

Universidad de las Ciencias Informáticas



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

Título:

*Análisis y diseño para el componente de captura de imágenes
faciales con múltiples modelos de cámaras digitales.*

Autor: “Diana Cantero Panteleeva.”

Tutor: “Ing. Yainier Labrada Nueva.”

Co-tutor: “Msc. Yurdiq Cervantes Mendoza.”

Asesor: “Lic. Miguel Ángel Hernández de la Rosa.”

La Habana. Junio, 2011.

“Año 53 de la Revolución”



“Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.”

Ernesto Che Guevara

Declaración de autoría

Declaro que soy el único autor(a) de este trabajo titulado: Análisis y diseño para el componente de captura de imágenes faciales con múltiples modelos de cámaras digitales y autorizo al Departamento de Biometría del Centro de Identificación y Seguridad Digital (CISED) de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

“Diana Cantero Panteleeva”

“Ing. Yainier Labrada Nueva”

Firma del autor

Firma del tutor

“Msc. Yurdik Cervantes Mendoza”

“Lic. Miguel Ángel Hernández de la Rosa”

Firma del cotutor

Firma del asesor

Datos de contacto

Nombre y Apellidos: Ing. Yainier Labrada Nueva.

Correo: ylabrada@uci.cu

Situación laboral: Profesor, Instructor Facultad 1.

Años de graduado: 3 años.

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Institución a la que pertenece: Universidad de las Ciencias Informáticas (UCI).

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Nombre y Apellidos: Msc. Yurdik Cervantes Mendoza.

Correo: yurdik@uci.cu

Situación laboral: Profesor Asistente, Facultad 1.

Años de graduado: 8 años.

Especialidad de graduación: Ingeniero Informático en el Instituto Superior Politécnico José Antonio Echeverría (CUJAE). Máster en Informática Aplicada con 5 años de experiencia en Biometría.

Institución a la que pertenece: Universidad de las Ciencias Informáticas (UCI).

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Nombre y Apellidos: Lic. Miguel Ángel Hernández de la Rosa.

Correo: miguelangel@uci.cu

Situación laboral: Profesor Asistente, Facultad 1.

Años de graduado: 4 años.

Especialidad de graduación: Licenciatura en Educación Especialidad Informática.

Institución a la que pertenece: Universidad de las Ciencias Informáticas (UCI).

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Dedicatoria

Dedico este trabajo a esa persona que tanto quiero y que me brindó su amor incondicional pese a cualquier circunstancia y me impulsó a terminar mis estudios, a ti abuela.

A mi madre Elena Panteleeva y mi padre Luis Cantero Corrales, por su sacrificio, cariño y por haber sabido guiarme por el buen camino.

A mi abuelo Luis Cantero Gutiérrez, que tanto me ha apoyado y se ha sacrificado por mí.

A mi hermana Karen Dianelys Cantero López, que tanto quiero.

A mi novio Ernesto Roja López, por el cariño y amor que me ha brindado.

A mis primas Dania y Deborah Carmenate, a mi madrastra Loly, mis tíos Ángel y Estherlily por quererme tal como soy.

Agradecimientos

Agradezco en primer lugar a mi familia, a mi abuela, abuelo, a mis padres, a mi hermana, a mi madrastra, tíos y primas que son mi razón de ser y que tanto me han apoyado en estos cinco años.

A mi novio, gracias por tu cariño, paciencia y sobre todo por enseñarme a confiar en mí.

A mis amigos, por tener la paciencia de soportarme.

A mis profesores de la UCI, que con su esfuerzo me han formado como ingeniera.

A mis tutores por los consejos y el apoyo ofrecido durante la realización de este trabajo.

A la Revolución por haberme dado la oportunidad de estudiar en la Universidad de las Ciencias Informáticas.

A todos los que de una forma u otra han contribuido a la realización del trabajo de diploma.

A todos los que influyeron en mi formación personal y profesional.

Resumen

En la actualidad, a pesar del desarrollo alcanzado en el acceso y comunicación con los dispositivos de captura de imágenes en los países del primer mundo, nuestro país no presenta un desarrollo en este tema en el que pueda emplear y explotar sus beneficios. El presente trabajo es el resultado de la investigación detallada de los aspectos relacionadas con el desarrollo de componentes de software, para la captura de imágenes faciales con múltiples modelos de cámaras digitales. Se estudian las tecnologías que se utilizan en los dispositivos de captura y en el acceso a estos, además se fundamentan los lenguajes y herramientas que se emplean. El mismo recoge los resultados obtenidos durante el desarrollo de la investigación, donde se utilizó la metodología Proceso Unificado de Rational (RUP), obteniéndose los modelos de análisis y de diseño, los cuales constituyen la entrada para un desarrollo futuro del componente. Se establecen las conclusiones obtenidas de la investigación y se plantean un conjunto de recomendaciones que deben ser tomadas para futuros usos y versiones del producto.

Palabras clave

Análisis, diseño, componente, captura de imágenes, cámaras digitales.

Tabla de contenido

Introducción	1
Capítulo 1: Fundamentación teórica	5
1.1 Introducción	5
1.2 Conceptos asociados a los dispositivos de captura.....	5
1.2.1 Vídeo.....	5
1.2.2 Dispositivos de captura	6
1.2.3 Procedimiento para la captura.....	7
1.3 Acceso a dispositivos	10
1.3.1 Tecnologías que se usan en el acceso a los dispositivos	10
1.4 Sistemas similares en el mundo que informatizan la captura de imágenes faciales	15
1.5 Caracterización de la plataforma de integración de dispositivos existente en el CISED	16
1.6 Desarrollo de componentes.....	19
1.6.1 Metodología	19
1.6.2 Herramientas CASE	22
1.6.3 Lenguaje de modelado	23
1.6.4 Marco de trabajo	23
1.6.6 Entorno de Desarrollo Integrado.....	24
1.7 Arquitectura de software	25
1.8 Patrones de diseño	26
1.9 Conclusiones	27
Capítulo 2: Características del sistema	28
2.1 Introducción	28
2.2 Descripción del sistema propuesto.....	28
2.3 Modelo del dominio	28
2.4 Requisitos	29
2.4.1 Captura de requisitos	30

2.4.2	Definición de requisitos	30
2.4.3	Técnica de validación	36
2.5	Modelo de casos de uso del sistema.....	37
2.5.1	Actores del sistema	37
2.5.2	Casos de Uso del sistema.....	37
2.5.3	Diagrama de Casos de Uso del sistema.....	38
2.5.4	Descripción de los CU	38
2.6	Conclusiones	43
Capítulo 3: Análisis y diseño		44
3.1	Introducción	44
3.2	Descripción de la arquitectura	44
3.3	Análisis	44
3.3.1	Modelo de análisis.....	45
3.3.2	Clase del análisis	45
3.3.3	Diagrama de clases del análisis	46
3.3.4	Diagramas de colaboración	48
3.4	Diseño.....	50
3.4.1	Patrones de diseño utilizados.....	50
3.4.2	Modelo de diseño	50
3.4.3	Clase del diseño.....	50
3.4.4	Diagrama de clases del diseño.....	51
3.4.5	Diagramas de secuencia	52
3.4.6	Descripción de las clases.	55
3.5	Modelo de despliegue	59
3.6	Conclusiones	60
Capítulo 4: Estudio de factibilidad y análisis de costos		61
4.1	Introducción	61
4.2	Planificación basada en casos de uso.....	61
4.3	Beneficios tangibles e intangibles.....	69
4.4	Análisis del costo	70

4.5 Análisis de costos y beneficios.....	71
4.6 Conclusiones	71
Conclusiones	72
Recomendaciones	73
Bibliografía referenciada	74
Bibliografía consultada.....	79
Anexos.....	83
Glosario	103

Índice de figuras.

Figura 1. Streaming.	8
Figura 2. Gráfico que muestra la arquitectura básica de WIA como una vía de dos capas entre las aplicaciones de imágenes y dispositivos.	12
Figura 3. Arquitectura en capas de la plataforma para el acceso a dispositivos.	17
Figura 4. Arquitectura de la plataforma de integración de dispositivos existente en el CISED.	18
Figura 5. Fases e iteraciones de la metodología RUP.	20
Figura 6. Modelo de dominio propuesto.	29
Figura 7. Diagrama de Casos de Uso.	38
Figura 8. Arquitectura en capas del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales.	44
Figura 9. Diagrama de clases del análisis del CU “Inicializar dispositivo de captura”.	47
Figura 10. Diagrama de clases del análisis del CU “Ajustar vídeo”.	47
Figura 11. Clase del análisis del CU “Capturar imagen”.	47
Figura 12. Diagrama de colaboración del CU “Inicializar dispositivo de captura”.	48
Figura 13. Diagrama de colaboración del CU “ Ajustar vídeo”.	49
Figura 14. Diagrama de colaboración del CU “Capturar imagen”.	49
Figura 15. Diagrama de clases del diseño.	52
Figura 16. Diagrama de secuencia del diseño para el CU “Inicializar dispositivo de captura” para cuando es un dispositivo “digital”.	53
Figura 17. Diagrama de secuencia del diseño para el CU “Inicializar dispositivo de captura” para cuando es un dispositivo “vídeo”.	53
Figura 18. Diagrama de secuencia del diseño para el CU” Ajustar vídeo”.	54
Figura 19. Diagrama de secuencia del CU “Capturar imagen” para cuando es un dispositivo “digital”.	54
Figura 20. Diagrama de secuencia del CU “Capturar imagen para cuando es un dispositivo “vídeo”.	55
Figura 21. Diagrama de despliegue del componente.	60
Figura 22. Proporción del rostro.	85

Índice de tablas.

Tabla 1. Requisitos funcionales.	31
Tabla 2. Requisitos no funcionales.	32
Tabla 3. Especificación del requisito “Verificar compatibilidad con la tecnología genérica WIA”.....	33
Tabla 4. Especificación del requisito “Visualizar vídeo”.....	34
Tabla 5. Especificación del requisito “Realizar zoom del lente de la cámara”.....	34
Tabla 6. Especificación del requisito “Seleccionar tipo de flash”.....	35
Tabla 7. Especificación del requisito “Captar la imagen”.....	36
Tabla 8. Especificación del requisito “Mostrar la imagen capturada”.....	36
Tabla 9. Actor del sistema.....	37
Tabla 10. Casos de Uso del sistema.....	37
Tabla 11. Resumen del CU “Inicializar el dispositivo de captura”.....	39
Tabla 12. Resumen del CU “Ajustar imagen”.....	39
Tabla 13. Resumen del CU “Capturar la imagen”.....	40
Tabla 14. Descripción extendida del CU “Inicializar dispositivo de captura”.....	41
Tabla 15. Descripción extendida del CU “Ajustar vídeo”.....	42
Tabla 16. Descripción extendida del CU “Capturar la imagen”.....	43
Tabla 17. Nombre, estereotipo y descripción de las clases del análisis.....	46
Tabla 18. Descripción de la clase “CaptadorFotoBase” del diseño.....	56
Tabla 19. Descripción de la clase “CamDigitalBase” del diseño.....	56
Tabla 20. Descripción de la clase “CamVideoBase” del diseño.....	57
Tabla 21. Descripción de la clase “VLCCamDigital” del diseño.....	57
Tabla 22. Descripción de la clase “WIACamDigital” del diseño.....	58
Tabla 23. Descripción de la clase “WIACamVideo” del diseño.....	58
Tabla 24. Descripción de la clase “VLCCamVideo” del diseño.....	58
Tabla 25. Descripción de la clase “IZoom” del diseño.....	59
Tabla 26. Descripción de la clase “IFlash” del diseño.....	59
Tabla 27. Descripción de la clase “VideoStream” del diseño.....	59
Tabla 28. Factor de peso de los actores sin ajustar.....	62

Tabla 29. Factor de peso de los casos de uso sin ajustar.....	63
Tabla 30. Peso de los factores de complejidad técnica.....	65
Tabla 31. Escala para evaluar los factores.....	66
Tabla 32. Peso de los factores ambientales.....	67
Tabla 33. Valores del esfuerzo.....	69
Tabla 34. Modelo de entrevista realizada.....	88
Tabla 35. Matriz de trazabilidad.....	91
Tabla 36: Lista de chequeo para requisitos.....	102

Introducción

Las cámaras digitales modernas tienen múltiples funciones y contienen algunos dispositivos capaces de grabar sonido y vídeo, además de fotografías. Los mercados están saturados porque los consumidores quieren más cámaras y los fabricantes tienen que luchar más, ya que las cámaras cada vez cuestan menos y vienen más equipadas. La atención se ha centrado, sobre todo, en las nuevas habilidades de estas que ahora son más rápidas e inteligentes. Muchas cámaras son capaces de detectar rostros en la fotografía y enfocarlos de forma automática, las mismas son usadas por sistemas biométricos como dispositivos de captura.

Muchas soluciones utilizan cámaras digitales para guardar la fotografía del usuario con un determinado fin. Para lograr la comunicación con las cámaras digitales desde una aplicación lo más fácil sería contar con el SDK (Software Development Kit o Paquete de Desarrollo de Software de sus siglas en inglés) del fabricante, pero es muy difícil encontrarlo de forma gratuita. Últimamente han aparecido en los sistemas operativos tecnologías genéricas, mediante la cual una aplicación puede comunicarse con múltiples tipos de hardware de adquisición de imágenes.

Actualmente en Cuba se utilizan cámaras digitales, ya sea en calles y en los distintos puntos de inspección de las carreteras, tiendas, hoteles, bancos entre otros, para tener un mejor control de la seguridad nacional.

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el centro de producción Geoinformática y Señales Digitales (GEySED) donde el objetivo fundamental es la comunicación con cámaras digitales de vigilancia y también el Centro de Identificación y Seguridad Digital (CISED), en el cual está ubicado el Departamento Biometría, donde se necesita desarrollar componentes de software para la interacción con diferentes dispositivos de captura para la obtención de muestras biométricas. En el CISED no existe un componente de software genérico de captura de imágenes faciales, que permita la interacción con múltiples modelos de cámaras digitales. En el mismo se ha hecho un esfuerzo para integrar múltiples tipos de dispositivos a sus aplicaciones, mediante la creación de una arquitectura de integración para los dispositivos, que permite que las aplicaciones tengan una interfaz genérica de acceso a estos, siempre y cuando se le implementen los plugins o componentes adecuados. Los plugins disponibles para la captura de imágenes faciales, han sido implementados usando SDKs y no tecnologías genéricas. También se han comprado componentes de software de captura de algunos fabricantes que incluyen hasta una llave de

hardware (dispositivo que se conecta por USB para autorizar al software a correr), con licencias muy caras y restrictivas.

En el proceso de desarrollo del componente de software para la captura de imágenes faciales, compatible con múltiples modelos de cámaras digitales, se identifica como **problemática**: Las soluciones de identificación que se desarrollan en el CISED soportan una cantidad limitada de modelos de cámaras digitales, restringiéndole la capacidad de comercialización a las entidades que se encargan de comprar los dispositivos, muchas veces eligiendo las cámaras digitales más caras o más calidad de la necesaria, incluyendo en más gastos para el CISED y el país, haciendo menos rentables los proyectos.

Para la realización de la investigación se hace necesario plantear el siguiente **problema científico**: ¿Cómo lograr que las soluciones de identificación de personas desarrolladas en el CISED interactúen con múltiples modelos de cámaras digitales?

El **objeto de estudio** para el presente trabajo queda enmarcado en: Componentes de software para la captura de imágenes faciales.

El **campo de acción** queda delimitado en: Componentes de software para la captura de imágenes faciales mediante el uso de tecnologías genéricas.

Objetivo general:

Realizar el análisis y diseño de un componente de software que utilice tecnologías genéricas para la captura de imágenes faciales con múltiples modelos de cámaras digitales, que se ajuste a la arquitectura de integración de dispositivos para identificación de personas existente en el CISED.

Se define como **idea a defender**: El desarrollo del análisis y diseño del componente de software para la captura de imágenes faciales basado en tecnologías genéricas, que se ajuste a la arquitectura de integración de dispositivos para la identificación de personas existente en el CISED, permitirá sentar la base para la implementación del componente de software compatible con múltiples modelos de cámaras digitales.

Tareas investigativas:

- 1- Sistematización de los fundamentos teóricos, referenciales acerca de componentes de software para la captura de imágenes faciales con múltiples modelos de cámaras digitales.
- 2- Estudio y fundamentación de las tecnologías a utilizar.
- 3- Caracterización de las tecnologías genéricas para la captura de imágenes, sin dejar de incluir:
 - Tecnología Windows Image Acquisition (WIA).

- Tecnología Streaming.
- Tecnologías vídeo mediante el uso de VideoLAN cliente (VLC).

4- Caracterización de la arquitectura de integración de dispositivos del CISED, haciendo énfasis en su posibilidad de extensión mediante la implementación de soporte para nuevos dispositivos.

5- Captación, definición y validación de los requerimientos del software.

6- Realización del análisis y diseño del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales.

7- Realización de un estudio de factibilidad y análisis de costos.

Métodos empleados en la investigación.

Métodos teóricos:

Análisis-Síntesis: Se consultan y se analizan las bibliografías referentes al desarrollo de componentes de software para la captura de imágenes faciales, y se hace uso de la información más adecuada para la presente investigación.

Modelación: Consiste en la representación del proceso que se estudia, ya que a través del modelado se puede obtener un entendimiento superior de la lógica del proceso a automatizar.

Métodos empíricos:

Entrevista: Se realiza a directivos del proyecto "Sistema de Vídeo Vigilancia" del centro GEySED, y a directivos y trabajadores del CISED con el objetivo de recopilar toda información relacionada con el sistema, lo que contribuyó al desarrollo del análisis y diseño del mismo.

Posibles resultados: Con la presente investigación se obtendrá como resultado, el análisis y diseño del componente de software para la captura de imágenes faciales, que se ajuste a la arquitectura de integración de dispositivos existentes en el CISED, el cual brindará soporte para la interacción con múltiples modelos de cámaras digitales, que aumentará la flexibilidad en la adquisición de las mismas, para la integración de soluciones de identificación.

El contenido de la investigación se estructura de la siguiente forma:

Capítulo 1 "Fundamentación teórica": Contiene la fundamentación teórica de la propuesta presentada, un estudio de las tecnologías relacionadas con los dispositivos de captura y las tecnologías genéricas de acceso a los mismos, de las cuales se hace una descripción, se exponen las ventajas, desventajas, limitantes, y se fundamenta la selección de cuál de estas se va a utilizar. Se realiza una caracterización de la arquitectura de integración de dispositivos existente en el CISED. Además se describen los aspectos

relacionados con el desarrollo de componentes de software, las tecnologías, la metodología de desarrollo de software, las herramientas, los lenguajes que se emplean y por qué de su utilización. Se hace un estudio de la arquitectura de software y sus estilos, y de los patrones de diseño.

Capítulo 2 “Características del sistema”: Se describe el negocio a través de un modelo de dominio y a partir de este, se comienza la descripción del sistema a desarrollar. Se realiza una especificación de los requisitos funcionales y no funcionales, que debe tener el componente. Se describen los actores y los casos de uso, haciéndose una descripción textual detallada de estos últimos y se muestra el diagrama de casos de uso.

Capítulo 3 “Análisis y diseño”: Se describe la arquitectura de software que se emplea. Se modelan los diagramas de clases de análisis, colaboración del análisis, clases del diseño y los diagramas de secuencia del diseño. Además se exponen los patrones que se utilizan y se muestra el diagrama de despliegue.

Capítulo 4 “Estudio de factibilidad y análisis de costos”: Se realiza un estudio de la factibilidad donde se emplea la estimación mediante el análisis por el método “Puntos de Casos de Uso”. Se realiza un análisis de los costos y beneficios, demostrándose si es factible o no el desarrollo de la investigación.

Capítulo 1: Fundamentación teórica

1.1 Introducción

En la actualidad los sistemas informáticos se han convertido en un aspecto fundamental del desarrollo en el mundo, por lo que se hace necesario el conocimiento de las tecnologías, metodologías y herramientas a la hora de hacer aplicaciones informáticas. El presente capítulo brinda una visión general de los aspectos relacionados con el proceso de desarrollo de componentes de software para la captura de imágenes faciales, con soporte para múltiples modelos de cámaras digitales. Se realiza un estudio de la arquitectura de integración de dispositivos existente en el CISED identificándose sus características. Se aborda acerca de tecnologías usadas en la actualidad y las usadas en el CISED, describiéndose las ventajas, desventajas y limitantes entre otras particularidades. Además se estudian los estilos arquitectónicos y los patrones de diseño.

1.2 Conceptos asociados a los dispositivos de captura

En el presente epígrafe se exponen algunos conceptos relacionados con los dispositivos de captura, que ayudan a entender el desarrollo de la investigación.

1.2.1 Vídeo

El vídeo se puede definir como aquel medio audiovisual destinado a la manipulación, registro y reproducción de sonidos e imágenes por procedimientos electromagnéticos de forma sincrónica y simultánea.(1)

Un vídeo digital es aquel que ha sido capturado mediante procedimientos y dispositivos informáticos, está expresado mediante cadenas de ceros y unos, que luego se procesarán y codificarán. En la actualidad existen muchos modelos de cámaras fotográficas digitales y de teléfonos móviles, que permiten la captura de vídeo digital a través de una cámara integrada, disponiendo de un software específico para almacenar los vídeos capturados en un ordenador.

La captura en vivo es la parte más importante que tienen los dispositivos de captura de imágenes ya que de este proceso depende la calidad de las mismas a la hora de tomarse la fotografía.

1.2.2 Dispositivos de captura

Llamamos dispositivos de captura a todos aquellos que nos permiten de una u otra forma introducir información en el ordenador. También se los suele llamar dispositivos digitalizadores, ya que convierten cierta información (luz, imagen, sonido, etc.) en información digital (en definitiva, en números binarios). (2) Entre los dispositivos de captura se encuentra el escáner, el cual convierte imágenes y documentos en información que puede interpretar una computadora.

La cámara digital es otro de los dispositivos de captura y que se aborda en el presente documento. Es un dispositivo muy versátil y de alta calidad.

La fotografía digital se basa fundamentalmente en el almacenamiento de las imágenes mediante sistemas o soportes que permiten el volcado de las fotos a un ordenador o proyector sin necesidad de recurrir al carrete tradicional. La gran ventaja estriba en la disponibilidad inmediata de las imágenes sin necesidad de un revelado. La fotografía digital se ha extendido hasta los teléfonos móviles. Hoy en día se encuentran en el mercado este tipo de dispositivos que incorporan una cámara digital. (3)

Las cámaras web (webcam) son los dispositivos de captura digital de vídeo de gama baja, que suelen estar destinados a las videoconferencias a través de Internet. También está la cámara de red (cámara IP) es una cámara que emite imágenes directamente a otros dispositivos que están conectados en red (Intranet o Internet), sin necesidad de un ordenador, una cámara de red incorpora su propio miniordenador, lo que le permite realizar una transmisión de vídeo por sí misma.

Los dispositivos de captura poseen características entre las cuales se encuentran:

Resolución

La resolución es la capacidad que tiene una cámara digital de capturar los detalles más pequeños. Mientras exista más definición en la foto entonces mayor es la resolución. Es el número total de píxeles que una cámara puede captar, se expresa en megapíxeles y depende de la potencia resolutive del objetivo.

De ella depende la calidad de la foto a la hora de visualizarla en la pantalla, por eso la importancia de contar con una cámara digital que tenga como mínimo 2 megapíxeles.

Conectividad

Múltiples son los sistemas para volcar imágenes hacia la computadora. La conexión mediante cable al puerto USB es la más usada, aunque algunas cámaras utilizan un puerto FireWire o Bluetooth.

Últimamente han aparecido algunos modelos que pueden conectarse a la computadora, vía red inalámbrica por el protocolo 802.11 (Wi-Fi).

Flash y zoom

Muchas veces es necesaria una fuente de iluminación para escenas de fotografías y es cuando generalmente se usa el flash, que es una fuente luz intensa. Generalmente las unidades de flash están constituidas por lámparas de xenón.

La mayoría de las cámaras digitales actuales cuentan con zoom óptico, que permite modificar la distancia focal regulando la ampliación de la imagen. El zoom digital suele ser un recorte de fotografía, se eliminan los márgenes agrandando la parte central de la imagen, en este proceso se usan menos píxeles por lo que la imagen va perdiendo calidad.

1.2.3 Procedimiento para la captura

Para obtener una buena imagen facial es necesario tener en cuenta varios aspectos, como son la iluminación, la posición de la persona, la distancia de la cámara digital a la persona, el color y ángulo del fondo entre otros, ya que la calidad de la muestra es fundamental pues de ahí depende que una imagen pueda ser procesada o no. En la presente investigación se estudian las directrices del Documento de viaje de lectura mecánica (la tercera edición 2008) de la Organización de Aviación Civil Internacional (4), de donde se extraen los pasos para conseguir una buena muestra biométrica, las mismas se encuentran en el [Anexo 1](#). También se estudia el Procedimiento para la captura de elementos biométricos que usa la Autoridad Certificadora (AC) que integran la ITFEA¹ (5), la cual se encuentra en el [Anexo 2](#) y el Proyecto de fin de carrera de César Reyes López, “Reconocimiento facial mediante visión artificial” (6) la cual se puede apreciar en el [Anexo 3](#). Tomándose como referencia las directrices del Documento de viaje de lectura mecánica, ya que propone una orientación para conseguir un buen retrato y que en nuestro caso pueda procesarse posteriormente.

1.2.4 Tecnologías que se usan en la captura y transmisión de vídeo

Según Pavón e Hidalgo la tecnología es: «El conjunto de conocimientos e información propios de una actividad que pueden ser utilizados en forma sistemática para el diseño, desarrollo, fabricación y

¹ Infraestructura tecnológica de firma electrónica avanzada.

comercialización de productos, o la prestación de servicios, incluyendo la aplicación adecuada de las técnicas asociadas a la gestión global.»(7)

En la actualidad existen múltiples tecnologías que se utilizan en la captura y transmisión de vídeo, entre ellas se encuentran:

Streaming

El streaming se refiere a una corriente continua, es decir sin interrupción. Esta tecnología se utiliza para aligerar la descarga y ejecución de audio y vídeo.

