

Universidad de las Ciencias Informáticas
Facultad 5
Realidad Virtual



**Simulador de Equipo de Rayos X para la
Aduana**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Javier Hechavarría Roldán

Tutor: Jaime Gonzalez Campistruz

Co-tutora: Lien Muguercia Torres

Ciudad de la Habana, junio del 2009

Agradecimientos

*A todas las personas que me han apoyado en la realización de este proyecto, a mi tutor Jaime,
y en especial a la Revolución, por permitirme estudiar en esta prestigiosa universidad*

Agradecimientos

Dedicatoria

.....este trabajo de diploma está dedicado especialmente a toda mi familia, en especial a mi hermana Janet, a mis padres René y Lourdes y a mi novia Beatriz por dejarme cumplir este sueño.

También no quiero pasar por alto a todos mis amigos de estos años, por el cual pasamos tantos buenos momentos, como también malos.....

Dedicatoria

Resumen

Con el objetivo de llevar a cabo un proceso eficiente de detección de indicios, la Aduana General de la República de Cuba tiene en cuenta la constante preparación y capacitación del personal que labora en terminales portuarias y aéreas. Hasta ahora esta preparación se llevaba a cabo mediante presentaciones y exámenes teóricos y, solo en ocasiones, se realizaba algunas prácticas en los salones de operación de la Aduana. De esta forma no se garantizaba en su totalidad la correcta capacitación de los inspectores, quedando el riesgo de que saltaran por alto determinadas ilegalidades.

Como parte de esta preparación, en el proyecto Entrenadores Aduaneros perteneciente al Polo Productivo de Realidad Virtual de la Facultad 5, se desarrolla un simulador de rayos X que a grandes rasgos debe manifestar el realismo de las imágenes y las funcionalidades de un equipo de este tipo. Dicha acción se desglosa entre otras funciones en: la aplicación de filtros a las imágenes, la representación por sonidos del ambiente de las terminales de aeropuertos y puertos y la transición de imágenes por el monitor.

El sistema resultante brindará un mayor acercamiento a las operaciones con los equipos de Rayos X que existen hoy en día en la Aduana, ayudando a que el operador de este equipo este más capacitado y vele por la seguridad de los pasajeros de una forma eficiente.

Palabras Clave: simulador de rayos X, filtro de imágenes, transición de imágenes

Índice

Índice de Figuras	9
Índice de Tablas	10
Introducción	1
Capítulo 1: Fundamentación Teórica	4
Introducción	4
1.1 . Simuladores.....	4
1.2. Características de los Simuladores	5
1.3. Etapas fundamentales del desarrollo de un Simulador	5
1.4. Tipos de Simuladores.....	6
➤ Simuladores de tránsito:.....	6
➤ Simuladores de cálculo:	6
➤ Simuladores de producto:	6
➤ Simuladores contra-producto:.....	7
➤ Simuladores de sistemas:	7
1.5. Ventajas y Desventajas de los Simuladores	7
1.6. Equipos de Rayos X.....	8
1.6.1. Principales partes componentes de los sistemas de rayos x.	8
1.6.2. Principios básicos de operación de los sistemas.....	9
1.6.3. Coloración asignada por los fabricantes de los equipos según la composición química de la materia.....	10
➤ La distribución de colores se realiza de la siguiente forma:.....	10
1.6.4. Funciones de realces de los equipos de RX	12
1.6.5. Equipos de RX empleados en Cuba	14
1.7. Representación de imágenes	15

Índice

1.8 Recomendaciones para el trabajo con un equipo de Rayos X.....	16
1.9 Algunos simuladores característicos	17
1.8. Conclusiones.....	18
Capítulo 2: Soluciones técnicas	19
Introducción	19
2.1. Especificación del simulador	19
2.2. Utilización de la Simple DirectMedia Layer (SDL).....	23
2.2.1. Ventajas	24
2.3. Herramientas y Tecnologías a utilizar	25
2.4. Conclusiones.....	25
Capítulo 3: Características del Sistema.	26
Introducción	26
3.1. Descripción del modelo de dominio	26
3.2. Modelo del Dominio.	26
3.2.1. Glosario de términos del Dominio.	27
3.3 Actores del sistema a automatizar.	27
3.4 . Requisitos Funcionales.....	28
3.5 Diagrama de Casos de Uso del Sistema.....	29
3.6 . Requisitos No Funcionales.	29
3.7 Descripciones de los casos de uso de sistema.	31
Capítulo 4: Diseño del Sistema.....	41
Introducción.	41
Modelo de Diseño.....	41
4.1. Diagramas de Clases de Diseño.	42
4.2. Diagramas de Secuencia del Diseño	45
4.2.1. Realización del caso de uso “Iniciar Transición de Imágenes”	45
4.2.2. Realización del caso de uso “Aplicar Filtros”	46

Índice

4.2.3.	Realización del caso de uso “Detener transición de imágenes”	¡Error! Marcador no definido.
4.2.4.	Realización del caso de uso “Marcar imágenes con sospechas”	47
4.2.5.	Realización del caso de uso “Ver resultado”	47
4.2.6.	Realización del caso de uso “Crear Perfil”	48
4.2.7.	Realización del caso de uso “Aplicar zoom a las imágenes”	48
4.2.8.	Realización del caso de uso “Controlar cambio de fase”	49
4.2.9.	Realización del caso de uso “Ver ayuda del funcionamiento del simulador”	¡Error! Marcador no definido.
4.2.10.	Realización del caso de uso “Configurar opciones”	50
4.2.11.	Realización del caso de uso “Cerrar aplicación”	¡Error! Marcador no definido.
4.2.12.	Realización del caso de uso “Alterar imágenes”	51
4.2.13.	Realización del caso de uso “Configurar imágenes según nivel de dificultad”	51
4.2.14.	Realización del caso de uso “Cargar Imágenes”	52
4.3.	Descripción de las clases	52
4.3.1.	Clase Controladora “Application_Control”	52
4.3.2.	Clase Controladora “State_Control”	54
4.3.3.	Clase Base “State”	55
4.3.4	Clase SubBase “Loading_State”	56
4.3.5	Clase SubBase “Menu_State”	57
4.3.6.	Clase SubBase “Game_State”	58
4.3.7.	Clase SubBase “CEvents”	59
4.3.8.	Clase SubBase “CControlObject”	60
4.3.9	Clase SubBase “CObject”	61
4.3.10.	Clase SubBase “CButton”	62
4.3.11	Clase SubBase “CXRaysView”	62
4.3.12	Clase SubBase “CDialogResult”	64

Índice

4.3.13 Clase Componente “CSprite”	66
4.3.14 Clase Componente “Cframe”	67
Capítulo 5: Implementación	69
Introducción	69
5.1. Estándar de Codificación	69
5.2. Diagrama de despliegue.	70
5.3. Diagramas de Componentes	71
5.3.1. Paquete del Simulador	71
5.3.2. Diagrama de Componente, paquete “Application”	72
5.3.3. Diagrama de Componente, paquete “Estados”	73
Conclusiones	74
Recomendaciones	75
Bibliografía	76
Glosario de Abreviaturas	77
Glosario de Términos	78

Índice de Figuras

FIGURA 1: IMAGEN ORIGINAL DE UN EQUIPAJE EN UN EQUIPO DE RAYOS X	8
FIGURA 2: BLANCO Y NEGRO.....	12
FIGURA 3: VIDEO INVERSO	13
FIGURA 4: INORGÁNICOS.....	13
FIGURA 5: SOLO ORGÁNICOS	14
FIGURA 6: EQUIPO DE RX HEIMANN.....	15
FIGURA 7: REPRESENTACIÓN DE PAQUETES Y SU DENSIDAD A TRAVÉS DE UN MONITOR DE EQUIPO DE RX.....	16
FIGURA 8: SIMULADOR DE RX SIMFOX	18
FIGURA 9: IMAGEN DE LA INTERFAZ DEL MENÚ	21
FIGURA 10: IMAGEN DE LA INTERFAZ DEL SIMULADOR DE EQUIPOS DE RAYOS X	22
FIGURA 11: DIAGRAMA CORRESPONDIENTE AL MODELO DEL DOMINIO.	26
FIGURA 12: DIAGRAMA DE CASOS DE USO DEL SISTEMA	29
FIGURA 13: CLASES CONTROLADORAS DEL DISEÑO	42
FIGURA 14: CLASES COMPONENTES DEL DISEÑO.....	43
FIGURA 15: CLASES HEREDERAS DEL DISEÑO	44
FIGURA 16: DIAGRAMA DE SECUENCIA “INICIAR TRANSICIÓN DE IMÁGENES”	45
FIGURA 17: DIAGRAMA DE SECUENCIA “APLICAR FILTROS”	46
FIGURA 18: DIAGRAMA DE SECUENCIA “DETENER TRANSICIÓN DE IMÁGENES”	¡ERROR!
MARCADOR NO DEFINIDO.	
FIGURA 19: DIAGRAMA DE SECUENCIA “MARCAR IMÁGENES CON SOSPECHAS”	47
FIGURA 20: DIAGRAMA DE SECUENCIA “VER RESULTADO”	47
FIGURA 21: DIAGRAMA DE SECUENCIA “CREAR PERFIL ”	48
FIGURA 22: DIAGRAMA DE SECUENCIA “APLICAR ZOOM A LAS IMÁGENES”	48
FIGURA 23: DIAGRAMA DE SECUENCIA “CONTROLAR CAMBIO DE FASE”	49
FIGURA 24: DIAGRAMA DE SECUENCIA “VER AYUDA DEL FUNCIONAMIENTO DEL SIMULADOR”	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA 25: DIAGRAMA DE SECUENCIA “CONFIGURAR OPCIONES”	50
FIGURA 26: DIAGRAMA DE SECUENCIA “CERRAR APLICACIÓN”	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA 27: DIAGRAMA DE SECUENCIA “ALTERAR IMÁGENES”	51
FIGURA 28: DIAGRAMA DE SECUENCIA “ALTERAR IMÁGENES”	51

Índice de Figuras y Tablas

FIGURA 29: <i>DIAGRAMA DE SECUENCIA “ALTERAR IMÁGENES”</i>	52
FIGURA 30: <i>DIAGRAMA DE DESPLIEGUE</i>	70
FIGURA 31: <i>PAQUETE DEL SIMULADOR</i>	71
FIGURA 32: <i>DIAGRAMA DE COMPONENTE “APPLICATION”</i>	72
FIGURA 33: <i>DIAGRAMA DE COMPONENTE “ESTADOS”</i>	73

Índice de Tablas

TABLA I. <i>DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA A AUTOMATIZAR</i>	28
TABLA II: <i>CU “INICIAR TRANSICIÓN DE IMÁGENES”</i>	32
TABLA III: <i>CU “APLICAR FILTRO A LAS IMÁGENES”</i>	33
TABLA IV: <i>CU “DETENER TRANSICIÓN DE IMÁGENES”</i> ¡ERROR! MARCADOR NO DEFINIDO.	
TABLA V: <i>CU “SELECCIONAR IMÁGENES”</i>	34
TABLA VI: <i>CU “VER RESULTADO”</i>	34
TABLA VII: <i>CU “CREAR PERFIL”</i>	35
TABLA VIII: <i>CU “APLICAR ZOOM A LAS IMÁGENES”</i>	36
TABLA IX: <i>CU “CONTROLAR CAMBIO DE FASE”</i>	31
TABLA X: <i>CU “VER AYUDA DEL FUNCIONAMIENTO DEL SIMULADOR”</i> ; ¡ERROR! MARCADOR NO DEFINIDO.	
TABLA XI: <i>CU “CONFIGURAR OPCIONES”</i>	37
TABLA XII: <i>CU “CERRAR APLICACIÓN”</i> ¡ERROR! MARCADOR NO DEFINIDO.	
TABLA XIII: <i>CU “ALTERAR EQUIPAJES”</i>	38
TABLA XIV: <i>CU “CARGAR IMÁGENES”</i>	39
TABLA XV: <i>CU “CONFIGURAR IMÁGENES SEGÚN NIVEL DE DIFICULTAD”</i>	39
TABLA XVI: <i>CLASE “APPLICATION_CONTROL”</i>	52
TABLA XVII: <i>CLASE “STATE_CONTROL”</i>	54
TABLA XVIII: <i>CLASE “STATE”</i>	55
TABLA XIX: <i>CLASE “LOADING_STATE”</i>	56

Índice de Figuras y Tablas

TABLA XX: CLASE “MENU_STATE”	57
TABLA XXI: CLASE “GAME_STATE”	58
TABLA XXII: CLASE “CEVENTS”	59
TABLA XXIII: CLASE “CCONTROBJECT”	60
TABLA XXIV: CLASE “COBJECT”	61
TABLA XXV: CLASE “CBUTTON”	62
TABLA XXVI: CLASE “CXRAYSVIEW”	62
TABLA XXVII: CLASE “CDIALOGRESULT”	64
TABLA XXVIII: “CLASE “CSPRITE”	66
TABLA XXIX: “CLASE “CFRAME”	67

Introducción

La Aduana General de la República es el sector encargado de velar por la seguridad nacional en los diferentes puertos y aeropuertos del país. De ellos depende en gran número, anular el tráfico de droga, de armas, ataques terroristas y otras acciones consideradas ilegales y de gran impacto negativo en la sociedad.

Es por ello que la constante preparación de su personal y la capacitación de aquellos que se unen a sus filas en la detección e identificación de ilegalidades, es de vital importancia y es uno de los objetivos principales de la organización.

Una de las potentes herramientas usadas por estos inspectores en su trabajo es el llamado Equipo de Rayos X, el cual es una especie de escaneador de equipajes con el cual los inspectores pueden identificar y detectar cualquier caso de entrada al país de elementos ilegales. Precisamente su manejo o uso necesita del inspector un previo conocimiento y entrenamiento con el equipo, de manera que pueda hacer uso de este sin ningún inconveniente.

Este proceso de entrenamiento y capacitación se lleva a cabo haciendo uso de presentaciones y test elaborados por especialistas de la Escuela Nacional de Formación Aduanera (ENFA) y en ocasiones, estos inspectores son trasladados a los salones de operaciones de la Aduana ya sea en los aeropuertos como en los puertos, a ejercitar sus conocimientos. Esto tiene de inconveniente que existe un alto riesgo de infiltración de ilegalidades, puesto que al tener personal no altamente capacitado en el terreno, puede ocasionarse un fallo y pasar por alto la introducción de droga, armas y en el peor de los casos, no detectar un ataque terrorista sobre nuestra sociedad.

De ahí que la jefatura de la entidad en cuestión decidiera desarrollar un sistema de realidad virtual con el objetivo de sustituir los medios de preparación existente por uno más convincente y le encargara la tarea del desarrollo de este sistema a un proyecto de la facultad 5 de la Universidad de las Ciencias Informáticas (UCI).

Hasta ahora, en la facultad se ha desarrollado diferentes sistemas de Realidad Virtual con diferentes propósitos y clientes, pero no se ha diseñado he implementado un sistema que simule el funcionamiento y comportamiento de un equipo de rayos X semejante a los usados por la Aduana.

Introducción

Estos argumentos constituyen la **Situación Problemática** de esta investigación por ello el **Problema Científico** identificado es: “¿Cómo lograr el desarrollo de un software que simule la actividad de un equipo de rayos x de la Aduana?”. Siendo el **Objeto de Estudio** “Simuladores Virtuales” y el **Campo de Acción** específicamente sería “Simulación de Equipos Rayos X para la detección de ilegalidades en las instituciones aduaneras.”

Una vez planteada la problemática principal se define como **Objetivo General** para el presente trabajo de diploma, “Crear un simulador de Equipos de Rayos X para la preparación y formación de los inspectores aduaneros en las instituciones de la Aduana.” Para dar cumplimiento al objetivo anteriormente planteado será necesario cumplimentar las siguientes **Tareas de Investigación**:

- Realizar el análisis de las tecnologías y herramientas que dan solución a las necesidades de la Aduana.
- Capturar los requisitos funcionales del simulador de equipos de Rayos X
- Realizar el análisis y diseño del simulador de equipos de Rayos X para la Aduana
- Realizar la implementación del simulador de equipos de Rayos X para la Aduana
- Garantizar la calidad y el funcionamiento de la propuesta de solución.
- Garantizar la implantación de la propuesta de solución.

Métodos Teóricos:

- Histórico-Lógico para estudiar la evolución y desarrollo de los simuladores, conociendo así los diferentes simuladores existentes en la actualidad, como el uso que se les da a dichas aplicaciones informáticas.
- Analítico-Sintético, haciendo énfasis en el análisis de las teorías, documentos, entre otros, que permite la extracción de los elementos más importantes para procesar la información y elaborar conclusiones. Este método permitirá la extracción de los requisitos tanto funcionales como no funcionales del sistema.

- Modelación Teórica: que no es más que el método que opera de forma práctica o teórica con un objeto, no de forma directa si no, utilizando cierto sistema intermedio. Se escoge este método para el desarrollo del análisis y diseño de una solución técnica propuesta, este método permite la creación de modelos, diagramas, propuestas, alternativas y estrategias.

Capítulo 1: Fundamentación Teórica

Introducción

Intentar simular los efectos de cualquier equipo, donde el individuo tenga algún tipo de contacto físico con sistemas automatizados, es para cualquier programador un gran desafío. El desarrollo de sistemas que sean óptimos y lo suficientemente reales, constituye el principal objetivo de los investigadores y una demanda en el campo de la realidad virtual.

Con la rápida evolución de la informática es posible simular y desarrollar un modelo virtual y esparcir de esta forma el realismo obtenido.

Los simuladores de Equipos de Rayos X, mayormente utilizados en puertos y aeropuertos son uno de estos fenómenos, que a la vista se podrá entender el significado de su desarrollo e implantación en estos lugares debido a que son de vital importancia para una mayor seguridad.

