}
else
{
    valor1=string_copy(atrib,0,igual-1);
    valor2=string_copy(atrib,igual+1,string_length(atrib));
    idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
    valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
    if(valor11==real(valor2))
        ds_list_add(valores,true);
    else
        ds_list_add(valores,false);
}

tmp=string_delete(tmp,1,i+1);
}
else
{
    ds_list_add(operadores,1);
    atrib=string_copy(tmp,0,o-1);
    menor=string_pos("<",atrib);
    mayor=string_pos(">",atrib);
    igual=string_pos("=",atrib);
    if(menor!=0)
    {
        valor1=string_copy(atrib,0,menor-1);
        valor2=string_copy(atrib,menor+1,string_length(atrib));
        idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
        valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
        if(valor11<real(valor2))
            ds_list_add(valores,true);
        else
            ds_list_add(valores,false);
    }
    else if(mayor!=0)
    {
        valor1=string_copy(atrib,0,mayor-1);
        valor2=string_copy(atrib,mayor+1,string_length(atrib));
        idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
        valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
        if(valor11>real(valor2))
            ds_list_add(valores,true);
        else
            ds_list_add(valores,false);
    }
    else
    {
        valor1=string_copy(atrib,0,igual-1);
        valor2=string_copy(atrib,igual+1,string_length(atrib));
        idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
        valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
        if(valor11==real(valor2))
            ds_list_add(valores,true);
        else
            ds_list_add(valores,false);
    }
    tmp=string_delete(tmp,1,o+1);
}
}

```

```

else if(i!=0||o!=0)
{
    if(i!=0)
    {
        ds_list_add(operadores,0);
        atrib=string_copy(tmp,0,i-1);
        menor=string_pos("<",atrib);
        mayor=string_pos(">",atrib);
        igual=string_pos("=",atrib);
        if(menor!=0)
        {
            valor1=string_copy(atrib,0,menor-1);
            valor2=string_copy(atrib,menor+1,string_length(atrib));
            idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
            valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
            if(valor11<real(valor2))
                ds_list_add(valores,true);
            else
                ds_list_add(valores,false);
        }
        else if(mayor!=0)
        {
            valor1=string_copy(atrib,0,mayor-1);
            valor2=string_copy(atrib,mayor+1,string_length(atrib));
            idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
            valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
            if(valor11>real(valor2))
                ds_list_add(valores,true);
            else
                ds_list_add(valores,false);
        }
        else
        {
            valor1=string_copy(atrib,0,igual-1);
            valor2=string_copy(atrib,igual+1,string_length(atrib));
            idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
            valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
            if(valor11==real(valor2))
                ds_list_add(valores,true);
            else
                ds_list_add(valores,false);
        }
        tmp=string_delete(tmp,1,i+1);
    }
    else
    {
        ds_list_add(operadores,1);
        atrib=string_copy(tmp,0,o-1);
        menor=string_pos("<",atrib);
        mayor=string_pos(">",atrib);
        igual=string_pos("=",atrib);
        if(menor!=0)
        {
            valor1=string_copy(atrib,0,menor-1);
            valor2=string_copy(atrib,menor+1,string_length(atrib));
            idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
            valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
            if(valor11<real(valor2))
                ds_list_add(valores,true);
            else
                ds_list_add(valores,false);
        }
        else if(mayor!=0)
        {
            valor1=string_copy(atrib,0,mayor-1);
            valor2=string_copy(atrib,mayor+1,string_length(atrib));
            idMT=ds_list_find_index(ds_list_find_value(argument1,0),valor1);
            valor11=ds_list_find_value(ds_list_find_value(argument1,1),idMT);
            if(valor11>real(valor2))
                ds_list_add(valores,true);
            else
                ds_list_add(valores,false);
        }
    }
}

```