Definición de streaming(8):

- Tecnología de transmisión a través de redes, de medios continuos (principalmente audio y vídeo).
- No existe descarga de información a un disco local.
- Se envía la información a través de la red y el cliente la reproduce en tiempo real al recibirla.

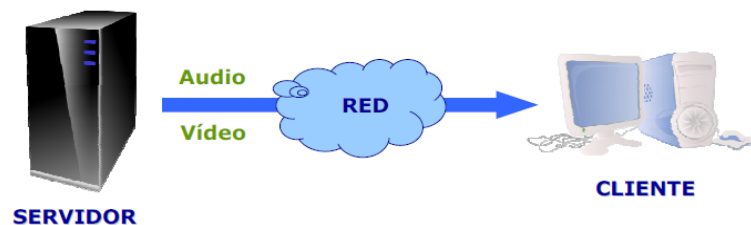


Figura 1. Streaming.

El funcionamiento del streaming es de la siguiente forma: El cliente se conecta con el servidor y este comienza a mandarle el fichero y mientras lo va recibiendo va construyendo un buffer² donde va guardando la información. Cuando el buffer ya contenga una parte del archivo, el cliente comienza a mostrarlo y a su vez continúa con la descarga. Si se sufren descensos de velocidad se utiliza la información que hay en el buffer y si el caso es que se cortara por un tiempo extendido, el buffer se vacía y la ejecución del archivo se corta hasta que se restaure la señal.

Una vez establecida la conexión con la cámara digital es necesario visualizar en la aplicación lo que está tomando el lente de la cámara, una de las formas de lograr este objetivo es por medio del streaming, para luego tomar la foto. Esta es una de las tecnologías más usadas en las cámaras web.

² La ubicación de la memoria de un dispositivo digital o una computadora que está reservada para el almacenamiento temporal de información.

Direct X

Microsoft DirectX es un conjunto avanzado de interfaces de programación de aplicaciones multimedia (API, por su sigla en inglés), desarrollado para sistemas operativos Microsoft Windows.

DirectX proporciona una plataforma estándar de desarrollo permitiendo a los programadores tener acceso a características específicas del hardware sin tener que escribir código de bajo nivel. Bajo Windows, si no existiera DirectX, los desarrolladores de software multimedia y juegos tendrían que incluir en sus productos drivers para cientos de tarjetas de vídeo, chips de sonido, joysticks, etc. Al utilizar DirectX, este proceso resulta más sencillo: ellos hacen que sus programas interactúen con DirectX, que hace de intermediario transmitiendo las órdenes al hardware del PC.(9)

Es un grupo de tecnologías que fueron diseñadas para convertir los equipos en plataformas ideales para la ejecución y visualización de aplicaciones con elementos multimedia como, gráficos, vídeo, animación 3D, audio.

VLC

El VideoLan es una solución de software para el streaming de vídeo, que fue creada por estudiantes del Ecole Centra de Paris (Francia) y bajo la licencia GNU puede ser desarrollado por todas las personas del mundo que deseen o necesiten hacerlo. Vídeo LAN presenta dos tipos de soluciones(10):

- VLS(VideoLAN servidor): En teoría es capaz de hacer el stream de archivos MPEG-1, MPEG-2 y MPEG-4 (ubicados en cualquier disco rígido o removible accesible), DVDs, canales digitales satelitales y canales digitales de televisión terrestre(que sean obtenidos por tarjetas DVB-S o DVB-T) y vídeos en vivo sobre redes (que sean obtenidos por una tarjeta de codificación MPEG). Trabaja bajo Linux o Windows.
- VLC(VideoLAN cliente): Este puede utilizarse al igual que el VLS como servidor para hacer el stream de archivos MPEG-1, MPEG-2 y MPEG-4, DVDs y vídeo en vivo sobre redes (Notar que comparado con el servidor, solo le faltan las posibilidades de streaming de canales digitales de satélite y canales de televisión terrestres digital). También puede ser usado como cliente para recibir, decodificar y mostrar stream de MPEG bajo múltiples sistemas operativos (Linux, Windows, Mac OS X, BeOS, *BSD, Solaris, Familiar Linux, Yopy/Linupy y QNX).

VideoLan Client es un reproductor que soporta muchos códecs de audio y vídeo, varios protocolos de streaming y diferentes formatos de archivos.

1.3 Acceso a dispositivos

Para interactuar y manejar el contenido de los dispositivos es necesario acceder a estos mediante diferentes tecnologías ya sean genéricas o específicas.

1.3.1 Tecnologías que se usan en el acceso a los dispositivos

Para desarrollar un componente para la captura de imágenes digitales se requiere el envío y captura de información entre las cámaras digitales y la computadora. Para este fin se utilizan tecnologías que facilitan el proceso.

En el mundo del acceso a dispositivos, el fabricante del hardware generalmente hace el software para este componente en específico, aunque últimamente suele suceder que se guían por estándares y los hacen con tecnologías genéricas (más adelante se explica el concepto), para tener compatibilidad con los demás fabricantes. Ya que ellos no saben qué uso se le va dar ni qué tipo de tecnologías van a usar, suelen brindar este software en tipo de componente, en alguna tecnología bien conocida como COM o API, que se explican a continuación.

COM

COM (Component Object Model) es una plataforma desarrollada por Microsoft para utilizarla, principalmente, en las distintas versiones del sistema operativo Windows. Existen implementaciones para otras plataformas. Se basa en crear componentes reutilizables y crear aplicaciones con esos componentes. Para ello proporciona servicios de comunicación entre procesos y creación dinámica de objetos desde cualquier lenguaje de programación que soporte esta tecnología. Una ventaja fundamental que ofrece esta tecnología es que los objetos están ya implementados en código binario, por lo que no es necesario disponer del código fuente original. Esta característica permite que se puedan integrar componentes escritos en distintos lenguajes, modificar las aplicaciones sin necesidad de recompilar los objetos, e incluso reubicarlos en otra máquina distinta.(11)

Pero trae el problema de que se hace más dependiente de la plataforma de destino a que se compiló.

API

Las API de Windows son bibliotecas de vínculos dinámicos (DLL) que forman parte del sistema operativo Windows. Se utilizan para realizar tareas cuando resulta difícil escribir procedimientos equivalentes.

La ventaja de utilizar las API de Windows en el código es que pueden ahorrar tiempo porque contienen numerosas funciones útiles ya escritas y listas para utilizar. La desventaja es que puede resultar difícil trabajar con las API de Windows y pueden ser implacables cuando las cosas van mal. (12)

Otras de las ventajas que obtenemos con la utilización de API es la gran funcionalidad que podemos darle a nuestra aplicación, y en segundo lugar la gran velocidad de proceso, ya que a menudo es mucho más rápido realizar una función a través de la API adecuada que por medio del lenguaje en sí mismo.

Tecnologías genéricas

Las tecnologías genéricas son las que están conformadas por los conocimientos técnicos comunes a todas las empresas industriales. Son accesibles para todos los que deseen utilizarlas y que tengan los conocimientos necesarios para comprenderlas, las mismas no pueden ser patentadas.

Uno de los tipos de periféricos que más auge ha adquirido con la generalización de la tecnología multimedia es el de adquisición de imágenes. Sin embargo, para que desde cualquier aplicación se pueda controlar este tipo de dispositivos, es necesario contar con un sistema de acceso común al hardware. Todos los dispositivos de captura de imágenes tienen peculiaridades propias, lo cual hace necesario que cada uno de ellos incluya un software específico para la transferencia al ordenador de las imágenes adquiridas mediante este hardware. En los últimos años ha existido un esfuerzo importante de estandarización de los dispositivos, lo que permite usar un mismo controlador para dispositivos de distintos fabricantes. Se hace evidente la necesidad de contar con un protocolo o con una API, mediante la cual una aplicación pudiese comunicarse con cualquier tipo de hardware de adquisición de imágenes. Esta necesidad es cubierta por las tecnologías TWAIN y WIA (las cuales se explican seguidamente).

TWAIN

TWAIN es un protocolo de interfaz multiplataforma que se utiliza para adquirir imágenes electrónicas que han sido capturadas por escáneres, cámaras digitales y paneles de captura de vídeo de marco.(13)

Es un protocolo de software y una API, que regula la comunicación entre aplicaciones del software y dispositivos de adquisición de imágenes tales como los escáneres y las cámaras digitales. Se diseñó para proveer una consistente y fácil integración de datos entre dispositivos de entrada sofisticados y aplicaciones de software. Es independiente del sistema operativo, existen implementaciones para varias plataformas y tecnologías de desarrollo y provee una interfaz que está bien definida.

WIA

Adquisición de Imágenes de Windows (WIA de sus siglas en inglés Windows Image Acquisition) es un modelo de driver y una API para los sistemas operativos más modernos de Microsoft Windows, que permite a las aplicaciones de gráficos comunicarse con dispositivos de imagen tales como escáneres, cámaras digitales y equipos de vídeo digital. Se implementa como un servicio bajo-demanda en Windows XP y posteriores sistemas operativos Windows. WIA proporciona un entorno de trabajo a través del cual los dispositivos pueden presentar sus capacidades únicas al sistema operativo y las aplicaciones pueden tomar ventaja de esas características.



Figura 2. Gráfico que muestra la arquitectura básica de WIA como una vía de dos capas entre las aplicaciones de imágenes y dispositivos.

La plataforma de WIA incluye un protocolo de adquisición de datos, un controlador de dispositivo y modelo de interfaz (DDI), un API y un dedicado servicio de WIA. La plataforma también incluye un conjunto de controladores incorporados en modo kernel que dan soporte a la comunicación con dispositivos de imágenes conectadas localmente a través de USB, serie / paralelo, interfaces SCSI³ y FireWire.(14)

³ De sus siglas en inglés Small Computer System Interface (Interfaz para sistemas de ordenadores pequeños).

"La arquitectura de WIA es un paso en la dirección correcta en proveer una mejor interfaz por la comunidad de la imagen digital. Intel también ve a WIA como una manera de hacer imágenes digitales más fácil de usar por los consumidores." (15)

Con el fin de dar soporte a aplicaciones TWAIN con capacidades privadas, WIA puede utilizar una técnica conocida como de paso a través de la funcionalidad. El mecanismo de transmisión se refiere a la forma en que una aplicación compatible TWAIN, se comunica con un driver de WIA, usando el gestor de origen de datos y la capa de compatibilidad con TWAIN como intermediario. Es importante tener en cuenta que la capacidad de TWAIN de paso a través de la funcionalidad se admite en Windows XP y versiones posteriores del sistema operativo. Si el controlador WIA no requiere el apoyo de TWAIN, usted no debe agregar esta funcionalidad a su driver. (16)

Otra de las tecnologías genéricas que se usan en la adquisición de imágenes es Still Image.

Still Image

Still Image es una API que constituye una interfaz estandarizada, que se utiliza principalmente para capturar imágenes fijas de dispositivos de captura de imágenes digitales. El soporte para este tipo de dispositivos para la captura de imágenes fijas se proporciona a través de proveedores individuales de hardware independiente (IHV). Las cámaras digitales usualmente soportan la clase de imágenes fijas que utiliza el Picture Transfer Protocol (PTP) o Media Transfer Protocol (MTP) para transferir imágenes a un ordenador.

La incorporación de imágenes fijas en las aplicaciones tiene las siguientes ventajas(17):

- Estandariza las interfaces de dispositivo: las API de los dispositivos de control de imagen fija son extremadamente independiente del dispositivo. Esto libera a los desarrolladores de aplicaciones de tener que escribir código especializado para todos los dispositivos que soportan.
- Simplifica el modelo de usuario: aplicaciones especializadas tienen una interfaz de usuario más intuitiva.
- No requiere un cambio en la arquitectura de la aplicación: Para mejorar las aplicaciones que utilizan imágenes fijas sólo se requieren cambios mínimos en los programas existentes. En la gran mayoría de los casos, las arquitecturas actuales de software, no cambia con la conversión STI.

La propuesta de solución de este trabajo, debe considerar cada una de estas tecnologías genéricas expuestas, así como de las tecnologías específicas que se enumeran a continuación.

Tecnología específica

Las tecnologías específicas como son los SDK y los drivers es otra vía para poder interactuar con el dispositivo de captura las cuales se explican a continuación.

SDK

Un SDK es un conjunto de herramientas y programas de desarrollo para crear un determinado software, plataforma de hardware, sistema operativo, dispositivo, entre otros. Los mismos contienen ejemplos y documentación.

El SDK puede estar atado a licencias que los hacen impropios para construir el software, ya que no pueden ser desarrolladas bajo una licencia incompatible. Por ejemplo, un SDK propietario probablemente será incompatible con el desarrollo del software libre y viceversa. En muchos casos es necesario pedir el SDK al productor del dispositivo para hacer alguna modificación y comúnmente hay que pagar por la obtención del mismo. El mismo es creado por las empresas que fabrican los dispositivos. Además de cumplir con las funciones principales que se necesitan para controlarlos, incluyen funciones específicas para una marca y modelo específico y no son compatibles con otras marcas. La ventaja de usarlos es que al ser elaborado por la misma empresa que lo fabrica es 100% funcional, dando una mayor velocidad y se aprovechan al máximo las funciones adicionales con las que suelen venir estos dispositivos por defecto. Aunque muchas veces las funcionalidades de los dispositivos es similar y no requeriría de tantas soluciones específicas.

Para contrarrestar algunas desventajas de los SDK como por ejemplo pagar por su obtención o la comunicación con su fabricante se pueden emplear las tecnologías genéricas ya que permiten la comunicación con distintas marcas y modelos.

Driver

Un driver o controlador posibilita que el sistema operativo de una computadora pueda entenderse con un equipamiento periférico, como es el caso de una impresora, una placa de vídeo, un mouse, un módem, etc. (18)

Un periférico puede tener varios controladores, ya sea porque son diferentes versiones, porque tienen diferentes funcionalidades o porque están hechos para sistemas operativos específicos.

En los SDKs generalmente vienen distribuidos los drivers del dispositivo y en las tecnologías genéricas vienen incluida una colección de drivers para dispositivo en el sistema operativo, o algún que otro driver genérico para una gran serie de dispositivos.

1.4 Sistemas similares en el mundo que informatizan la captura de imágenes faciales

En el mundo existe una gran cantidad de software que interactúa con algún tipo de dispositivo de captura de imágenes desde una computadora, cada uno con determinadas características y condiciones. No se encontraron referencias de alguno que ofreciera el código, y la mayoría están sujetos a un SDK, otra desventaja de estos software, es que todos son aplicaciones que están hechas para ser usadas por usuarios y no para integrarse en el desarrollo de aplicaciones como son los componentes.

Ejemplos de algunos de ellos son:

Cam2Com: Permite controlar de forma remoto las cámaras digitales (Olympus o Canon) desde la computadora. Controla los parámetros de la toma y puede controlar varias cámaras simultáneamente.

Canon Camera PTP WIA Driver for Windows Me: Este software le permite a la computadora comunicarse con la cámara digital cuando el modo de comunicación de la cámara esta en Imprimir / PTP⁴.

NIKON CAMERA CONTROL PPPRO 2: Permite tomar la foto y cambiar todos los parámetros de la cámara.

PSRemote: Permite un alto grado de control de la cámara Canon PowerShot desde una computadora, y la visualización en vivo.

WxAstroCapture: Este programa permite realizar capturas de imágenes con webcam. Puede usarse para realizar autoguiado en tomas de larga exposición. Permite abrir varias instancias para fotografiar y guiar al mismo tiempo.

Selene: Es un programa para utilizar una webcam, para la captura de Vídeo e imágenes.

Astrosnap Pro 2.1: Programa para realizar capturas de imágenes con webcam. Dispone de autoenfoco, tratamiento automático de Flats y Darks, grabación en BMP y JPG.

CyberLink YouCam 4.0: Este programa graba secuencias de vídeo y captura imágenes. Permite trabajar en tiempo real o con grabaciones de vídeo. El mismo incluye un catálogo muy variado de filtros y efectos.

El **XLS2** es un componente utilizado en las soluciones exportables del centro CISED, que automatiza el proceso de captura de imágenes faciales, huellas y firmas. Realiza detección y recorte de rostro bajo estándares de la OACI⁵ con ajustes de iluminación y otras técnicas de post-procesamiento. Lo provee la

⁴ De sus siglas en inglés (Picture Transfer Protocol).

⁵ Organización de Aviación Civil Internacional.

compañía francesa SAGEM⁶, y depende de una licencia y llave de hardware que lo hacen dependiente a un solo modelo de dispositivo. Cuesta alrededor de 100.00 USD por estación. Se utiliza como un componente aislado de la arquitectura de dispositivos del CISED, lo cual representa otra limitante.

En Cuba no se han encontrado referencias acerca de alguna empresa que se dedique al desarrollo de componentes para la comunicación con cámaras digitales. En la Universidad de las Ciencias Informáticas, el Centro GEySED, se comunica con cámaras digitales de vigilancia (cámaras IP). También en la UCI se desarrolló el Sistema Autónomo de Identificación, Migración y Extranjería (SAIME) implantado en Venezuela, el cual se comunica con la cámara digital PowerShoot A520 mediante su SDK, también en el CISED, específicamente el Departamento de Identidad Cuba se cuenta con una aplicación que interactúa a través del SDK con la cámara Canon SX110 IS. En la presente investigación se tomará como modelo de estudio la aplicación existente en el Departamento de Identidad Cuba y la aplicación Cam2Com, esta última interactúa con cámaras Olympus y Canon (también utilizando SDK).

Se desea realizar el análisis y diseño de un componente que no requiera el uso de un SDK, ya que el mismo constituye un impedimento pues es muy difícil encontrarlo de forma gratuita, sino que utilizando tecnologías genéricas, se pueda comunicar con distintos modelos de cámara digital que soporte dicha tecnología.

1.5 Caracterización de la plataforma de integración de dispositivos existente en el CISED

El CISED cuenta con la existencia de una plataforma para integrar el acceso a dispositivos que permite la comunicación con dispositivos de captura de firmas, huellas, imágenes, acceso a tarjetas inteligentes y con lectores de página completa.

La misma presenta las siguientes características: integrable, extensible, configurable y de interfaz amigable. Brinda la posibilidad de integrar cualquier aplicación o módulo de forma fácil y segura. Permite la incorporación de nuevos dispositivos en la medida que vaya surgiendo la necesidad de nuevos módulos o la comunicación con otros equipos con iguales características pero de diferentes fabricantes, sin hacer cambios en la aplicación. Posibilita que a través de configuraciones se definan los dispositivos habilitados para la captura, para que no sea necesario recompilar la aplicación en el caso que se precise la utilización

⁶ Compañía de Aplicaciones Generales de Electricidad y Mecánica.

de otro dispositivo, para la realización de las mismas operaciones de captura. Provee los componentes gráficos necesarios en caso de que el dispositivo de captura requiera de una retroalimentación gráfica. La plataforma de acceso a los dispositivos fue desarrollada en Microsoft .Net Framework v1.1.4, con C# como lenguaje y en el entorno de desarrollo Microsoft Visual Studio .Net 2003. El mismo está diseñado para que la estructura básica sea común para cada tipo de dispositivo. Presenta un conjunto de interfaces que definen el comportamiento general de todos los dispositivos con respecto a la funcionalidad, propiedades y eventos de notificación, luego están definidas un conjunto más grande de interfaces que implementan las anteriores y agregan funcionalidades propias en dependencia del dispositivo de captura. Para el diseño de la plataforma para integrar el acceso a dispositivos, la arquitectura que se utilizó fue en capas, la cual comprende tres de estas: una que es la de interfaces genéricas para la definición de las capacidades de cada tipo de dispositivo, la capa de implementación de los wrapper⁷ para cada dispositivo y la capa de interacción con cada SDK o API. En el mismo cada wrapper puede verse como un componente.

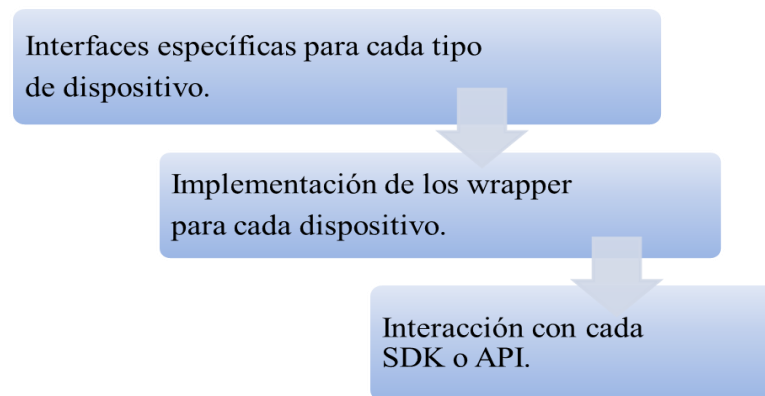


Figura 3. Arquitectura en capas de la plataforma para el acceso a dispositivos.

Seguidamente se muestra el diagrama de clases de dicha plataforma.

⁷ Conjunto de clases que controla el acceso a otras clases.

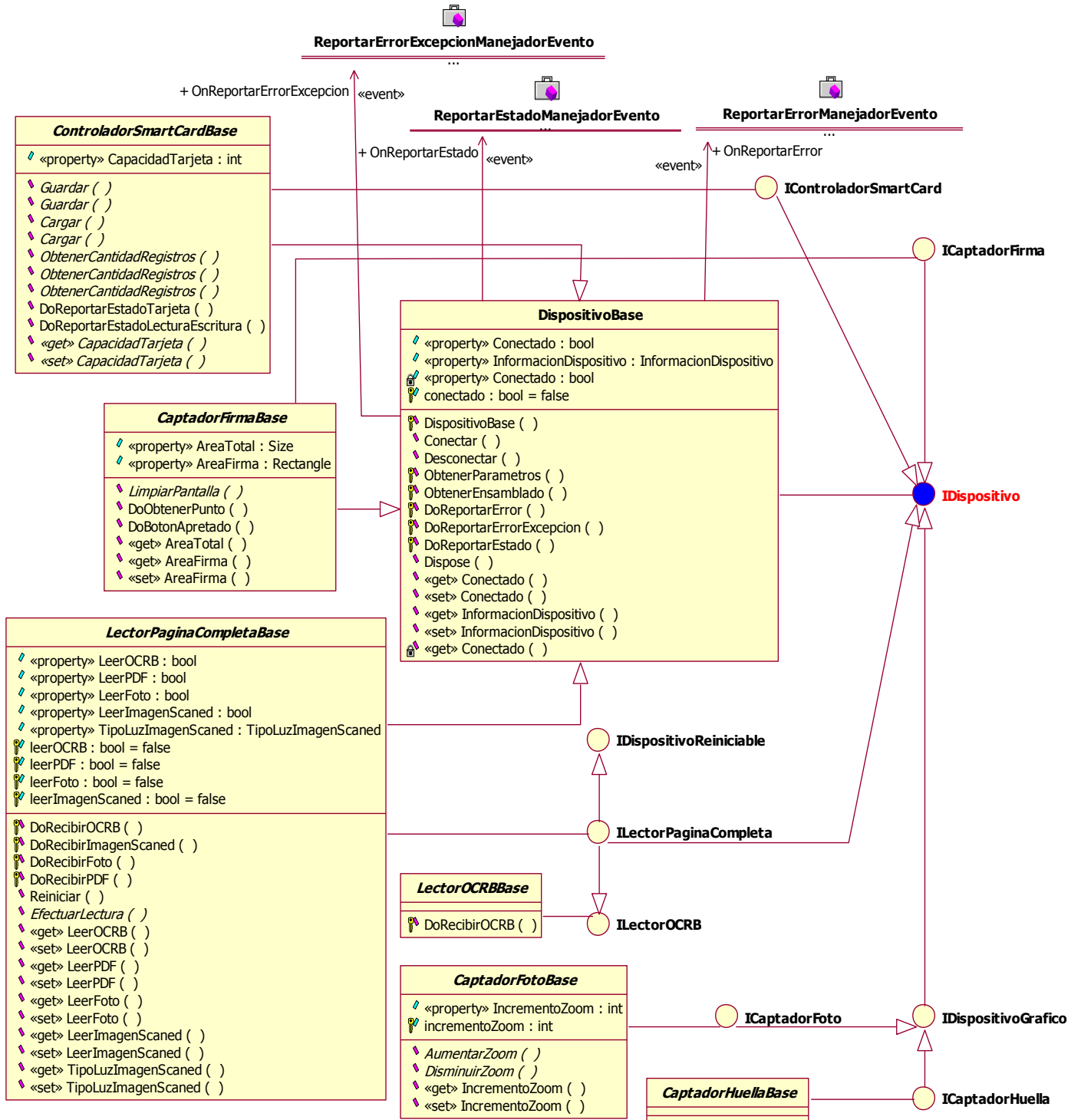


Figura 4. Arquitectura de la plataforma de integración de dispositivos existente en el CISED.

1.6 Desarrollo de componentes

Aunque existen varias definiciones de componente de software en la presente investigación se utiliza la definición de Szyperski, que plantea que el mismo es: Un componente de software es una unidad de composición con interfaces contractualmente especificadas y explícitas sólo con dependencias dentro de un contexto. Un componente de software puede ser desplegado independientemente y es sujeto a la composición de terceros. (19)

Para que un elemento pueda ser clasificado como un componente debe cumplir con determinadas características como que sea identificable, que pueda ser remplazado por otro componente, con acceso solamente a través de su interfaz, sus servicios no varíen, bien documentado, genérico, reutilizado dinámicamente e independiente de plataforma.

En el presente epígrafe se describen las tecnologías para el desarrollo de componentes de software. Se exponen las características más relevantes, con la finalidad de ayudar al entendimiento de porqué su selección, y para realizar la selección de las que se usan en la investigación y futuro desarrollo del componente.