En este capítulo se profundizará acerca de los diferentes métodos de simulación y visualización existentes en el mundo, así como conceptos generales de las leyes químicas que rigen a los Rayos X aplicados a los bolsos que pasan una vez por el equipo.

1.1 . Simuladores

Los simuladores permiten la construcción de modelos virtuales que interaccionan con el usuario de igual forma o transmitiéndole las mismas sensaciones que el modelo real. Desde los grandes simuladores utilizados en la aeronáutica para el entrenamiento militar hasta los de riesgos, desarrollados en los años 80, su empleo se ha acrecentado en igual forma que los costes de hardware y su desarrollo se ha reducido de forma considerable.

Un simulador es una aplicación informática que simula el comportamiento y propiedades reales de un producto o sistema, y de los elementos que interactúan con él: usuarios, eventos externos, otros productos o sistemas.

1.2. Características de los Simuladores

El avance y desarrollo de las aplicaciones de simulación constituye un proceso de gran complejidad con respecto a las soluciones convencionales, y requiere para su tratamiento, del aporte de especialistas o expertos en las diferentes áreas de conocimiento.

De esta forma, dicho proceso, se apoya sobre la base de tres ejes de desarrollo complementarios. (2)

- **Educativo o Instruccional:** cuyo aporte se centra en la definición de las necesidades educativas primordiales, a cubrir por el aplicativo. Asimismo, debe plasmar la conducta de entrada y campo vital de la población objeto.
- **Comunicacional:** define la interfaz del usuario (zona de interacción usuario-aplicativo). La usabilidad y accesibilidad del sistema, aplicada a entornos altamente interactivos, constituye un factor preponderante en este apartado.
- **Soporte Tecnológico:** cuyo objetivo primordial es determinar la adecuación de las posibilidades tecnológicas de desarrollo, integrando los ejes temáticos antes mencionados.

1.3. Etapas fundamentales del desarrollo de un Simulador

Los simuladores resultan hoy en día una herramienta muy útil, donde los clientes se sienten atraídos por estos avances tecnológicos, para lograr tales propósitos su desarrollo se semienta o caracteriza por las siguientes etapas: (1)

- Fase de concepto: para definir la funcionalidad y usabilidad de un producto, solo o como parte de un sistema complejo; para analizar y evaluar un proceso antes de estar diseñarlo o implantarlo.
- Fase de diseño del producto: para elaborar un prototipo y verificarlo.
- Fase de implantación de un producto: para aprender, medir y mejorar rutas y métodos.
- Fase de venta de un producto: para presentación funcional, disponible en forma de aplicación informática, tanto hacia técnicos como compradores.

1.4. Tipos de Simuladores

La constante evolución y desarrollo de las tecnologías de la información ha permitido que se haya empezado a crear programas de simulación con fines de entretenimiento. Sin embargo, se ha profundizado el uso de estos en áreas tales como la psicología, la política, fenómenos sociales y de negocios entre otros, con el objetivo de lograr una mejor preparación de sus trabajadores. Para visualizar mejor el desarrollo alcanzado se describen a continuación algunos tipos de simuladores:

➤ **Simuladores de tránsito:**

Los simuladores de tránsito son programas en tiempo real, que permiten prever las reacciones de los conductores frente a accidentes y definir las distintas alternativas a tener en cuenta para la evacuación. Para su desarrollo se obtiene información estadística del tránsito actual y del pasado. (1)

➤ **Simuladores de cálculo:**

Mediante la definición de parámetros de funcionamiento, los simuladores de cálculo simulan de forma automática cualquier periodo real de trabajo de un sistema en períodos de tiempo reducidos, donde han tenido lugar los mismos eventos que en el sistema real, con una medición y evaluación de los recursos empleados. Se generan informes automáticos en tiempo real y logs de incidencias. (1)

➤ **Simuladores de producto:**

Son muy útiles para asegurar funcionalidad y usabilidad. El usuario actúa sobre los elementos gráficos de la aplicación que simulan particularidades del sistema real. El cursor cambia de aspecto cuando pasa sobre elementos accionables. En la fase de venta, permiten presentar el producto mediante una aplicación informática (1).

➤ **Simuladores contra-producto:**

Son muy ventajosos para asegurar una correcta funcionalidad del producto en fase de diseño, y sirven de ayuda durante las pruebas unitarias del producto. El simulador actúa, frente al producto a verificar, figurando el entorno real en el que se va a encontrar, creando incluso situaciones de “máxima carga” que garanticen su correcta respuesta en casos muy desfavorables. (1)

➤ **Simuladores de sistemas:**

Permiten analizar el comportamiento corporativo de varios productos que interactúan entre sí. Engloban el concepto de simulador de producto e incorporan la posibilidad de detección de bloqueos mutuos, esperas infinitas y otros comportamientos no deseados. (1)

1.5. Ventajas y Desventajas de los Simuladores

El funcionamiento de los simuladores está basado en un tipo de aprendizaje de forma experimental. Es por ello que el actor se convierte prácticamente en un alumno, que obtiene conocimiento por medio de la interacción con un micro mundo virtual, logrando de esta forma simular una situación del mundo real.

Ventajas:

- Permite al alumno manipular un número reducido de objetos dentro del proceso simulador.
- Hace necesario que el alumno es esfuerce para identificar diferentes situaciones que se le presenten durante la simulación.
- El problema llega a ser un reto, por lo que el alumno logra la fase de motivación.

Desventajas:

- En ciertas ocasiones los casos referentes a los simuladores no se pueden medir debido a su alta complejidad. Es imprescindible que se establezca una relación

En muchos de los Centros Educativos no se cuenta con estas nuevas tecnologías por la falta de recursos económicos y materiales para poder ser aplicados en todas las aulas.

Foto detectores.

Estos elementos de captación están distribuidos en arco en el exterior del túnel, ensamblados en tarjetas de circuito impreso que agrupan a una buena cantidad de ellos y constituyen el elemento primario para la captación de los Rx que se obtienen después de atravesar el túnel y su contenido.

Para los sistemas en blanco y negro se emplea una sola capa de foto detectores, pero para los sistemas a color se utilizan dos capas, quedando superpuestos dos foto detectores, se mide entonces la diferencia de penetraciones, a estos sistemas se les llama multielemento o multienergía. Esta segunda capa de foto detectores se monta delante y se cubre su parte trasera con un filtro de cobre, es más sensible en el rango de baja energía espectral de la radiación X.

El filtro de cobre evita el paso de esa energía baja hacia la otra tarjeta detectora (tarjeta de alta).

Los elementos de comunicación.

Permiten visualizar las imágenes (monitores), ejercer las funciones (teclado) y señalar operaciones o alertas. En general se tienen uno o dos monitores para mostrar imágenes en blanco y negro y a color, los teclados tienen indicadores de Rx activado, dirección de movimiento de la estera, elementos de encendido, parada de emergencia y teclas de funciones de realce.

1.6.2. Principios básicos de operación de los sistemas.

El principio de operación de los sistemas es la detección por transmisión. El generador de rayos X emite un haz que pasa por el túnel de inspección, donde atraviesa los bultos presentes y va a chocar con el arreglo de foto detectores.

En este camino los rayos X son más o menos absorbidos por los objetos que se encuentran en el túnel del equipo, otra porción del haz llega directo a los foto detectores y una parte al chocar con los equipajes se dispersa.

Algo de esta dispersión pudiera escapar al exterior del túnel por lo que se toman medidas de protección ubicando las cortinas plomadas en ambos lados del túnel de inspección.

Al encender el equipo se produce una generación de rayos X que permite realizar una normalización de los valores extremos de luz en el túnel, con este vacío.

Para que se generen rayos X solo cuando se inspeccionan bultos, a la entrada del túnel hay una barrera de luz infrarroja (4) y otra muy cerca de la zona por donde barre el haz, la generación ocurre cuando el bulto pasa el segundo haz y hasta que este quede nuevamente libre, la inspección del bulto se hace línea a línea, sincronizada con el movimiento del objeto. Cuando el haz es cortado por el bulto, este absorbe radiación de acuerdo al espesor y naturaleza de los objetos, por lo que a los foto detectores llega el residuo.

El resultado de la detección es una señal de voltaje (4) que se convierte en palabra digital y es guardado en memorias para completar las imágenes que son tratadas electrónicamente y finalmente mostradas en un monitor, este proceso se repite para cada bulto, y en el monitor aparecen nuevas imágenes las que desplazan a las anteriores, por tanto el tiempo de inspección y análisis por parte del operador es limitado y depende de la separación que exista entre un bulto y otro, es por ello necesario la separación de 30 a 50 cm entre cada uno, además si se necesita de más tiempo para hacer una correcta valoración de la imagen hay que detener la estera, o sea, pulsar STOP.

1.6.3. Coloración asignada por los fabricantes de los equipos según la composición química de la materia.

Durante el proceso de revisión de equipaje, una de las características que se manifiesta de los equipos de RX, es la coloración de los objetos que se están analizando, detalle muy importante en cuanto a la detección de elementos ilegales que se estén transportando.

➤ La distribución de colores se realiza de la siguiente forma:

- El color **rojo** se emplea en funciones especiales.
- Los elementos que no pueden ser penetrados aparecen en **negro**.
- El color *blanco* se corresponde con la no absorción. (4)
- **Naranja**: Número atómico menor que 11. Elementos más ligeros, hidrógeno, carbono, nitrógeno, oxígeno y sus compuestos moleculares; explosivos como la nitroglicerina, plásticos como el acrílico, papel, textiles, alimentos, madera, agua y narcóticos.

Capítulo 1: Fundamentación Teórica

Materiales orgánicos.

- **Verde:** Número atómico entre 11 y 18. Elementos semipesados o mixtos, como el aluminio, sodio, cloro y cloruro sódico.
- **Azul:** Número atómico mayor que 18. Elementos pesados como los metales, titanio, cromo, hierro, níquel, cobre, zinc, estaño, plomo, oro, plata.

Materiales inorgánicos.

Las (4) sustancias rara vez aparecen puras en la naturaleza y usualmente lo que se tiene es una mezcla de materiales. Al inspeccionar los equipajes además de la mezcla de materiales ocurre la superposición de objetos, de ahí que se prefiera hablar de números atómicos efectivos (Zeff), los que son proporcionales a la densidad física media de los materiales.

Debido a esto aparece el concepto de **número atómico ponderado**, mediante el cual se define que color se asigna a cada compuesto.

Un ejemplo muy simple es con el agua:

$$H_2O = [2(1) + 1(8)] / 3 = 3$$

Donde:

2 - Cantidad de átomos de hidrógeno.

1 - Número atómico del hidrógeno.

1 - Cantidad de átomos de oxígeno.

8 - Número atómico del oxígeno.

3 - Cantidad total de elementos químicos.

3 - Número atómico ponderado.

Entonces el agua se representa en color **naranja**.

Cuando existe superposición, predomina el color de aquel componente que proporciona mayor absorción; por eso las superposiciones aparecen en cualquiera de los tres colores dependiendo del grosor de cada capa.

1.6.4. Funciones de realces de los equipos de RX

Las funciones de realce son un grupo de posibilidades que el fabricante pone a disposición de los operadores para poder hacer un estudio más real y objetivo de la imagen que se están analizando.

Los sistemas instalados brindan a los operadores un conjunto de funciones de realce que facilitan la inspección y evaluación de las imágenes radioscópicas. También le brindan al operador funciones especiales (6) que le ayudan a tomar decisiones más precisas al inspeccionar y evaluar una imagen.

A continuación se detallan algunas funciones de realces:

Blanco y Negro: Tipo de filtro que se le aplica a las imágenes en los equipos de rayos x, su función es simplemente que mientras más claro se muestre el objeto, menos absorción de rayos por su baja densidad

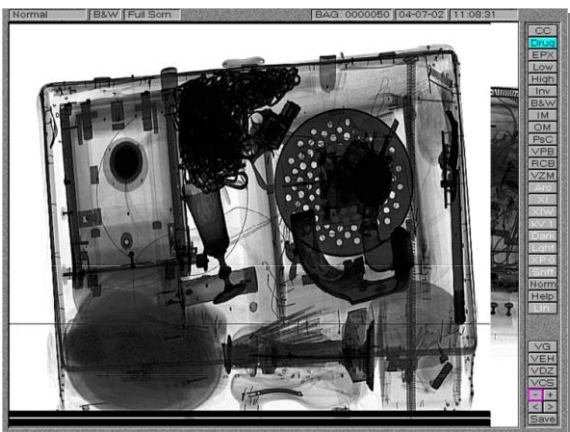


Figura 2: Blanco y Negro

Video Inverso: Tipo de filtro que se le aplica a las imágenes en los equipos de rayos x, su función es invertir el brillo de la imagen, el negro se convierte en blanco, el naranja brillante en opaco, el azul brillante en azul opaco. Función muy útil para detectar pequeños hilos metálicos

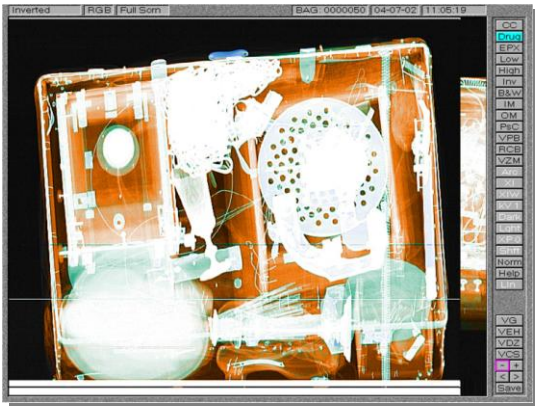


Figura 3: *Video Inverso*

Inorgánicos: Tipo de filtro que se le aplica a las imágenes en los equipos de rayos x, su función es mostrar solo los materiales inorgánicos y los mixtos más densos, colorea los objetos en azul y verde. Se eliminan los tonos naranjas

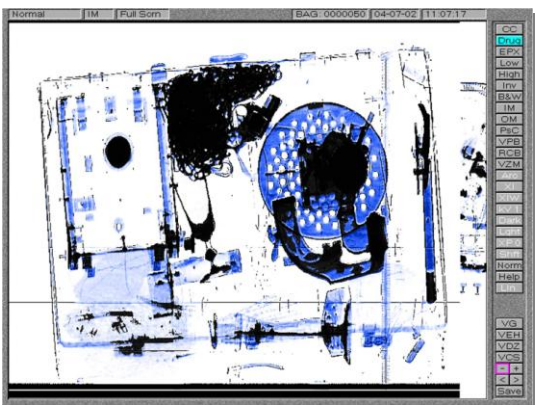


Figura 4: *Inorgánicos*

Orgánicos: Tipo de filtro que se le emplea a las imágenes en los equipos de rayos x, su función es que atenúa los materiales inorgánicos y los mixtos más densos. Elimina los materiales inorgánicos

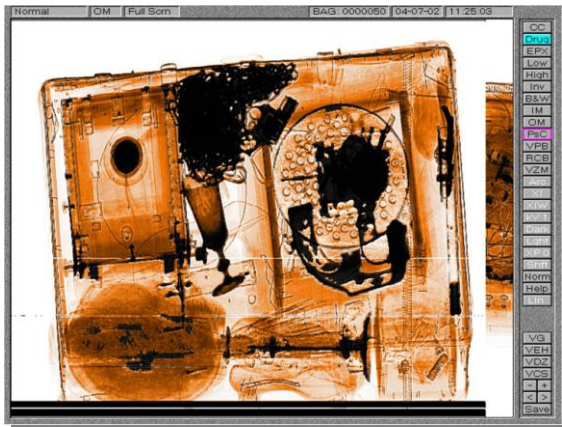


Figura 5: Solo Orgánicos

Todas las funciones de realce proporcionan una información mejorada de la imagen de rayos X. La información primaria brinda:

1. La Forma (Contornos y Tamaños)
2. La Densidad (Claridad y Oscuridad)
3. El Tipo de Sustancia o Material

1.6.5. Equipos de RX empleados en Cuba

Los sistemas instalados en Cuba son de fabricación inglesa, alemana y china denominándose de la siguiente forma:

➤ **RAPISCAN:**

Tecnología inglesa, fabricados por "Corrigan Instrumentation", comenzaron a instalarse en 1995. El arco foto detector es plegado y emplea un sistema multienergía; presenta imagen a color y blanco y negro; el procesamiento de las imágenes es computarizado y son del tipo:

- Mark I y Mark III - Serie 100 (Trent 1)
- Mark I, Mark III y Mark VIII - Serie 300 (Trent 2)
- Mark I, Mark III y Mark VIII - Serie 500 (EPX)
- Mark I, equipo utilizado en chequeo de equipaje no facturado.

- Mark III y Mark VIII, equipos utilizados en el chequeo de equipaje facturado.

- **HEIMANN (HiMat):**

Tecnología alemana, presenta imagen a color y blanco y negro, foto detectores plegados y emplea el sistema multienergía. Los más antiguos utilizan Z80 como procesador de imagen y sistema modular de tarjetas. Los más modernos emplean computadoras para el procesamiento de imágenes. Son del tipo Mark I y Mark III.