```

        else
            ds_list_add(valores, false);
    }
    else
    {
        valor1=string_copy(atrib, 0, igual-1);
        valor2=string_copy(atrib, igual+1, string_length(atrib));
        idMT=ds_list_find_index(ds_list_find_value(argument1, 0), valor1);
        valor11=ds_list_find_value(ds_list_find_value(argument1, 1), idMT);
        if(valor11==real(valor2))
            ds_list_add(valores, true);
        else
            ds_list_add(valores, false);
    }
    tmp=string_delete(tmp, 1, c+1);
}
)
else
{
    atrib=string_copy(tmp, 0, string_length(tmp));
    menor=string_pos("<", atrib);
    mayor=string_pos(">", atrib);
    igual=string_pos("=", atrib);
    assign=string_pos("-", atrib);
    if(menor!=0)
    {
        valor1=string_copy(atrib, 0, menor-1);
        valor2=string_copy(atrib, menor+1, string_length(atrib));
        idMT=ds_list_find_index(ds_list_find_value(argument1, 0), valor1);
        valor11=ds_list_find_value(ds_list_find_value(argument1, 1), idMT);
        if(valor11<real(valor2))
            ds_list_add(valores, true);
        else
            ds_list_add(valores, false);
    }
    else if(mayor!=0)
    {
        valor1=string_copy(atrib, 0, mayor-1);
        valor2=string_copy(atrib, mayor+1, string_length(atrib));
        idMT=ds_list_find_index(ds_list_find_value(argument1, 0), valor1);
        valor11=ds_list_find_value(ds_list_find_value(argument1, 1), idMT);
        if(valor11>real(valor2))
            ds_list_add(valores, true);
        else
            ds_list_add(valores, false);
    }
    else if(igual!=0)
    {
        valor1=string_copy(atrib, 0, igual-1);
        valor2=string_copy(atrib, igual+1, string_length(atrib));
        idMT=ds_list_find_index(ds_list_find_value(argument1, 0), valor1);
        valor11=ds_list_find_value(ds_list_find_value(argument1, 1), idMT);
        if(valor11==real(valor2))
            ds_list_add(valores, true);
        else
            ds_list_add(valores, false);
    }
    else
    {
        valor2=string_copy(atrib, assign+1, string_length(atrib));
        return real(valor2);
    }
    tmp=string_delete(tmp, 1, string_length(tmp));
}
}
if(ds_list_size(operadores) !=0)
{
    tmpvalor=ds_list_create();
    ban=0;

```

```

for(i=0;i<ds_list_size(operadores);i+=1)
{
    if(ds_list_find_value(operadores,i)==1)
        ban=1;
}
if(ban==1)
{
    for(i=0;i<ds_list_size(operadores);i+=1)
    {
        operador=ds_list_find_value(operadores,i);
        if(operador==1)
        {
            if(ds_list_find_value(valores,i)|| ds_list_find_value(valores,i+1))
                ds_list_replace(valores,i+1,true);
            else
                ds_list_replace(valores,i+1,false);
            ds_list_replace(valores,i,-1);
        }
    }
    for(i=0;i<ds_list_size(valores);i+=1)
    {
        if(ds_list_find_value(valores,i)!=-1)
            ds_list_add(tmpvalor,ds_list_find_value(valores,i))
    }
    if(ds_list_size(tmpvalor)>1)
    {
        for(i=0;i<ds_list_size(tmpvalor)-1;i+=1)
        {
            if(ds_list_find_value(tmpvalor,i)&& ds_list_find_value(tmpvalor,i+1))
                ds_list_replace(tmpvalor,i+1,true);
            else
                ds_list_replace(tmpvalor,i+1,false);
        }
    }
    return ds_list_find_value(tmpvalor,ds_list_size(tmpvalor)-1);
}
else
{
    for(i=0;i<ds_list_size(operadores);i+=1)
    {
        if(ds_list_find_value(valores,i)&& ds_list_find_value(valores,i+1))
            ds_list_replace(valores,i+1,true);
        else
            ds_list_replace(valores,i+1,false);
    }
    return ds_list_find_value(valores,ds_list_size(valores)-1);
}
return ds_list_find_value(valores,ds_list_size(valores)-1);

/*****
* Nombre      : DetAtrib
* Prototipo   : String DetAtrib(List)
* Uso         : C=DetAtrib(ds_list_find_value(BC,indice));
* Descripción: Esta función retorna el nombre del atributo dentro
* del consecuente de una regla.
*****/
con=ds_list_find_value(argument0,0);
assign=string_pos("-",con);
atrib=string_copy(con,0,assign-1);
return atrib;

```

```

|*****
* Nombre      : PoneValorEnMT                               *
* Prototipo   : void PoneValorEnMT(List,Int,String,Int)    *
* Uso        : PoneValorEnMT(MT,Evalua(con,MT),C,indice);  *
* Descripción: Esta función actualiza los valores dentro de MT para un *
* atributo dado.                                           *
*****/
/*PoneValorEnMT(MT,Evalua(BC(i,1)),C,i)*/
idMT=ds_list_find_index(ds_list_find_value(argument0,0),argument2);
ds_list_replace(ds_list_find_value(argument0,1),idMT,argument1);
ds_list_replace(ds_list_find_value(argument0,2),idMT,argument3);