1.6.1 Metodología

Para obtener un producto que cumpla con todas las especificaciones y esté en el tiempo estimado, se necesita una metodología que guíe el proceso de desarrollo de software. Una metodología es la que define *quién debe hacer qué, cuándo y cómo*. Es un proceso donde se realizan tareas para obtener un producto a través de un conjunto de actividades definidas previamente.

Una metodología de desarrollo de software permite producir organizada y económicamente software de alta calidad, siguiendo una serie de pasos donde se utilizan un conjunto de técnicas, notación y normas de documentación preestablecidas. En la actualidad no existe una metodología universal que se le pueda aplicar a todos los proyectos y su selección depende de las características de cada proyecto. A continuación se hace un breve resumen de las características fundamentales de dos de las metodologías más utilizadas actualmente: la metodología Proceso Unificado de Rational (RUP) y la metodología Extreme Programming (XP).

RUP

Su propósito es asegurar la producción de software con una alta calidad que se ajuste a las necesidades de sus usuarios finales. Reconoce como un hecho que los requerimientos cambian en el desarrollo del

proyecto y define un enfoque flexible para controlar estos. Tratar de definir los requerimientos de forma total al inicio del proyecto, es una decisión muy riesgosa por lo que el enfoque adoptado en esta metodología asegura un control eficaz del riesgo del alcance, además se genera una gran cantidad de artefactos, por lo que conlleva bastante documentación.

En RUP las actividades se realizan en a través de 9 flujos de trabajos y 4 fases. Ver figura 5.

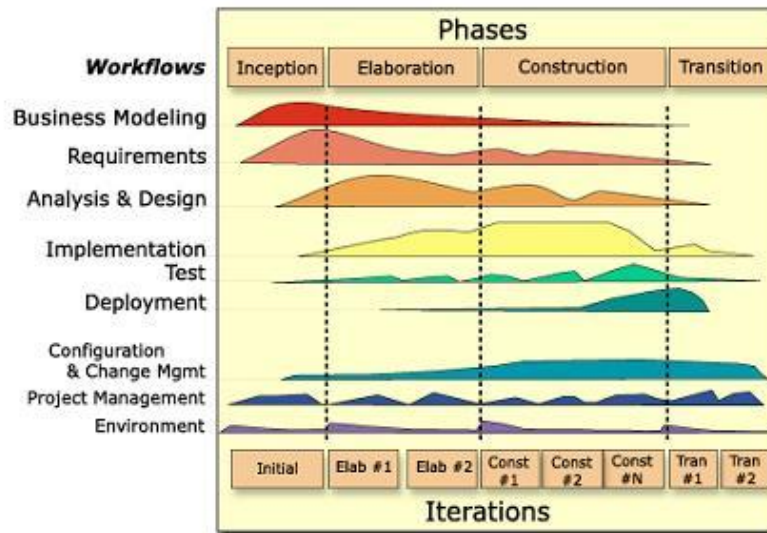


Figura 5. Fases e iteraciones de la metodología RUP.

Las fases son(20):

Fase de concepción: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones.

Fase de elaboración: Los casos de uso seleccionados para desarrollarse en esta fase permiten definir la arquitectura del sistema, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar del problema y comienza la ejecución del plan de manejo de riesgos, según las prioridades definidas en él.

Fase de construcción: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requerimientos pendientes, administrar el cambio de los artefactos construidos, ejecutar el plan de administración de recursos y mejoras en el proceso de desarrollo para el proyecto.

Fase de transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto al inicio del mismo.

Las características fundamentales de RUP son:

- Dirigido por casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

XP

Extreme Programming es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer. Es capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto. La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. El manejo del cambio se convierte en parte sustantiva del proceso, el código fuente pertenece a todos, solo emplea un mínimo de organización y con poca documentación, aumentando la productividad.

La programación extrema tiene dos objetivos: la satisfacción del cliente y potenciar al máximo el trabajo en grupo. Surge ideada por Kent Beck, como proceso de creación de software diferente al convencional. En palabras de Beck: «Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.» (21)

Las fases de la metodología XP son: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

La metodología de desarrollo que se emplea es RUP ya que dentro de sus características se encuentra que genera mucha documentación, que es precisamente lo necesario para la posterior implementación del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales.

1.6.2 Herramientas CASE

Hoy es impensable desarrollar software sin utilizar un proceso soportado por herramientas. Algunas de estas herramientas se dirigen principalmente a mejorar la calidad, como es el caso de las herramientas CASE (Computer Aided Software Engineering, en español Ingeniería de Software Asistida por Ordenador). Entre algunas de las ventajas de su utilización se encuentra que se aumenta considerablemente la calidad de las aplicaciones desarrolladas, se pueden producir de forma rápida prototipos de las aplicaciones, los cuales se pueden probar, corregir e incorporar a la aplicación final y aumenta considerablemente la productividad. Rational Rose y Visual Paradigm, son ejemplos de este tipo de herramienta.

Rational Rose

La herramienta Rational Rose es una herramienta CASE basada en UML (más adelante se explican sus características), proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente. Permite crear los diagramas que se van generando durante el proceso de Ingeniería en el desarrollo del software, además posee un gran número de estereotipos predefinidos que facilitan el proceso de modelación. Es completamente compatible con la metodología RUP.

Entre las características que presenta se encuentra que genera código Ada, ANSI C++, C++, CORBA, Java y Visual Basic, modela con UML para trabajo en el diseño de bases de datos y permite la integración con otras herramientas de desarrollo de Rational.

Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo completo del desarrollo de software. Proporciona una construcción de la aplicación más rápida, con mayor calidad y a un menor costo.

Visual Paradigm ofrece muchos beneficios entre los que se encuentran(22):

- La navegación intuitiva entre el código y el modelo visual.
- Poderoso generador de informes PDF⁸/HTML.
- Tiempo real en la demanda.
- Un ambiente modelador visual superior.
- Sofisticado diseño de diagramas.

⁸ Acrónimo del inglés *portable document format*, formato de documento portátil.

La herramienta CASE que se utiliza es Rational Rose debido a que es una herramienta de fácil entendimiento y de la cual se tiene mayor conocimiento pues fue la estudiada y empleada en las asignaturas de Ingeniería del Software 1 e Ingeniería del Software 2.

1.6.3 Lenguaje de modelado

UML es un lenguaje visual de modelado para “visualizar, especificar, construir y documentar los artefactos de un sistema de software”.(23)

UML no es un método o un proceso, sino un lenguaje para especificar. No especifica que metodología usar y se puede emplear en una gran variedad de formas para soportar determinada metodología. Son muchos los beneficios que se obtienen al utilizarlo, entre ellos se encuentran la presencia una mayor efectividad y productividad, el tiempo invertido se minimiza, y la detección de errores se agiliza.

1.6.4 Marco de trabajo

La plataforma para la integración de dispositivos existente en el CISED fue desarrollado en Microsoft .Net Framework v1.1.4, con C# como lenguaje y en el entorno de desarrollo Microsoft Visual Studio .Net 2003. Windows .NET Framework es el componente de Microsoft Windows que permite crear y ejecutar la próxima generación de aplicaciones de software y servicios Web XML, componentes que facilitan la integración compartiendo datos e información a través de la red mediante protocolos estándar independientes de la plataforma.(24)

1.6.5 Lenguaje de programación

C# es un lenguaje orientado a objetos y soporta todas las características propias de este paradigma: encapsulamiento, herencia y polimorfismo.

Las principales características que presenta son(25):

- Sencillez: C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET.
- Modernidad: C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones y que en otros lenguajes como Java o C++ hay que simular, como un tipo básico *decimal* que permita realizar operaciones de alta precisión con reales de 128 bits (muy útil en el mundo financiero), la inclusión de una instrucción

foreach que permita recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, la inclusión de un tipo básico *string* para representar cadenas o la distinción de un tipo *bool* específico para representar valores lógicos.

- Orientación a objetos: como hijo directo de C++, C# cuenta con la mayoría de las características de orientación a objetos de aquél. La entidad de primer nivel es, como en todos los lenguajes de esta categoría, la clase, definición a partir de la cual se crearían objetos y derivarían nuevas clases. La encapsulación, la herencia y el polimorfismo son, por tanto, los tres pilares básicos de la orientación a objetos en C# como en C++.
- Orientación a componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente *propiedades* (similares a campos de acceso controlado), *eventos* (asociación controlada de funciones de respuesta a notificaciones) o *atributos* (información sobre un tipo o sus miembros).
- Compatible: Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados Platform Invocation Services (PInvoke), la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLLs de la API Win32.

1.6.6 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE de su significado en inglés Integrated Development Environment) es un programa destinado a los programadores que está compuesto por una serie de herramientas que le facilita el trabajo a los desarrolladores, como un editor de texto, depuradores de código, compilador, entre otros. Este programa puede ser utilizado con un único lenguaje de programación o puede permitir ser utilizado por varios de estos. Entre ellos se encuentra el Visual Studio que es el IDE utilizado en el desarrollo del framework para la integración de dispositivos.

Microsoft Visual Studio es un IDE para sistemas operativos Windows. Soporta varios lenguajes como son: Visual C++, ASP .NET, Visual C#, Visual Basic. Su suite de productos ofrece una interfaz visual para la identificación de un programa como un servicio Web y ofrece formularios para la construcción de una interfaz de usuario.

Visual Studio .NET es la herramienta de desarrollo multilenguaje más completa para construir e integrar rápidamente aplicaciones y servicios Web XML. Aumenta de un modo extraordinario la productividad de los desarrolladores y crea nuevas oportunidades de negocio. En su diseño se han integrado a fondo los estándares y protocolos de Internet, como XML y SOAP, por lo que Visual Studio .NET simplifica considerablemente el ciclo de vida del desarrollo de aplicaciones.(26)

1.7 Arquitectura de software

En el presente epígrafe se aborda acerca de la arquitectura del software. Según Paul Clements en 1996 él expresó: «La AS⁹ es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.»

Los estilos arquitectónicos son los que definen la estructura de un sistema de software y se componen de subsistemas con sus responsabilidades.

Estilo arquitectónico: Caracteriza una familia de sistemas que están relacionados por compartir propiedades estructurales y funcionales. También puede definirse como la descripción de los tipos componente y de los patrones de interacción entre ellos. (28)

El estilo arquitectónico que se usó en la plataforma para la integración de dispositivos fue la arquitectura en capas el cual está comprendido dentro de la arquitectura de llamada y respuesta (call and return), que les permite a los diseñadores conseguir una estructura de un programa que sea fácil de modificar y de escalar.

Arquitecturas en capas: Los sistemas o arquitecturas en capas constituyen uno de los estilos que aparecen con mayor frecuencia mencionados como categorías mayores del catálogo, o, por el contrario, como una de las posibles encarnaciones de algún estilo más envolvente.(29)

Se define como una organización jerárquica de manera que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. Las ventajas de este estilo son muchas entre ellas se encuentra que proporciona una amplia reutilización.

⁹ Arquitectura del software.

1.8 Patrones de diseño

Un patrón es una solución a un problema en un contexto dado, codifica conocimiento específico acumulado por la experiencia de desarrolladores en un dominio. Cada patrón describe un problema que ocurre frecuentemente en nuestro ambiente, y el núcleo de la solución a dicho problema, de modo tal que pueda ser utilizada reiteradamente. Es por ello que un patrón es una pareja de problema-solución representada por un nombre y aplicable a varios contextos; además, viene acompañado de una sugerencia sobre la manera de usarlo en situaciones nuevas.

Según Booch: "Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles".

El libro "Design Patterns: Elements of Reusable Object-Oriented Software" de Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, 1995, Addison-Wesley (también conocido por GOF) encierra los patrones más famosos. En el mismo se clasifican de acuerdo a su propósito en: creacionales (abstraen el proceso de creación de instancias), estructurales (se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño) y de comportamiento (atañen a los algoritmos y a la asignación de responsabilidades entre objetos).

Otro conjunto de patrones muy usado son los patrones GRASP¹⁰. Se puede decir que son una guía para encontrar otros patrones de diseño.

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Dentro de estos se encuentran:

- Experto: El mismo plantea que la responsabilidad de realizar una tarea es de la clase que tiene los atributos o datos relacionados con la misma. Un sistema puede tener cientos de clases y miles de responsabilidades, si se asignan las mismas correctamente puede convertirse en clases sencillas y fáciles de comprender.

¹⁰ De sus siglas de General Responsibility Assignment Software Patterns y en español sería Patrones Generales de Software para Asignar Responsabilidades.

- Creador: Es el que guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito fundamental del mismo es encontrar un creador para conectar con el objeto producido en cualquier evento.
- Alta cohesión: Es la medida de la fuerza con la que las clases relacionan y enfocan sus responsabilidades. Es también el enfoque de las responsabilidades de un elemento, los mismos deben desempeñar una labor en el sistema que no sea realizada por el resto de los elementos. Si tenemos una baja cohesión posiblemente ese elemento realice más trabajo del necesario. Se asignan las responsabilidades a las clases de manera que todos sus métodos tengan un comportamiento bien definido.
- Bajo acoplamiento: Es la medida de la fuerza con que un elemento confía en otros elementos. Es un error muy común asignar demasiado nivel de acoplamiento con el resto de los componentes del sistema. Cada clase está acoplada (relacionada) a las clases estrictamente necesarias.
- Controlador: Generalmente un controlador delega en otros objetos el trabajo que se necesita hacer, es decir controla toda la actividad. El mismo aumenta el potencial para reutilizar las interfaces.

1.9 Conclusiones

Se puede apreciar que las tendencias generales en las tecnologías de captura de imágenes son a usar dispositivos altamente dependientes de un SDK. En los componentes de captura de imágenes del CISED no se usan las tecnologías genéricas, por lo que el soporte de múltiples modelos de dispositivos de captura de imágenes faciales es limitado. Se decide utilizar, a partir de las características descritas, como metodología de desarrollo RUP, atendiendo a sus características y a las facilidades que aporta; como herramienta de modelado se selecciona Rational Rose Enterprise, como tecnología se decide utilizar la API WIA, VLC y vídeo Streaming, por las características que anteriormente se exponen. Como plataforma para el desarrollo se selecciona Visual Studio 2003 ya que el componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales se integra al framework existente en el CISED para integrar el acceso a dispositivos el cual utiliza como IDE el Visual Studio y como lenguaje de programación se utilizará C#. Se estudia acerca de la arquitectura del software, específicamente acerca de la arquitectura en capas y acerca de los patrones de diseño posibles a utilizar.

Capítulo 2: Características del sistema

2.1 Introducción

En el presente capítulo se describen las características del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales. Se comienza con un estudio de los procesos, modelando la estructura de los mismos en un modelo de dominio y se describen los conceptos fundamentales presentes en el mismo. Luego se especifican las necesidades del sistema mediante los requisitos funcionales y no funcionales. También se identifican los actores y cómo interactúan con el sistema mediante los casos de usos, que son representados en los diagramas, finalmente se hace una detallada descripción de cada uno de ellos.

2.2 Descripción del sistema propuesto

Se propone el análisis y diseño de un componente para capturar imágenes faciales con múltiples modelos de cámaras digitales utilizando tecnologías genéricas, en específico la tecnología WIA. El mismo debe formar parte de la arquitectura de integración de dispositivos existente en el CISED para el desarrollo de sistemas de identificación.

El componente debe poseer tres funciones fundamentales:

- La inicialización del dispositivo de captura se realiza mediante la tecnología WIA, aquí se verifica si el dispositivo soporta la tecnología y en caso de que soporte la misma, el sistema se comunica con el dispositivo, y se visualiza el vídeo en tiempo real.
- El ajuste del vídeo consiste en brindar la posibilidad de aumentar o disminuir el zoom y de modificar el tipo de flash que se desee utilizar.
- La captura de la imagen se realiza cuando se solicita capturar una imagen en un momento determinado y se muestra en la pantalla.

2.3 Modelo del dominio

El modelo de dominio permite mostrar al usuario, los principales conceptos que se operan en el dominio del desarrollo del sistema. De esta forma los usuarios utilizan un vocabulario común, para lograr entender el contenido en que se enmarca el mismo, y por tanto también contribuir a la comprensión de los

requisitos. Se describe mediante diagramas UML (especialmente mediante diagramas de clases). Un modelo de dominio no representa objetos del software.

A partir de una breve descripción de los procesos del negocio expuestos, se puede observar que el mismo tiene un bajo nivel de estructuración por lo que conviene la utilización de un modelo de dominio. A continuación se describen los conceptos involucrados en el proceso de captura de imágenes faciales con múltiples modelos de cámaras digitales, y seguidamente se representa el diagrama del Modelo de Dominio mediante un diagrama de clases:

- **Sistema_Externo:** Plataforma para integrar el acceso a dispositivos existente en el CISED, que interactúa con el componente para la captura de imágenes faciales.
- **Componente para la captura de imágenes faciales:** Conjunto de clases que brinda funcionalidades para la comunicación con el dispositivo de captura.
- **Dispositivo de captura:** Es aquel dispositivo (en este caso una cámara digital), que permite introducir información en el ordenador.
- **Imagen facial:** Imagen del rostro capturada por el dispositivo de captura.

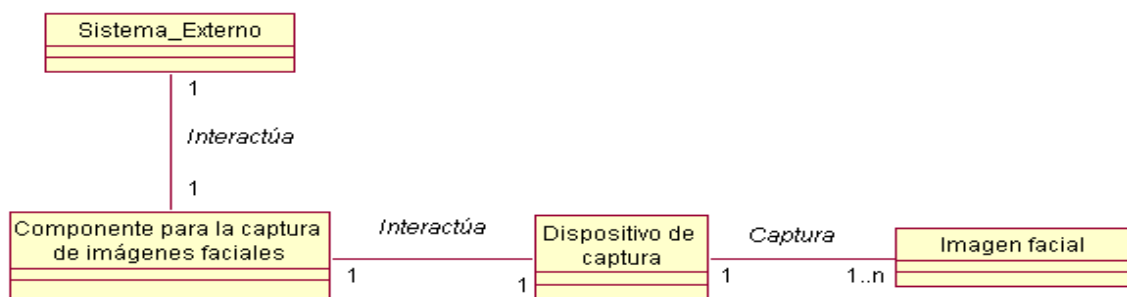


Figura 6. Modelo de dominio propuesto.

2.4 Requisitos

El desarrollo de un software generalmente trae diversidad de opiniones entre desarrolladores y el usuario final, es por ello que es necesario llegar a un acuerdo, y definir los requisitos de un software es fundamental para desarrollarlo.

Este es uno de los flujos de trabajo más importantes, porque en él se establece qué es lo que tiene que hacer exactamente el sistema que construyamos. En esta línea los requisitos son el contrato que

debemos cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos. (30)

El proceso de especificación de requisitos se divide en tres grandes actividades:

- 1- Captura de requisitos.
- 2- Definición de requisitos.
- 3- Validación de requisitos.

2.4.1 Captura de requisitos

La captura de requisitos es la actividad fundamental en el proceso de especificación de requisitos, por tanto es necesario realizar la obtención de información de forma eficiente. Existen diferentes técnicas para estos procesos como son: revisión de documentos, entrevistas, sistemas existentes, tormenta de ideas, mapa de conceptos, cuestionarios, comparación de terminología, entre otras.

La técnica para la captura de requisitos que se utiliza en la presente investigación son las entrevistas (ver [Anexo 4](#)) y las tormentas de ideas. Las entrevistas se emplean ya que son formas eficientes, rápidas y concisas de obtener la información. La persona que entrevista generalmente es el analista y debe centrarse en los temas de interés, no desviarse y no deben ser entrevistas improvisadas, deben prepararse con antelación para evitar omitir detalles importantes. La tormenta de ideas se aplica por su sencillez y su facilidad de empleo cuando se encuentra reunido un grupo de personas que se relacionan con el sistema y cada uno aporta sus criterios e ideas.

2.4.2 Definición de requisitos

La segunda actividad dentro del proceso de especificación de requisitos es la definición de los mismos existiendo un gran número de técnicas propuestas. Ellas se dedican a documentarlos de forma que esté al alcance de todos los involucrados, algunas son: lenguaje natural, plantillas o patrones, casos de uso y glosarios de términos.

La técnica de definición que se emplea son las plantillas, pues es una técnica sencilla descrita en un lenguaje natural y que elimina la ambigüedad.

Los requisitos se dividen en dos grupos, los funcionales y no funcionales. Los cuales se definen a continuación.

Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con que propiedades o cualidades se relacionen.(31)

A continuación se muestran los requisitos funcionales definidos para el componente.

Número	Requerimiento
RF 1	Inicializar dispositivo de captura.
RF 1.1	Verificar compatibilidad con la tecnología genérica WIA.
RF 1.2	Visualizar vídeo.
RF 2	Ajustar vídeo.
RF 2.1	Realizar zoom del lente de la cámara.
RF 2.2	Seleccionar tipo de flash.
RF 3	Capturar imagen.
RF 3.1	Captar la imagen.
RF 3.2	Mostrar la imagen capturada.

Tabla 1. Requisitos funcionales.

Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requisitos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. (31)

A continuación se exponen los requisitos no funcionales del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales.

Número	Requisitos
Diseño e implementación	
RNF 1	Utilizar C# como lenguaje de programación.
RNF 2	Utilizar la herramienta IDE de desarrollo Visual Studio 2003.
RNF 3	Utilizar la herramienta CASE Rational Rose Enterprise Edition.
RNF 4	Realizar el modelado con UML.
RNF 5	El marco de trabajo base de desarrollo que se utiliza es Microsoft .Net Framework v1.1.4.
RNF 6	La arquitectura que se utiliza es en capas.
Hardware	
RNF 7	Se debe contar con 512 Mb de memoria RAM o superior.
RNF 8	Tarjeta de vídeo de 64Mb o superior.
RNF 9	Microprocesador de 1 GHz o superior.
RNF 10	Disco duro con 1GB libre.
RNF 11	Debe tener puerto USB y utilizar la interfaz de comunicación USB 1.1 o superior.
RNF 12	Se debe contar con al menos un dispositivo de captura que posea como mínimo 2 megapíxeles de resolución.
RNF 13	Se debe contar con al menos un dispositivo de iluminación.
Software	
RNF 14	Sistema Operativo Windows 95 o Superior
Rendimiento	
RNF 15	Realizar una captura de una imagen en menos de 5 segundos.

Tabla 2. Requisitos no funcionales.

Descripción completa de los requisitos funcionales.

- *Requisito funcional “Verificar compatibilidad con la tecnología genérica WIA”:* Consiste en que el componente propuesto sea capaz de verificar que el dispositivo de captura que se ha conectado, sea compatible con la tecnología WIA.

Conceptos tratados	Conceptos	Atributos
	Compatibilidad	-
Precondiciones	Precondiciones	Pre-requisitos
	Debe existir un dispositivo de adquisición de imágenes conectado a la computadora.	No procede.
Descripción	El usuario solicita iniciar la comunicación con el dispositivo de adquisición de imágenes. Se verifica la compatibilidad con la tecnología WIA.	
Validaciones	Se valida a través de revisiones, matriz de trazabilidad y auditorías con la utilización de una lista de chequeo.	
Post-condiciones	Se ha comenzado la interacción con el dispositivo de adquisición de imágenes.	
Post-requisitos	Visualizar vídeo en tiempo real.	

Tabla 3. Especificación del requisito “Verificar compatibilidad con la tecnología genérica WIA”.

- *Requisito funcional “Visualizar vídeo”*: Consiste en que después de haber iniciado la comunicación con el dispositivo de adquisición de imágenes, se logre visualizar el vídeo en tiempo real en la computadora a la misma vez que se visualiza en la cámara digital.

Conceptos tratados	Conceptos	Atributos
	Vídeo	-
Precondiciones	Precondiciones	Pre-requisitos
	Exista comunicación con algún dispositivo de adquisición de imágenes.	Verificar compatibilidad con la tecnología genérica WIA.
Descripción	Inmediatamente después de efectuar la conexión con el dispositivo de captura de imágenes se comienza la visualización en tiempo real del vídeo en la computadora, utilizando para ello las facilidades que brinda WIA o mediante la utilización del VLC.	
Validaciones	Se valida a través de revisiones, matriz de trazabilidad y auditorías con la	

	utilización de una lista de chequeo.
Post-condiciones	Se visualiza el vídeo en tiempo real de lo que el dispositivo de adquisición de imágenes está captando en ese momento.
Post-requisitos	No procede

Tabla 4. Especificación del requisito “Visualizar vídeo”.

- *Requisito funcional “Realizar zoom del lente de la cámara”*: Consiste en ofrecer la opción al usuario de poder acercar o alejar el vídeo a conveniencia de lo que le es realmente necesario o lo que desea obtener al usuario.

Conceptos tratados	Conceptos	Atributos
	Zoom del lente de la cámara.	-
Precondiciones	Precondiciones	Pre-requisitos
	Exista una visualización en tiempo real de lo que el dispositivo de adquisición de imágenes está captando.	Visualizar vídeo.
Descripción	Se selecciona aumentar zoom según lo deseado. Se selecciona disminuir zoom según lo deseado.	
Validaciones	Se valida a través de revisiones, matriz de trazabilidad y auditorías con la utilización de una lista de chequeo.	
Post-condiciones	Se ajusta el zoom del vídeo visualizado.	
Post-requisitos	No procede.	

Tabla 5. Especificación del requisito “Realizar zoom del lente de la cámara”.

- *Requisito funcional “Seleccionar tipo de flash”*: Consiste en ofrecer al usuario las siguientes tres opciones: siempre utilizar flash, nunca utilizarlo o hacerlo de forma automática.

Conceptos tratados	Conceptos	Atributos
	Tipo de flash.	Sin flash.