Figura 6: *Equipo de RX Heimann*

1.7. Representación de imágenes

La imagen que se obtiene como resultado de la irradiación se representa con los colores anteriormente explicados, en los que se diferencian los grados de luminosidad, tonos y composición química, desde el blanco hasta el negro, según sea la absorción de los objetos irradiados. La imagen además brinda la forma y densidad de los objetos observados. (4)



Figura 7: Representación de paquetes y su densidad a través de un monitor de equipo de RX

1.8 Recomendaciones para el trabajo con un equipo de Rayos X

- La valoración de las imágenes no se debe hacer bajo presión de tiempo. La obtención de los estándares de seguridad depende principalmente de su trabajo. Detenga la cinta si no ha tenido tiempo suficiente para valorar correctamente la imagen.
- Considere sospechosos aquellos objetos que no son identificables y realice un control posterior.
- Será necesario realizar siempre un control posterior de aquellos objetos que aparecen en negro y que no pueden ser aclarados con ninguna de las funciones de procesamiento de imágenes, los objetos peligrosos pueden quedar ocultos. (4)
- No hay que realizar evaluaciones totales de la imagen, basándose únicamente en pequeños detalles reconocibles.
- Tenga en cuenta siempre que su equipo de RX no reconoce objetos automáticamente, ponga su máxima atención.
- Para identificar objetos desconocidos, puede ser necesario irradiar de nuevo el bulto desde un ángulo distinto, para esto el bulto se gira y se transporta en dirección contraria hasta volver a la entrada del túnel.

En ocasiones se hace muy necesario para lograr una interpretación correcta de las imágenes, auxiliarse de las posibilidades que brindan los equipos de RX, estas posibilidades son conocidas como funciones de realce y se encuentran en el panel de control directamente en las teclas duras (equipos Heimann) o en el panel de control y en la zona de la derecha de la pantalla del monitor conocidas como teclas blandas en los (equipos de la Corrigan).

1.9 Algunos simuladores característicos

En el mundo existen un sin número de programas o simuladores con características similares a las de un equipo de RX. Por solo mencionar algún ejemplo, a continuación se describe brevemente un simulador muy utilizado y eficaz, por su alto nivel de enseñanza y aceptación, no solo en el mundo de la informática sino también por su gran aporte a los servicios aduaneros. (5)

SIMFOX

El simulador del entrenamiento de la radiografía simula exactamente las máquinas de radiografía usando las firmas tomadas del ordenador central. Las imágenes se exhiben en su escala original y conservan su equilibrio original del color.

SIMFOX es el único sistema de entrenamiento de la radiografía disponible que permite el uso del teclado real de la máquina de radiografía del anfitrión. Si procede, **SIMFOX** se puede instalar para utilizar el teclado real de la máquina.

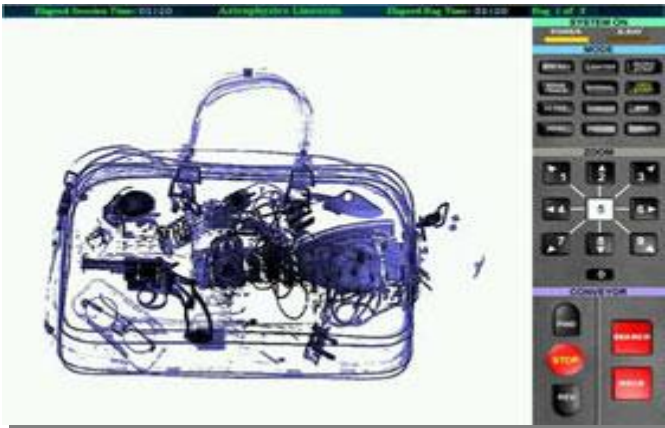


Figura 8: *Simulador de RX SIMFOX*

Funcionamiento de SIMFOX

Cuenta con un técnico o administrador del simulador, que es el que se encarga de crear distintas sesiones, donde cada sesión está representada por diversas imágenes. Estas imágenes pueden estar conformadas por un bolso y varios objetos. Las mismas serán presentadas al usuario (en este caso el estudiante) en orden aleatorio, luego se le irá realizando al estudiante una evaluación dada su reacción a diferentes imágenes, donde al final emitirá un resultado que revelará si tuvo un correcto desenvolvimiento en la sesión donde se trabajó. (5)

1.8. Conclusiones

Tras la investigación de los conceptos elementales de los simuladores, como de los distintos tipos de simuladores existentes hasta la actualidad, se puede llegar a la conclusión de que la utilización y perfeccionamientos de estos simuladores, ha ido prosperando y en pleno crecimiento, de modo que la utilización de nuevas técnicas y nuevas herramientas de desarrollo, crearán mejoramientos visuales y apariciones de nuevos métodos.

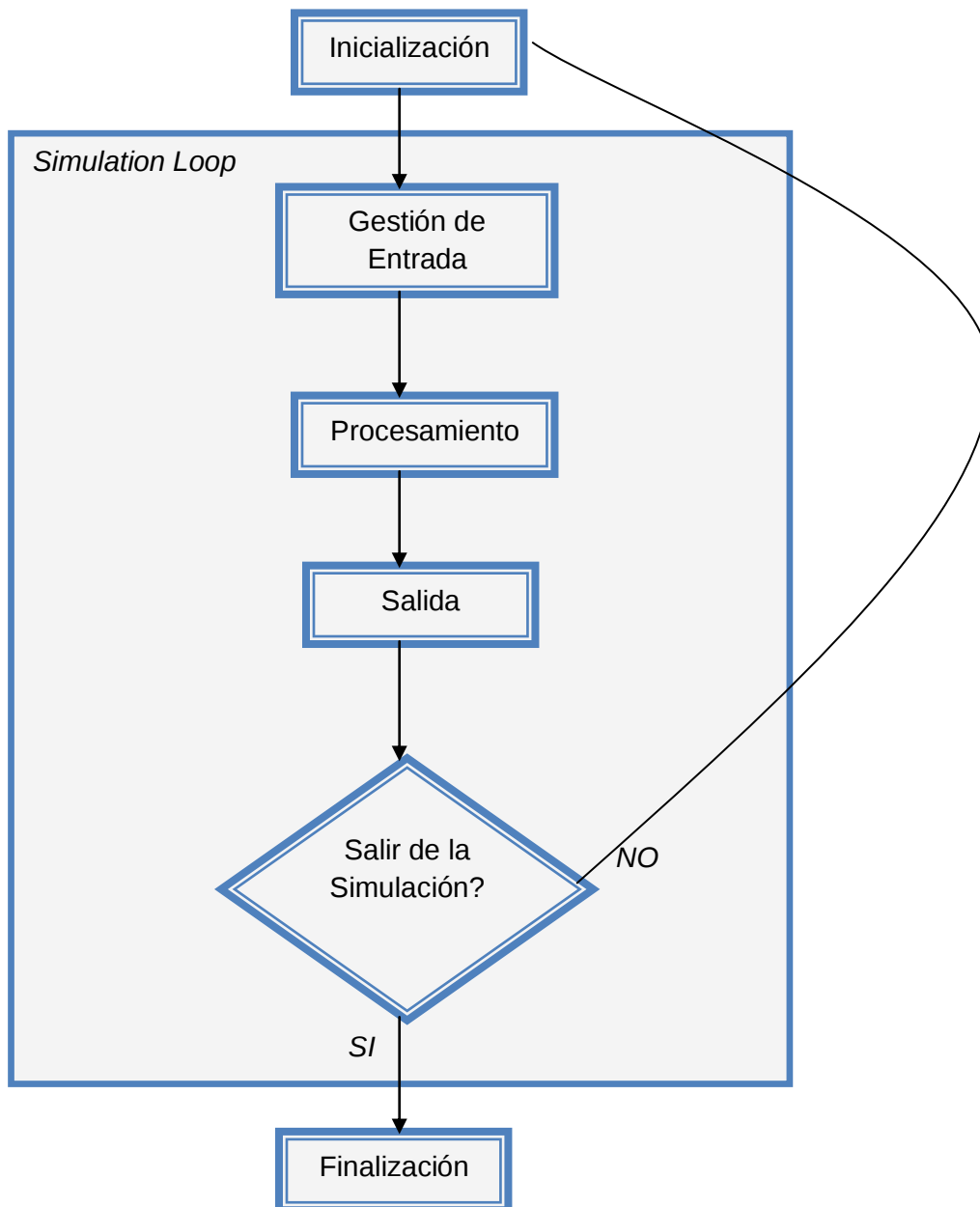
Capítulo 2: Soluciones técnicas

Introducción

Este capítulo describe las herramientas y bibliotecas utilizadas para la confección del software propuesto en el presente proyecto de tesis. Identificando características que permitan que el mismo constituya un verdadero simulador para el entrenamiento del personal de seguridad de la Aduana de la República de Cuba.

2.1. Especificación del simulador

Existen distintas formas para realizar esta tarea. Se dedicará un tiempo a plantear lo que se quiere hacer en el simulador. Se va a seguir un esquema que servirá para especificar los requisitos. Este simulador tendrá entre sus características primordiales, lograr que el usuario, que interactúe con la aplicación se encuentre motivado. Es por esta vía en donde se encuentra el objetivo, así se logra que el producto sea más atractivo y llamativo.



Lógica del simulador

La interfaz entre la aplicación y el hardware lo proporciona la biblioteca SDL. Esta brinda la interfaz con el teclado, el ratón, el hardware gráfico y los dispositivos de sonido. Para interactuar con el sistema operativo se utilizarán funciones propias del lenguaje C++

Todas las ventanas de la aplicación podrán ser mostradas a pantalla en modo normal, donde van a existir dos tipos de ventanas con las que el usuario puede interactuar con la aplicación (7)

- Ventana del Menú
- Ventana de Simulación



Figura 9: Imagen de la Interfaz del Menú

A continuación se especificarán características utilizadas en la propuesta de solución.

Primeramente se mostrará un menú en donde el sistema le dará la posibilidad al usuario de elegir la opción que desee, ya sea empezar la partida o salir del simulador. Este menú debe de tener un ítem o efecto que nos permita conocer sobre que opción el usuario está situado. Una vez que el usuario oprima el botón de empezar partida, el sistema carga una ventana que sería en donde se cumplirán los principales requisitos del simulador, Primeramente, se quiere lograr de que una vez que el usuario este en la simulación real de un equipo de rayos x, se vayan mostrando imágenes y que el usuario determine si son sospechosas o no. También a estas imágenes se le aplicaría distintas funciones de realce o filtros, que esto

permite dar una panorámica diferente a las imágenes para una mejor detección de objetos sospechosos, esto se logra cargando imágenes correspondiente a la original, pero ya concebida con la función de realce aplicada. Unas de sus funcionalidades aplicadas también, es la de aplicarle zoom a las imágenes, así el que este interactuando con la aplicación, podrá detallar aún más las ilustraciones, ubicándose así, en una región específica de la imagen. También permitirá marcar aquellas imágenes que el usuario defina como sospechosas, una vez concluida la transición de imágenes el usuario, en este caso el inspector, tiene la posibilidad de mostrar si su selección fue correcta o no, mostrándose en pantalla una ventana con las imágenes que fueron seleccionadas por el inspector y las que son realmente sospechosas, emitiendo así una evaluación final.

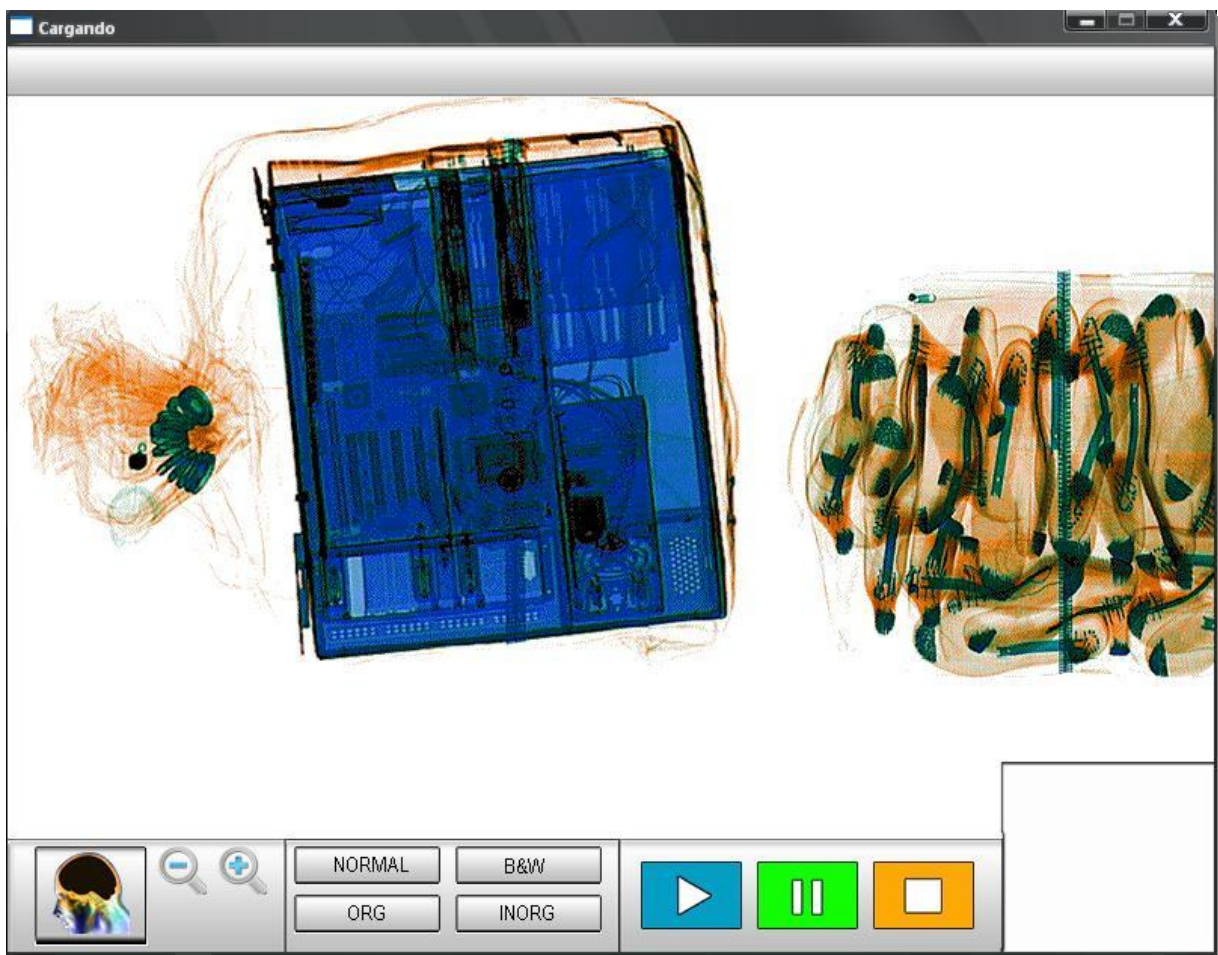


Figura 10: Imagen de la Interfaz del Simulador de Equipos de Rayos X

Para argumentar de manera general y tratar sobre el lenguaje y bibliotecas que se usan, se pondrá a su consideración una serie de parámetros convincentes de manera más explicativa.

2.2. Utilización de la Simple DirectMedia Layer (SDL)

En Cuba se está generalizando un método más económico y seguro, que es el de emigrar completamente a software libre tratando de desviar la atención del software propietario. Por tales razones, La Aduana de la República de Cuba como pionera en esta migración no se queda atrás, por lo que se acordó preferiblemente trabajar con SDL. Es una biblioteca multimedia multiplataforma, es decir, todas las aplicaciones que se desarrollan con esta biblioteca pueden ser compiladas, sin cambiar nada en el código, en varios sistemas diferentes como Windows, Linux, etc. Está diseñada para proporcionar acceso de bajo nivel al audio, teclado, ratón y a los dispositivos de video 2D. Es muy utilizada en diferentes tipos de aplicaciones entre ellas se encuentra un gran número juegos y adaptaciones de juegos entre plataformas.

El principal motivo para elegir SDL es que es la mejor API para desarrollo de videojuegos disponible para Linux, esto posibilita que el simulador vaya a tratarse a modo de un videojuego por lo que va a ser de gran utilidad tocar algunas de las características de los videojuegos como su lógica y arquitectura.

Uno de los aspectos importantes por los que se ha dirigido las miradas a esta biblioteca es su licencia. Permite desarrollar aplicaciones sin tener que estar pendientes de los aspectos legales que envuelven a la biblioteca.

Que la licencia sea un tema importante no quiere decir que no sea una herramienta potente. En los últimos años ha sido premiada varias veces como la mejor biblioteca libre para el desarrollo de videojuegos lo que pone de manifiesto la importancia de esta biblioteca en el mundo libre.

Resumiendo, utilizando esta biblioteca se consigue no atar el desarrollo de la aplicación a ningún sistema. Permite coger un programa que se ha desarrollado para Linux y compilarlo para Mac, Palms y viceversa.

2.2.1. Ventajas del uso de SDL

Dos de las grandes alternativas a SDL para realizar tareas parecidas son DirectX y OpenGL, estas no son soluciones definitivas, ya que presentan diferentes problemas.

OpenGL está especializada en gráficos 3D y no proporciona acceso a otros recursos del sistema, como puede ser el teclado o el joystick, fundamentales en la programación de videojuegos. DirectX es bastante completo, ya que proporciona gráficos en 2D y 3D, entrada y salida, comunicación entre redes; el problema radica que sólo es compatible con sistemas Windows además de que su uso en la programación de videojuegos es algo engorrosa si nunca se ha trabajado con las APIs de Microsoft Windows(c).

Las ventajas de SDL pueden agruparse en: (8)

- **Estabilidad:** Una de las condiciones de la licencia de SDL es que no puede ser utilizado por empresas o particulares sin dar soporte al desarrollo de la API. Con esto se consigue la corrección de errores y la aportación de mejoras que favorecen la robustez de la API. El desarrollo de SDL es incremental de forma que se congela una versión estable mientras que se crea una versión nueva con todas las características aportadas con el fin de ser testeadas y aprobadas para una versión posterior que será considerada estable.
- **Simplicidad** SDL ha sido diseñada para tener un API simple, que permita que haya la menor cantidad de código entre lo que quieres hacer y el resultado final.
- **Flexibilidad** La biblioteca es multiplataforma. Las aplicaciones que se creen van a correr en Win32, BeOS y Linux sin tener que cambiar ni una sólo línea de código. Otro aspecto de la flexibilidad es que aparte de que el código es totalmente multiplataforma puedes consultar cualquier aspecto de la implementación por si te puede aportar alguna idea o quieres mejorar algo. Nada está bloqueado, puedes ver cómo está implementada cada una de sus partes.