|*****
* Nombre      : MarcaRegla                                 *
* Prototipo   : void MarcaRegla(List)                     *
* Uso        : MarcaRegla(ds_list_find_value(BC,indice)); *
* Descripción: Esta función marca con 1 como usada a la regla enviada *
*****/
ds_list_replace(argument0,2,1);

```

ANEXO B

ENTREVISTA CON EL EXPERTO

En este apartado se presenta el cuestionario de preguntas utilizado en las entrevistas efectuadas con el experto en epidemiología Dr. José Ochoa de la Torre, el mismo fue realizado con la finalidad de recolectar información necesaria para el desarrollo de nuestro proyecto, además se hace una breve explicación de cómo fue utilizada dicha información.

El cuestionario consiste en un conjunto de preguntas abiertas relacionadas con la enfermedad del Dengue y con los requerimientos de la aplicación. La información que se obtuvo a través del mismo durante las entrevistas fue debidamente procesada y analizada con el objetivo de armar posteriormente el conjunto de reglas que utiliza el motor de inferencias de nuestra aplicación. La información más relevante fue estructurada y presentada en este documento de tal manera que el lector capte las ideas principales en las que se basa nuestro proyecto.

CUESTIONARIO

1. ¿Qué es el dengue?
2. ¿Cuáles son los síntomas?
3. ¿Cómo actuar ante la presencia de esos síntomas?
4. ¿Existen varios tipos de dengue? ¿En qué se diferencian?
5. ¿Cómo se transmite esta enfermedad?
6. ¿Se puede contagiar de persona a persona?
7. ¿Cómo se reproduce el mosquito transmisor del dengue?

8. ¿La picadura de cualquier tipo de mosquito provoca Dengue?
9. ¿Qué medidas de prevención se deben tomar para prevenir esta enfermedad?
10. ¿Qué tipos de criaderos son los que prefieren los mosquitos transmisores del dengue para reproducirse?
11. ¿Cuáles son los diferentes factores que hacen que un ambiente sea ideal para la reproducción del mosquito?
12. ¿Qué antecedentes ha tenido esta enfermedad en nuestro país?
13. ¿En qué tipo de ambientes es mayor o menor el peligro de que se desarrolle y se transmita la enfermedad?
14. ¿Qué tipo de información se maneja para realizar un control epidemiológico de dengue?
15. Uno de los objetivos de nuestro proyecto es desarrollar una herramienta educativa y de concientización que permita a la gente saber qué medidas tomar para prevenir el dengue. Con respecto a esto ¿cuáles serían sus expectativas y sugerencias sobre lo que debería tener el sistema a desarrollar?
16. Otro de los objetivos de nuestro proyecto es desarrollar una aplicación sencilla para el usuario. ¿Qué tipo de información relacionada con la enfermedad se debería manejar para cumplir con este objetivo?
17. La aplicación a desarrollar básicamente es un juego, el mismo que involucrará información relacionada con el Dengue, ¿qué tipo de información le gustaría ver dentro del juego?
18. ¿A qué se refiere con ambientes reales?
19. ¿Qué tipo de ambientes le gustaría ver en el juego?
20. Según su criterio ¿qué personajes deberían estar presentes dentro del juego?
¿Por qué?

ANEXO C

INSTRUMENTO UTILIZADO PARA MEDIR EL NIVEL DE ACEPTACIÓN DE LA APLICACIÓN

Objetivo:

Este cuestionario busca obtener información sobre el nivel de aceptación de la información mostrada en el juego. Los datos vertidos en este cuestionario serán totalmente confidenciales.

1.-Información general del entrevistado

Edad: _____	Unidad Educativa: _____
Género: F <input type="checkbox"/>	M <input type="checkbox"/>

Para la siguiente sección utilice la siguiente escala para valorar cada enunciado

1. TOTALMENTE EN DESACUERDO
2. PARCIALMENTE EN DESACUERDO
3. NI DE ACUERDO NI DESACUERDO
4. PARCIALMENTE DE ACUERDO
5. TOTALMENTE DE ACUERDO

Marca con (X) la alternativa deseada, por ejemplo si marcas 5, indicarás que estás totalmente de acuerdo con dicho enunciado.

2.- Con relación a la información presentada en el juego

	1	2	3	4	5
La información presentada fue suficiente para aprender sobre el dengue.					
La información presentada me permitió aprender sobre cómo prevenir el dengue.					
Se me hizo sencillo entender la información presentada en el juego.					
Luego de usar el juego, sé cómo combatir una epidemia de dengue.					
Las instrucciones presentadas son suficientes.					

3.- En relación a los mundos del juego

	1	2	3	4	5
Identifiqué con facilidad los diferentes personajes y objetos dentro del juego					
Distinguí con facilidad las instrucciones del juego					