		Con flash automático. Usando siempre flash.
Precondiciones	Precondiciones	Pre-requisitos
	Exista una visualización en tiempo real de lo que el dispositivo de adquisición de imágenes está captando.	Visualizar vídeo.
Descripción	Se selecciona el tipo de flash que desee. Puede ser sin la utilización del flash o escogiendo para que sea de forma automática o utilizándolo en cada captura el mismo.	
Validaciones	Se valida a través de revisiones, matriz de trazabilidad y auditorías con la utilización de una lista de chequeo.	
Post-condiciones	Se selecciona el tipo de flash.	
Post-requisitos	No procede	

Tabla 6. Especificación del requisito “Seleccionar tipo de flash”.

- *Requisito funcional “Captar la imagen”*: Consiste en ofrecer la opción de capturar en un momento determinado la imagen visualizada en el vídeo en tiempo real.

Conceptos tratados	Conceptos	Atributos
	Imagen	-
Precondiciones	Precondiciones	Pre-requisitos
	Exista una visualización en tiempo real de lo que el dispositivo de adquisición de imágenes está captando.	Visualizar vídeo.
Descripción	Se selecciona el momento que se desee capturar la imagen visualizada en el	

	vídeo en tiempo real. Se captura la imagen que se visualiza en ese momento determinado.
Validaciones	Se valida a través de revisiones, matriz de trazabilidad y auditorías con la utilización de una lista de chequeo.
Post-condiciones	Se captura la imagen.
Post-requisitos	Mostrar la imagen capturada.

Tabla 7. Especificación del requisito “Captar la imagen”.

- *Requisito funcional “Mostrar la imagen capturada”*: Consiste en visualizar la imagen que se obtuvo de la captura de la imagen.

Conceptos tratados	Conceptos	Atributos
	Imagen capturada	-
Precondiciones	Precondiciones	Pre-requisitos
	Halla sido capturada la imagen	Captar la imagen.
Descripción	Se muestra la imagen que fue capturada.	
Validaciones	Se valida a través de revisiones, matriz de trazabilidad y auditorías con la utilización de una lista de chequeo.	
Post-condiciones	Se muestra la imagen capturada.	
Post-requisitos	No procede.	

Tabla 8. Especificación del requisito “Mostrar la imagen capturada”.

2.4.3 Técnica de validación

La validación de los requisitos se realiza para demostrar que los requisitos realmente son lo que desea el cliente. Algunas de las técnicas para la validación son: revisiones, auditorías, matrices de trazabilidad y prototipos.

Finalmente, la técnica de validación que se utilizan son las revisiones ([Anexo 5](#)), la matriz de trazabilidad ([Anexo 6](#)) y las auditorías con la utilización de una lista de chequeo ([Anexo 7](#)).

2.5 Modelo de Casos de Uso del sistema

Una vez capturados los requisitos funcionales del sistema, estos se representan mediante el Diagrama de Casos de Uso del sistema. Primeramente es necesario definir cuáles son los actores que interactúan con el sistema y los casos de uso que representan todas las funcionalidades del mismo.

2.5.1 Actores del sistema

Los actores del sistema no son parte de él. Pueden intercambiar información con él. Pueden ser un recipiente pasivo de información. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado. (31)

En la presente investigación se define un único actor que en un momento determinado realiza un grupo de acciones, el mismo se justifica a continuación:

Actores	Justificación
Sistema_Externo	Es la plataforma para integrar el acceso a dispositivos existente en el CISED que interactúa directamente con el componente usando sus funcionalidades.

Tabla 9. Actor del sistema.

2.5.2 Casos de Uso del sistema

Los Casos de Uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. (31)

A continuación se exponen los casos de uso obtenidos.

Número	Caso de Uso	Prioridad
CU_1	Inicializar dispositivo de captura.	Crítico.
CU_2	Ajustar vídeo.	Crítico.
CU_2	Capturar imagen.	Crítico.

Tabla 10. Casos de Uso del sistema.

2.5.3 Diagrama de Casos de Uso del sistema

Un Diagrama de Casos de Uso del sistema representa gráficamente a los procesos y su interacción con los actores. Cada caso de uso debe comunicarse con al menos un actor, si no aparece ningún actor que se comuniquen con un caso de uso esto indica error en el modelo de caso de uso o en los requisitos planteados.(31)

A continuación se muestra el Diagrama de Casos de Uso (DCUS) propuesto.

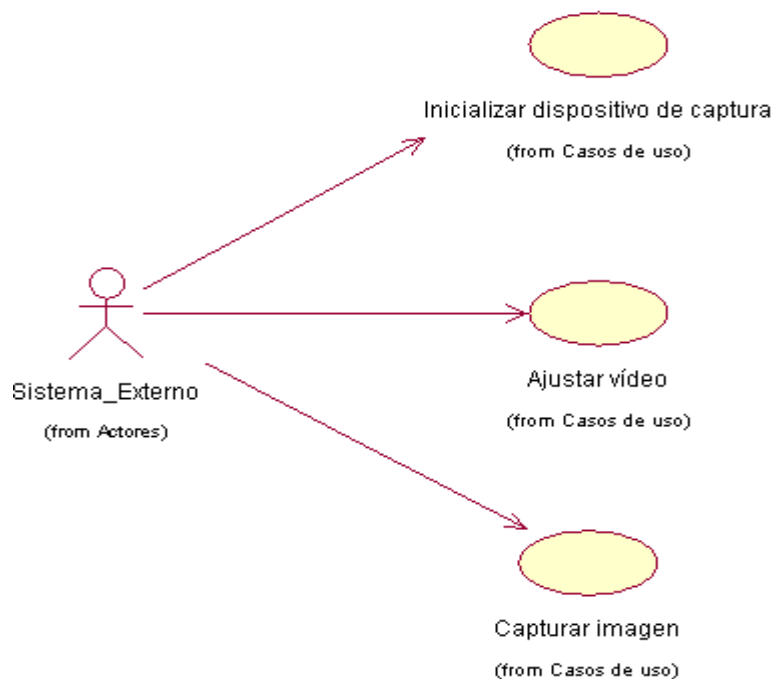


Figura 7. Diagrama de Casos de Uso.

2.5.4 Descripción de los CU

Para comprender exactamente cada Caso de Uso no es suficiente con el Diagrama de Casos de Uso, es por ello que se hace necesaria una descripción textual, la misma puede hacerse de dos formas: breve o extendida.

A continuación se describen los Casos de Uso del sistema en forma breve:

Resumen del CU “Inicializar dispositivo de captura”.

Caso de Uso:	Inicializar dispositivo de captura.
Actor :	Sistema_Externo.
Descripción:	Se conecta con del dispositivo de captura verificando que soporte la tecnología WIA y luego se visualiza el vídeo en tiempo real.
Referencia:	RF 1.1- Verificar compatibilidad con la tecnología genérica WIA. RF 1.2- Visualizar vídeo.

Tabla 11. Resumen del CU “Inicializar el dispositivo de captura”.

Resumen del CU “Ajustar imagen”.

Caso de Uso:	Ajustar vídeo.
Actor :	Sistema_Externo.
Descripción:	Se le brinda la posibilidad al Sistema_Externo de modificar el zoom aumentándolo o disminuyéndolo en medida de acercarse o alejarse más al objetivo deseado, y modificar el flash del vídeo que se visualiza, de acuerdo al grado de luz que se desee o necesite.
Referencia:	RF 2.1- Realizar zoom del lente de la cámara. RF 2.2- Seleccionar tipo de flash.

Tabla 12. Resumen del CU “Ajustar imagen”.

Resumen del CU “Capturar la imagen”.

Caso de Uso:	Capturar imagen.
---------------------	------------------

Actor :	Sistema_Externo.
Descripción:	Se le ofrece al Sistema_Externo la opción de capturar la imagen que desee en el vídeo mostrado. Luego de la captura, esta imagen es mostrada.
Referencia:	RF 3.1- Captar la imagen. RF 3.2- Mostrar la imagen capturada.

Tabla 13. Resumen del CU “Capturar la imagen”.

Seguidamente se exponen las descripciones de los Casos de Uso en forma extendida.

Descripción extendida del CU “Inicializar dispositivo de captura”.

Caso de Uso:	Inicializar dispositivo de captura.
Actores:	Sistema_Externo.
Resumen:	El caso de uso se inicia cuando el Sistema_Externo solicita la conexión con un dispositivo de captura. Seguidamente se verifica la compatibilidad del dispositivo con la tecnología genérica WIA, y en caso de que el dispositivo la soporte, el sistema muestra el vídeo en tiempo real. En caso de que el dispositivo de captura no la soporte se notificará al Sistema_Externo.
Precondiciones:	
Referencias:	RF 1.1- Verificar compatibilidad con la tecnología genérica WIA. RF 1.2- Visualizar vídeo.
Prioridad:	Crítico.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema

1. Solicita conexión con el dispositivo de captura	2. Verifica compatibilidad con la tecnología WIA. 3. El sistema se comunica con el dispositivo de captura. (Ver el Flujo Alterno 3.1.)
	4. El sistema visualiza el vídeo en tiempo real.
Flujos alternos	
Flujos alternativo 1 “ Cancelar”	
Acción del actor	Respuesta del sistema
	3.1 En caso de error el sistema muestra un mensaje indicando que el dispositivo no soporta la tecnología WIA.
Poscondiciones:	Se creó la conexión entre el sistema y el dispositivo de captura.

Tabla 14. Descripción extendida del CU “Inicializar dispositivo de captura”.

Descripción extendida del CU “Ajustar vídeo”.

Caso de Uso:	Ajustar vídeo.
Actores:	Sistema_Externo.
Resumen:	El caso de uso se inicia cuando se le brindan las opciones a Sistema_Externo de aumentar y disminuir zoom, y de seleccionar el tipo de flash que desee utilizar.
Precondiciones:	Esté inicializado el dispositivo de captura.
Referencias:	RF 2.1- Realizar zoom del lente de la cámara. RF 2.2- Seleccionar tipo de flash.
Prioridad:	Crítico.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. Sistema_Externo solicita aumentar zoom.	2. Aumenta zoom al vídeo visualizado (Ver flujo alternativo 2.1).
3. Sistema_Externo solicita disminuir	4. Disminuye zoom al vídeo visualizado (Ver flujo alternativo

zoom.	4.1).
	3. Se muestra el vídeo con el zoom modificado.
5. El Sistema_Externo solicita modificar el tipo de flash según el escogido.	6. Habilita el tipo de flash solicitado.
	7. Se muestra el vídeo.
Flujos alternos	
Flujos alterno 1 “ Cancelar”	
Acción del actor	Respuesta del sistema
2.1. Sistema_Externo solicita disminuir el zoom.	2.2 El sistema disminuye el zoom.
4.1. Sistema_Externo solicita aumentar el zoom.	4.2 El sistema aumenta el zoom.
Poscondiciones:	Se ajustó el vídeo.

Tabla 15. Descripción extendida del CU “Ajustar vídeo”.

Descripción extendida del CU “Capturar la imagen”.

Caso de Uso:	Capturar la imagen.
Actores:	Sistema_Externo.
Resumen:	El caso de uso se inicia cuando el Sistema_Externo manda a capturar la imagen, y se muestra si el mismo decide que no se corresponde con una buena muestra facial solicita capturar otra foto.
Precondiciones:	Esté inicializado el dispositivo de captura.
Referencias:	RF 3.1- Captar la imagen. RF 3.2- Mostrar la imagen capturada.

Prioridad:	Crítico.
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
1. El Sistema_Externo solicita capturar la imagen.	2. El sistema captura la imagen en el momento determinado.
	3. El sistema muestra la imagen correspondiente (Ver flujo alterno 3.1).
Flujos alternos	
Acción del actor	Respuesta del sistema
3.1 En caso de inconformidad, el Sistema_Externo solicita capturar nuevamente la imagen.	3.2 El sistema captura nuevamente la imagen en el momento determinado.
Poscondiciones:	Se capturó la imagen.

Tabla 16. Descripción extendida del CU “Capturar la imagen”.

2.6 Conclusiones

En el presente capítulo se obtiene un mayor entendimiento de los procesos de negocio, definiendo así los requisitos funcionales y no funcionales, lo que permite concretar y lograr un mejor entendimiento de las actividades a automatizar y especificar las cualidades con las que debe cumplir el sistema. Se construyen artefactos propuestos por la metodología de desarrollo del software RUP, definidos para la fase de inicio. Se tiene una visión más acertada del futuro análisis y diseño del componente de captura de imágenes faciales con múltiples modelos de cámaras digitales. Con todos los elementos expuestos en el presente capítulo, se puede proceder a desarrollar el análisis y diseño del componente.

Capítulo 3: Análisis y diseño

3.1 Introducción

En el presente capítulo se profundiza en el análisis y diseño del sistema. Después de las definiciones y descripciones abordadas en los capítulos anteriores, se hace necesario definir cómo se desarrolla. En el mismo se precisa el modelo de análisis con los diagramas de clases del análisis del sistema correspondientes, además se exponen los diagramas de interacción de los mismos. Luego se expone el modelo de diseño, en el mismo se desarrolla el diagrama de clases del diseño teniendo en cuenta patrones, finalmente se modelan los diagramas de interacción mostrando una descripción gráfica de la interacción entre los actores y el sistema. Finalmente se realiza una descripción de las clases presentes en el diagrama de clases del diseño.

3.2 Descripción de la arquitectura

La arquitectura del componente de software para la captura de imágenes faciales con múltiples modelos de cámaras digitales se rige por la arquitectura de la plataforma a la cual se integra, por lo cual también está constituido por capas, en este caso no presenta la capa de interfaces para la definición de las capacidades de cada tipo de dispositivo, ya que el componente es para la captura de imágenes, pero si presenta la capa de implementación del wrapper y la capa de interacción con la API WIA.

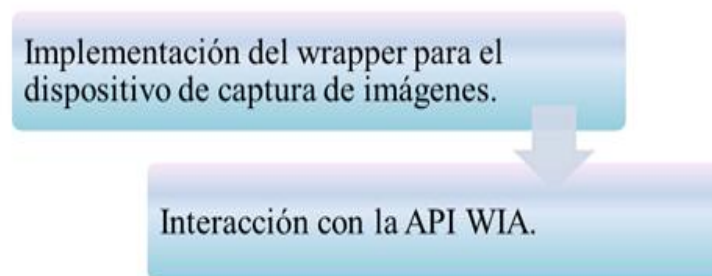


Figura 8. Arquitectura en capas del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales.

3.3 Análisis

Durante el análisis, se estudian los requisitos que fueron descritos en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y

una descripción de los mismos que sea fácil de mantener y que ayude a estructurar todo el sistema, incluyendo su arquitectura.(32)

3.3.1 Modelo de análisis


En el modelo de análisis no se tiene en cuenta el lenguaje de programación, ni la plataforma en la que se ejecutará la aplicación, ni otras características que afectan al sistema, ya que el objetivo del análisis es comprender los requisitos. Se obtiene una visión del sistema que se enfoca a qué hace el sistema, por lo que sólo se interesa por los requisitos funcionales. El modelo de análisis puede considerarse una primera aproximación al modelo de diseño, aunque es un modelo propio.

Es importante hacer notar que el modelo de análisis hace abstracciones; evita resolver algunos problemas y tratar algunos requisitos que pensamos que es mejor posponer al diseño y a la implementación. Debido a esto no siempre se puede conservar la estructura proporcionada por el análisis, sino que debemos negociar y comprometer durante el diseño y la implementación.(33)

Durante el análisis, los 6 requisitos funcionales capturados en el capítulo anterior se especifican de manera más precisa y estructurada.

3.3.2 Clase del análisis

Una clase del análisis siempre se corresponde a uno de tres estereotipos básicos: interfaz, control o entidad (ver tabla 17). Los estereotipos constituyen un método consistente para identificar y describir las clases del análisis.

Nombre	Estereotipo	Descripción
Entidad	 nombre_entidad	Estas clases son utilizadas para modelar información que posee larga vida y que es a menudo persistente. Poseen la información acerca de algún fenómeno (persona, objeto del mundo real o suceso). Generalmente se derivan de una clase del dominio o del negocio. Las clases entidad reflejan la información de un modo que beneficia a los desarrolladores a la hora de diseñar e implementar el sistema. Un objeto de entidad no ha de ser necesariamente pasivo y puede tener en ocasiones un comportamiento complejo relativo a la información que representa. Los objetos

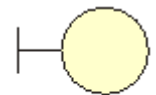

		de entidad aíslan los cambios en la información que representan. Las clases de entidad suelen mostrar una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema.(33)
Interfaz	 <p>nombre_interfaz</p>	Se utilizan para modelar la interacción entre el sistema y sus actores (usuarios, y sistemas externos). Modelan las partes del sistema que dependen de sus actores. No es necesario que describan cómo se ejecuta físicamente la interacción, ya que esto se considerará en las actividades de diseño e implementación.(33)
Control	 <p>nombre_control</p>	Las clases de control representan coordinación, secuencia, transacciones y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Los aspectos dinámicos del sistema se modelan con clases de control, debido a que ellas manejan y coordina las acciones y los flujos de control principales, y delegan trabajo a otros objetos (es decir objetos de interfaz y de entidad). Las clases de control no encapsulan aspectos relacionados con las interacciones, con los actores, ni tampoco aspectos relacionados con información de larga vida y persistente gestionada por el sistema; esto lo encapsulan las clases de interfaz y de entidad, respectivamente.(33)

Tabla 17. Nombre, estereotipo y descripción de las clases del análisis.

3.3.3 Diagrama de clases del análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo. (32)

Este modelo define realizaciones de casos de uso, y cada una de ellas representa el análisis de un caso de uso del modelo. A continuación se muestran los diagramas de clase del análisis para los 3 Casos de Uso obtenidos en el capítulo anterior.

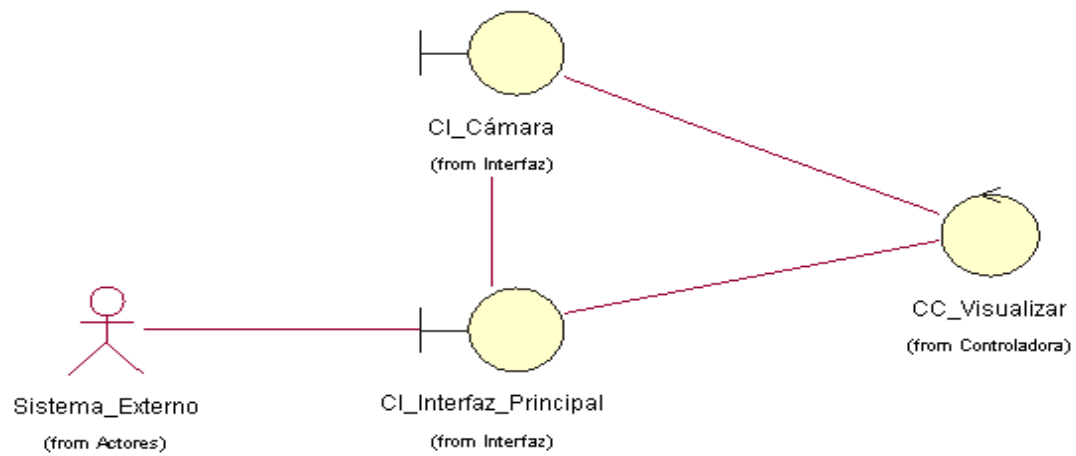


Figura 9. Diagrama de clases del análisis del CU “Inicializar dispositivo de captura”.

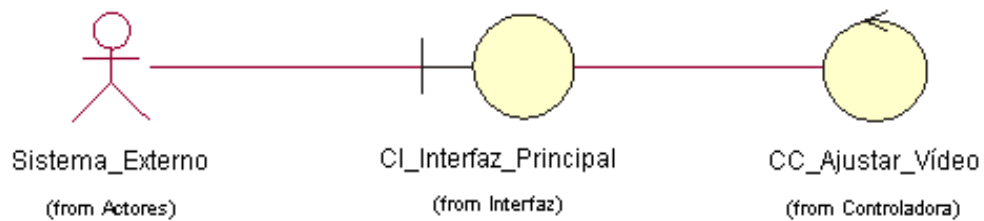


Figura 10. Diagrama de clases del análisis del CU “Ajustar vídeo”.

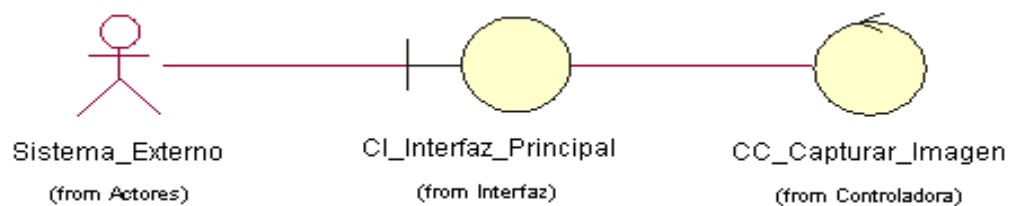


Figura 11. Clase del análisis del CU “Capturar imagen”.

3.3.4 Diagramas de colaboración

Cuando un actor invoca al caso de uso mediante el envío de algún tipo de mensajes al sistema, comienza una secuencia de acciones en un caso de uso. En el análisis se prefiere utilizar los diagramas de colaboración.

Una colaboración modela los objetos y los enlaces significativos dentro de una interacción. Los objetos y los enlaces son significativos solamente en el contexto proporcionado por la interacción. Un rol describe un objeto, y un rol en la asociación describe un enlace dentro de una colaboración. Un diagrama de colaboración muestra los roles en la interacción en una disposición geométrica. Los mensajes se muestran como flechas, ligadas a las líneas de la relación, que conectan a los roles. La secuencia de mensajes, se indica con los números secuenciales que preceden a las descripciones del mensaje.(32)

A continuación se muestran los diagramas de colaboración correspondientes.

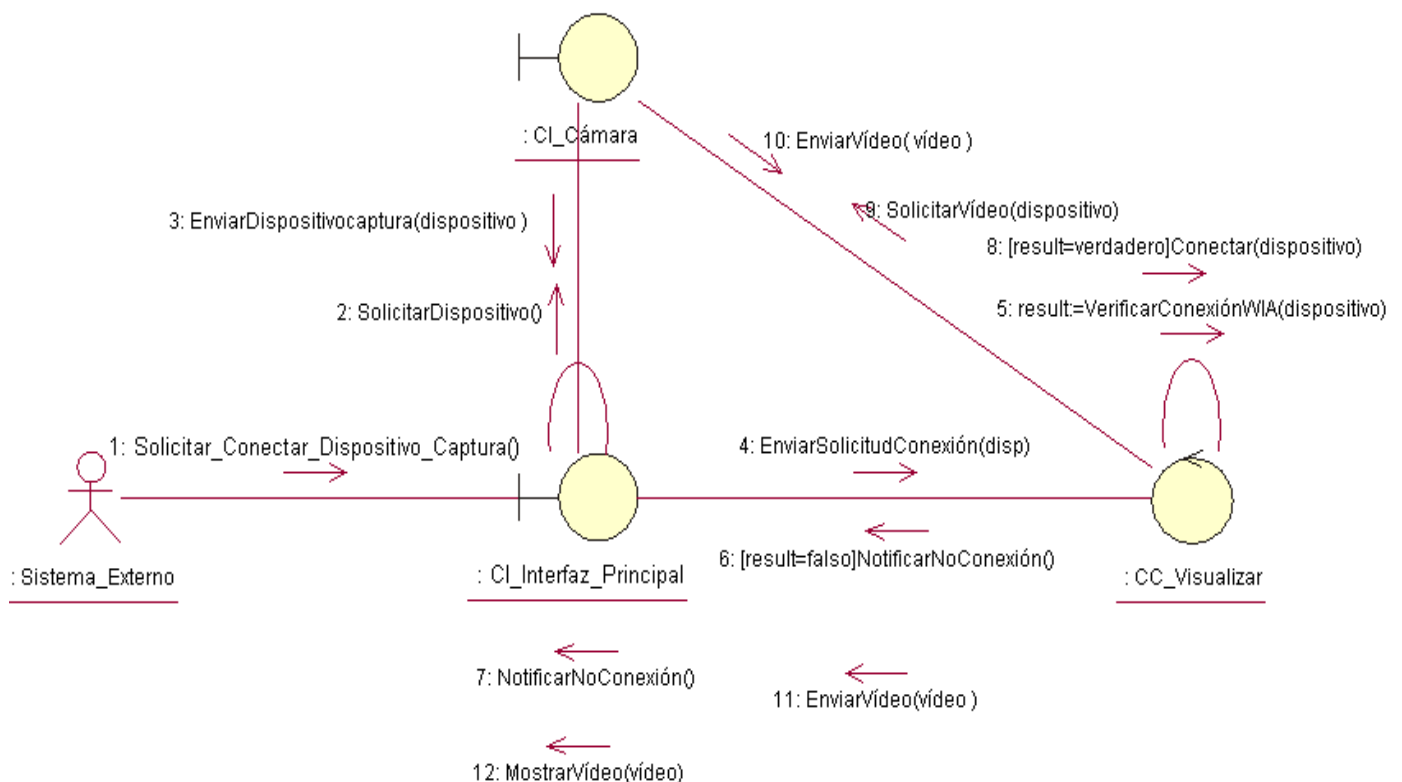


Figura 12. Diagrama de colaboración del CU “Inicializar dispositivo de captura”.