Resumiendo, libSDL es portable, intuitiva y estable como una roca (así la define su creador). Se puede confiar en ella y las restricciones son mínimas. Comparada con sus rivales del mundo de la programación de videojuegos es mucho más asequible en todos los sentidos para adentrarnos en el mundo de la programación de videojuegos.

2.3. Herramientas y Tecnologías a utilizar

Después de haber realizado un estudio de las tendencias actuales y de las principales características de las tecnologías más usadas en la actualidad, se propone utilizar para el desarrollo de la propuesta de solución Visual Paradigm como herramienta libre para el aprendizaje y el trabajo con UML. También se trabajará con la ya nombrada libSDL, como herramienta de desarrollo se utilizará Visual Studio 2008 Team System de Windows y CodeBlock de Linux y como lenguaje de programación C++.

2.4. Conclusiones

En este capítulo se abordó las soluciones técnicas que se irán desarrollando en el transcurso de los siguientes capítulos para darle solución a nuestro problema. Por lo que se explotará la propuesta utilizando la biblioteca SDL para facilitar el manejo de las imágenes en la simulación del equipo de Rayos X.

Capítulo 3: Características del Sistema.

Introducción

En este capítulo se realiza una descripción del objeto de estudio. Se utiliza el modelo de dominio para describir los procesos del negocio, se definen las funcionalidades del software a través de los requisitos funcionales y no funcionales; los actores y el diagrama de casos de usos del sistema, finalmente se realiza la descripción de los casos de usos.

3.1. Descripción del modelo de dominio

Debido a la poca estructuración de los procesos del negocio que tienen que ver con el objeto de estudio y para que se pueda entender el contexto en que se emplaza el sistema, se describe el funcionamiento de la aplicación mediante una serie de conceptos, entidades y sus relaciones, agrupándolos en un modelo de dominio.

3.2. Modelo del Dominio.

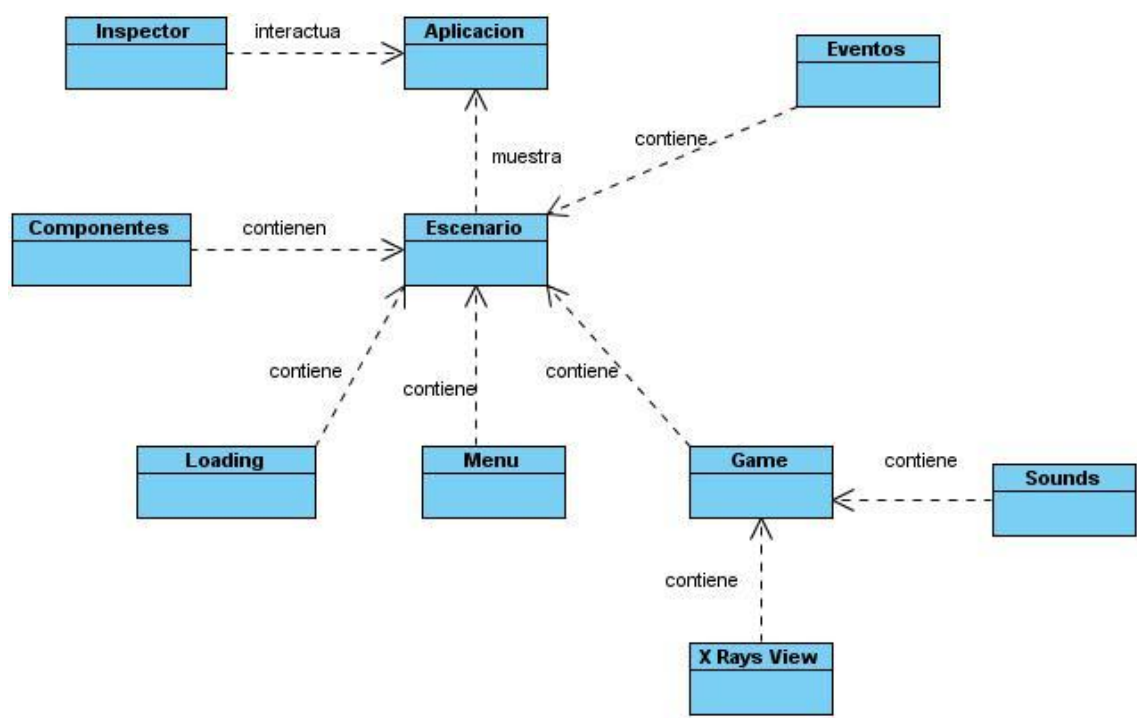


Figura 11: Diagrama correspondiente al modelo del dominio.

3.2.1. Glosario de términos del Dominio.

- **Inspector:** Este sería el usuario que está interactuando con la aplicación
- **Aplicación:** Es la clase principal que interrelaciona todos los aspectos del juego.
- **Escenario:** Esta clase hace de interfaz entre las diferentes escenas de la aplicación para la navegación de ellas que no son otras que Menú, Juego y Configuración.
- **Eventos:** Esta clase nos permite gestionar la entrada tanto por teclado como por mouse
- **Componentes:** En esta clase se almacenan todos los componentes que vamos a utilizar en el sistema
- **Menú:** Esta clase controla el menú principal de la aplicación y sus distintas opciones.
- **Game:** Esta clase controla los aspectos relevantes de la aplicación en tiempo de juego.
- **X_Rays_View:** Es un componente que controla la transición de imágenes, se encarga de cargar y pintar las imágenes en la pantalla
- **Sounds:** Esta clase nos permite gestionar el sonido que entre en el estado que se desee reproducir algún sonido o música.

3.3 Actores del sistema a automatizar.

Un actor es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso. Por lo general estimula el sistema con eventos de entradas o recibe algo de él. O sea, es un rol de un usuario, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema.

Capítulo 3: Características del Sistema

Tabla I. Descripción de los actores del sistema a automatizar

Nombre del actor	Descripción
Inspector	Representa al usuario que interactúa con el sistema con el objetivo de entrenarse.
Técnico	Representa al administrador del sistema, el cual se le está permitido ajustar la configuración del sistema, para el inspector.

3.4 . Requisitos Funcionales.

En este apartado se deben de responder a la pregunta sobre, qué debe de hacer la aplicación o sistema.

1. Inicializar Menú (UI).
 - a. Cargar sonidos a utilizar en el menú
 - b. Cargar texturas a utilizar en el menú.
2. Inicializar sistema de captura de eventos de mouse
3. Comenzar partida.
 - a. Cargar sonido de la simulación
 - b. Inicializar interface del simulador
 - c. Cargar las imágenes a utilizar
 - d. Preparar el estado inicial del sistema de transición de imágenes.
 - e. Inicializar variables de uso en el sistema
 - i. Lista de imágenes sospechosas.
4. Finalizar el sistema.
 - a. Eliminar imágenes y texturas cargadas en memoria.
 - b. Eliminar sonidos cargados en memoria
 - c. Destruir variables y listas creadas.
5. Configurar imágenes por niveles de dificultad.
6. Crear perfil.
7. Alterar imágenes.
8. Cargar imágenes.
9. Controlar cambio de fase.
10. Ver ayuda del funcionamiento del simulador
11. Configurar opciones
12. Iniciar transición de imágenes.
13. Aplicar filtros a las imágenes.
14. Marcar imagen como sospechosa

15. Mostrar lista de sospechosa
16. Aplicar zoom
17. Emitir una evaluación
18. Detener transición de imágenes.
19. Salir de la Simulación.

3.5 Diagrama de Casos de Uso del Sistema.

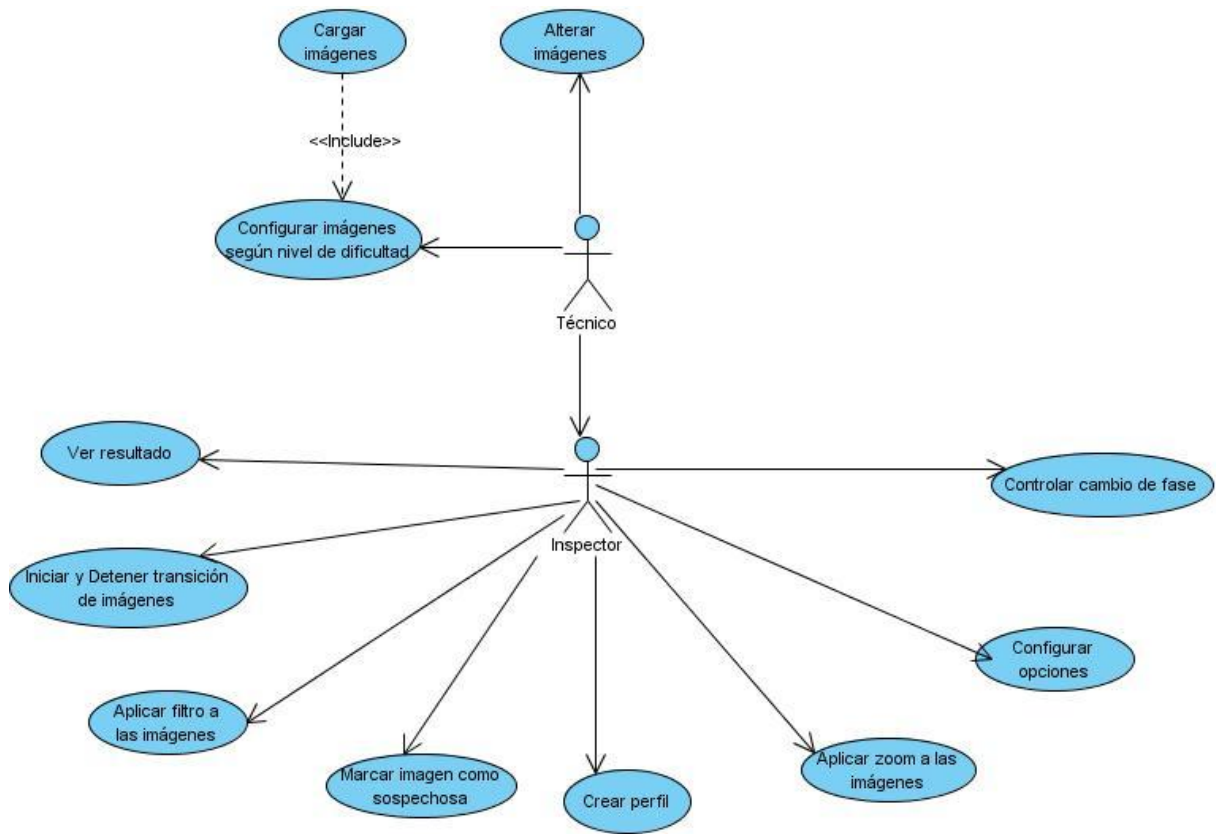


Figura 12: Diagrama de casos de uso del sistema

3.6 . Requisitos No Funcionales.

➤ Software

- Se debe disponer de sistemas operativos tanto como Ubuntu, Debian como hasta Windows XP para la instalación de la aplicación

➤ **Hardware**

Se requieren máquinas con los siguientes requisitos:

- Procesador Pentium 3 o superior.
- 128 Mb de RAM.
- 200 Mb de capacidad del disco duro.

➤ **Seguridad**

- Disponibilidad: Evitar que el programa deje funcionar, que el usuario del ordenador lo interrumpa o lo detenga.

➤ **Requisitos de Rendimiento**

Tenemos que indicar los requisitos relativos al rendimiento de la aplicación tal como tiempos de respuesta y otros aspectos. La aplicación se puede considerar de tiempo real blando, ya que se establece períodos que deben de cumplirse para una correcta funcionalidad de la aplicación.

➤ **Restricciones de diseño**

Ahora se especificarán aquellas restricciones que se hayan identificado e influyan en el diseño del sistema.

En el diseño de la aplicación tiene que primar los tiempos de respuesta sobre el consumo de recursos de espacio como la memoria principal o secundaria. Esta es la principal restricción que tendrá el diseño del simulador. Las demás características tendrán que ser diseñadas basándose en este principio.

El grado de cohesión y acoplamiento de las clases puede verse afectado por esta restricción. Es necesario evaluar estos parámetros con el objetivo anteriormente expuesto.

➤ Atributos del sistema software

En este apartado se deben de especificar todos los atributos con los que debe de cumplir la aplicación.

Uno de los requisitos principales de la aplicación es que sea portable entre los sistemas compatibles con SDL. No se deberá utilizar código dependiente del sistema operativo donde se desarrolle la aplicación. Debe de ser un código mantenible y ampliable que se pueda mejorar en futuras versiones del mismo perfeccionando todos los aspectos que sean necesarios o que se quieran modificar.

La aplicación debe ser robusta y fiable desde su diseño. La seguridad no es un tema relevante ya que no se va a requerir ningún tipo de dato que sea conflictivo o susceptible de ser protegido.

3.7 Descripciones de los casos de uso de sistema.

Tabla II: CU “Controlar cambio de fase”

Caso de Uso:	Controlar cambio de fase
Actores:	Inspector, Técnico
Resumen:	El caso de uso inicia una vez que el actor ha cambiado de escena
Precondiciones	El Técnico tiene que haber configurado las imágenes
Casos de usos asociados	“Configurar opciones” “ Ver ayuda del funcionamiento del simulador” “ Cerrar aplicación”
Flujo Normal de Eventos	

Capítulo 3: Características del Sistema

Acción del Actor	Respuesta del Negocio
1-El actor presiona el botón ayuda 2-El actor presiona el botón comenzar. 3- El actor presiona el botón de salir	1.1- El sistema muestra una escena con la descripción del funcionamiento del software 2.1- El sistema muestra la escena del entrenamiento. 3.1- El sistema cierra la ventana correspondiente al menú, y sale definitivamente de la aplicación
Flujo Alternativo de Eventos A	
1-El actor presiona el botón atrás	1.1- El sistema muestra la ventana del menú
Flujo Alternativo de Eventos B	
2-El actor puede interactuar con las opciones.	2.1 El sistema ejecuta la opción seleccionada.

Tabla III: CU “Iniciar transición de imágenes”

Caso de Uso:	Iniciar y Detener transición de imágenes
Actores:	Inspector, Técnico
Resumen:	El caso de uso comienza cuando el actor presiona el botón <i>play</i> para que inicie la transición de imágenes y cuando selecciona el botón <i>pause</i> , termina cuando se detienen la transición de imágenes

Capítulo 3: Características del Sistema

Casos de usos asociados	-----
Precondiciones	Se tiene que haber iniciado el cambio de fase
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1-El actor presiona el botón de <i>Play</i> . 2-El actor presiona el botón <i>Pause</i>	1.1-El sistema muestra en pantalla una ventana con las imágenes que serán identificadas por el usuario. 2.1 El sistema detiene la transición de imágenes.

Tabla IV: CU “Aplicar filtro a las imágenes”

Caso de Uso:	Aplicar filtro a las imágenes
Actores:	Inspector, Técnico
Resumen:	El caso de uso comienza cuando el actor presiona el botón de cambiar filtro a la imagen, y termina cuando el sistema ejecuta la acción
Casos de usos asociados	
Precondiciones	Se tiene que haber iniciado el cambio de fase y se tiene que haber iniciado la transición de imágenes
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1- el actor presiona el botón de aplicar filtro	1.1- El sistema muestra la imagen con el filtro seleccionado

Capítulo 3: Características del Sistema

Tabla V: CU “Seleccionar imágenes”

Caso de Uso:	Seleccionar imágenes	
Actores:	Inspector, Técnico	
Resumen:	El caso de uso comienza cuando el actor observa la transición de imágenes y selecciona una imagen con algún objeto sospechoso y culmina cuando la imagen seleccionada se guarda en una lista de seleccionadas.	
Precondiciones	Se tiene que haber iniciado el cambio de fase y que se haya iniciado o detenido la transición de imágenes.	
Casos de usos asociados		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1- El actor selecciona la imagen que el crea que es sospechosa	1.1-	El sistema captura el evento de la imagen
	1.2-	Guarda las imágenes en una lista de imágenes seleccionadas.

Tabla VI: CU “Ver resultado”

Caso de Uso:	Ver resultado
Actores:	Inspector, Técnico
Resumen:	El caso de uso comienza cuando el actor selecciona el botón correspondiente a imágenes sospechosas y culmina cuando muestra una ventana con las imágenes seleccionadas, las realmente

Capítulo 3: Características del Sistema

	sospechosas y el resultado.
Precondiciones	Se tiene que haber iniciado el cambio de fase y que se haya seleccionado al menos una imagen.
Casos de usos asociados	CU “Seleccionar Imágenes”
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1- El actor presiona el botón de imágenes sospechosa.	1.1-El sistema muestra un cuadro de diálogo con la lista de imágenes seleccionadas por el usuario.
2-El actor verificará si se equivocó en seleccionar las imágenes	2.1-Las Imágenes aparecerán estáticas en la ventana. 2.2-Se mostrarán además las imágenes que realmente contengan objetos ilícitos. 2.3-Emitirá una evaluación final.

Tabla VII: CU “Crear perfil”

Caso de Uso:	Crear perfil
Actores:	Inspector, Técnico
Resumen:	El caso de uso inicia una vez que el actor ejecuta la aplicación, le saldrá una ventana para que cree su perfil.
Precondiciones	Se tiene que haber ejecutado la aplicación.
Casos de usos	CU “Controlar cambio de fase”

Capítulo 3: Características del Sistema

asociados	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1-El actor inicia la aplicación	1.1- El sistema muestra la ventana de crear perfil
2-El actor escribe su nombre en el componente correspondiente	2.1- El sistema guarda los datos
3- El actor escribe su rango en el componente correspondiente.	3.1- El sistema guarda e interpreta el dato del componente.