Distinguí con facilidad los botones para jugar					
Te gustaron los colores usados en el juego					

Te gustó la música utilizada en el juego					
Las animaciones permitieron entender cómo jugar					
Se me hizo fácil jugar					

	SI	NO
¿Recomendarías este juego a tus amigos?		
¿Te gustaría volver a utilizar este juego?		

ANEXO D

FOTOS DE LA PRUEBA PILOTO



ANEXO E

REQUISITOS PARA INSTALAR LA APLICACIÓN

Para ejecutar una aplicación desarrollada en Game Maker se requieren los siguientes componentes [11]:

- Una PC moderna con procesador Pentium y sistema operativo Windows 2000, Me, XP, Vista, Windows 7, Mac OS X 10.5 o Mac OS X 10.6.
- DirectX 8 o superior.
- Tarjeta gráfica compatible con DirectX 8 con al menos 32 MB de memoria, resolución de al menos 800x600 y colores de 16 bits o 32 bits.
- Tarjeta de sonido compatible con DirectX 8.0 o superior y que la versión 8.0 o superior de DirectX esté instalada en tu ordenador.
- Un DEP (Data Execution Prevention) puede bloquear la ejecución del entorno de desarrollo de Game Maker 8. DEP puede ser deshabilitado, establecida en los programas y servicios de Windows, o una excepción DEP debe ser creado para Game_Maker.exe.

GLOSARIO

Aplicación: Programa informático que permite a un usuario utilizar una computadora con un fin específico.

Interactivo: Característica que permite establecer comunicación entre una aplicación y un usuario.

Multimedia: Sistema u objeto que utiliza varios medios como videos, imágenes, sonido, y texto para transmitir información.

Inteligencia Artificial: Rama de la computación que se ocupa de construir hardware y software con comportamiento inteligente.

Dengue: Enfermedad infecciosa transmitida por la picadura del mosquito hembra Aedes Aegypti.

Sistema Experto: Software que simula el comportamiento de un experto humano en la solución de un problema específico.

Regla de inferencia: Estructura de representación del conocimiento que tiene la forma Si <condición> entonces <acción>

Algoritmo: Conjunto de pasos ordenados que se siguen para resolver un problema en particular.

Script: Porción de código de un programa que realiza una tarea específica.

Escenario: En este caso término que hace referencia a cada uno de los niveles o mundos dentro del juego.

BIBLIOGRAFÍA

- [1] Bar, M., El Aedes aegypti y la transmisión del Dengue, 2001, <http://exa.unne.edu.ar/biologia/artropodos/El%20Aedes%20aegypti%20y%20la%20transmision%20del%20dengue.pdf>, fecha de consulta Octubre del 2011
- [2] Matilde, C., Sistemas Basados en Reglas, 2004, http://ai.frm.utn.edu.ar/micesari//files/SISTEMAS_BASADOS_REGLAS.pdf, fecha de consulta Septiembre del 2011
- [3] Casas, S., Exploración de Reglas de Inferencia para Automatizar la Refactorización Aspectual, 2008 <http://www.oocities.org/espanol/profeprog2/INVPAPER27.pdf>, fecha de consulta Septiembre del 2011
- [4] Gamarra, A., & Gamarra, J. D., Inteligencia Artificial. Diseño de Sistemas Basados en Reglas con Encadenamiento hacia adelante, 2006 <http://es.scribd.com/doc/42091193/03-Encadenamiento-hacia-adelante>, fecha de consulta Octubre del 2011
- [5] Márquez, J., Introducción a los Sistemas Expertos, 2004 <http://redcientifica.com/doc/doc199908210001.html>, fecha de consulta Octubre del 2011
- [6] Umich, M., Programación Lógica, 2005 <http://www.google.com.ec/url?sa=t&rct=j&q=sistemas%20expertos%20filetype%3Apt&source=web&cd=2&ved=0CC0QFjAB&url=http%3A%2F%2Fisc.fie.umich.mx%2F~jjap%2Fpl.ppt&ei=B3QsT5GEB8Pu0gG->, fecha de consulta Noviembre del 2011
- [7] Castillo, E., Gutiérrez, J., & Hadi, A., Sistemas expertos y modelos de redes probabilísticas, 1997, <http://personales.unican.es/gutierjm/papers/BookCGH.pdf>, fecha de consulta Septiembre del 2011

- [8] Gutiérrez, P., Sistemas Expertos Basados en Reglas, 2008
<http://personales.unican.es/gutierjm/cursos/expertos/Reglas.pdf>, fecha de consulta Agosto del 2011
- [9] C.A.E.I.E Instituto Técnico , Campaña para erradicar el dengue , (2010). <http://www.campanaparaerradicareldengueelcaie.blogspot.com> , fecha de consulta Diciembre del 2011
- [10] Overmars, M., Designing Games with game Maker, 2006,
<http://gamemaker.info/en/manual>, fecha de consulta Octubre del 2011
- [11] Cominotti, P., Motor de Inferencias para Sistemas Basados. en Conocimientos con Arquitectura SaaS, 2004, Recuperado el 24 de Agosto de 2011,
<https://docs.google.com/viewer?a=v&q=cache:MKKc6B9IQ-sJ:openodmanphp.googlecode.com/svn-> , fecha de consulta Noviembre del 2011