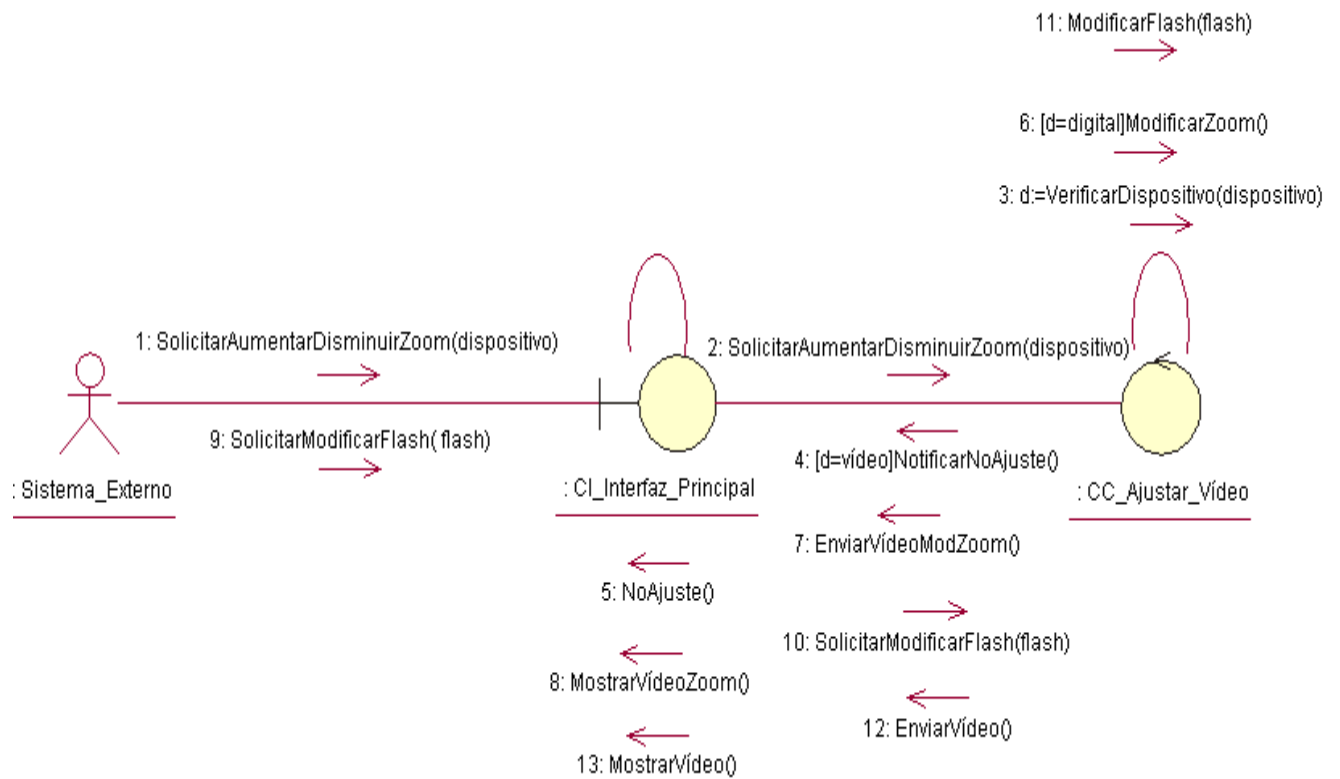


Figura 13. Diagrama de colaboración del CU "Ajustar vídeo".

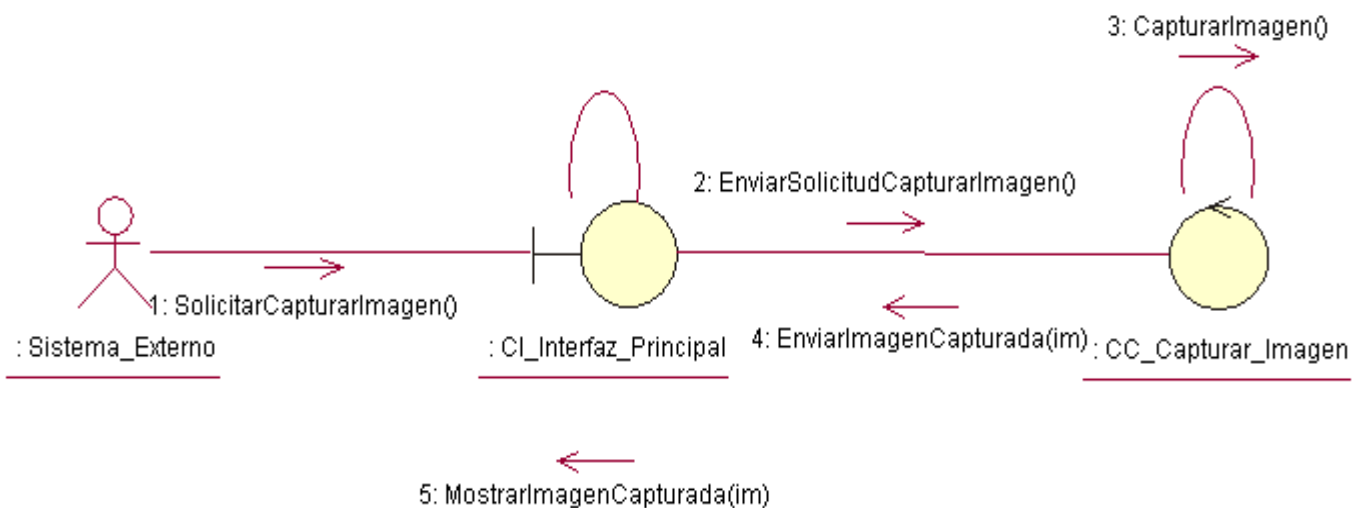


Figura 14. Diagrama de colaboración del CU "Capturar imagen".

3.4 Diseño

Teniendo en cuenta la propuesta de solución descrita anteriormente, es necesario definir cómo se desarrollará el sistema. De esta operación se encarga el flujo de trabajo de diseño.

El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable y sólida y a crear un plano del modelo de implementación.(33)

En el contexto de la ingeniería del software, Mostow sugiere que el propósito del diseño es construir un sistema que(34):

- Satisface determinada (quizás informal) especificación funcional.
- Se ajusta a las limitaciones impuestas por el medio de destino.
- Respeta requisitos implícitos o explícitos sobre rendimiento y utilización de recursos.
- Satisface criterios de diseño implícitos o explícitos sobre la forma del artefacto.
- Satisface restricciones sobre el propio proceso de diseño, tales como su longitud o coste, o las herramientas disponibles para realizar el diseño.

3.4.1 Patrones de diseño utilizados

En el diseño de la propuesta de solución, se tienen en cuenta esencialmente el patrón Alta cohesión que se aplica en todas las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden, y el patrón Bajo acoplamiento, ya que cada clase se comunica con un número pequeño de clases.

3.4.2 Modelo de diseño

El modelo de diseño constituye un modelo de objetos que describe la realización física de los casos de uso. Se centra en cómo cumple el sistema sus objetivos, cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas en el entorno de la implementación, tienen impacto en el sistema a considerar. El modelo de diseño sirve de abstracción de la implementación del sistema.

3.4.3 Clase del diseño

Una clase del diseño es una abstracción de una clase en la implementación del sistema. El lenguaje utilizado para especificar una clase del diseño es el mismo lenguaje de programación elegido que se

utiliza en la implementación. Los métodos que se especifican en los diagramas de clases del diseño se corresponden con el método en la implementación de las clases, y las relaciones que se establecen entre las clases en la implementación tienen usualmente un significado directo.

3.4.4 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación.

Para realizar el diseño de las clases se siguen una serie de pautas de codificación las mismas son:

- Comenzar el nombre de cada interfaz con la letra “I”, por ejemplo: IZoom.
- No repetir el nombre de una clase.
- Utilizar la palabra “Base” en cada clase padre, por ejemplo: CamDigitalBase.
- Utilizar una notación en la cual el comienzo de cada palabra sea con una letra mayúscula y restantes letras sean minúsculas, por ejemplo: CaptadorFotoBase.

Después de mostrar algunas de las principales características que tienen las clases del diseño se muestra el diagrama de clases del diseño del componente propuesto. En el mismo se puede observar cómo quedan definidas las clases con sus atributos, funcionalidades y cómo se relacionan entre ellas, con el subsistema externo y cómo se usan los patrones de diseño anteriormente explicados.

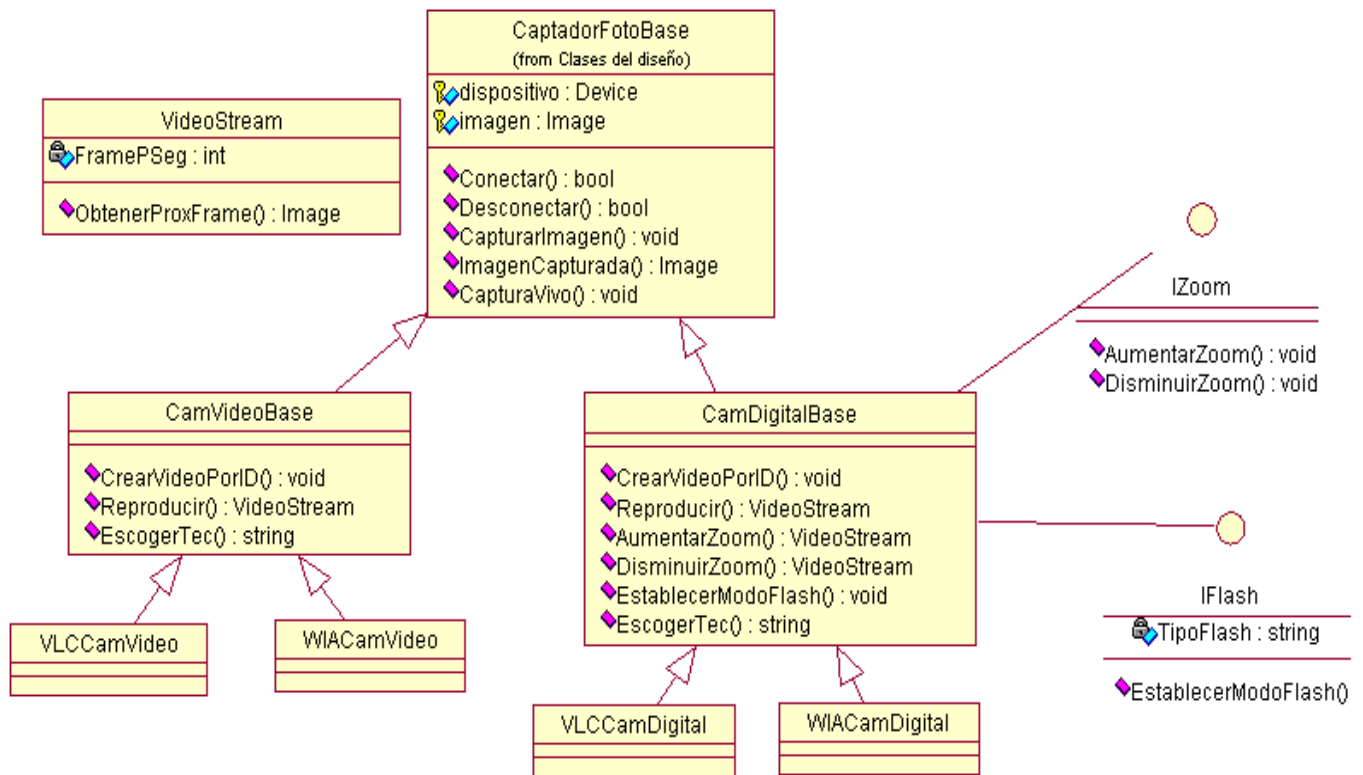


Figura 15. Diagrama de clases del diseño.

3.4.5 Diagramas de secuencia

En el diseño es preferible representar la secuencia de las acciones de un caso de uso, con diagramas de secuencia, ya que el centro de atención principal es encontrar secuencias de interacciones detalladas y ordenadas en el tiempo.

Un diagrama de secuencia representa una interacción como un gráfico bidimensional. La dimensión vertical es el eje de tiempo, que avanza hacia abajo de la página. La dimensión horizontal muestra los roles de clasificador que representan objetos individuales en la colaboración. Cada rol de clasificador se representa mediante una columna vertical-línea de vida. Durante el tiempo que existe un objeto, el rol se muestra por una línea discontinua. Durante el tiempo que dura una activación de un procedimiento en el objeto, la línea de vida se dibuja como una línea doble.(32)

A continuación se muestran los diagramas de secuencia del diseño para cada uno de los casos de uso.

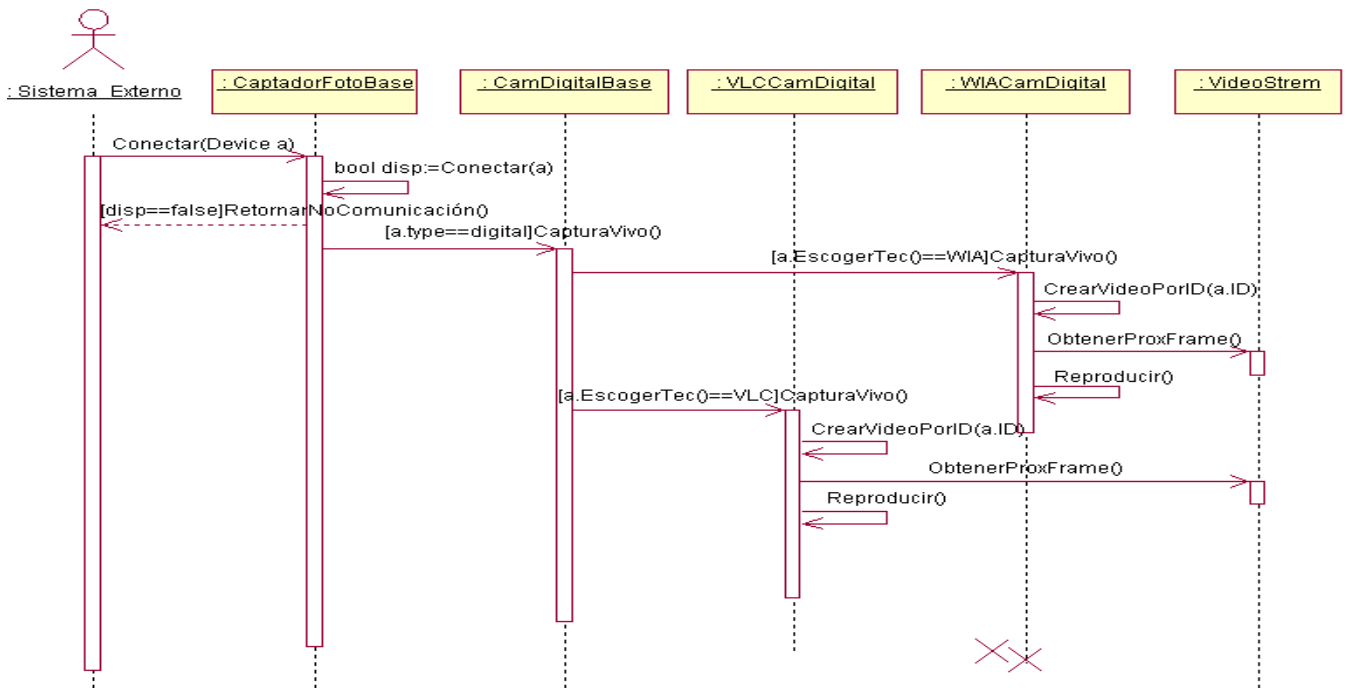


Figura 16. Diagrama de secuencia del diseño para el CU “Iniciar dispositivo de captura” para cuando es un dispositivo “digital”.

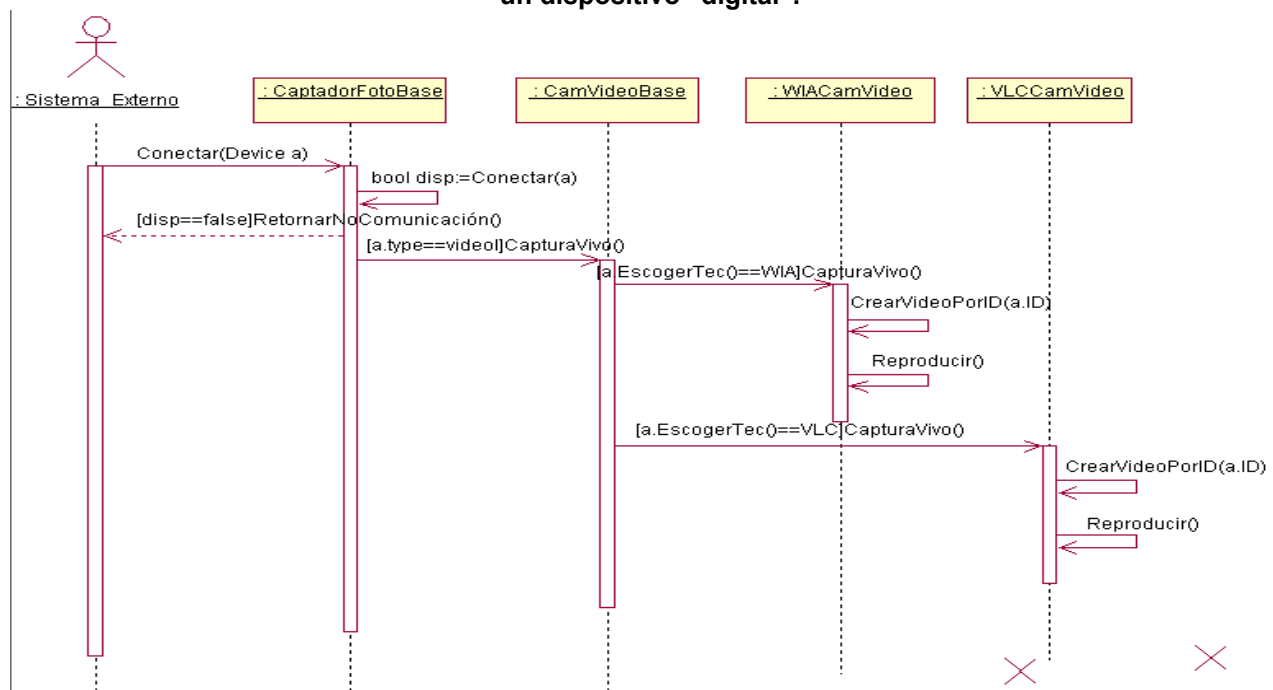


Figura 17. Diagrama de secuencia del diseño para el CU “Iniciar dispositivo de captura” para cuando es un dispositivo “vídeo”.

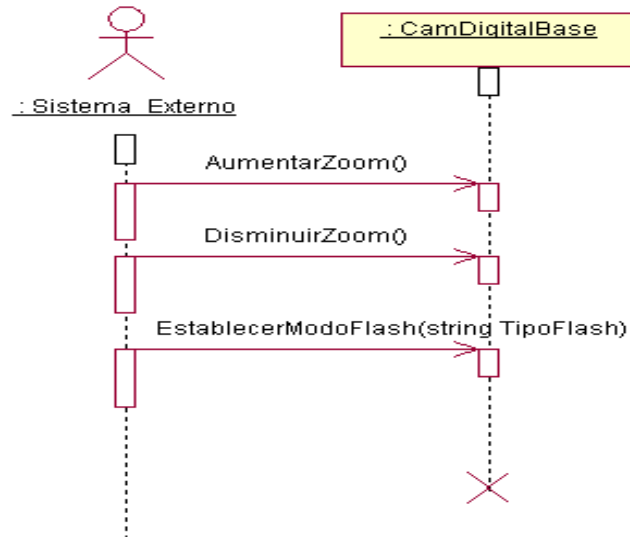


Figura 18. Diagrama de secuencia del diseño para el CU "Ajustar vídeo".

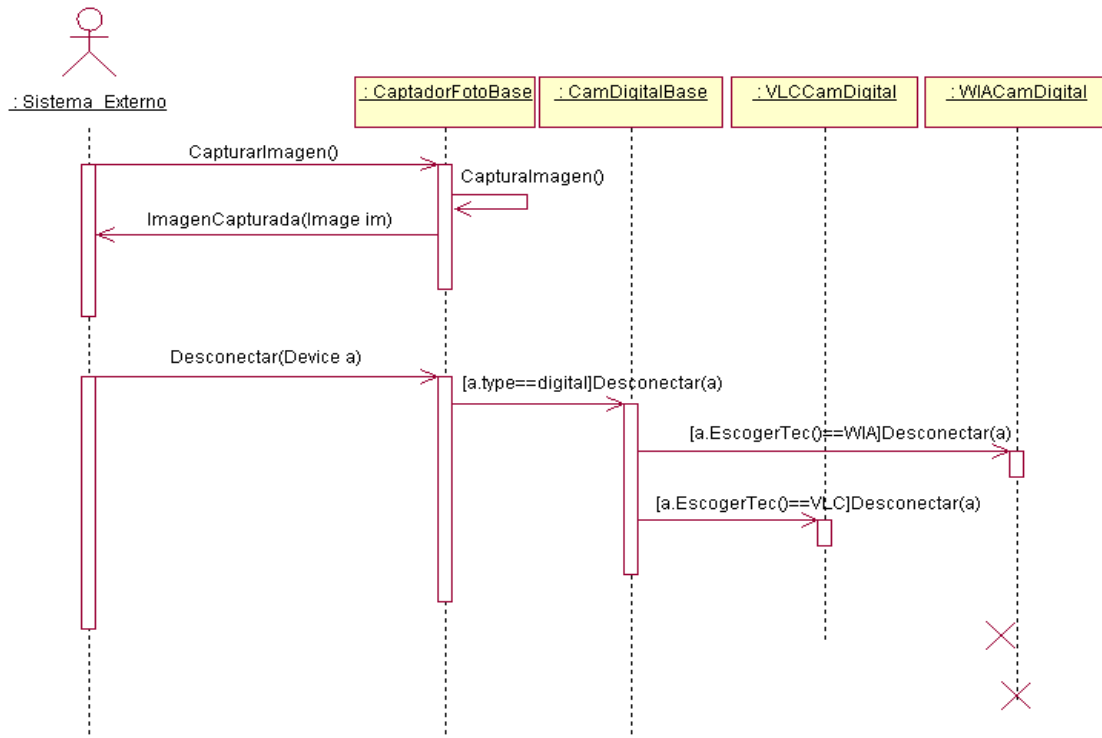


Figura 19. Diagrama de secuencia del CU "Capturar imagen" para cuando es un dispositivo "digital".

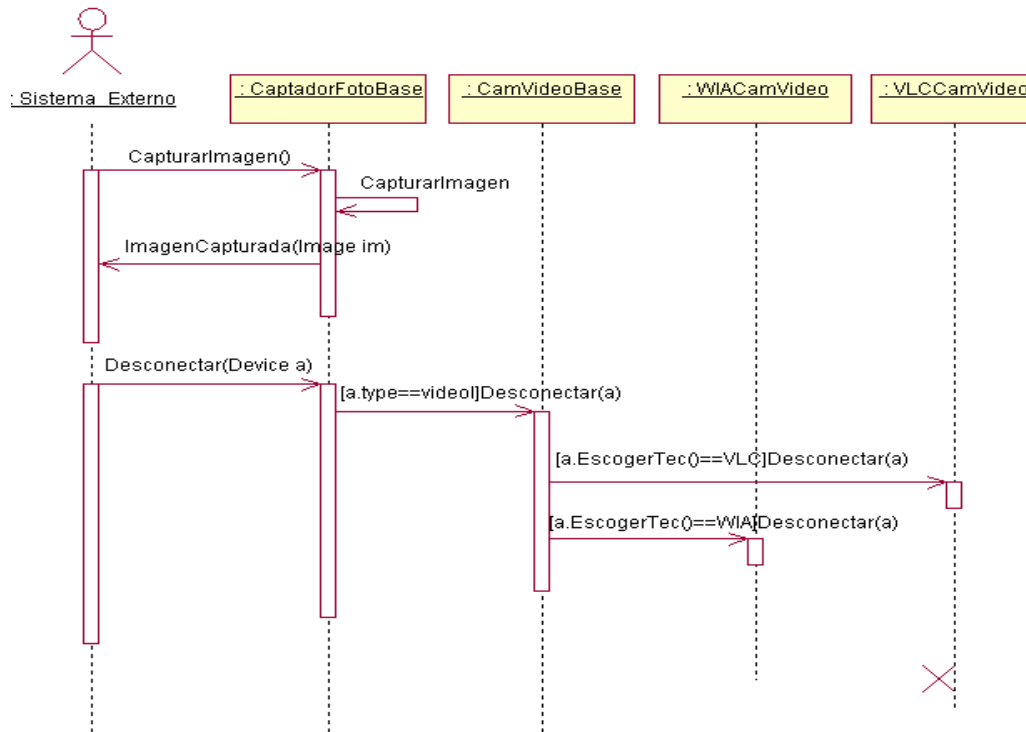


Figura 20. Diagrama de secuencia del CU “Capturar imagen” para cuando es un dispositivo “vídeo”.

Los diagramas de colaboración y los diagramas de secuencia muestran interacciones, pero acentúan aspectos diferentes. Los diagramas de secuencia muestran secuencias de tiempo claramente pero no muestran explícitamente relaciones de objetos. Los diagramas de colaboración muestran las relaciones entre objetos claramente, pero las secuencias de tiempo se deben obtener a partir de números de secuencia. Los diagramas de secuencia son a menudo más útiles para mostrar escenarios; los diagramas de colaboración son a menudo más útiles para mostrar el diseño detallado de procedimientos.(35)

3.4.6 Descripción de las clases.

Seguidamente se exponen descripciones más detalladas de las clases presentes en el diagrama de clases del diseño.

Nombre	CaptadorFotoBase
Tipo de clase	Controladora

Atributo	Tipo
dispositivo	Device
imagen	Image
Para cada responsabilidad	
Nombre:	Conectar()
Descripción:	Operación para conectar el dispositivo de captura utilizando la tecnología WIA.
Nombre:	CapturarImagen()
Descripción:	Operación para capturar la imagen.
Nombre:	ImagenCapturada
Descripción:	Operación para mostrar la imagen capturada.
Nombre:	CapturaVivo()
Descripción:	Operación para mostrar el vídeo en tiempo real.

Tabla 18. Descripción de la clase “CaptadorFotoBase” del diseño.

Nombre	CamDigitalBase
Tipo de clase	Controladora
Atributo	Tipo
Para cada responsabilidad	
Nombre:	AumentarZoom()
Descripción:	Operación para aumentar el zoom del vídeo.
Nombre:	DisminuirZoom()
Descripción:	Operación para disminuir el zoom del vídeo
Nombre:	CrearVideoPorID()
Descripción:	Operación para crear el vídeo.
Nombre:	Reproducir()
Descripción:	Operación para reproducir el vídeo.
Nombre:	EscogerTec()
Descripción:	Operación para escoger si se quiere utilizar WIA o VLC para la reproducción del vídeo.