Tabla VIII: CU “Aplicar zoom a las imágenes”

Caso de Uso:	Aplicar zoom a las imágenes
Actores:	Inspector, Técnico
Resumen:	El caso de uso se inicia una vez que el actor aplica zoom a las imágenes y termina con la ejecución del mismo
Precondiciones	Se tiene que haber iniciado la transición de imágenes
Casos de usos asociados	CU “Detener transición de imágenes”
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio

Capítulo 3: Características del Sistema

1-El actor presiona el botón de aplicar zoom.	1.1-El sistema escala las dimensiones de las imágenes. 1.2-El sistema muestra las imágenes escaladas.
---	--

Tabla IX: CU “Configurar opciones”

Caso de Uso:	Configurar opciones
Actores:	Inspector, Técnico
Resumen:	El caso de uso comienza cuando el actor presiona en el botón de configuración entre las opciones del menú y culmina cuando se muestra las opciones de complejidad
Precondiciones	Se tiene que haber iniciado el cambio de fase
Casos de usos asociados	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1-El actor aprieta el botón de configuración en la ventana del menú.	1.1-El sistema muestra una escena.
2- El actor escoge la opción de complejidad bajo	2.1- El sistema guarda la complejidad escogida.
3- El actor escoge la opción de complejidad	3.1- El sistema guarda la complejidad escogida.

Capítulo 3: Características del Sistema

medio. 4- El actor escoge la opción de complejidad alto.	4.1- El sistema guarda la complejidad escogida.
---	---

Tabla X: CU “Alterar Equipajes”

Caso de Uso:	Alterar Equipajes	
Actores:	Técnico	
Resumen:	El caso de uso comienza cuando el técnico presiona en el botón de configuraciones y termina una vez que concluye de hacerle los cambios a los equipajes	
Precondiciones	Se tiene que haber iniciado el cambio de fase	
Casos de usos asociados		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1-El técnico selecciona las imágenes o la imagen a alterar	1.1-El sistema muestra una serie de objetos, para adaptarlos a las imágenes.	
2-El técnico elije los objetos q desee para editar la imagen seleccionada	2.1-El sistema muestra las imágenes previamente editada.	

Capítulo 3: Características del Sistema

Tabla XI: CU “Cargar imágenes”

Caso de Uso:	Cargar imágenes	
Actores:	Técnico	
Resumen:	El caso de uso comienza cuando el técnico cargar las imágenes y culmina una vez que se terminan de cargar todas las imágenes.	
Precondiciones	Haberse iniciado la interfaz del menú.	
Casos de usos asociados		
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1- El técnico selecciona la opción de cargar imágenes.	1.1- El sistema carga las imágenes.	

Tabla XII: CU “Configurar imágenes según nivel de dificultad”

Caso de Uso:	Configurar imágenes según nivel de dificultad	
Actores:	Técnico	
Resumen:	El caso de uso comienza cuando a los niveles de complejidad le configuras las imágenes y termina con la ejecución del mismo.	
Precondiciones	Haberse iniciado la interfaz del menú y deben estar cargadas las imágenes.	

Capítulo 3: Características del Sistema

Casos de usos asociados	“Cargar Imágenes”	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1- El técnico selecciona la imagen y la agrega al nivel de complejidad correspondiente.	1.1- El sistema almacena las imágenes que se seleccionaron en cada uno de los niveles.	

Capítulo 4: Diseño del Sistema

Introducción.

A continuación se presentaran los diagramas de clase del diseño más importante en el Diseño, con las relaciones existentes tanto de composición como de herencia, además se muestran los diagramas de secuencia de los casos de usos que intervienen en el primer ciclo de desarrollo del software.

Modelo de Diseño.

En este epígrafe se representa el diagrama de clase de diseño y los diagramas de secuencia por cada caso de uso del sistema. De esta forma, se estará conformando la arquitectura del software a implementar

4.1. Diagramas de Clases de Diseño.

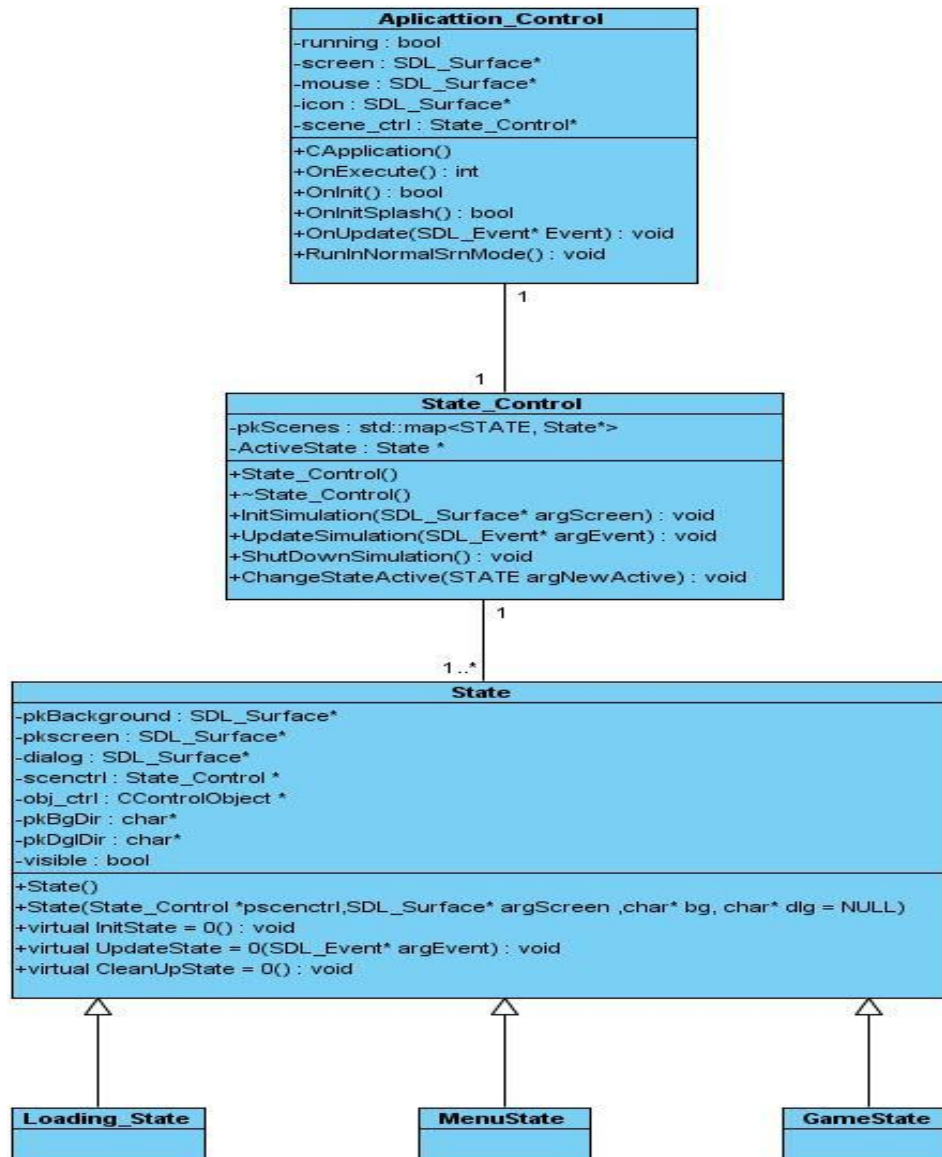


Figura 13: Paquete Controlador

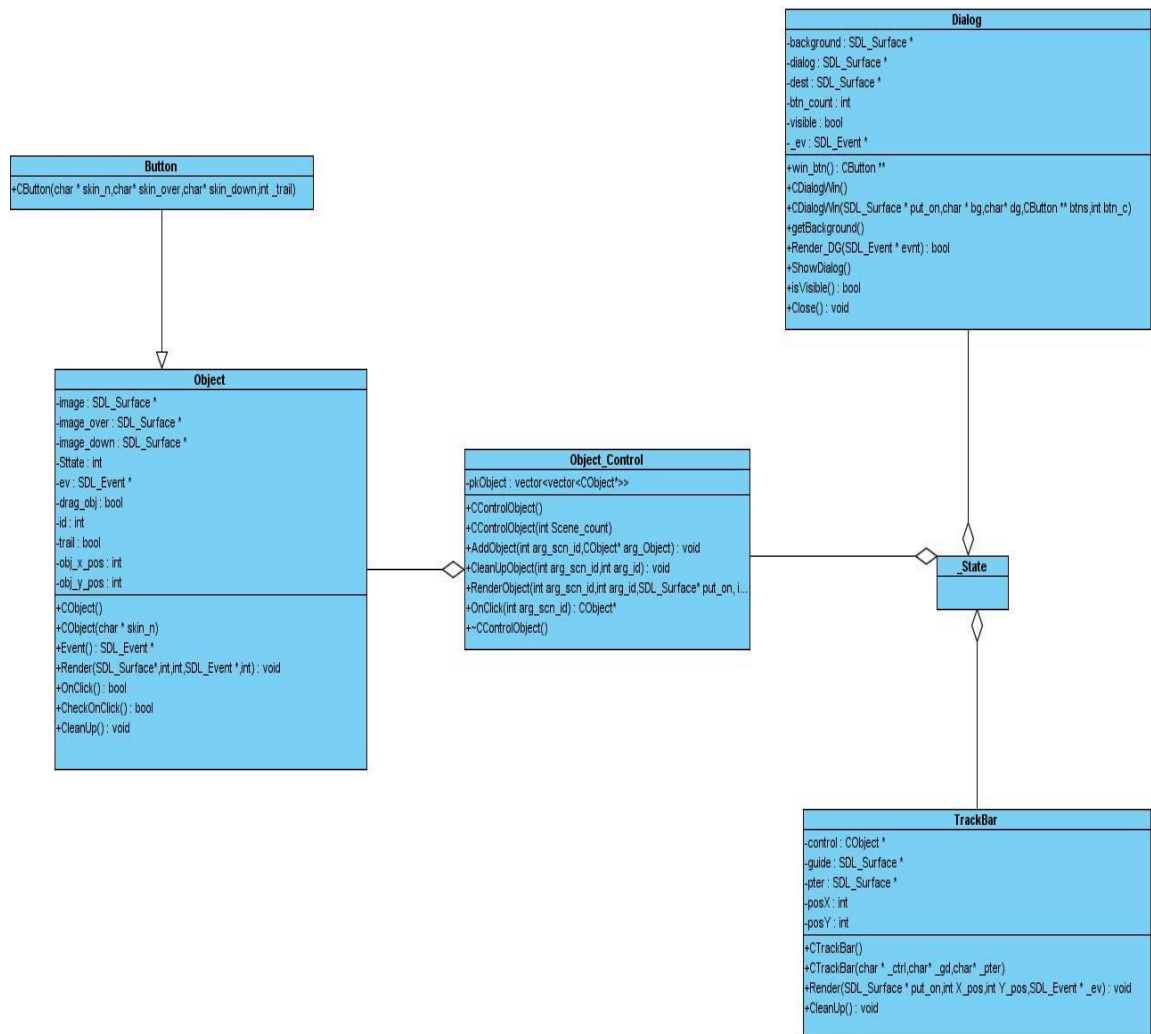


Figura 14: Paquete Componentes

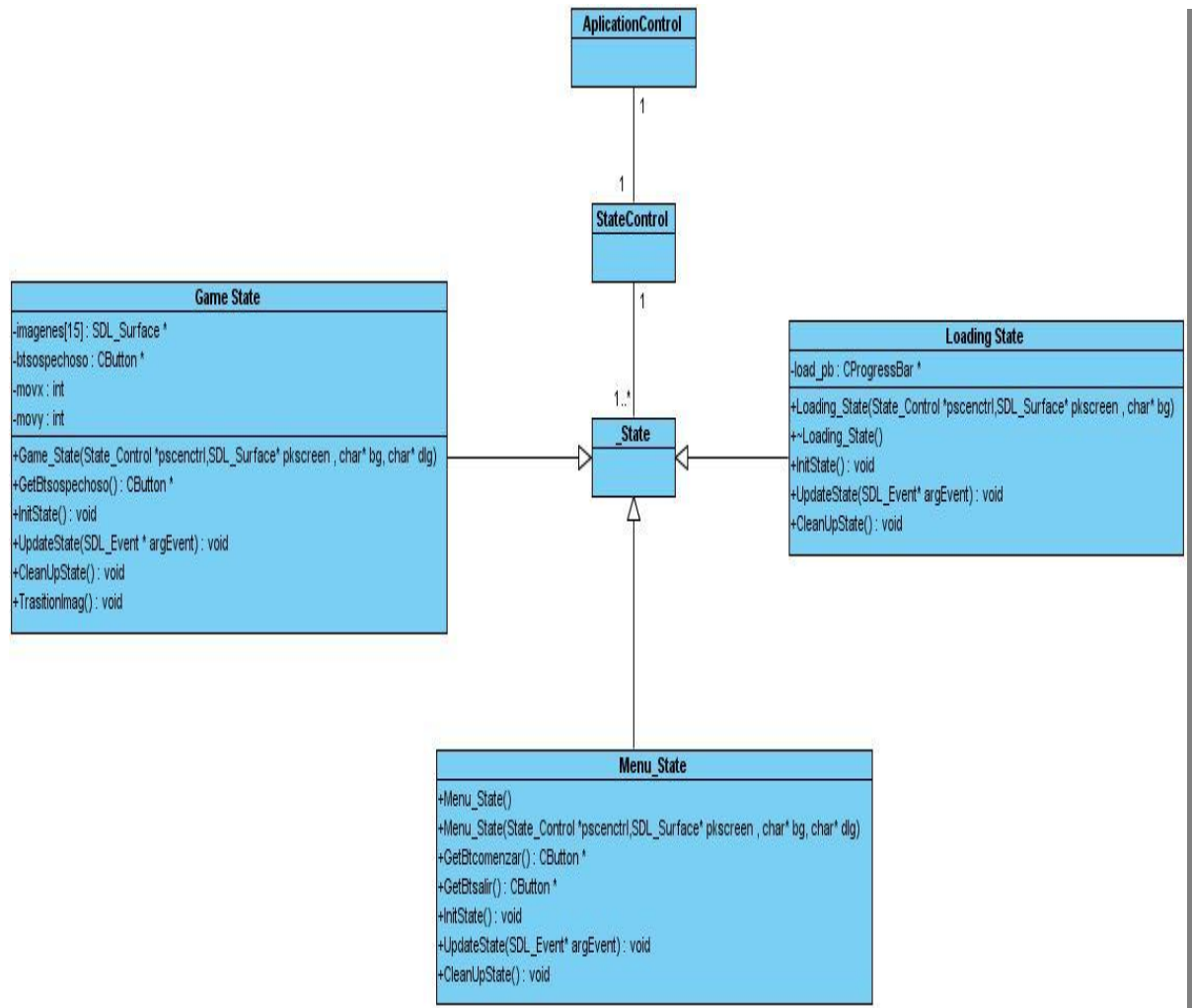


Figura 15: Paquete de Estados

4.2. Diagramas de Secuencia del Diseño

4.2.1. Realización del caso de uso “Iniciar y Detener Transición de Imágenes”

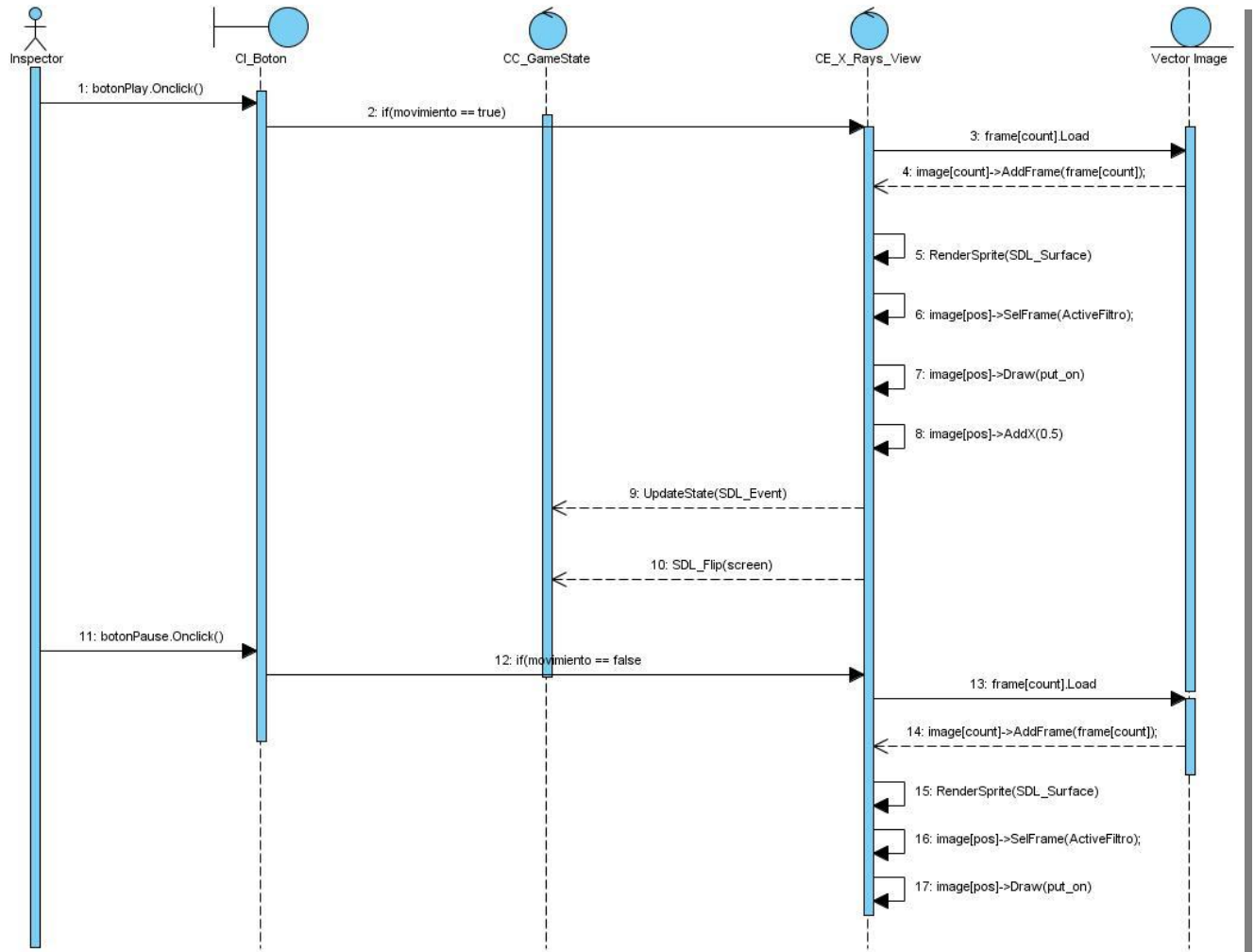


Figura 16: Diagrama de Secuencia “Iniciar y Detener Transición de Imágenes”