Tabla 19. Descripción de la clase “CamDigitalBase” del diseño.

Nombre	CamVideoBase
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	CrearVideoPorID()
Descripción:	Operación para crear el vídeo.
Nombre:	Reproducir()
Descripción:	Operación para reproducir el vídeo.
Nombre:	EscogerTec()
Descripción:	Operación para escoger si se quiere utilizar WIA o VLC para la reproducción del vídeo.

Tabla 20. Descripción de la clase “CamVideoBase” del diseño.

Nombre	VLCCamDigital
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	CapturaVivo()
Descripción:	Implementación específica para mostrar el vídeo en tiempo real.
Nombre:	CrearVideoPorID()
Descripción:	Implementación específica para crear el vídeo.
Nombre:	Reproducir()
Descripción:	Implementación específica para reproducir el vídeo.

Tabla 21. Descripción de la clase “VLCCamDigital” del diseño.

Nombre	WIACamDigital
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	CapturaVivo()
Descripción:	Implementación específica para mostrar el vídeo en tiempo real.
Nombre:	CrearVideoPorID()

Descripción:	Implementación específica para crear el vídeo.
Nombre:	Reproducir()
Descripción:	Implementación específica para reproducir el vídeo.

Tabla 22. Descripción de la clase “WIAmCamDigital” del diseño.

Nombre	WIAmCamVideo
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	CapturaVivo()
Descripción:	Implementación específica para mostrar el vídeo en tiempo real.
Nombre:	CrearVideoPorID()
Descripción:	Implementación específica para crear el vídeo.
Nombre:	Reproducir()
Descripción:	Implementación específica para reproducir el vídeo.

Tabla 23. Descripción de la clase “WIAmCamVideo” del diseño.

Nombre	VLCCamVideo
Tipo de clase	Controladora
Para cada responsabilidad	
Nombre:	CapturaVivo()
Descripción:	Implementación específica para mostrar el vídeo en tiempo real.
Nombre:	CrearVideoPorID()
Descripción:	Implementación específica para crear el vídeo.
Nombre:	Reproducir()
Descripción:	Implementación específica para reproducir el vídeo.

Tabla 24. Descripción de la clase “VLCCamVideo” del diseño.

Nombre	IZoom
Tipo de clase	Interfaz
Para cada responsabilidad	
Nombre:	AumentarZoom()

Descripción:	Aumenta el zoom del vídeo visualizado.
Nombre:	DisminuirZoom()
Descripción:	Disminuye el zoom del vídeo visualizado.

Tabla 25. Descripción de la clase “IZoom” del diseño.

Nombre	IFlash
Tipo de clase	Interfaz
Atributo	Tipo
TipoFlash	string
Para cada responsabilidad	
Nombre:	EstablecerModoFlash()
Descripción:	Modifica el tipo de flash.

Tabla 26. Descripción de la clase “IFlash” del diseño.

Nombre	VideoStream
Tipo de clase	Controladora
Atributo	Tipo
FramePSeg	int
Para cada responsabilidad	
Nombre:	ObtenerProxFrame()
Descripción:	Devuelve la imagen en ese instante.

Tabla 27. Descripción de la clase “VideoStream” del diseño.

3.5 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. Cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo hardware similar. El modelo de despliegue en sí mismo representa una correspondencia entre la arquitectura de software y la arquitectura del sistema (el hardware) (33)

Seguidamente se muestra el diagrama de despliegue del componente para la captura de imágenes faciales. En el mismo un nodo representa la computadora donde se encuentra la plataforma para integrar el acceso a dispositivos y al cual se integrará el componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales (PC_Captación de imágenes faciales), comunicándose mediante USB con el nodo del dispositivo (Dispositivo de captura de imágenes).

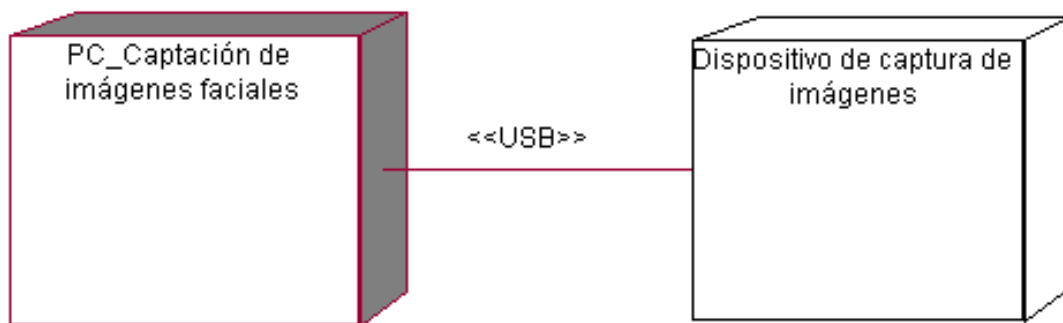


Figura 21. Diagrama de despliegue del componente.

3.6 Conclusiones

En el capítulo se realiza la justificación de la arquitectura utilizada. Se expone el modelo de análisis, se muestran los diagramas de clases del análisis y colaboración para los casos de uso. Se muestra el modelo de diseño, el diagrama de clases del diseño que explica la lógica del componente, a la cual se le aplican patrones con el objetivo de obtener un mejor diseño, además se construyen los diagramas de secuencia del diseño para cada caso de uso. Se hizo una descripción de las clases utilizadas y se especificaron los atributos y operaciones de cada una. De esta forma, queda hecho el análisis y diseño del componente permitiendo dar paso a la realización de la implementación.

Capítulo 4: Estudio de factibilidad y análisis de costos

4.1 Introducción

En el presente capítulo se hace un estudio sobre la factibilidad y análisis de costos del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales, donde se emplea la estimación del mismo a través de la planificación basada en casos de uso. En la realización de un proyecto es necesario el análisis del tiempo de desarrollo en meses, según la cantidad de personas que trabajarán en el mismo, también es necesario conocer el costo total así como los gastos económicos que implicaría. Dicho análisis es esencial para la planificación del equipo de trabajo, para fijar una fecha de terminación del producto y para conocer si es factible o no la realización de la investigación.

4.2 Planificación basada en casos de uso

La metodología empleada para el desarrollo del análisis y diseño del componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales es RUP, donde se especifican las funcionalidades de los sistemas mediante la utilización de casos de uso, lo cual nos posibilita la utilización para el análisis de la factibilidad de la estimación mediante el análisis por Puntos de Casos de Uso.

Dicha estimación es un método propuesto originalmente por Gustav Karner y posteriormente refinado por muchos autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto, mediante la asignación de pesos a un cierto número de factores que lo afectan, para poder contabilizar el tiempo total estimado para el proyecto a partir de estos factores.

Paso 1. Identificar los puntos de casos de usos desajustados

$$UUCP = UAW + UUCW$$

Dónde:

UUCP: Puntos de casos de uso sin ajustar.

UAW: Factor de peso de los actores sin ajustar.

UUCW: Factor de peso de los casos de uso sin ajustar.

Paso 2. Cálculo del factor de peso de los actores sin ajustar (UAW)

El total del Factor de Peso de los Actores sin ajustar (UAW) se calcula mediante un análisis de la cantidad de actores presentes en el sistema, y la complejidad de cada uno de ellos a través de diferentes criterios, los cuales se muestran en la tabla a continuación:

Tipo	Descripción	Peso	Actores	Cant*peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API).	1	1	1*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo, o una persona interactuando a través de una interfaz en modo texto.	2	0	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI).	3	0	0*3
Total				1

Tabla 28. Factor de peso de los actores sin ajustar.

$$UAW = \sum (\text{Peso} * \text{Actores})$$

$$UAW = (1*1) + (0*2) + (0*3)$$

$$UAW = 1$$

Paso 3. Cálculo del factor de peso de los casos de uso sin ajustar (UUCW)

El valor del Factor de Peso de los Casos de Uso sin ajustar (UUCW), se calcula mediante un estudio de los casos de uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad se define teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción es una secuencia de actividades completa.

Los criterios se muestran en la tabla a continuación:

Tipo	Descripción	Peso	Caso de uso	CantCU	CantCU * peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	1. Inicializar dispositivo de captura. 2. Ajustar vídeo. 3. Capturar la imagen.	3	15
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10		0	0
Complejo	El Caso de Uso contiene más de 8 transacciones.	15		0	0
Total					15

Tabla 29. Factor de peso de los casos de uso sin ajustar.

$$UUCW = \sum (\text{Peso} * \text{CantCU})$$

$$UUCW = (3*5) + (0*10) + (0*15)$$

$$UUCW = 15$$

Paso 4 Determinar los puntos de caso de uso sin ajustar (UUCP)

$$UUCP = UAW + UUCW$$

$$UUCP = 1 + 15$$

$$UUCP = 16$$

Paso 5. Ajustar los puntos de casos de uso

Una vez que se obtienen los Puntos de Casos de Uso sin ajustar, se debe ajustar mediante la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

Dónde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Paso 6. Cálculo del factor de complejidad técnica

Para calcular el factor de complejidad técnica (TCF), se cuantifica un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica en un valor entre 0 y 5 donde 0 significa que es un aporte irrelevante y 5 un aporte muy relevante.

Las fórmulas para este punto son:

$$TFactor = Sum (Valor * Peso)$$

$$TCF = 0.6 + (0.01 * TFactor)$$

Peso de los factores de complejidad técnica:

Factor	Descripción	Peso	Valor	Valor*Peso
T1	Sistema distribuido.	2	3	6
T2	Objetivos de performance o tiempo de respuesta.	1	4	4
T3	Eficiencia del usuario final.	1	4	4
T4	Procesamiento interno complejo.	1	4	4
T5	El código debe ser reutilizable.	1	4	4
T6	Facilidad de instalación.	0.5	4	2

Capítulo 4: Estudio de factibilidad y análisis de costos

T7	Facilidad de uso.	0.5	4	2
T8	Portabilidad.	2	4	8
T9	Facilidad de cambio.	1	4	4
T10	Concurrencia.	1	4	4
T11	Incluye objetivos especiales de seguridad.	1	1	1
T12	Provee acceso directo a terceras partes.	1	4	4
T13	Se requiere facilidades especiales de entrenamiento a usuario.	1	3	3
Total(TFactor)				50

Tabla 30. Peso de los factores de complejidad técnica.

Cada uno de estos puntos se debe evaluar según la siguiente escala:

Descripción	Valor
No presente o sin influencia.	0
Influencia incidental o presencia incidental.	1
Influencia moderada o presencia moderada.	2
Influencia media o presencia media.	3

Influencia significativa o presencia significativa.	4
Fuerte influencia o fuerte presencia.	5

Tabla 31. Escala para evaluar los factores.

$$TCF = 0.6 + (0.01 * 50)$$

$$TCF = 1.1$$

Paso 7. Cálculo del factor de ambiente

Para calcular el factor ambiente (EF) se cuantifica con un valor desde a 0 a 5, donde el 0 significa aporte irrelevante y el 5 significa aporte muy relevante. Este factor está relacionado con las habilidades y experiencia del grupo de personas que trabajan en el desarrollo del componente.

Las fórmulas para este punto son:

$$EFactor = \text{Sum (Valor * Peso)}$$

$$EF = 1.4 + (-0.03 * EFactor)$$

Peso de los factores ambientales:

Factor	Descripción	Peso	Valor	Valor*Peso
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	5	7.5
E2	Experiencia en la aplicación.	0.5	2	1
E3	Experiencia en orientación a objetos.	1	3	3
E4	Capacidad del analista líder.	0.5	4	2

E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	4	8
E7	Personal part-time.	-1	4	-4
E8	Dificultad del lenguaje de programación.	-1	5	-5
Total(EFactor)				17.5

Tabla 32. Peso de los factores ambientales.

$$EF = 1.4 + (-0.03 * 17.5)$$

$$EF = 0.875$$

Paso 8 Determinar los puntos de caso de uso ajustados (UCP)

Finalmente, los Puntos de Casos de Uso ajustados resultan:

$$UCP = UUCP * TCF * EF$$

$$UCP = 16 * 1.1 * 0.875$$

$$UCP = 15.4$$

Paso 9. Calcular esfuerzo de implementación

$$E = UCP * CF$$

Dónde:

E: esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

Paso 10. Cálculo del factor de conversión

Para calcular el factor de conversión se cuentan cuántos valores de los que afectan el factor ambiente (E1... E6) están por debajo de la media (3) y los que están por arriba de la media para los restantes (E7 y E8). Si el total es dos o menos se utiliza el factor de conversión 20 Horas-Hombre / Punto de Casos de uso, si el total es tres o cuatro se utiliza el factor de conversión 28 Horas-Hombre / Punto de Casos de uso y si el total es mayor o igual que cinco se recomienda efectuar cambios en el proyecto ya que se considera que el riesgo de fracaso del mismo es demasiado alto. (36)

CF = 28 horas-hombre (ya que el total de los valores que afectan el factor ambiente es igual a tres).

Luego:

$$E = UCP * CF$$

$$E = 15.4 * 28 \text{ horas-hombre}$$

$$E = 431,2 \text{ horas-hombre}$$

Paso 10. Esfuerzo de todo el proyecto

El esfuerzo calculado anteriormente (E) representa una parte del esfuerzo de todo el proyecto, generalmente se considera que es un 40%, el mismo se refiere al esfuerzo para la implementación de las funcionalidades especificadas en los casos de uso.

Actividad	% Esfuerzo	Valor esfuerzo
Análisis	10%	107.8 horas-hombre
Diseño	20%	215.6 horas-hombre
<i>Implementación</i>	<i>40%</i>	<i>431,2 horas-hombre</i>
Prueba	15%	161.7 horas-hombre

Sobrecarga	15%	161.7 horas-hombre
Total	100%	1078 horas-hombre

Tabla 33. Valores del esfuerzo.

Suponiendo que la jornada laboral del proyecto es de 7 horas y un mes tiene como promedio 30 días, de los que 24 son días laborables, la cantidad de horas que puede trabajar una persona en 1 mes es 168 horas. Si el esfuerzo de todo el proyecto es de 1078 horas-hombre y por cada 168 horas se tiene 1 mes, eso daría un esfuerzo de 6,42 mes-hombre. De los resultados obtenidos se llega a la conclusión que con 1 hombre trabajando en el proyecto el mismo se desarrolla en aproximadamente seis y medio meses.

El análisis y diseño se realiza con un esfuerzo de 323.4 horas-hombre, que daría un tiempo de trabajo de 1,925 mes-hombres. Es decir el análisis y diseño se realiza en dos meses. Valorando que primeramente tiene que hacerse una fundamentación teórica, realizar entrevistas, luego realizar un levantamiento de requisitos y definir los casos de uso (debido al empleo de la metodología RUP), y que para esto se necesita un aproximado de 6 meses, para un total de 8 meses.

4.3 Beneficios tangibles e intangibles

Los beneficios pueden ser tangibles e intangibles. Los beneficios tangibles son los que reportan ventajas económicas cuantificables y los beneficios intangibles son los que reportan beneficios organizativos de funcionamiento o eficiencia.

Los beneficios que se reportan hasta el momento son intangibles:

- Se cuenta con un diseño de un componente que permite la comunicación con múltiples modelos de cámaras digitales.
- Contribuye a la calidad de las soluciones de identificación que se desarrollan en el CISED.
- Aumenta la experiencia del equipo de trabajo, con respecto a las tecnologías que se estudian en la fundamentación teórica.

Con el desarrollo y funcionamiento del componente automatizado para la captura de imágenes faciales con múltiples modelos de cámaras digitales se evidencian otros beneficios en este caso tangibles.

Entre los beneficios tangibles que se obtendrán serán:

- La dirección de la UCI decidirá la comercialización o no del software una vez terminado, incurriendo en beneficios y aportes en la economía de la misma y del país.
- Traerá menos gastos para la UCI y el país, ya que se contará con más opciones de compra de cámaras digitales.
- Aumentará la flexibilidad a la empresa comercializadora Albet, en la adquisición de cámaras digitales para la integración de soluciones de identificación.

4.4 Análisis del costo

El análisis del costo es esencial para determinar si vale el trabajo o no, desarrollar la investigación, aparte ayuda a determinar el costo del proyecto y su mantenimiento ya que todos los proyectos informáticos llevan incluido costo monetario, recursos y esfuerzo.

Seguidamente se muestran una serie de indicadores que se utilizan en el cálculo del costo de la investigación.

CH = Cantidad de hombres.

SM = Salario mensual.

MT = Meses de trabajo.

CP = Costo del proyecto.

DPC = Depreciación de computadora.

GC = Gasto de corriente.

CantP = Cantidad de PC.

Asumiendo que el salario básico de un estudiante (SM) es de \$100.00 mensual, la depreciación de la computadora (DPC) es de \$10.00 por mes y el gasto de corriente de la computadora (GC) es de \$15.00 por mes y solo se usa una computadora (CantP).

$$CP = (SM*CH + DPC*CantP + GC*CantP)*MT$$

$$CP = (100*1 + 10*1 + 15*1)*8$$

$$CP = 125 *$$

$$CP = 1000$$

De los resultados obtenidos se interpreta que con un estudiante trabajando en la investigación, el mismo se desarrolla en aproximadamente 8 meses y que su costo total se estima que sea \$1000.00.

4.5 Análisis de costos y beneficios

El diseño del componente para la captura de imágenes faciales mediante la utilización de tecnologías genéricas, propiciará la comunicación con más modelos de cámaras digitales, ya que actualmente se permite la interacción con dos, beneficiando las soluciones de identificación existentes en el CISED, además propiciará un componente que se ajuste a la arquitectura de integración de dispositivos existente en el mismo, y se cuenta con los integrantes, herramientas y tecnologías necesarias para la implementación del componente, contribuyendo esto que no se tenga que incurrir en gastos de hardware o software.

Además la universidad puede vender la plataforma de integración de dispositivos, o un sistema completo que contenga la plataforma, o puede vender solamente el componente. En general son varios los beneficios intangibles y tangibles como ya vimos en epígrafes anteriores.

Teniendo en cuenta el análisis realizado de los beneficios que reporta el diseño del sistema y que no fue necesario realizar inversiones en equipos técnicos para el desarrollo del producto, se concluye que la investigación es factible.

4.6 Conclusiones

La factibilidad de un proyecto es un paso muy importante en su desarrollo pues a partir de este punto se decide si se continuará el proyecto o no. En el presente capítulo se realiza el estudio de factibilidad mediante el método por Punto de Casos de Uso al componente propuesto, además se realiza un análisis del costo y de los beneficios del proyecto, teniendo en cuenta el costo estimado para llevarlo a cabo y los beneficios que reporta y que reportará al ser implantado. Se llegó a la conclusión de que es factible realizar el análisis y diseño del componente por los beneficios que reporta.

Conclusiones

Primeramente se llegó a la conclusión de que es necesario el desarrollo de un componente para la captura de imágenes faciales ya que estos son cotizados a precios muy altos en el mundo, convirtiéndose en casi inaccesibles para Cuba. Es por ello la necesidad de diseñarlo y que se integre a la plataforma de integración de dispositivos existente en el CISED.

Fueron definidos los requerimientos funcionales y no funcionales, estructurando los primeros en el modelo de casos de uso del sistema y describiéndose los mismos para una mejor comprensión. Luego se realizó el análisis y diseño del componente cumpliendo con los patrones de diseño utilizados, generándose artefactos como los diagramas de clases del análisis, los diagramas de clases del diseño, diagramas de secuencia del diseño, entre otros. Se efectuó un análisis de la factibilidad de la investigación, de los costos y de los beneficios, con lo cual se obtuvo el costo estimado de la realización de la misma concluyendo que es factible y beneficioso para la UCI. El componente que se propone dará solución a la problemática que le dio origen anteriormente y se concluye que el objetivo propuesto para el presente trabajo ha sido cumplido satisfactoriamente.

Seguidamente se realizan una serie de recomendaciones que han de tomarse en cuenta para la continuación de la investigación y el desarrollo de los próximos ciclos de desarrollo.

Recomendaciones

Una vez culminada la realización del presente trabajo investigativo, y alcanzado el objetivo propuesto en el mismo, se sugieren las siguientes recomendaciones:

- Realizar la implementación del componente propuesto con el diseño realizado, de modo que se obtenga un componente genérico, adaptable a la arquitectura de integración de dispositivos existente en el CISED y que pueda usarse mediante la realización de pequeños cambios y reconfiguraciones en empresas e instituciones de nuestro país y en cualquier país del mundo.
- Continuar el desarrollo del componente, mediante la realización de nuevas iteraciones e incluyendo otras funcionalidades como aumentar y disminuir el brillo, el contraste y el color.
- Continuar extendiendo el componente mediante el uso de otras tecnologías genéricas.

Bibliografía referenciada

1. GARCÍA, J. L. G.; CANO, M. J. S., et al. La televisión y el Vídeo. Nº Disponible en: http://www.uhu.es/62024/factoria/2006_2007/edu_musical/materiales/4-TV_y_Video.pdf
2. STRIZINEC, G. Fotografía digital. Editado por: Ra-Ma. Libros de Informática, Computación y Empresa.
3. BARCO NEBRED, L.; IGLESIAS HERNÁNDEZ, C., et al. Fotografía digital frente a fotografía convencional en dermatología. Piel, 2001, vol. 16, nº 2, p. 60-62 Disponible en: <http://www.elsevier.es/es/revistas/piel-21/fotografia-digital-frente-fotografia-convencional-dermatologia-10022518-editorial-2001>
4. Documento de viaje de lectura mecánica Tercera edición ed. Organización de Aviación Civil Internacional, 2008. Vol. 1, 124 p. ISBN 978-92-9231-339-5 Disponible en: http://www.google.es/url?Sa=t&source=web&cd=1&ved=0cbkqfjaa&url=http%3A%2F%2Fwww2.icao.int%2Fen%2FMRTD%2Fdownloads%2Fdoc%25209303%2Fdoc%25209303%2520Spanish%2Fdoc%25209303%2520Part%25203%2520Vol%25201.pdf&rct=j&q=Documento%20de%20viaje%20de%20lectura%20mec%C3%a1nica%20Tercera%20edici%C3%b3n&ei=Tr34TbHxBIHs0gGBraCuDw&usq=afqjcnfm1zfbos_tkscwxxotgs5njoqxhw&cad=rja
5. AVANZADA, S. D. F. E. Procedimiento para la captura de elementos biométricos. 2006, nº Disponible en: http://www.google.es/url?Sa=t&source=web&cd=1&ved=0ccoqfjaa&url=http%3A%2F%2Fwww.cidge.gob.mx%2Fdoc%2FF8Pc_cap_ebi.pdf&rct=j&q=PROCEDIMIENTO%20PARA%20LA%20CAPTURA%20DE%20ELEMENTOS%20BIOM%C3%89TRICOS&ei=bi7tze4iil00gghvbwfca&usq=afqjcnquepswlmnkam_QIM01ls_nql_uaa&cad=rja
6. REYES LÓPEZ, C. Reconocimiento facial mediante visión artificial. Tutor: Camacho, D. E. F. Ingeniería de telecomunicación, Departamento de Ingeniería de sistemas y automática. Universidad de Sevilla, 2005 Disponible en: http://www.google.es/url?Sa=t&source=web&cd=2&ved=0cdiqfjab&url=http%3A%2F%2Fbibing.us.es%2Fproyectos%2Fabreproy%2F11025%2Ffichero%2Freconocimiento%2bfacial%2bmediante%2bvisi%25f3n%2bartificial%25252fportada.pdf&rct=j&q=Reconocimiento%20facial%20mediante%20visi%C3%b3n%20artificial&ei=vl74tcvrc4hs0gqbracudw&usq=afqjcnh7ffk4jsjun6a_afhf3wxqws36tw&cad=rja

7. PAVÓN, J. Y HIDALGO, A. La dimensión estratégica de la innovación tecnológica nº p. 17. Disponible en: http://www.google.es/url?Sa=t&source=web&cd=10&ved=0cfwqfjaj&url=http%3A%2F%2Faprendeonline.udea.edu.co%2Fms%2Fmoodle%2Fmod%2Fresource%2Fview.php%3Finpopup%3Dtrue%26id%3D74077&rct=j&q=Tecnolog%C3%adas%20Gen%C3%a9ricas&ei=w4t9tbbafcw40qgggothaaw&usq=afqjcne1qzux9_n4oo92ypcihp7cocnb_A&cad=rja
8. ALONSO, F. J. S. Tecnologías de Streaming. Oviedo, 2009, 56 p. Disponible en: <http://www.atc.uniovi.es/teleco/5tm/archives/8streaming.pdf>
9. ORTEGA GALLARDO, G. Seguimiento de Objetos Móviles en Tiempo Real aplicando el Filtro de Partículas. Ingeniería técnica en informática de sistemas Departamento de Informática, Estadística y Telemática. Universidad Rey Juan Carlos, 2005 Disponible en: http://www.google.es/url?Sa=t&source=web&cd=4&ved=0cdmqfjad&url=http%3A%2F%2Fwww.etsii.urjc.es%2F~jjpantrigo%2fpcfs%2fseguimientotporealjun2005.pdf&rct=j&q=Seguimiento%20de%20Objetos%20M%C3%b3viles%20en%20Tiempo%20Real%20aplicando%20el%20Filtro%20de%20Part%C3%adculas&ei=NL_4tyvcl-bb0ghsp_iudw&usq=afqjcnepipdc6hyoiv4anypfuskj9eeazq&cad=rja
10. DALLA VECCHIA, G.; CORBO, D., et al. Modelado de la topología de una red multicast: Análisis e implementación. 2005, nº p. 38. Disponible en: http://iie.fing.edu.uy/ense/assign/perfredes/trabajos/trabajos_2004/inferenciadearbolesmulticast/inferenciadearbolesmulticast.pdf
11. MATEOS, M. R.; FUENTE, A. I., et al. Mateos, m. R.; fuente, a. I., et al. Integración de componentes COM de matlab/simulink en el entorno Case xbdk, para el modelado de sistemas de conformación de haz matlab/simulink COM Component Integration for xbdk Case environment, Oriented to beamforming applications Ingeniare. Revista chilena de ingeniería, 2009, vol. 17, nº 1, p. 129-135. Disponible en: http://www.scielo.cl/scielo.php?Pid=S0718-33052009000100014&script=sci_pdf&tlng=es
12. Application Programing Interface [Consultado en: enero de 2011]. Disponible en: <http://msdn.microsoft.com/es-es/library/172wfck9%28v=vs.80%29.aspx>
13. MICROSOFT, S. WD2000: Compatible con Microsoft Word 2000 admite TWAIN escáneres y cámaras [Consultado el: enero de 2011]. Disponible en: <http://support.microsoft.com/kb/240183/es>

14. Windows Image Acquisition [Consultado en: enero de 2011]. Disponible en: [http://msdn.microsoft.com/en-us/library/ms630368\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms630368(v=vs.85).aspx)
15. WHITESIDE, D. Microsoft Working With Imaging Industry to Deliver Enhanced Imaging Functionality. Los ángeles - Abril 7, 1999 n° Disponible en: <http://www.microsoft.com/presspass/press/1999/apr99/wnimagpr.msp>
16. TWAIN. [Consultado en: enero de 2011]. Disponible en: [http://msdn.microsoft.com/en-us/library/ff548446\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ff548446(v=vs.85).aspx)
17. Still Image [Consultado en: enero de 2011]. Disponible en: <http://msdn.microsoft.com/en-us/library/cc836543%28v=VS.85%29.aspx>
18. GUGLIELMETTI, M. Definición de driver. El 11 de Febrero de 2005 ed. MASTERMAGAZINE, publicado el: enero del 2011 de 2005, última actualización: enero del 2011. Disponible en: <http://www.mastermagazine.info/termino/?S=driver>
19. MONTILVA, J.; ARAPÉ, N., et al. Desarrollo de Software Basado en Componentes. En 2003. P. 9. Disponible en: <http://ws-01.ula.ve/ingenieria/jonas/Productos/Publicaciones/Congresos/CAC03%20Desarrollo%20de%20componentes.pdf>
20. CASTRO GIL, R. A. Estructura básica del proceso unificado de desarrollo de software. Universidad Icesi: 2004, Disponible en: http://www.google.es/url?Sa=t&source=web&cd=4&ved=0cdmqfjad&url=http%3A%2F%2Fwww.icesi.edu.co%2Fbiblioteca_digital%2Fbitstream%2Fitem%2F399%2F1%2Fcastro_estructura-bas-puds.pdf&rct=j&q=proceso%20unificado%20de%20desarrollo%20&ei=czi9teevppcw0gha8ittaw&usq=afqjcnqfwdec6kqm4dxbebzis_medgc4q&cad=rja.
21. FERNÁNDEZ ESCRIBANO, G. Introducción a Extreme Programming. En Ingeniería del Software 2. 2002 Disponible en: <http://www.google.es/url?Sa=t&source=web&cd=1&ved=0cbwqfjaa&url=http%3A%2F%2Fwww.dsi.uclm.es%2Fasignaturas%2F42551%2Ftrabajosanteriores%2Ftrabajo-XP.pdf&rct=j&q=%20Introduccion%20a%20Extreme%20Programing.&ei=P8D4TZX8D6LN0gGt48WuDw&usq=afqjcnej740ftdzkzmywpbpmqaazdiqb0q&cad=rja>.

22. CUERVO, M. C. Y MORENO, O. Y. B. Herramientas libres para modelar software Free tools to model software. Revista Facultad de Ingeniería, 2005, vol. 14, nº p. 6. Disponible en: http://ervisual.hostoi.com/pdf/er_tecnico.pdf
23. GÉNOVA, G.; FUENTES, J. M., et al. Evaluación de herramientas CASE para UML. Nº.
24. SÁNCHEZ, S. H. Y ROMÁN, F. B.NET Framework. 2005, nº Disponible en: <http://zarza.fis.usal.es/~fgarcia/docencia/poo/04-05/Trabajos/Framework.NET.pdf>
25. VICENTE, M. Á. G. Y MOHEDANO, M. M. Introducción a C#. 2005, nº Disponible en: http://zarza.usal.es/~fgarcia/docencia/poo/04-05/Trabajos/C_Sharp.pdf
26. TEMPLEMAN, J. Y VITTER, D. Visual Studio. NET. 2002, nº ISSN 8441513805 Disponible en: <http://dialnet.unirioja.es/servlet/libro?Codigo=319553>
27. Visual studio 2008. [Consultado en: enero de 2011]. (MSDN). Disponible en: <http://msdn.microsoft.com/es-es/vstudio/aa700830.aspx>
28. CRISTIÁ, M. Introducción a la Arquitectura de Software. 2008, nº p. 8. Disponible en: <http://www.google.es/url?Sa=t&source=web&cd=4&ved=0CC8QFjAD&url=http%3A%2F%2Fwww.willydev.net%2Finsitecreation%2Fv1.0%2Fwillycrawler%2F2008.05.01.articulo.introduccion%2520a%2520la%2520arquitectura%2520de%2520soft.pdf&rct=j&q=Introduccion%20ala%20arquitectura%20del%20software&ei=m2ujtzyjispg0qgs5z2gbq&usq=afqjcnqz2tnsisamudu1kesuxfpntatnkw&cad=rja>
29. KICILLOF, C. R. N. Estilos arquitectónicos. Nº Disponible en: <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>
30. MARTÍNEZ, A. Y MARTÍNEZ, R. Guía a Rational Unified Process Albacete: Universidad de Castilla la Mancha. Disponible en: <http://www.google.es/url?Sa=t&source=web&cd=1&ved=0cbkqfjaa&url=http%3A%2F%2Fwww.dsi.uclm.es%2Fasignaturas%2F42551%2Ftrabajosanteriores%2Ftrabajo-Guia%2520RUP.pdf&rct=j&q=%20MART%20C%20R.%20Gu%20a%20Rational%20Unified%20Process%20%20Albacete%20%20Universid%20de%20Castilla%20la%20Mancha&ei=fsp4tci-BKLN0qGz48WuDw&usq=afqjcnhiwlnpx2kba4jn3jhpm8hspheo0a&cad=rja>
31. Introducción a la Disciplina de Requisitos de RUP. En 2010. Disponible en: <http://eva.uci.cu/mod/resource/view.php?id=34734>

32. Introducción a la disciplina de Análisis y Diseño. En 2010. Disponible en: <http://eva.uci.cu/mod/resource/view.php?id=35381>
33. JACOBSON, I.; BOOCH, G., et al. El Proceso Unificado de Desarrollo de Software Editado por: Wesley, A. 2000. Disponible en: [http://eva.uci.cu/mod/resource/view.php?id=21621&subdir=/El proceso unificado de desarrollo de software](http://eva.uci.cu/mod/resource/view.php?id=21621&subdir=/El%20proceso%20unificado%20de%20desarrollo%20de%20softw)
34. MOSTOW, J. S. Toward Better Models of the Design Process. Editado por: Aimugaiine. 1985. Vol. 1. Disponible en: http://www.google.es/url?Sa=t&source=web&cd=1&ved=0cciqfjaa&url=http%3A%2F%2Fwww.aaai.org%2Fdocs%2Findex.php%2Faimagazine%2Farticle%2Fview%2F468&rct=j&q=Toward%20Better%20Models%20of%20the%20Design%20Process&ei=PMX4TZyFubB0gHsp_iudw&usg=afqjcnhcd4gh6oaua3frg9vatjimkdfq&cad=rja
35. JACOBSON, I.; BOOCH, G., et al. El lenguaje Unificado de Modelado. Manual de referencia. Editado por: Wesley, A. 528 p Disponible en: [http://eva.uci.cu/mod/resource/view.php?id=21621&subdir=/El lenguaje unificado de modelado. Manual de Referencia](http://eva.uci.cu/mod/resource/view.php?id=21621&subdir=/El%20lenguaje%20unificado%20de%20modelado.%20Manual%20de%20Referencia)
36. PERALTA, M. Estimación del esfuerzo basada en casos de uso. Reportes Técnicos en Ingeniería de Software. Buenos Aires-Argentina, 2004, vol. 6, nº 1, p. 1-16 Disponible en: <http://pisuerga.inf.ubu.es/rcobos/anis/estimacion-del-esfuerzo-basada-en-casos-de-usos.pdf>
37. SOFTWARE, M. D. I. D. Requisitos de Sistemas de Software. Validación de requisitos En Ingeniería de software. Disponible en: http://www.google.es/url?Sa=t&source=web&cd=9&ved=0cfuqfjai&url=http%3A%2F%2Fis.fi.upm.es%2Fdocs%2Fmasterti%2FARS%2Fdocs%2Fmanual_M2C1U11.pdf&rct=j&q=Requisitos%20de%20Sistemas%20de%20Software.%20Validaci%C3%B3n%20de%20requisitos%20&ei=8MX4Tf2rA-iy0qhnvpmudw&usg=afqjcnf0h8enexjjbuxyexqjzambnw58w&cad=rja

Bibliografía consultada

1. Component Object Model [Consultado en: enero del 2011] Disponible en: <http://msdn.microsoft.com/es-es/library/bd9cdfyx%28v=VS.80%29.aspx>
2. CANÓS, J.; LETELIER, P., et al. Metodologías Ágiles en el desarrollo de Software. Metodologías Ágiles en el Desarrollo de Software, nº p. 1. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.4553&rep=rep1&type=pdf#page=9>
3. ARIAS CHAVES, M. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. Universidad de Costa Rica: 2005, Disponible en: <http://www.google.es/url?sa=t&source=web&cd=6&ved=0CEQQFjAF&url=http%3A%2F%2Fredalyc.uaemex.mx%2Fpdf%2F666%2F66661111.pdf&rct=j&q=T%C3%A9nicas%20utilizadas%20en%20las%20actividades%20de%20la%20ingenier%C3%ADa%20de%20requisitos&ei=mavATdXTAoXWtQOOzbzqBw&usq=AFQjCNHOpSucJ8t9VkmR9Nrkm7ZsEzA2Vg&cad=rja>
4. BARTOLOMÉ, A. Vídeo digital. 2003, nº p. 10. Disponible en: http://dialnet.unirioja.es/servlet/dfichero_articulo?codigo=755200&orden=77210
5. CALERO SOLÍS, M. Una explicación de la programación extrema (XP). En 2003 Disponible en: http://www.google.es/url?sa=t&source=web&cd=1&ved=0CBkQFjAA&url=http%3A%2F%2Fwww.willydev.net%2Fdescargas%2Fprev%2FExplicaXP.pdf&rct=j&q=Una%20explicaci%C3%B3n%20de%20la%20programaci%C3%B3n%20extrema%20%28XP%29%20&ei=usv4TfmrCuiy0QHNVpmuDw&usq=AFQjCNEsTjXb0ziWCqYBy_k2mQxcf5MUqw&cad=rja
6. CENATAV. (2004), (Citado en: enero del 2011). Disponible en: <http://www.cenatav.co.cu./es/publicaciones.html>
7. CORPORATION, M. Introduction to WIA 2.0. 2007, Disponible en: <http://download.microsoft.com/download/0/0/b/00bba048-35e6-4e5b-a3dc-36da83cbb0d1/WIA20-intro.doc>
8. DE JAVA, O. Java y la Programación Orientada a Objetos. nº, Disponible en: http://www.javahispano.org/contenidos/archivo/37/poo_java.pdf .
9. DURAN-LIMON, H.; MEDA-CAMPAÑA, M. E., et al. Un marco de trabajo de una fábrica de software para el reuso del diseño arquitectónico y de componentes de software. 2008, Disponible en: http://tecnociencia.uach.mx/numeros/v2n1/data/marco_de_trabajo_de_una_fabrica_de_sw.pdf

10. ENTERPRISE, G. S. G. R. R. [Consultado en: enero del 2011 Disponible en: <http://www.rational.com.ar/herramientas/roseenterprise.html>]
11. ESCORSA, P. y DE LA PUERTA, E. La estrategia tecnológica de la empresa: una visión de conjunto. Inteligencia competitiva, nº Disponible en: <http://www.temarium.com/serlibre/recursos/pdf/79059.Inteligencia%20Competitiva.Lecturas.pdf#page=5>
12. FERNÁNDEZ-BOZA, J. Fotografía digital: ventajas e inconvenientes. Rev Esp Ortod, 2004, vol. 34, nº p. 335-341. Disponible en: http://www.revistadeortodoncia.com/files/2004_34_4_335-341.pdf
13. GRACIA, J. Patrones de diseño. Diseño de software orientado a datos. Mayo 2005 ed. publicado en: enero 2011 de 2005, última actualización: enero 2011. Disponible en: <http://www.ingenierossoftware.com/analisisydiseno/patrones-diseno.php>
14. IEEE. ESPECIFICACIONES DE LOS REQUISITOS DEL SOFTWARE 1998, Disponible en: http://www.google.es/url?sa=t&source=web&cd=1&ved=0CBwQFjAA&url=http%3A%2F%2Fwww.ctr.unica.n.es%2Fasignaturas%2Fis1%2FIEEE830_esp.pdf&rct=j&q=IEEE.%20ESPECIFICACIONES%20DE%20LOS%20REQUISITOS%20.%201998&ei=vY_MTc6GDJKztwfcz-2OBQ&usq=AFQjCNFQjS-GUNF2_qjHn-7suw9L6thw3g&cad=rja
15. JOVER, J. N. La ciencia y la tecnología como procesos sociales. Universidad de la Habana, 1998, nº p. 1-21. Disponible en: <http://www.indernet.cu/indernet/Provincias/hlg/documetos/textos/P%20S%20DE%20LA%20CIENCIA%20Y%20LA%20TECNOLOGIA/P%20S%20DE%20LA%20CIENCIA%20Y%20LA%20TECNOLOGIA.pdf>
16. M.J. ESCALONA, N. K. Ingeniería de Requisitos en Aplicaciones para la Web: Un estudio comparativo Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Disponible en: http://www.sistemas.edu.bo/lalgado/sis3390/Requisitos/Ingenier_a%20de%20Requisitos%20en%20Aplicaciones%20Web%20-%20Un%20estudio%20compatativo.pdf
27. MARTÍNEZ, A. M. Y. R. Guía a Rational Unified Process Universidad de Castilla la Mancha Disponible en: <http://www.google.es/url?sa=t&source=web&cd=1&ved=0CBcQFjAA&url=http%3A%2F%2Fwww.info-ab.uclm.es%2Fasignaturas%2F42551%2FtrabajosAnteriores%2FTrabajo-Guia%2520RUP.pdf&rct=j&q=Alejandro%20Mart%C3%ADnez%20y%20Ra%C3%BAI%20Mart%C3%ADnez%20.Gu%C3%ADa%20a%20Rational%20Unified%20Process%20%20&ei=4zSITZzcDOTW0QHZisjoC>

[A&usg=AFQjCNFT9n8vfRkCjza8yA7FBqoVxvo-gA&cad=rja](#)

18. MONCAYO, G. Curso Fundamentos de Programación en Java. nº. Disponible en: http://www.e-socrates.org/pluginfile.php/42767/mod_resource/content/0/MaterialNov2010/Capitulo1.pdf
19. MORENO, Á. y RENÉ, M. Procesamiento y gestión digital de la información. Biblioteca Universitaria, vol. 9, nº 2, p. 8. ISSN 0187-750X. Disponible en: <http://revistas.unam.mx/index.php/rbu/article/viewFile/24997/23435>
20. MORENO, A. M. y SÁNCHEZ-SEGURA, M. Patrones de Usabilidad: Mejora de la Usabilidad del Software desde el momento de Arquitectónico. VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD'03). Alicante, 2003, nº. Disponible en: <http://grise.upm.es/miembros/ana/docs/jis03-0.pdf>
21. ORTEGA, S. S. y GÜIL, F. M. Fotografía en tricología. Piel, 2004, vol. 19, nº 4, p. 177-178. Disponible en: <http://www.elsevier.es/sites/default/files/elsevier/pdf/21/21v19n04a13059514pdf001.pdf>
22. PEÑALVO, F. J. G.; GARCÍA, M. N. M., et al. ADAM CASE. nº. Disponible en: <http://avellano.usal.es/~nines/d.publica/adamcase.pdf>
23. PRESSMAN, R. S. Ingeniería del Software. Un Enfoque Práctico. Editado por: Ince, M. G. H. A. P. D.
24. Ingeniería del Software. Un Enfoque Práctico. En Ince, A. P. D. (editor). Mc Graw Hill, Quinta Edición, p. 614.
25. REYNOSO, C. B. Introducción a la Arquitectura de Software. Documento de la Universidad de Buenos Aires. 2004, vol. 21, nº 01, p. 2009. Disponible en: <http://carlosreynoso.com.ar/wp-content/plugins/google-document-embedder/pdf.php%3Ffile%3Dcarlosreynoso.com.ar/archivos/arquitectura/Arquitectura-software.pdf%26fn%3DArquitectura-software.pdf>
26. ROVERE, M. Redes nómades, algunas reflexiones desde una práctica de intervención institucional nº. Disponible en: http://guajiros.udea.edu.co/fnsp/cvsp/politicaspUBLICAS/0010.rovere_redes_nomades.pdf
27. RUP, C. I. I. E. L. P. U. R. INGENIERIA DE PROCESOS DE SOFTWARE. nº. Disponible en: http://www.oocities.org/hwfiestasb/CatedrasUNP/IngenieriaProcesosSoftware/Syllabus_IPS.doc
28. SCHMIDT, D. C. Pattern-oriented Software Architecture: A system of patterns. John Wiley & Sons Inc, 1996. vol. 1, ISBN 0471958697.

29. SECO, J. A. G. El lenguaje de programación C#. Documento en formato digital, 2001, n°. Disponible en: http://campus.instituto.almagro.ort.edu.ar/analista-sistemas-taller-4/descargar/repositorioarchivo/30864/C_sharp_El_lenguaje_de_programacion_C.pdf
30. TEAMS, D. Rational Unified Process, Best Practices for Software. n° Disponible en: http://www.google.es/url?sa=t&source=web&cd=15&ved=0CDQQFjAEOAo&url=http%3A%2F%2Fwww.ibm.com%2Fdeveloperworks%2Ffractional%2Flibrary%2Fcontent%2F03July%2F1000%2F1251%2F1251_bestpractices_TP026B.pdf&rct=j&q=RUP&ei=-5F9TbvMI4aE0QHcwdnTAW&usq=AFQiCNGblel67UmHOuE3zjxke6cH_hlBIA&cad=rja
31. THOMAS, H. De las tecnologías apropiadas a las tecnologías sociales. Conceptos/estrategias/diseños/acciones. Ponencia presentada en la 1ra Jornada sobre Tecnologías Sociales, Programa Consejo de la Demanda de Actores Sociales-Ministerio de Ciencia, Tecnología e Innovación Productiva, Buenos Aires, 2009, n°. Disponible en: http://www2.mincyt.gob.ar/documentos/Jornadas_tecno_soc_Hernan_Thomas.pdf
32. VILLAVICENCIO, V.; CARLOS, J., et al. Diseño Y Desarrollo De Un Sistemas De Señalética Digital (Digital Signage) Usando Volcado De Vídeo. 2006, n°. Disponible en: <https://www.dspace.espol.edu.ec/handle/123456789/3039>

Anexos

Anexo 1

Directrices ilustrativas para retratos en los DVLM

Las ilustraciones de las páginas siguientes proporcionan orientación para la captación de fotografías que han de emplearse como retrato del titular en los DVLM (Documento de viaje de lectura mecánica – término genérico que abarca pasaportes, visados y tarjetas)

- 1.1 Deberá mostrar una toma de primer plano de la cabeza y los hombros.
- 1.2 La fotografía debe captarse de modo que una línea horizontal imaginaria entre los centros de los ojos resulte paralela al borde superior de la imagen.
- 1.3 El rostro debe estar enfocado con total nitidez y no deberán aparecer imperfecciones como manchas de tinta o arrugas.
- 1.4 La fotografía deberá mostrar al sujeto en pose frontal y mirando directamente a la cámara con expresión neutra y boca cerrada.
- 1.5 La distancia desde el mentón hasta la coronilla (coronilla es la parte superior de la cabeza si no hubiera cabello) ocupará del 70% al 80% de la dimensión vertical de la fotografía.
- 1.6 Los ojos deben estar abiertos y no debe haber cabello que los oculte.
- 1.7 Si el sujeto usa gafas, la fotografía debe mostrar los ojos claramente sin reflejos luminosos en ellas. Las gafas no tendrán lentes coloreadas. De ser posible se evitarán las monturas (armazones) gruesas y se asegurará de que éstas no cubren parte alguna de los ojos.
- 1.8 No se permiten tocados, cabello, peinados o adornos faciales que oculten el rostro.
- 1.9 La fotografía tendrá un trasfondo monocromático de color claro.
- 1.10 No habrán otras personas u objetos en la fotografía.
2. Iluminación, exposición y equilibrio cromático
 - 2.1 La iluminación será uniforme sin sombras o reflejos sobre el rostro o en el trasfondo.
 - 2.2 Los ojos del sujeto no deben aparecer rojos.
 - 2.3 La fotografía debe tener brillo y contrastes apropiados
 - 2.4 Cuando la fotografía sea a color, el proceso de iluminación y fotográfico debe equilibrar los colores para producir tonos cutáneos naturales.

Anexo 2

Fotografía

2.1 Pose

Se deberá utilizar la pose frontal o de cara completa, verificando que la persona se encuentre bien sentada, recargada y con la mirada al frente.

2.2 Enfoque

- En las imágenes faciales, la cara del sujeto siempre debe estar enfocada de la nariz hacia las orejas.
- La distancia a la que se debe encontrar la cámara no será mayor a un metro.
- La imagen facial en el momento de ser capturada debe cumplir las siguientes condiciones:
 - Una línea vertical imaginaria entre el punto medio horizontal de la boca que pase por el puente de la nariz y que coincida con el centro horizontal de la imagen, como se muestra por la línea *AA* de la Figura 22.
 - Una línea horizontal imaginaria a través del centro de los ojos de la persona se situará aproximadamente en el 55% de la distancia vertical hacia arriba desde el borde inferior de la imagen capturada, como se muestra por la línea *BB* de la Figura 22.
 - El ancho de la cabeza de la persona ocupará aproximadamente el 50% del total del ancho de la fotografía. Este ancho será la distancia horizontal entre los puntos medianos de dos líneas verticales imaginarias. Cada línea imaginaria será dibujada entre los lóbulos superiores y más bajos de cada oído y colocada donde el oído externo conecta con la cabeza, como se muestra por la línea *CC* de la Figura 22.

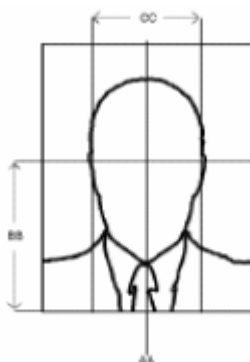


Figura 22. Proporción del rostro.

2.3 Iluminación

Para tomar la fotografía se recomienda una iluminación suficiente y se deben utilizar las técnicas adecuadas para tratar de eliminar cualquier tipo de sombra o manchas que pudieran aparecer.

2.4 Fondo

Deberá usarse al 18 % de gris en una superficie totalmente plana.

2.5 Relación de aspecto

La relación de aspecto de la imagen “ancho:largo” estará de acuerdo con el estándar universal de la cámara fotográfica de 3:4.

La manipulación de la imagen original capturada antes de la compresión y almacenamiento es permitida, a condición de que la técnica de la modificación mantenga la relación de aspecto especificado de 3:4 y la imagen sea almacenada con la misma relación.

2.6 Formato de la imagen

La fotografía deberá ser capturada en formato JPEG.

2.7 Bordes

No se permitirá ningún borde o marco de color que rodee la imagen facial del retrato.

2.8 Resolución

Un mínimo de resolución de 480 x 600 píxeles para la imagen digital (480 horizontal o ancho x 600 vertical o alto).

Anexo 3

3. Componentes del sistema

3.2. Entorno de trabajo

3.2.1. Fuente de iluminación

En todo proceso de visión artificial es fundamental la elección del más adecuado tipo de fuente de iluminación, ya que la obtención de resultados satisfactorios dependerá en gran medida de ello. Puede disminuirse de manera significativa efectos indeseados como imágenes con un bajo contraste, brillos y sombras. Un sistema de luces apropiado simplifica la imagen del objeto a estudiar aportando información de mayor calidad para la detección y extracción del mismo, haciendo innecesaria la aplicación de algoritmos de corrección en el proceso de visión.

El trabajar en entornos poco estructurados dificulta la aplicación de los procesos y técnicas de visión, por lo que se intenta que las condiciones de iluminación permanezcan invariantes el mayor tiempo posible.

En un entorno de trabajo pueden diferenciarse dos tipos de fuentes de iluminación:

- Fuente de iluminación directa.
- Fuente de iluminación arbitraria.

3.2.1.1. Fuente de iluminación directa

Si se tiene una fuente de iluminación directa, la luz incide directamente sobre la cara del usuario intentándose así reducir al máximo los efectos de sombra puesto que se mejora sensiblemente el contraste de la imagen. Sin embargo aparecen otros problemas como brillos y zonas oscuras indeseadas

si el foco de luz no se encuentra en posición y orientación cercana a la cámara web de captura o si el plano de la cara del usuario no está completamente perpendicular al haz de luz del foco emisor. Por tanto, puede falsearse el proceso de captura de imágenes. Además hay que tener en cuenta que un foco de luz apuntando directamente sobre la cara del usuario resulta molesto, disminuyendo el grado de aceptación del sistema de reconocimiento facial.