4.2.2. Realización del caso de uso “Aplicar Filtros”

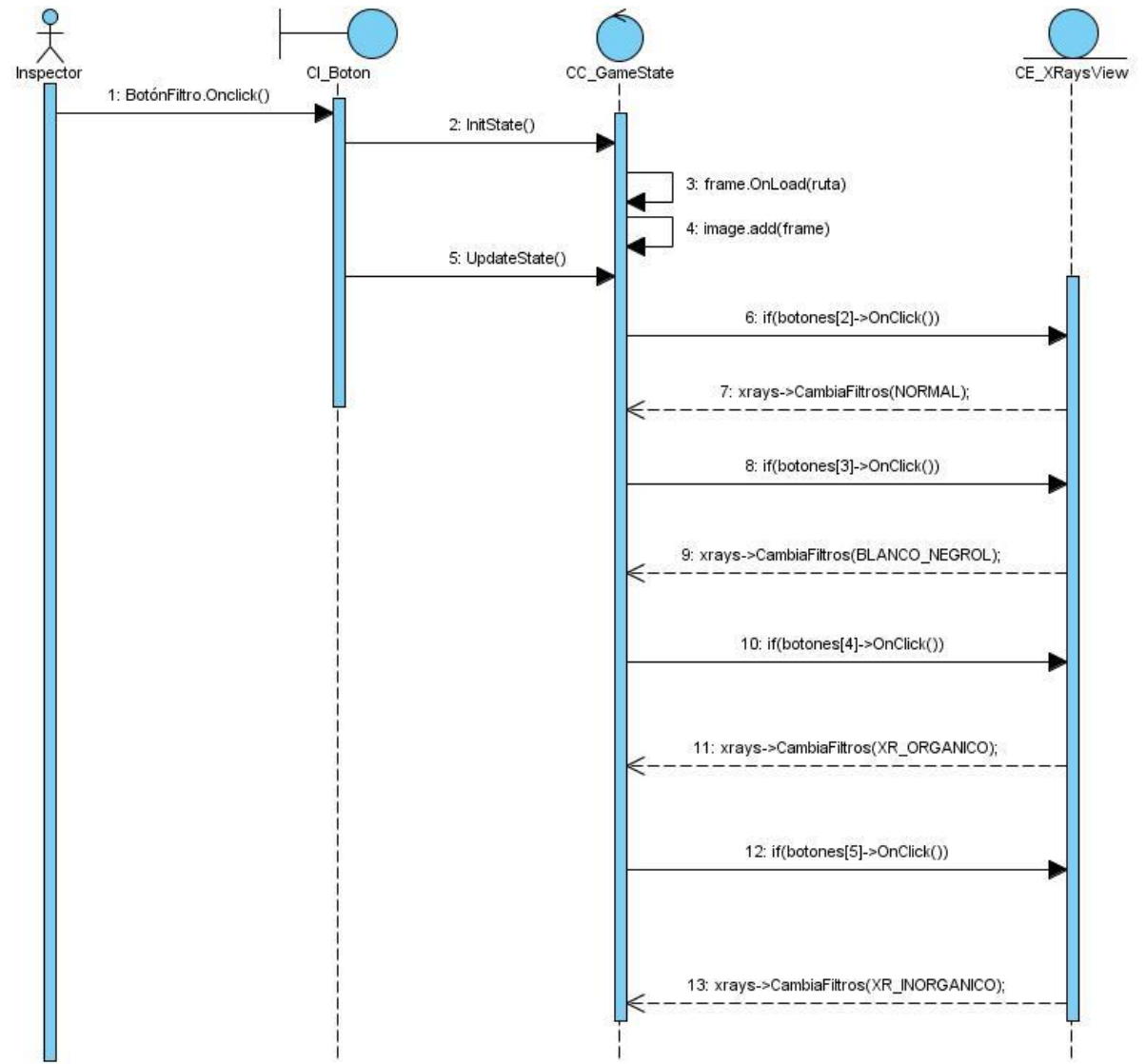


Figura 17: Diagrama de Secuencia “Aplicar Filtros”

4.2.3. Realización del caso de uso “Marcar imágenes con sospechas”

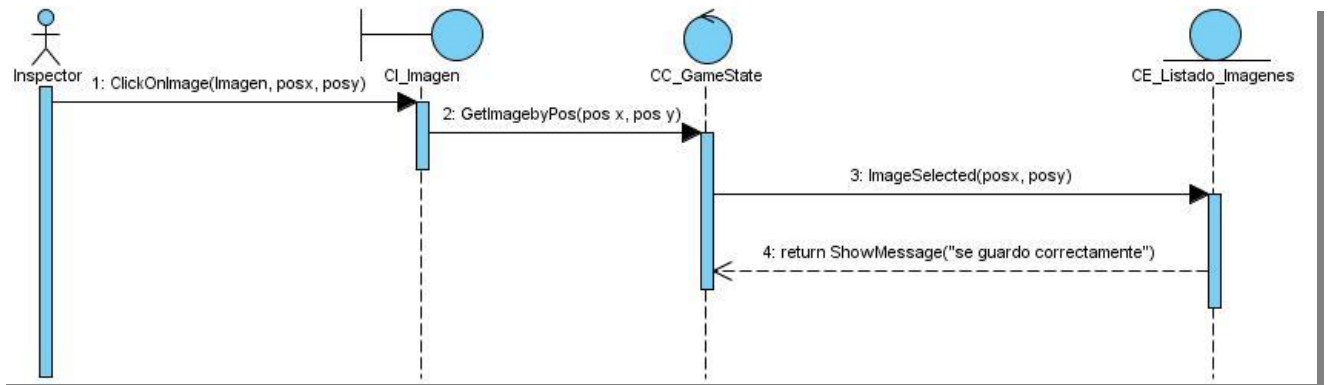


Figura 18: Diagrama de Secuencia “Marcar imágenes con sospechas”

4.2.4. Realización del caso de uso “Ver resultado”

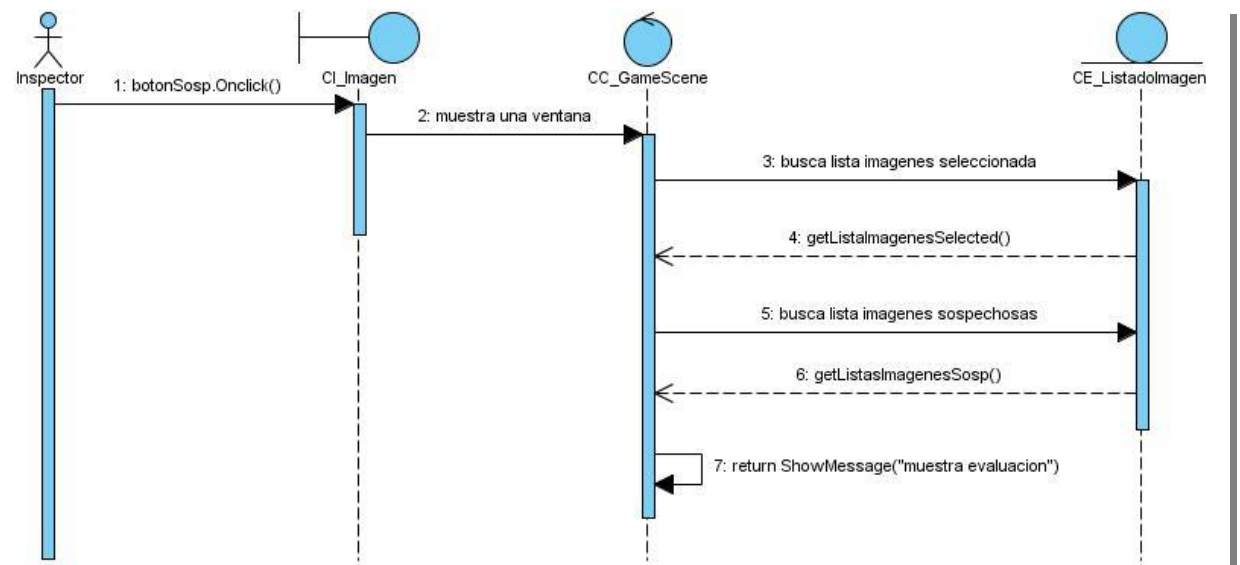


Figura 19: Diagrama de Secuencia “Ver resultado”

4.2.5. Realización del caso de uso “Crear Perfil”

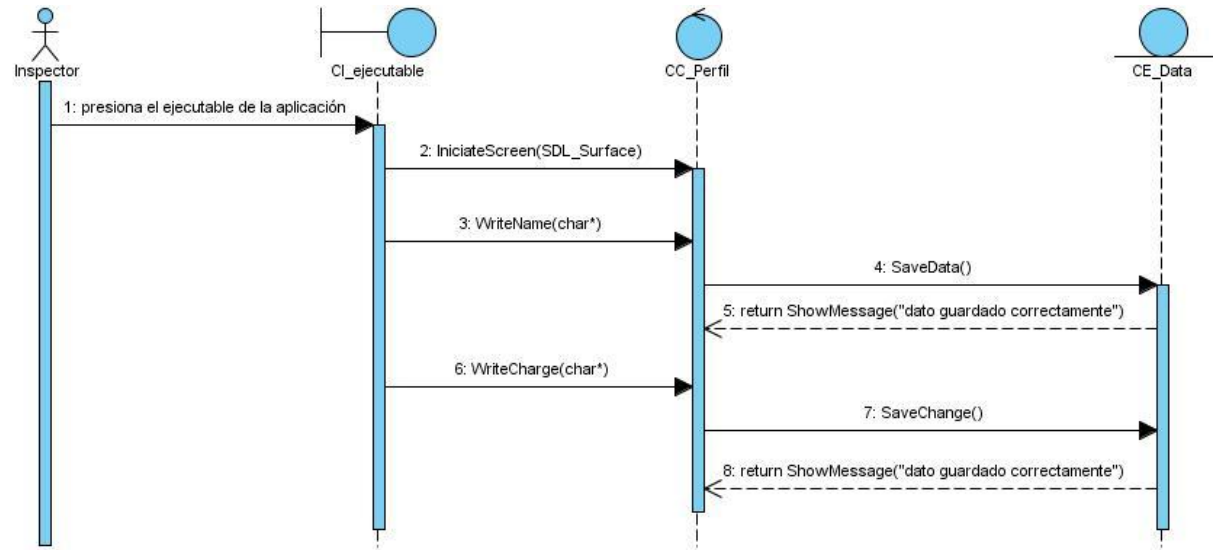


Figura 20: Diagrama de Secuencia “Crear Perfil”

4.2.6. Realización del caso de uso “Aplicar zoom a las imágenes”

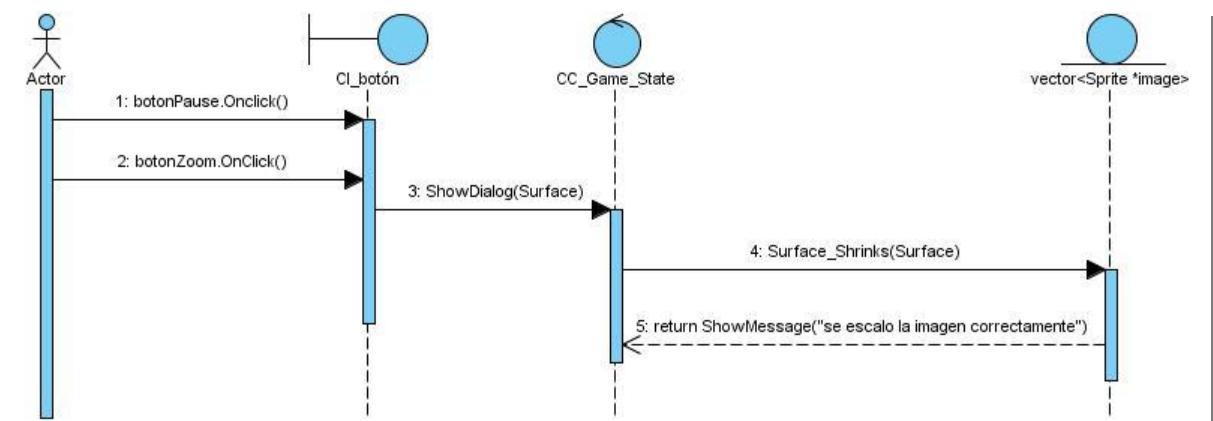


Figura 21: Diagrama de Secuencia “Aplicar zoom a las imágenes”

4.2.7. Realización del caso de uso “Controlar cambio de fase”

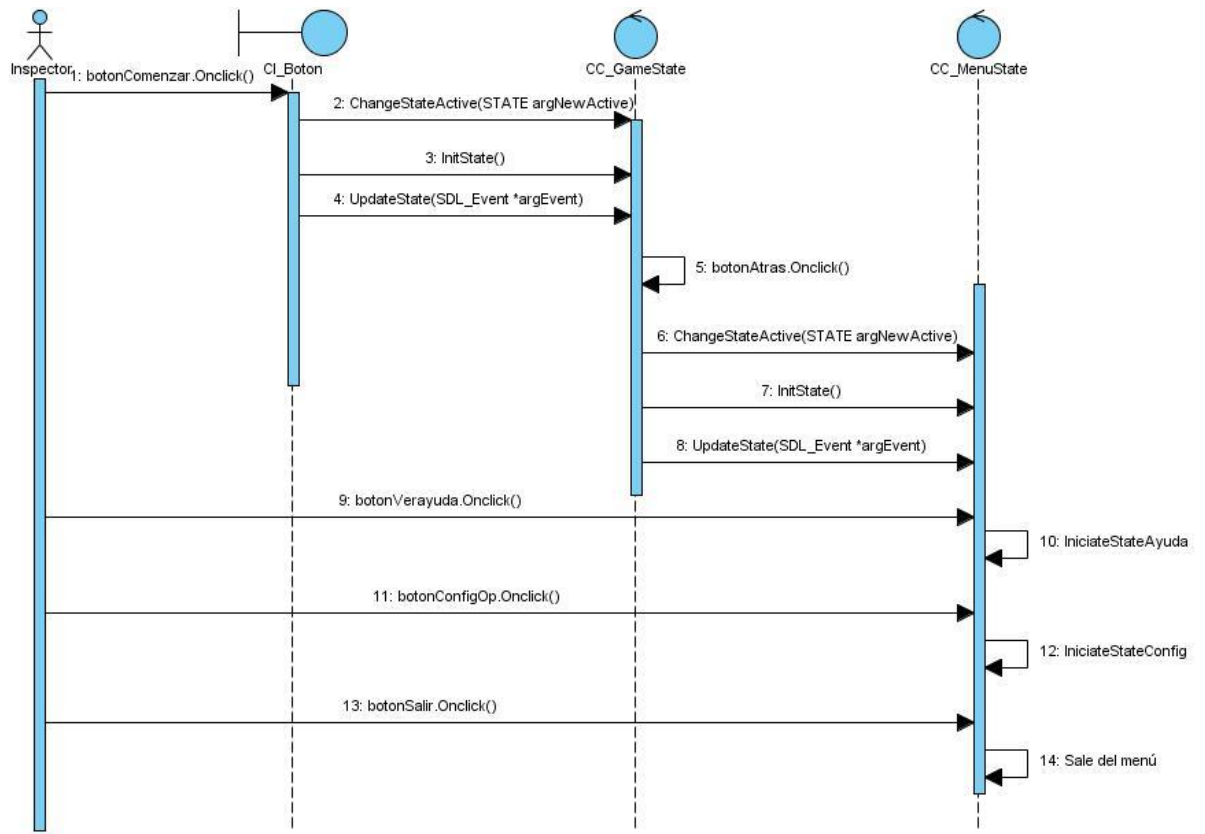


Figura 22: Diagrama de Secuencia “Controlar cambio de fase”

4.2.8. Realización del caso de uso “Configurar opciones”

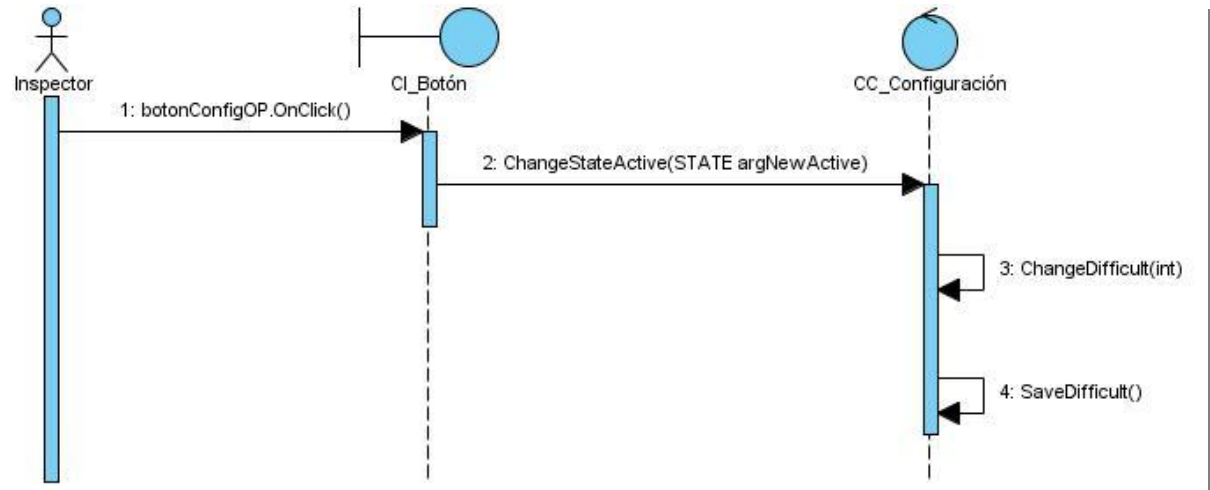


Figura 23: Diagrama de Secuencia “Configurar opciones”

4.2.9. Realización del caso de uso “Alterar imágenes”

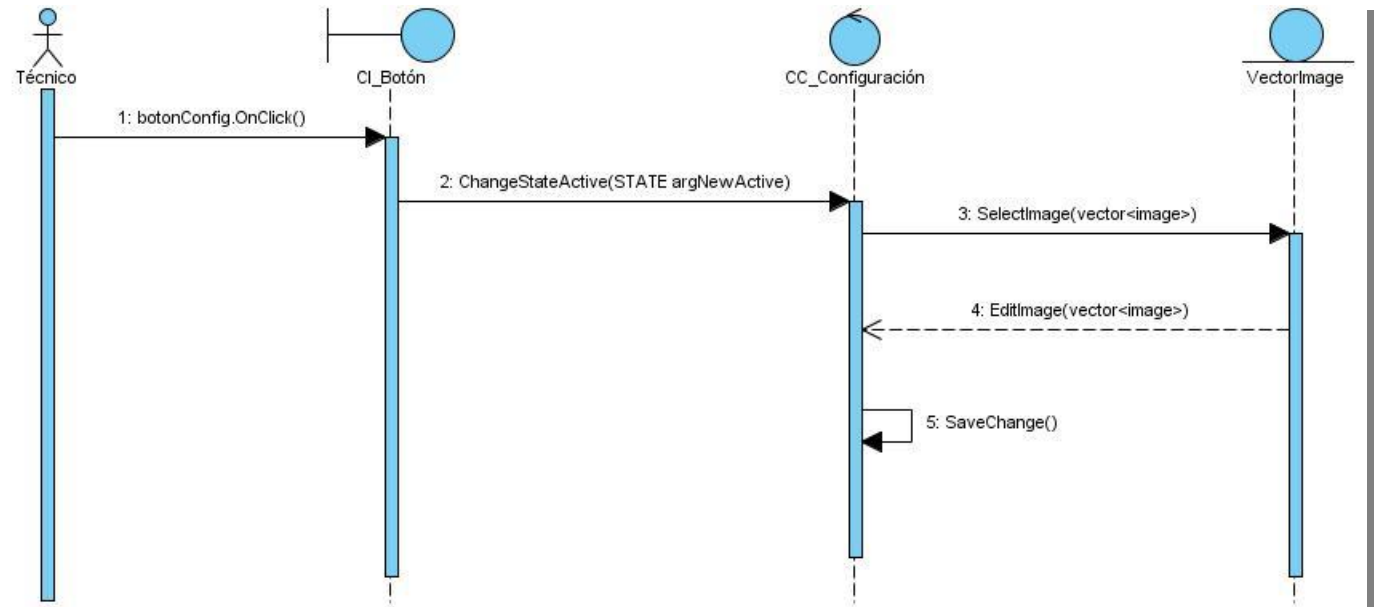


Figura 24: Diagrama de Secuencia “Alterar imágenes”

4.2.10. Realización del caso de uso “Configurar imágenes según nivel de dificultad”

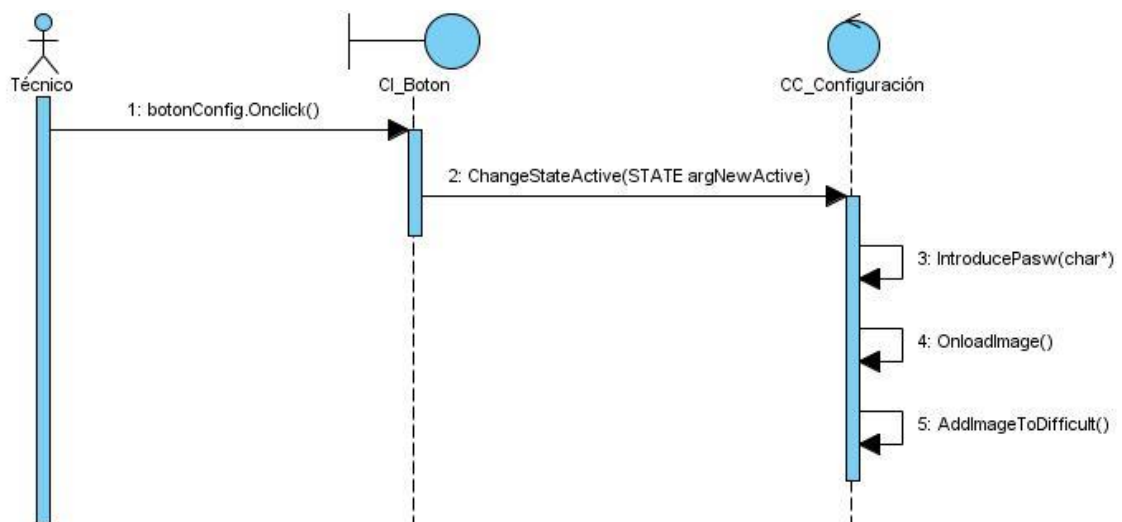


Figura 25: Diagrama de Secuencia “Alterar imágenes”

4.2.11. Realización del caso de uso “Cargar Imágenes”

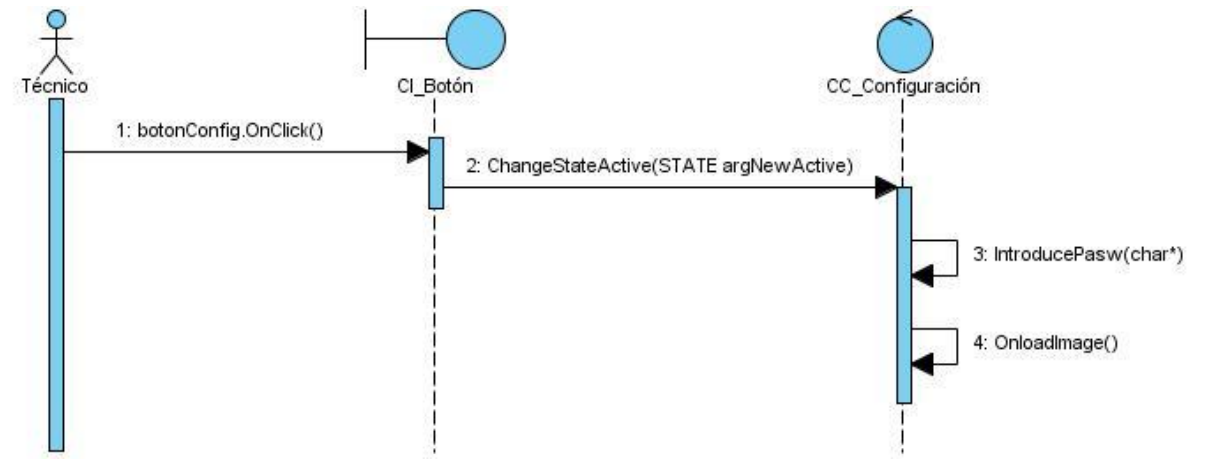


Figura 26: Diagrama de Secuencia “Alterar imágenes”

4.3. Descripción de las clases

4.3.1. Clase Controladora “Application_Control”

Tabla XIII: Clase “Application_Control”

Nombre: Application_Control	
Tipo de Clase: Controladora	
Atributo:	Tipo:
screen	SDL_Surface*

icon	SDL_Surface*
mouse	SDL_Surface*
scene_ctrl	State_Control*
running	bool
Para cada Responsabilidad:	
Nombre:	int OnExecute()
Descripción:	Se encarga de controlar la ejecución
Nombre:	bool OnInit()
Descripción:	Inicializa los estados de la Aplicación
Nombre:	bool OnInitSplash()
Descripción:	Inicializa los subsistemas a utilizar con SDL
Nombre:	void OnUpdate(SDL_Event* Event)
Descripción:	Pinta las escenas
Nombre:	CApplication()
Descripción:	Constructor de la Clase
Nombre:	void RunInFullSrnmode()
Descripción:	Función de configuración del FullScreen

4.3.2. Clase Controladora “State_Control”

Tabla XIV: Clase “State_Control”

Nombre: State_Control	
Tipo de Clase: Controladora	
Atributo:	Tipo:
pkScenes	std::map<STATE, State*>
ActiveState	State *
Para cada Responsabilidad:	
Nombre:	State_Control()
Descripción:	Constructor de la clase
Nombre:	~State_Control(void)
Descripción:	Destructor de la clase
Nombre:	void InitSimulation(SDL_Surface* argScreen)
Descripción:	Inicializa los objetos e imágenes del estado activo
Nombre:	void UpdateSimulation(SDL_Event* argEvent)
Descripción:	Pinta la escena del estado activo
Nombre:	void ShutDownSimulation()
Descripción:	Libera las superficies cargadas
Nombre:	void ChangeStateActive(STATE argNewActive)

Descripción:	Controla los cambios de estados de la aplicación
Nombre:	State * GetActiveState()
Descripción:	Devuelve el estado activo

4.3.3. Clase Base “State”

Tabla XV: Clase “State”

Nombre: State	
Tipo de Clase: Base	
Atributo:	Tipo:
pkBackground	SDL_Surface*
pkscreen	SDL_Surface*
Event	SDL_Event
m_posx, m_posy	int
scenctrl	State_Control *
visible	bool
bs_running	bool
Para cada Responsabilidad:	
Nombre:	State()
Descripción:	Constructor de la clase
Nombre:	SDL_Surface*& getBackground()

Descripción:	Me devuelve el background según el tipo de State
Nombre:	<code>SDL_Surface*& getDialog()</code>
Descripción:	Me devuelve un dialog que se utiliza en el State Game
Nombre:	<code>bool IsRunnig()</code>
Descripción:	Método para comprobar si está corriendo la aplicación
Nombre:	<code>virtual void InitState() = 0</code>
Descripción:	Método polimórfico, redefinido en las clases herederas para cargar las imágenes y los botones a mostrar
Nombre:	<code>virtual void UpdateState(SDL_Event *argEvent) = 0</code>
Descripción:	Método polimórfico, redefinido en las clases herederas para pintar las imágenes y los botones a mostrar
Nombre:	<code>virtual void CleanUpState() = 0</code>
Descripción:	Método polimórfico, redefinido en las clases herederas para limpiar todas las imágenes y botones de la memoria

4.3.4 Clase SubBase “Loading_State”

Tabla XVI: Clase “Loading_State”

Nombre: Loading_State	
Tipo de Clase: SubBase	
Atributo:	Tipo:
Para cada Responsabilidad:	

Nombre:	Loading_State(State_Control *pscncntrl,SDL_Surface* pkscreen , char* bg, char* dlg)
Descripción:	Constructor de la clase
Nombre:	void InitState()
Descripción:	Carga los elementos del estado Loading_State
Nombre:	void UpdateState(SDL_Event *argEvent)
Descripción:	Pinta los elementos del estado Loading_State
Nombre:	void CleanUpState()
Descripción:	Libera los elementos del estado Loading_State de la memoria

4.3.5 Clase SubBase “Menu_State”

Tabla XVII: Clase “Menu_State”

Nombre: Menu_State	
Tipo de Clase: SubBase	
Atributo:	Tipo:
btcomenzar	CButton *
btsalir	CButton *
Para cada Responsabilidad:	
Nombre:	Menu_State(State_Control *pscncntrl,SDL_Surface* pkscreen , char* bg, char* dlg)
Descripción:	Constructor de la clase

Capítulo 4: Diseño del Sistema

Nombre:	<code>void InitState()</code>
Descripción:	Carga los elementos del estado Menu_State
Nombre:	<code>void UpdateState(SDL_Event *argEvent)</code>
Descripción:	Pinta los elementos del estado Menu_State
Nombre:	<code>void CleanUpState()</code>
Descripción:	Libera los elementos del estado de la memoria
Nombre:	<code>void OnLButtonDown(int mx, int my)</code>
Descripción:	Método redefinido de la clase CEvent para hacer uso de los eventos del mouse

4.3.6. Clase SubBase “Game_State”

Tabla XVIII: Clase “Game_State”

Nombre: Game_State	
Tipo de Clase: SubBase	
Atributo:	Tipo:
botones[9]	CButton *
mover	<code>bool</code>
bw	<code>bool</code>
org	<code>bool</code>
inorg	<code>bool</code>

Para cada Responsabilidad:	
Nombre:	<code>Game_State(State_Control *psctrl,SDL_Surface* pkscreen , char* bg, char* dlg)</code>
Descripción:	Constructor de la clase
Nombre:	<code>void initState()</code>
Descripción:	Carga los elementos del estado <code>Game_State</code>
Nombre:	<code>void UpdateState(SDL_Event *argEvent)</code>
Descripción:	Pinta los elementos del estado <code>Game_State</code>
Nombre:	<code>void CleanUpState()</code>
Descripción:	Libera los elementos del estado de la memoria
Nombre:	<code>void OnLButtonDown(int mx,int my)</code>
Descripción:	Método redefinido de la clase <code>CEvent</code> para hacer uso de los eventos del mouse

4.3.7. Clase SubBase “CEvents”

Tabla XIX: Clase “CEvents”

Nombre: CEvents	
Tipo de Clase: Base	
Atributo:	Tipo:

Para cada Responsabilidad:	
Nombre:	<code>CEvent ()</code>
Descripción:	Constructor de la clase
Nombre:	<code>virtual void OnEvent(SDL_Event* Event)</code>
Descripción:	Se encarga de trabajar todos los eventos en la aplicación
Nombre:	<code>virtual void OnLButtonDown(int mX, int mY, CButton* pkButon[9]);</code>
Descripción:	Actúa con el evento del click izquierdo del mouse

4.3.8. Clase SubBase “CControlObject”

Tabla XX: Clase “CControlObject”

Nombre: CControlObject	
Tipo de Clase: Controladora	
Atributo:	Tipo:
<code>pkObject</code>	<code>vector<vector<CObject*>></code>
Para cada Responsabilidad:	
Nombre:	<code>CControlObject ()</code>
Descripción:	Constructor de la clase
Nombre:	<code>void AddObject(int arg_scn_id,CObject* arg_Object)</code>
Descripción:	Adiciona objetos a la estructura vector, me garantiza tener estos objetos almacenados, para luego poder tratarlos

Nombre:	<code>void CleanObjectList()</code>
Descripción:	Me limpia los objetos de la estructura definida.