3.2.1.2. Fuente de iluminación arbitraria

Generalmente, la iluminación arbitraria del entorno de trabajo no suele ser recomendable en procesos de visión por computador ya que las imágenes obtenidas presentan un bajo contraste y un nivel de sombras inadecuado. Las imágenes obtenidas, más si se trabaja en exteriores, pueden variar enormemente de una a otra hora del día, dificultando tanto el proceso de captación como el de reconocimiento ya que influyen tanto factores climáticos como otras fuentes de iluminación no naturales.

3.2.1.3. Iluminación del entorno de trabajo

Vistas la particularidades de los diferentes tipos de iluminación y teniendo en cuenta que una de las características iniciales del proyecto es que se asumía que las condiciones de iluminación iban a ser constantes, para la iluminación del entorno de trabajo se ha elegido una mezcla de ambos tipos de iluminación, es decir, se cuenta con diferentes focos de luz directa, pero no se prescinde de la iluminación indirecta y arbitraria, ya sea natural o artificial, procurándose sin embargo que la zona donde el usuario se establezca para que se le capture una imagen de una cara tenga una variabilidad de iluminación prácticamente nula al igual que el fondo.

3.2.2. Entorno estructurado

Este sistema de reconocimiento facial, y en particular la parte de detección de la cara de usuario, ha sido diseñado para trabajar en un entorno mínimamente no estructurado, es decir, a la condición de invariabilidad de la iluminación se suma la invariabilidad del entorno. Si la iluminación es uniforme y constante, el fondo que pueda aparecer en la imagen capturada de la cara del usuario se verá también constante y por tanto no afectará al resultado.

Existen sin embargo otros parámetros que sí interfieren más en el sistema.

- Distancia de la cara del usuario a la cámara. Para personas de distintas edades y estaturas la distancia de la cara a la cámara puede verse alterada significativamente, por lo que será misión del software de extracción y normalización de la cara hacer que este efecto sea inapreciable.
- Posición variable del plano de la cara del sujeto respecto del plano de la cámara, pudiéndose tener por tanto diferentes orientaciones de una misma cara de usuario e incluso no llegar a detectar una cara en una imagen.

Sin embargo, la robustez del sistema diseñado hace posible que pequeñas variaciones de todas estas características del y en el entorno de trabajo no influyan en los resultados.

Anexo 4

Modelo de la entrevista realizada a directivos de los centros CISED y GEySED.

Entrevista realizada a directivos del centro CISED y GEySED.	
Preguntas	Respuestas
¿Existe en el CISED algún sistema con funcionalidades similares a las que posee el componente propuesto?	
¿En caso afirmativo: Diga si el sistema fue desarrollado utilizando el SDK de la cámara digital o utilizando tecnologías genéricas, o ambas?	
¿Diga cuáles son las funcionalidades que brinda dicho sistema?	
¿Conoce de algún sistema implementado en la Universidad de las Ciencias Informáticas que posea funcionalidades similares a las que posee el componente propuesto?	
¿En caso afirmativo: Diga si el sistema fue desarrollado utilizando el SDK de la cámara digital o utilizando tecnologías genéricas, o ambas?	
¿Diga cuáles son las funcionalidades que brinda dicho sistema?	

Tabla 34. Modelo de entrevista realizada.

Anexo 5

Revisión de requisitos

La revisión de los requisitos es una de las técnicas de validación más utilizadas. Ellas permiten descubrir una gran cantidad de defectos en los mismos, reducir el costo de desarrollo y reducir el tiempo de pruebas.

En la revisión normalmente se realizan reuniones entre los implicados en el sistema y las mismas se realizan habitualmente siguiendo un procedimiento compuesto por siete pasos(37):

- **Preparar el plan de la revisión.** Este plan debería incluir al menos lo siguiente: Las tareas a realizar (esto es, las que se indican a continuación), la planificación temporal y las personas que participarán en la revisión.
- **Distribuir los documentos a revisar.** Habitualmente, el único documento a revisar será el documento de especificación. Junto con este documento, también es necesario remitir a los participantes en la revisión todos aquellos documentos que ayuden a comprender adecuadamente el documento de especificación. Típicamente, tales documentos son los documentos referenciados y los anexos a la especificación.
- **Preparar la reunión.** Esta tarea es muy distinta, dependiendo de quien la realice: o Para el analista promotor de la reunión, esta tarea es fundamentalmente logística. Básicamente consiste en reservar una sala donde realizar la revisión, solicitar los materiales que sean necesarios (desde papel y lápiz a cañones de proyección), preparar justificantes del tiempo empleado por el personal participante (en caso de que sea preciso), etc. o Para los analistas participantes, la preparación de la reunión es, probablemente, la más importante de las tareas de revisión. Durante esta tarea se deben leer cuidadosamente los documentos recibidos y anotar todas aquellos defectos (o cosas que parezcan defectos) con la finalidad de ponerlos de manifiesto durante la reunión posterior.
- **Realizar la reunión de revisión.** Básicamente, El formato de esta reunión puede ser muy diverso: Puede oscilar desde una total falta de control hasta grupos muy formalizados y sujetos a protocolos de actuación.
- **Identificar los defectos y acciones a realizar.** La lista de defectos y acciones recomendadas es el documento final obtenido en las revisiones de requisitos.

- **Realizar las correcciones que sean precisas a los documentos revisados.** El analista promotor de la reunión (esto es, el analista encargado del documento de especificación) debe evaluar y, si lo estima conveniente, llevar a cabo, las acciones recomendadas que han surgido de la reunión de revisión.
- **Informar de las modificaciones realizadas a los participantes en la reunión.** Una vez que los defectos en la especificación han sido subsanados, debería enviarse un breve informe de las tareas realizadas, y una copia corregida de los documentos de especificación, a los participantes en la reunión para su visto bueno.

Se realizaron varias reuniones con los clientes, en las mismas se desarrollaron todos los pasos anteriormente explicados encontrando en cada una de las reuniones inconsistencias, las cuales fueron subsanadas correctamente.

Anexo 6

Matriz de trazabilidad

La matriz de trazabilidad es otra técnica que se emplea para validar los requerimientos. A continuación se muestra una tabla diseñada específicamente para hacer corresponder los requisitos funcionales (RF) con los objetivos del sistema.

RF \ Objetivos del sistema	Inicialización del dispositivo de captura	Ajuste del vídeo	Captura de la imagen
Verificar compatibilidad con la tecnología genérica WIA.	X		
Pre-visualizar Vídeo.	X		
Realizar zoom del lente de la cámara.		X	
Seleccionar tipo de flash.		X	
Captar la imagen.			X

Mostrar la imagen capturada.			X
------------------------------	--	--	---

Tabla 35. Matriz de trazabilidad.

Como se puede apreciar en la matriz de trazabilidad todos los requisitos identificados están relacionados con las funciones que debe tener el componente. De esta forma se asegura que todos los requerimientos sean implementados y que no existan inconsistencias.

Anexo 7

Lista de Chequeo para validar el listado de requisitos

Seguidamente se muestra una propuesta de lista de chequeo aplicada a los requisitos con el objetivo de verificar que estén correctos y no presenten inconsistencias.

El objetivo general de la lista de chequeo que se mostrará a continuación es evaluar la especificación de los requisitos en cuanto a su calidad. Al concluir la calificación de todos los elementos relacionados con el componente para la captura de imágenes faciales se tienen recogidos en el documento los puntos eficientes y los ineficientes que tienen los requisitos capturados.

La presente planilla es aplicable a todos los requisitos que inicien el proceso de desarrollo del software para el componente para la captura de imágenes faciales con múltiples modelos de cámaras digitales.

Definiciones, Acrónimos y Abreviaturas

Nivel: Importancia del aspecto a evaluar

Eval: Evaluación

NP: No Procede

Comentario: Es obligatorio en las respuestas negativas.

Las evaluaciones serán:

(--) **Incorrectos** – no es entendible. (0)

(-) **Ambiguos** – no es claro. (2)

(+) **Preciso** – se entiende lo que quiere decir. (4)

(++) **Correctos** – está correctamente. (5)

El peso de la pregunta será:

(!!) Muy importante (5)

(!) Menos importante (3)

Nivel	Evaluación	Eval.	NP	Comentario
A	Claridad ¿Los requisitos se escriben en lengua comprensible para el usuario/cliente?	5		
A	¿Hay requisitos que tienen más de una interpretación?	5		
A	¿Cada requisito característico del producto final se describe con una terminología única?	5		
A	¿Hay un glosario en el cual los requisitos significativos y específicos están en términos definidos y claros?	5		
A	¿Se entienden los requisitos para ponerlos en ejecución por un grupo independiente?	5		
A	Completo ¿El requisito tiene un contenido específico?	5		
A	Están especificados los cambios posibles a los requisitos.	4		
A	La probabilidad de cambios está especificada para cada requisito.	4		
B	Del contraste contra diversas fuentes no surgen errores u omisiones.	5		
B	¿Se etiquetan todas las figuras, tablas y diagramas?	5		
B	¿Se hace una remisión de todas las figuras, tablas y diagramas?	5		
A	¿Se definen todos los términos en cada uno de los requisitos determinados?	4		
B	¿Se ponen en un índice todos los términos	4		

	identificados?			
B	¿Todas las unidades de medida están bien definidas?	5		
A	¿Existen áreas donde está incompleta la información debido a que el desarrollo aún no ha comenzado a especificarse?	4		
A	¿La información que falta se define en el requisito?	4		
A	¿Algún requisito necesita una especificación detallada?	5		
A	¿Algún requisito necesita ser menos especificado?	4		
B	¿Todos los requisitos se describen ellos mismos?		NP	Se hizo una especificación de requerimientos por cada uno de los requisitos
A	¿Están incluidos todos los requisitos relacionados con la funcionalidad?	5		
A	¿Hay requisitos que produzcan inquietud?	5		
A	¿Están incluidos todos los requisitos relacionados con el funcionamiento?	5		
A	¿Están incluidos todos los requisitos relacionados con los apremios del diseño?	5		
A	¿Están incluidos todos los requisitos relacionados con sus cualidades?	5		
A	¿Están incluidos todos los requisitos relacionados con las interfaces externas?		NP	Dado que los requerimientos son de un componente de software que estará integrado dentro de un sistema.

A	¿Están incluidos todos los requisitos relacionados con las bases de datos?		NP	Dado que el componente de software no se relaciona con bases de datos.
A	¿Están incluidos todos los requisitos relacionados con el software?	5		
A	¿Están incluidos todos los requisitos relacionados con la comunicación?	5		
A	¿Están incluidos todos los requisitos relacionados con el hardware?	5		
A	¿Están incluidos todos los requisitos relacionados con las entradas?	5		
A	¿Están incluidos todos los requisitos relacionados con las salidas?	5		
B	¿Están incluidos todos los requisitos relacionados con la divulgación?		NP	Dado que el componente no posee ese requisito.
A	¿Están incluidos todos los requisitos relacionados con la seguridad?		NP	Dado que el componente no posee ese requisito.
A	¿Están incluidos todos los requisitos relacionados con la capacidad de mantenimiento?		NP	Dado que el componente no posee ese requisito.
A	¿Están incluidos todos los requisitos relacionados con la instalación?		NP	Dado que el componente no posee ese requisito.
B	¿Están incluidos todos los requisitos relacionados con la criticalidad?		NP	Dado que el componente no posee ese requerimiento.
A	¿Están incluidos todos los requisitos relacionados con las limitaciones de permanencia?		NP	Dado que el componente no posee ese requisito.
A	¿Se especifican los posibles cambios de los requisitos?		NP	Dado que el componente no posee ese requisito.
A	¿Se especifica para cada requisito la probabilidad del cambio?		NP	Dado que el componente no posee ese requisito.

A	Consistencia ¿Hay requisitos que describen el mismo objeto que estén en conflicto con otros requisitos con respecto a la terminología?	5		
A	¿Hay requisitos que describen al mismo objeto que estén en conflicto con respecto a las características?	5		
A	¿Hay requisitos que describan dos o más acciones que estén en conflicto temporalmente?	5		
A	Cada frase aporta a la especificación.	5		
A	Rastreabilidad ¿Los requisitos que fueron especificados por los usuarios fueron rastreados varias veces?	5		
A	¿Los requisitos que están registrados en el documento o que fueron registrados por alguna persona fueron rastreados varias veces?	5		
A	¿Todos los requisitos son detectados en el documento del diseño?	5		
A	¿Los requisitos fueron rastreados en todos los módulos?		NP	Dado que el componente de software no posee módulos.
A	Comprobabilidad ¿Existen requisitos que sean imposibles de cumplir?	5		
A	¿Para cada requisito hay un proceso que es ejecutado por un ser humano o una máquina para verificar el requisito?	5		
B	¿Hay requisitos que sean expresados en términos comprobables en la última fase?	5		
A	Modificabilidad	5		

	¿En el documento los requisitos se organizan de manera clara y lógica?			
A	¿La organización está adherida a un estándar aceptado?	5		
A	¿La redundancia es mínima y sólo se debe a distintos niveles de abstracción o detalle?	5		
A	Contenido General ¿Cada requisito es relevante al problema y a su solución?	5		
A	¿Está definido cualquier requisito en el diseño realmente?	5		
A	¿Está definido cualquier requisito con detalle en la verificación?	5		
A	¿Está definido cualquier requisito con detalle en la gerencia del proyecto?	5		
B	¿Tiene una sección de introducción?		NP	Debido a que el componente de software no requiere de dicha sección.
B	¿Tiene una sección de la descripción general?		NP	Debido a que el componente de software no requiere de dicha sección.
B	¿Tiene una sección del alcance?		NP	Debido a que el componente de software no requiere de dicha sección.
A	¿Tiene las definiciones, siglas, y una sección de las abreviaturas?	5		
A	¿Tiene una sección específica de los requisitos?	5		
B	¿Tiene una sección de la perspectiva del producto?		NP	Debido a que el componente de software no requiere de

				dicha sección.
B	¿Tiene una sección de las funciones del producto?		NP	Debido a que el componente de software no requiere de dicha sección.
B	¿Tiene una sección de las características del usuario?		NP	Debido a que el componente de software no requiere de dicha sección.
B	¿Tiene una sección general de los apremios?		NP	Debido a que el componente de software no requiere de dicha sección.
B	¿Tiene una sección de las dependencias?		NP	Debido a que el componente de software no requiere de dicha sección.
B	¿Están todos los apéndices necesarios?	5		
A	¿Están todas las tablas necesarias?	5		
A	¿Están todos los diagramas necesarios?	5		
A	Entradas ¿Se especifican todas las fuentes de la entrada?	5		
A	¿Se especifican todos los requisitos de la exactitud de la entrada?	5		
B	¿Se especifican todas las frecuencias de la entrada?	5		
B	¿Se especifican todos los formatos de la entrada?	4		
B	¿Se especifican todos los tipos de valores en las salidas?	4		
B	¿Se especifican todos los formatos de la salida?	4		
B	¿Se especifican las frecuencias de las salidas?	4		
A	Funciones	5		

	¿Son todas las funciones del software especificadas?			
A	¿Todas las entradas se especifican para cada función?	5		
A	¿Todos los aspectos del producto se especifican para cada función?	5		
A	¿Todas las salidas se especifican para cada función?	5		
A	¿Todos los requisitos de funcionamiento se especifican para cada función?	5		
A	¿Está todo el diseño especificado para cada función?	5		
A	¿Todas las cualidades se especifican para cada función?		NP	Dado que el componente no posee ese requisito.
A	¿Todos los requisitos de la seguridad especificados para cada función?		NP	Dado que el componente no posee ese requisito.
A	¿Todos los requisitos de la capacidad de mantenimiento se especifican para cada función?		NP	Dado que el componente no posee ese requisito.
A	¿Todos los requisitos de la base de datos se especifican para cada función?		NP	Dado que el componente no posee relación con bases de datos.
B	¿Todos los requisitos operacionales se especifican para cada función?		NP	Dado que el componente no posee ese requisito.
B	¿Todos los requisitos de la instalación se especifican para cada función?		NP	Dado que el componente no posee ese requisito.
A	Interfaces Externas ¿Se especifican todas las interfaces utilizadas?	5		
A	¿Se especifican todas las interfaces del hardware?	5		

A	¿Se especifican todas las interfaces del software?	5		
A	¿Se especifican todas las interfaces de comunicaciones?	5		
A	¿Se especifican todos los requisitos del diseño de interfaz?		NP	Dado que el componente de software no posee una interfaz para diseñar.
A	¿Se especifican todos los requisitos de la seguridad de interfaz?		NP	Dado que el componente de software no posee dicho requisito.
A	¿Se especifican todos los requisitos de la capacidad de mantenimiento de la interfaz?		NP	Dado que el componente de software no posee dicho requisito.
A	¿Se especifican todas las interfaces de interacción de humano – computador?		NP	Dado que en el componente de software no existen interfaces de interacción humano-computador.
A	Interfaces Internas ¿Se han identificado todas las interfaces internas?	5		
A	¿Se han especificado todas las características internas de las interfaces?	5		
A	Sincronización ¿Se especifican todos los tiempos de transformación?	5		
A	¿Se especifican todas las tarifas de transferencia de datos?	5		
A	¿Se especifica todo el sistema con tarifas?	5		
A	Hardware	5		

	¿Se especifica la memoria mínima?			
A	¿Se especifica el almacenamiento mínimo?	5		
A	¿Se especifica la memoria máxima?	5		
A	¿Se especifica el almacenamiento mínimo?	5		
A	Software ¿Se especifica el software requerido y el sistema operativo?	5		
A	¿Se especifican todas las unidades requeridas del software?	5		
A	¿Se especifican todos los productos de software comparados que deben ser utilizados con el sistema?	5		
A	Comunicaciones ¿Se especifica la red?		NP	Debido a que para desarrollar el componente de software no es necesaria la red.
A	¿Se especifican los protocolos requeridos?		NP	Debido a que para desarrollar el componente de software no es necesaria la red.
A	¿Se especifica la capacidad requerida de la red?		NP	Debido a que para desarrollar el componente de software no es necesaria la red.
A	¿Se especifica la tarifa del rendimiento de procesamiento de la red, requerido y estimado?		NP	Debido a que para desarrollar el componente de software no es necesaria la red.
A	¿Se especifica el mínimo estimado de las		NP	Debido a que para

	conexiones de red?			desarrollar el componente de software no es necesaria la red.
A	¿Se especifica los requisitos de funcionamiento mínimo de la red?		NP	Debido a que para desarrollar el componente de software no es necesaria la red.
A	¿Se especifican los requerimientos de funcionamiento máximo de la red?		NP	Debido a que para desarrollar el componente de software no es necesaria la red.
A	¿Se especifican los requerimientos de funcionamiento óptimos para la red?		NP	Debido a que para desarrollar el componente de software no es necesaria la red.
A	Trazabilidad Cada requisito obedece a una necesidad específica de usuario.	5		
A	Cada requisito tiene su origen en una fuente (documento o persona) específica.	5		
A	Cada requisito se puede rastrear hacia delante su incorporación al diseño.	5		
A	Cada requisito se puede rastrear hacia delante su incorporación en determinados módulos.	5		
A	Verificabilidad Cada requisito es implementable.	5		
A	Para cada requisito existe un procedimiento que, ejecutado por una persona o máquina, permite verificar si se cumple	5		

A	Hay algún requisito que se va a expresar en términos verificables más adelante.	5		
A	Priorización Cada requisito tiene asignado un nivel de prioridad asignado por el Cliente/Usuario para guiar las negociaciones/compromisos.	5		

Tabla 36: Lista de chequeo para requisitos.

Glosario

Algoritmo: Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad

Arquitectura: Organización de los diversos elementos constitutivos de un sistema informático.

BeOS: Es un sistema operativo para PC desarrollado por Be Incorporated en 1990, orientado principalmente a proveer alto rendimiento en aplicaciones multimedia. A pesar de la creencia común fomentada por la inclusión de la interfaz de comandos Bash en el sistema operativo, el diseño de BeOS no estaba basado en UNIX.

Bluetooth: Es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

CD: El disco compacto (conocido popularmente como CD, por la sigla en inglés de *Compact Disc*) es un soporte digital óptico utilizado para almacenar cualquier tipo de información (audio, vídeo, documentos y otros datos).

Digital Vídeo Broadcasting by Satellite (DVB-S): Es un sistema que permite incrementar la capacidad de transmisión de datos y televisión digital a través de un satélite UH11 usando el formato MPEG2. La estructura permite mezclar en una misma trama un gran número de servicios de Vídeo, audio y datos.

Disco rígido: Es un dispositivo de almacenamiento de datos no volátil que emplea un sistema de grabación magnética para almacenar datos digitales.

DVD: Es un formato de almacenamiento óptico que puede usarse para guardar datos. Su nombre proviene de la sigla de Digital Versatile Disc o Disco Versátil Digital. Se asemeja a los discos compactos (CD) en cuanto a sus dimensiones físicas, pero está codificado en un formato distinto y a una densidad mucho mayor.

DVB-T (Digital Vídeo Broadcasting – Terrestrial, en castellano Difusión de Vídeo Digital - Terrestre) es el estándar para la transmisión de televisión digital terrestre creado por la organización europea DVB. Este sistema transmite audio, Vídeo y otros datos a través de un flujo MPEG-2.

Eclipse: Escrito en Java aunque su principal uso es como IDE para Java, este es un lenguaje neutral. El soporte para desarrollo en Java es proveído por un componente enchufado o plugin, pero además están disponibles plugins para otros lenguajes, como C/C++, Cobol, C#, etc.

FireWire: El IEEE 1394 (conocido como FireWire por Apple Inc. y como i.Link por Sony) es un estándar multiplataforma para entrada/salida de datos en serie a gran velocidad. Suele utilizarse para la interconexión de dispositivos digitales como cámaras digitales y Vídeocámaras a computadoras.

Hardware: Corresponde a todas las partes tangibles de una computadora.

HTML: Siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP: HyperText Transfer Protocolo, protocolo cliente-servidor utilizado para el intercambio de páginas Web.

IDE: Entorno de desarrollo integrado, pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

Ingeniería de software: Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

Kernel: Se refiere al núcleo de un sistema operativo.

Mac OS X: Mac OS X es un sistema operativo desarrollado y comercializado por Apple Inc. que ha sido incluido en su gama de computadoras Macintosh desde 2002. Es el sucesor del Mac OS 9 (la versión final del Mac OS Classic), el sistema operativo de Apple desde 1984.

Media Transfer Protocol: Media Transfer Protocol (MTP) es un conjunto de extensiones a PTP (Picture Transfer Protocol) creado por Microsoft, para permitir al protocolo su uso con otros dispositivos además de cámaras digitales, como por ejemplo reproductores de audio digitales, y otros dispositivos digitales portátiles.

MPEG-1: Es el nombre de un grupo de estándares de codificación de audio y Vídeo normalizados por el grupo MPEG (Moving Pictures Experts Group). *MPEG-1 vídeo* se utiliza en el formato Vídeo CD. La calidad de salida con la tasa de compresión usual usada en VCD es similar a la de un cassette vídeo VHS doméstico.

MPEG-2: Es similar a MPEG-1, pero también proporciona soporte para vídeo entrelazado (el formato utilizado por las televisiones.) MPEG-2 vídeo no está optimizado para bajas tasas de bits (menores que 1 Mbit/s), pero supera en desempeño a MPEG-1 a 3 Mbit/s y superiores.

MPEG-4: Toma muchas de las características de MPEG-1 y MPEG-2 así como de otros estándares relacionados, tales como soporte de VRML (Virtual Reality Modeling Language) extendido para

Visualización 3D, archivos compuestos en orientación a objetos (incluyendo objetos audio, vídeo y VRML), soporte para la gestión de Derechos Digitales externos y variados tipos de interactividad.

Multiplataforma: es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

Periféricos: Se denomina periféricos a los aparatos o dispositivos auxiliares e independientes conectados a la unidad central de procesamiento de una computadora.

Picture Transfer Protocol: El Protocolo de transferencia de imágenes (PTP) o Picture Transfer Protocol por su sigla en inglés, es un protocolo ampliamente difundido y desarrollado por la Asociación Internacional de la Industria de Imágenes (*International Imaging Industry Association*). Este protocolo permite la transferencia de imágenes desde cámaras digitales hacia computadoras o hacia otros dispositivos digitales sin la necesidad de instalar controladores. El protocolo ha sido estandarizado como ISO 15740.

Píxel: Un píxel o pixel, plural píxeles es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

Plugin: Aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica y es ejecutada por la aplicación principal.

Protocolos: Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.

Puerto USB: Es un puerto que sirve para conectar periféricos a un ordenador. El USB puede conectar varios tipos de dispositivos como pueden ser: mouse, teclados, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, discos duros externos, entre otros.

Puertos serie / paralelo: Un puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

QNX: Es un sistema operativo de tiempo real basado en Unix. QNX está basado en una estructura de micronúcleo, que proporciona características de estabilidad avanzadas frente a fallos de dispositivos, aplicaciones, etc.

RAM: memoria RAM (Memoria de Acceso Aleatorio) es donde el computador guarda los datos que está utilizando en el momento presente.

Software: se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica.

Software Libre: es el software que una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente. El software libre suele estar disponible gratuitamente.

Solaris: Solaris es un sistema operativo de tipo Unix desarrollado desde 1992 inicialmente por Sun Microsystems y actualmente por Oracle Corporation como sucesor de SunOS. Es un sistema certificado oficialmente como versión de Unix

Windows: Microsoft Windows es el nombre de una serie de sistemas operativos desarrollados por Microsoft desde 1981. Las versiones más recientes de Windows son Windows 7 para equipos de escritorio, Windows Server 2008 R2 para servidores y Windows Phone 7 para dispositivos móviles.

Wi-Fi: es una marca de la *Wi-Fi Alliance* la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local.

Wrapper: Un Wrapper es un programa que controla el acceso a un segundo programa. El Wrapper literalmente cubre la identidad de este segundo programa, obteniendo con esto un más alto nivel de seguridad.