4.3.9 Clase SubBase “CObject”

Tabla XXI: Clase “CObject”

Nombre: CObject	
Tipo de Clase: Controladora	
Atributo:	Tipo:
image	SDL_Surface *
image_over	SDL_Surface *
image_down	SDL_Surface *
ev	SDL_Event *
trail	bool
Para cada Responsabilidad:	
Nombre:	<code>CObject(char * skin_n, char* skin_over, char* skin_down, int is_trail)</code>
Descripción:	Constructor de la clase
Nombre:	<code>void AddObject(int arg_scn_id, CObject* arg_Object)</code>
Descripción:	Adiciona objetos a la estructura vector, me garantiza tener estos objetos almacenados, para luego poder tratarlos
Nombre:	<code>void CleanObjectList()</code>

Descripción:	Me limpia los objetos de la estructura definida.
---------------------	--

4.3.10. Clase SubBase “CButton”

Tabla XXII: Clase “CButton”

Nombre: CButton	
Tipo de Clase: SubBase	
Atributo:	Tipo:
Para cada Responsabilidad:	
Nombre:	CButton(char * skin_n, char* skin_over, char* skin_down, int _trail)
Descripción:	Constructor de la clase

4.3.11 Clase SubBase “CXRayView”

Tabla XXIII: Clase “CXRayView”

Nombre: CXRayView	
Tipo de Clase: Componente	

Capítulo 4: Diseño del Sistema

Atributo:		Tipo:
imagenes[15]		SDL_Surface *
imagenesBN[15]		SDL_Surface *
imagenesORG[15]		SDL_Surface *
imagenesINORG[15]		SDL_Surface *
dialg		SDL_Surface *
dest		SDL_Surface *
visible		bool
movx, movy		double
Para cada Responsabilidad:		
Nombre:	CX RaysView()	
Descripción:	Constructor de la clase	
Nombre:	void Inicie()	
Descripción:	Se encarga de cargar todas las imágenes a utilizar en el Game_State	
Nombre:	void RenderSprite(SDL_Surface * put_on)	
Descripción:	Su funcionalidad es la de pintar las imágenes reales de un equipo de rayos x	
Nombre:	bool ClickOnImage(CSprite *e, int x, int y)	
Descripción:	Su funcionalidad es la de verificar si fue marcada la imagen	
Nombre:	CSprite* GetImagebyPos(int x, int y)	

Capítulo 4: Diseño del Sistema

Descripción:	Su funcionalidad es la de devolver la imagen que fue marcada
Nombre:	<code>void UpdateXR(SDL_Surface * put_on)</code>
Descripción:	Su funcionalidad es la de mostrar las imágenes de un equipo de rayos x en pantalla
Nombre:	<code>void CleanUp()</code>
Descripción:	Libera las imágenes de la memoria del sistema
Nombre:	<code>void TrasitionImag(bool me_muevo)</code>
Descripción:	Logra el movimiento de las imágenes en el Game_State
Nombre:	<code>void setFiltro(FILTRO filter)</code>
Descripción:	Este método se encarga de cambiar al filtro que se active en algún momento de la simulación

4.3.12 Clase SubBase “CDialogResult”

Tabla XXIV: Clase “CDialogResult”

Nombre: CDialogResult	
Tipo de Clase: Componente	
Atributo:	Tipo:

Capítulo 4: Diseño del Sistema

background	SDL_Surface *
visible	bool
_ev	SDL_Event *
close	CButton *
rays	CXRaysView *

Para cada Responsabilidad:

Nombre:	CDialogWin()
Descripción:	Constructor de la clase
Nombre:	void Iniciate_DG()
Descripción:	Se encarga de cargar todas las imágenes a utilizar en el DialogResult
Nombre:	void Render_DG(SDL_Surface * put_on, SDL_Event * _event)
Descripción:	Su funcionalidad es la de pintar las Imágenes a utilizar en este componente
Nombre:	void ClearDialog()
Descripción:	Su funcionalidad es la de limpiar todo el componente
Nombre:	void OnLeftClick(int mx, int my)
Descripción:	Controla todos los eventos pasado por el mouse en el componente

4.3.13 Clase Componente “CSprite”

Tabla XXV: “Clase “CSprite”

Nombre: CSprite	
Tipo de Clase: Componente	
Atributo:	Tipo:
Sprite	Cframe *
nframes	int
posx	double
posy	int
estado	int
Para cada Responsabilidad:	
Nombre:	CSprite()
Descripción:	Constructor de la clase
Nombre:	void AddFrame(Cframe frame)
Descripción:	Se encarga de adicionar un frame al sprite, cada frame sería una imagen y como tengo cuatro filtros lo que se le adiciona son cuatro frames
Nombre:	void SelFrame(int nf)
Descripción:	Su funcionalidad es la de pintar la posición del frame
Nombre:	void AddX(double c)
Descripción:	Su funcionalidad es la de aumentar el eje x, para lograr el movimiento del frame

Nombre:	<code>void Draw(SDL_Surface *superficie)</code>
Descripción:	Se encarga de pintar cada uno de los frames del sprite
Nombre:	<code>void Escalar(double height, double weight)</code>
Descripción:	Se encarga de disminuir o aumentar las dimensiones de la imagen, según los valores pasados por parámetros

4.3.14 Clase Componente “Cframe”

Tabla XXVI: “Clase “Cframe”

Nombre: Cframe	
Tipo de Clase: Componente	
Atributo:	Tipo:
Img	SDL_Surface *
Para cada Responsabilidad:	

Capítulo 4: Diseño del Sistema

Nombre:	<code>Cframe(void)</code>
Descripción:	Constructor de la clase
Nombre:	<code>void Load(char*Path)</code>
Descripción:	Se encarga de cargar el fichero al frame
Nombre:	<code>void Unload()</code>
Descripción:	Su funcionalidad es la de limpiar la superficie de la imagen del frame

Capítulo 5: Implementación

Introducción

Esta etapa del proyecto, constituye el paso del diseño de clases a la creación de componentes físicos .h y .cpp correspondiente a la implementación en C++

5.1. Estándar de Codificación

Constantes: Los nombres de las constantes se escriben completamente con mayúsculas, en caso de que el nombre este compuesto por más de una palabra, estas son separadas por el carácter “_”.

Variables: Las variables comenzarán con una letra minúscula, y a continuación la palabra o palabras que conformen el nombre de la variable se mantendrá en minúsculas. Y aquellas variables que sean para representar una ventana, un fondo o un dialogo todas van a empezar con las iniciales pk, siguiendo a continuación la definición en minúscula.

Clases: Los nombres de las clases se escriben con letra inicial mayúscula, en caso de tener un nombre conformado por más de una palabra estas se escribirán una a continuación de la otra comenzando siempre cada palabra con una letra mayúscula y el resto en minúscula.

Métodos miembros de clases: Los nombres de los métodos miembros de una clase comenzarán con letra inicial minúscula, en caso de que el nombre este compuesto por más de una palabra la primera palabra será escrita en minúscula completamente y el resto comenzará con letra mayúscula.

Nombres de enumerados: Los enumerados comenzarán con la letra “E” y a continuación la palabra o palabras que conformen el nombre, cada una comenzando con letra mayúscula y el resto en minúsculas. Los valores de los enumerados cumplen con las mismas reglas que las constantes.

5.2. Diagrama de despliegue.

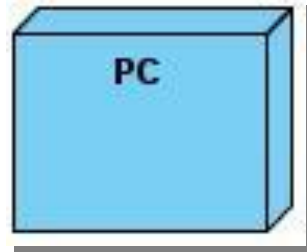


Figura 27: *Diagrama de Despliegue*

5.3. Diagramas de Componentes

5.3.1. Paquete del Simulador

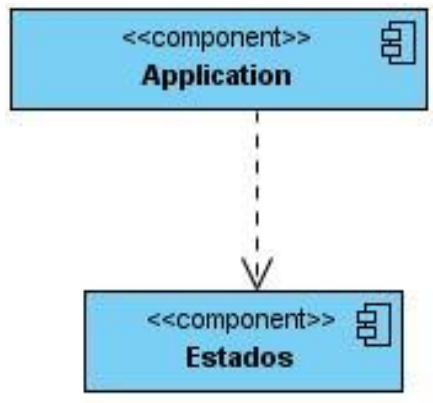


Figura 28: *Paquete del Simulador*

5.3.2. Diagrama de Componente, paquete “Application”

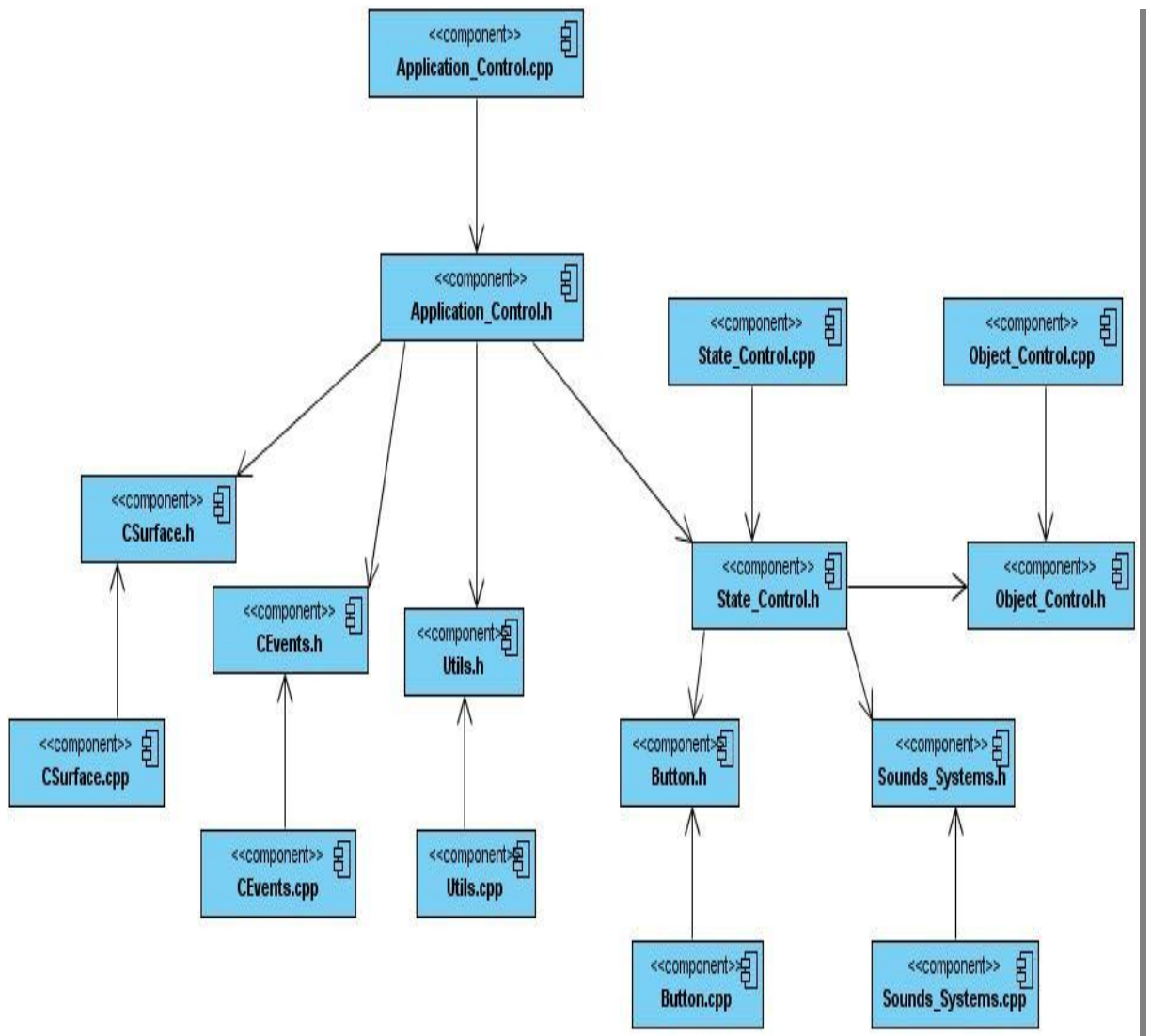


Figura 29: Diagrama de Componente “Application”

5.3.3. Diagrama de Componente, paquete “Estados”

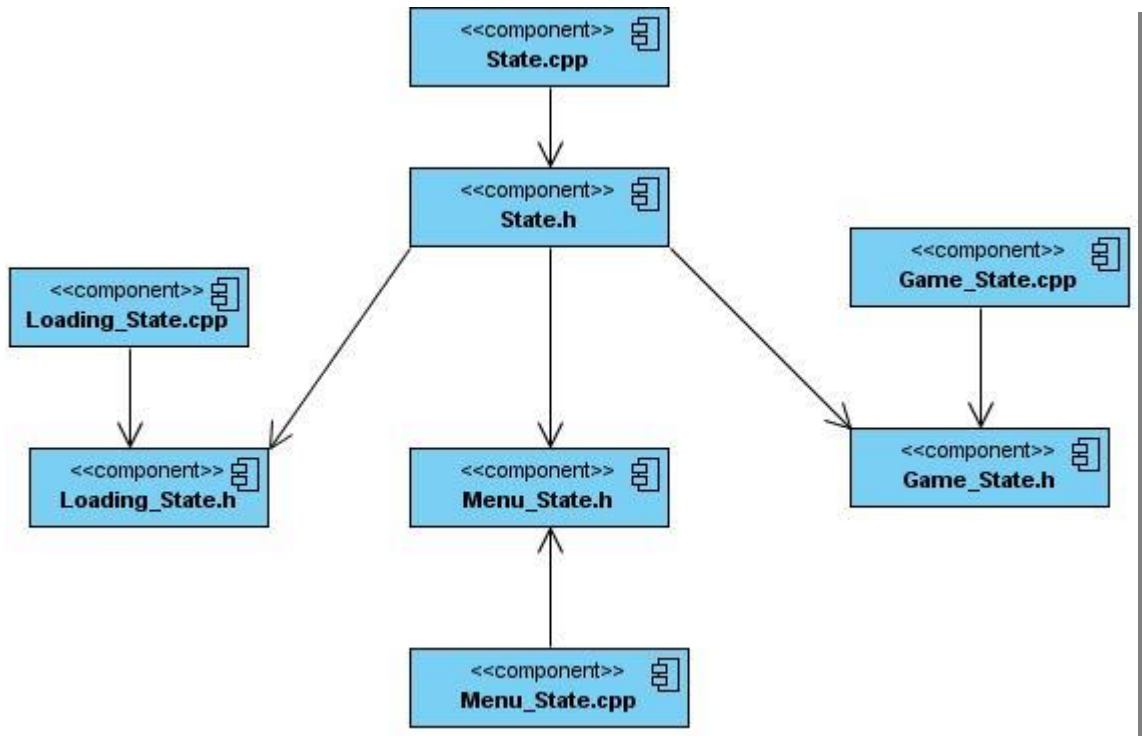


Figura 30: Diagrama de Componente “Estados”

En este capítulo se mostraron los diagramas pertenecientes al flujo de trabajo de Implementación. A partir de este momento están sentadas todas las condiciones necesarias para proceder a la etapa de programación para la siguiente fase del software.

Conclusiones

Tras la realización de esta investigación se puede arribar a la conclusión fundamental de que la puesta en práctica de la aplicación desarrollada permite el entrenamiento del personal aduanero que labora en los aeropuertos y puertos del país.

El simulador implementado posee una interfaz agradable e interactiva que permite a cualquier operador aduanero hacer uso del mismo sin complicaciones idiomáticas o de simbología.

En cuanto a su estructura arquitectónica se puede concluir que cuenta con un diseño flexible que permite realizar transformaciones en el simulador, para futuras funcionalidades visuales.

Recomendaciones

Extender la biblioteca de las imágenes del software, mediante la conexión a bases de datos para garantizar la diversidad y la exactitud en la muestra de los objetos visualizados.

Crear un mecanismo para la configuración y actualización de la galería de imágenes. Crear perfiles de usuario en el producto de software en concordancia con las jerarquías que existen en la Aduana General de la República.

Se recomienda además insertar la funcionalidad de asignación de evaluación al usuario y un sistema de reporte estadístico.

Por la novedad en las soluciones que brinda y el carácter pionero que tiene el software en el desarrollo de este tipo de aplicaciones en el país, se propone su utilización en las agencias aduaneras de Cuba y la creación de software con características generales para su comercialización.

Bibliografía

1. **De Antonio, Angélica y Villalobos, M. y Luna, E.** Cuándo y cómo usar la Realidad Virtual en la enseñanza. Universidad de Guadalajara : s.n., 2000.
2. **Universidad de Zaragoza (Pedro Cerbuna 12, 50009 ZARAGOZA-ESPAÑA.** www.unizar.es/. [En línea] 2009. <http://www.unizar.es/otri/doc/spinoff/news/simulacion.pdf>.
3. <http://www.grupoe4.com/grupoe4/index.html>. [En línea] Grupo E4, 2002. <http://www.grupoe4.com/grupoe4/simuladores.html>.
4. **LTD, Renful Premier Technologies Renful Aviation Security.** renful.co.uk. [En línea] 2003. <http://www.renful.co.uk/index.php?ID=sim>.
5. **Cerda, Instructor de Rx: Miguel E.González de la.** ANALISIS DE IMÁGENES RADIOSCOPICAS. Habana : s.n., 2007.
6. **Rucker, Rudy.** *Software Engineering and Computer Games* . . San Jose, California : s.n., 2002.
7. **Alba, Antonio García.** [uca.es](http://www.uca.es). [En línea] 2008 . <http://www.uca.es/softwarelibre/wikiSDL>.
8. **Peris, R. J.** *Taller de Tratamiento de Imágenes*. 2005.
9. **León, M. P.** *Curso de Superación de Operadores de Rayos X*. [ppt] Habana. : E.N.F.A, 2004.
10. *Quadratica:uk*. [En línea] Europarc Innovation Centre, Innovation Way, 2004.

Glosario de Abreviaturas

SRV: Sistema de Realidad Virtual

ENFA: Escuela nacional de Formación Aduanera

UCI: Universidad de las Ciencias Informáticas

RX: Rayos X

X-ray on: Luces señalizadores en los equipos de Rayos X en la Aduana

SIMFOX: Simulador de entrenamiento de radiografía

SDL: Simple DirectMedia Layer. Biblioteca de gráficos 2D, además de que es abierta

API: Application Programmer's Interface (Interfaces para programadores de aplicaciones)

OpenGL: Open Graphics Library (Biblioteca de gráficos abierta)

DirectX: Proporciona gráficos en 2D y 3D

BeOS: Sistema Operativo desarrollado por Be Incorporated.

Win32: Versión del API de Windows de 32 bits

Windows: Sistema Operativo desarrollado e incorporado por Microsoft

Linux: Núcleo del Sistema Operativo libre denominado GNU/Linux

UML: Unified Modeling Language (Lenguaje Unificado de Modelado)

Glosario de Términos

Simulación: "La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término, experiencias con el mismo, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias -dentro de los límites impuestos por un cierto criterio o un conjunto de ellos - para el funcionamiento del sistema" (1)

Bibliotecas: Conjunto de subprogramas y clases utilizados para desarrollar una aplicación informática.

Tecnología: Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.

Texturas: Proporciona riquezas visuales a las imágenes

Eventos: Es una acción realizada por el usuario de una interfaz de usuario utilizando el mouse

Filtros: Se encarga de darle coloraciones diferentes a las imágenes

Zoom: Escala regiones de una imagen en específico tanto hacia adentro de la misma, como hacia afuera

Joystick: Un joystick o palanca de mando es un dispositivo de control de dos o tres ejes que se usa desde una computadora